The Vernacular of the S-Matrix

Jacob L. Bourjaily

Lecce, Italy
International School on Amplitudes and Cosmology,
Holography and Positive Geometries

The Niels Bohr
International Academy

The Vernacular of the S-Matrix

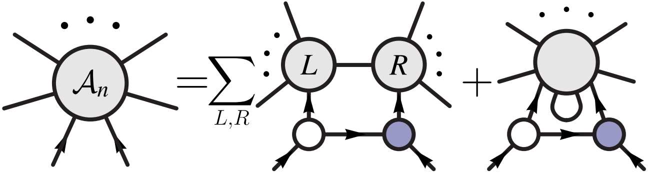
Jacob L. Bourjaily

Lecce, Italy
International School on Amplitudes and Cosmology,
Holography and Positive Geometries

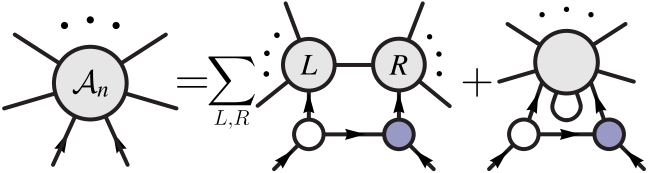
The Niels Bohr
International Academy

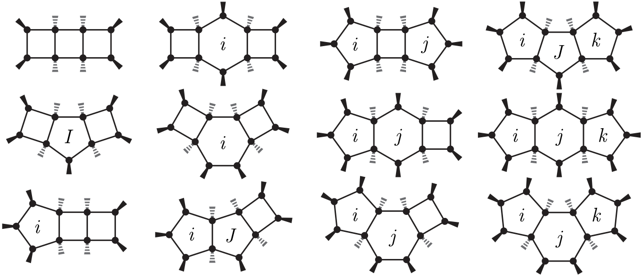
Enormous Advances in Understanding Scattering Amplitudes

Enormous Advances in Understanding Scattering Amplitudes

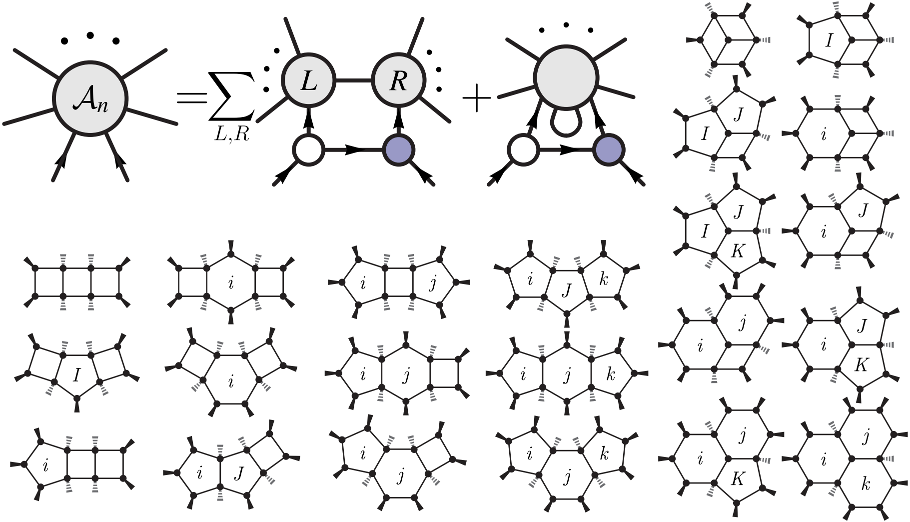


Enormous Advances in Understanding Scattering Amplitudes

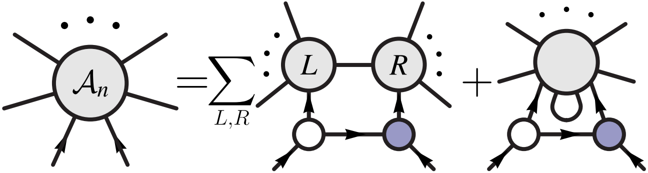


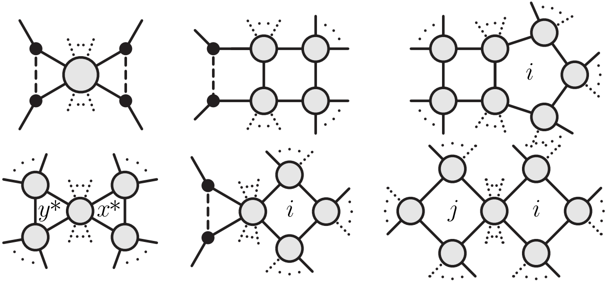


Enormous Advances in Understanding Scattering Amplitudes

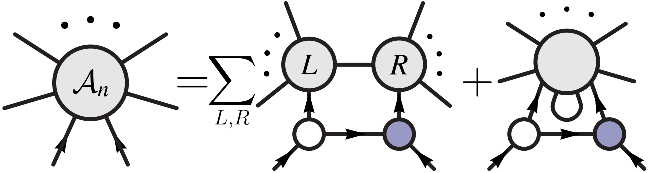


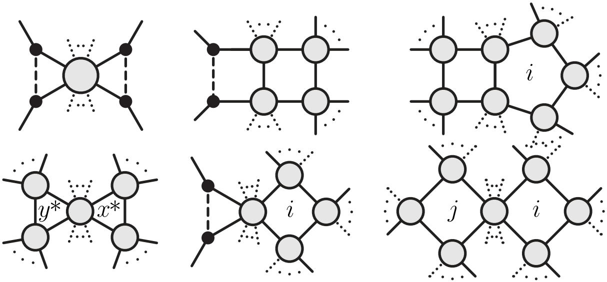
Enormous Advances in Understanding Scattering Amplitudes



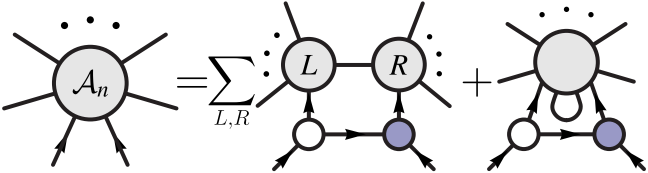


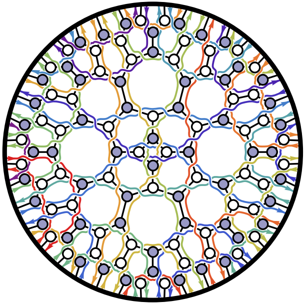
Enormous Advances in Understanding Scattering Amplitudes

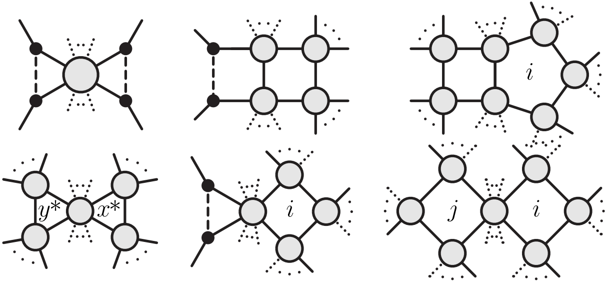




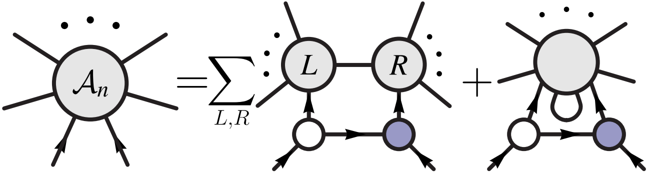
Enormous Advances in Understanding Scattering Amplitudes

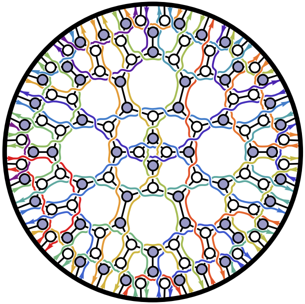


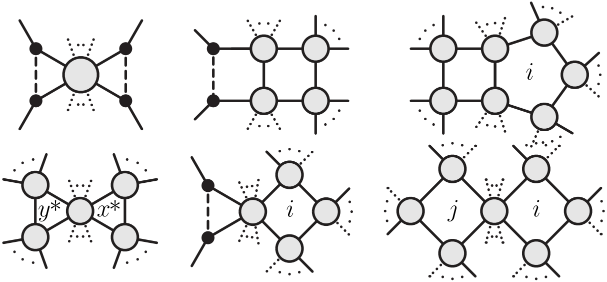




Enormous Advances in Understanding Scattering Amplitudes

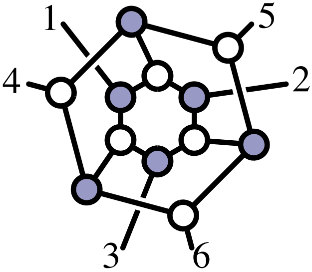






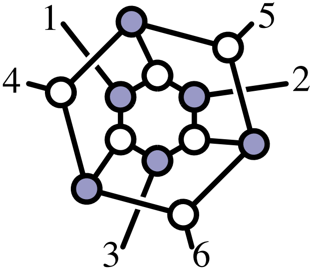
On-Shell Physics/Grassmannian Geometry Correspondence

On-Shell Physics/Grassmannian Geometry Correspondence



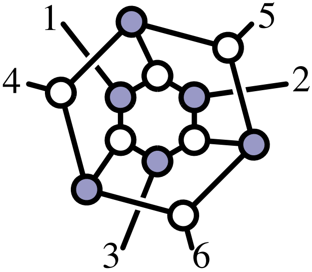
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v$$



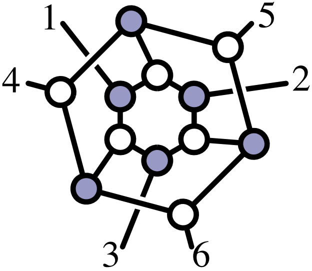
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



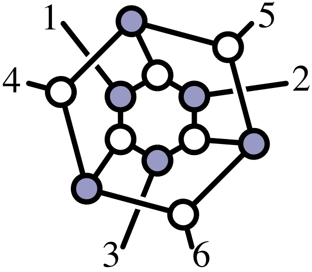
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



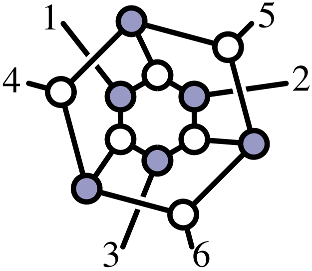
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



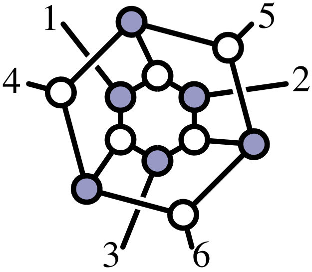
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C, p, h)$$



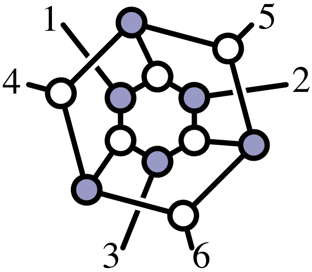
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



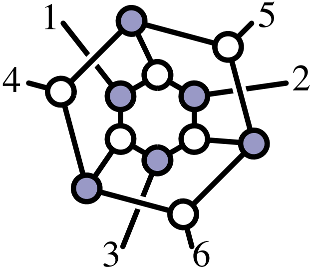
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C, p, h)$$



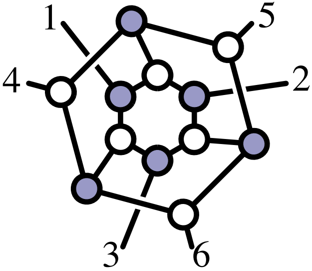
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C,p,h)$$



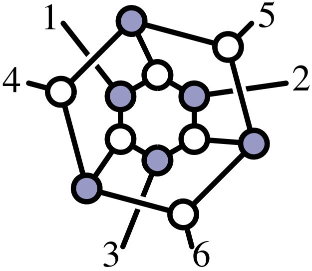
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



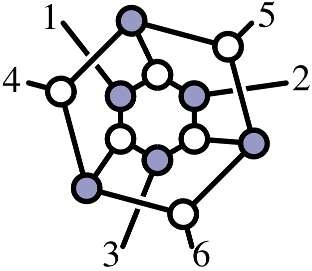
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



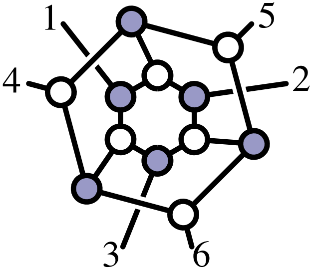
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C\ \delta(C,p,h)$$



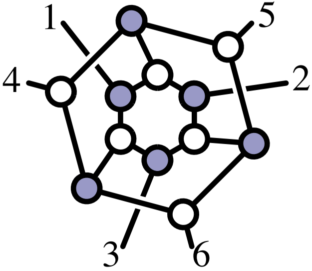
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$



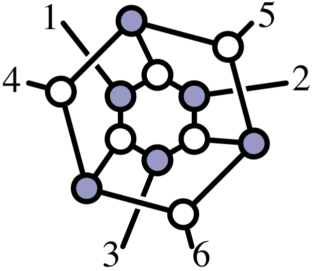
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



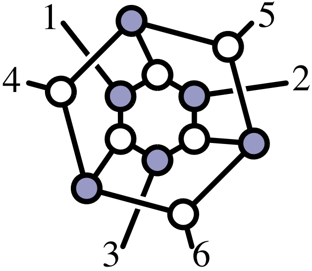
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



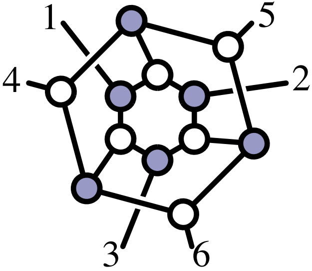
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



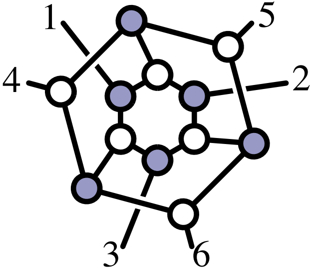
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \ \delta(C, p, h)$$



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$



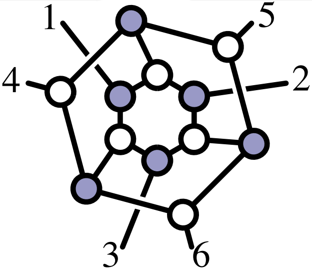
On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

$\Leftrightarrow$

Grassmannian Geometry



On-Shell Physics/Grassmannian Geometry Correspondence

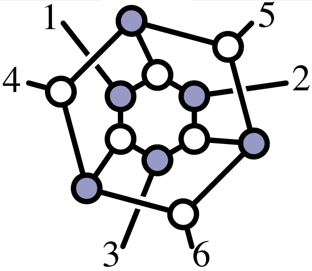
$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

On-Shell Physics

- on-shell diagrams

$\Leftrightarrow$

Grassmannian Geometry



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

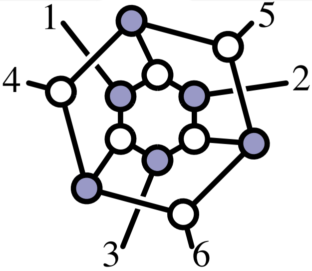
On-Shell Physics

- on-shell diagrams

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

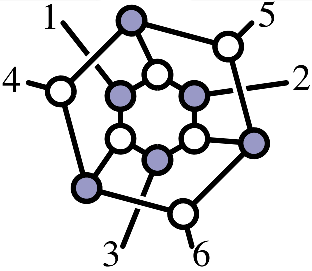
On-Shell Physics

- on-shell diagrams

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

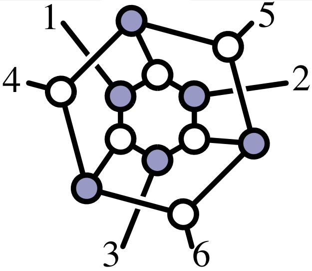
On-Shell Physics

- on-shell diagrams

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams

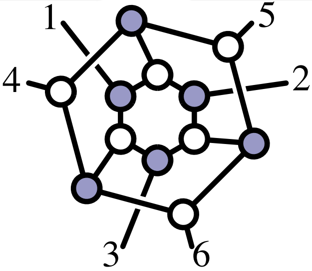
- physical symmetries

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$

- volume-preserving diffeomorphisms



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \ \delta(C, p, h)$$

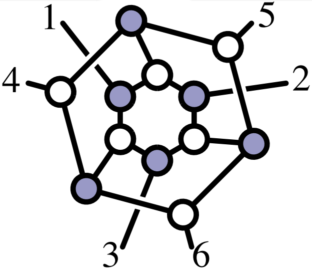
On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k, n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \ \delta(C, p, h)$$

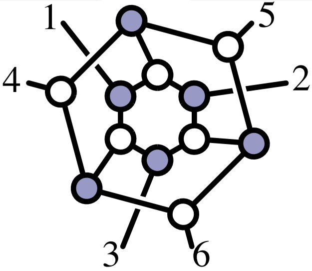
On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k, n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

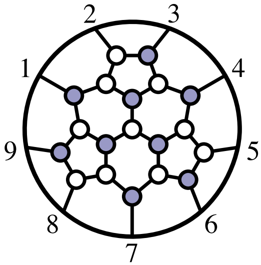
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

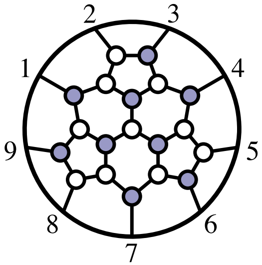
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

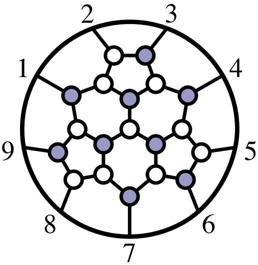
On-Shell Physics: planar $\mathcal{N}\!=\!4$

- on-shell diagrams
  - bi-colored, **un**directed
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \left(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \right) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C, p, h)$$

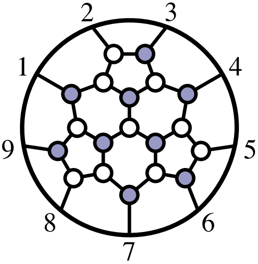
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

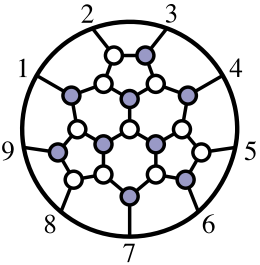
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C,p,h)$$

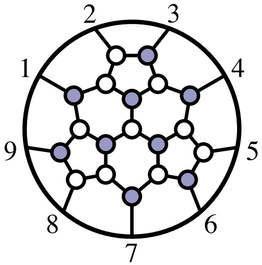
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \;\equiv\; \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

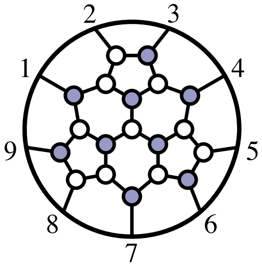
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

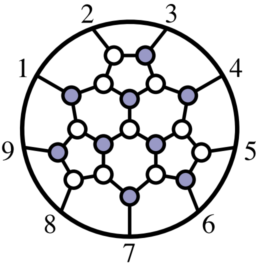
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

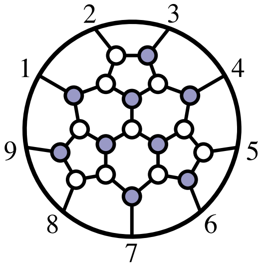
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}} \right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C,p,h)$$

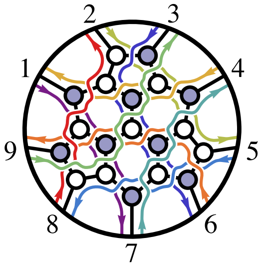
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \;\equiv\; \begin{pmatrix} 1 & \alpha_8 & \alpha_5{+}\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4{+}\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3{+}\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1}\wedge\cdots\wedge\frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

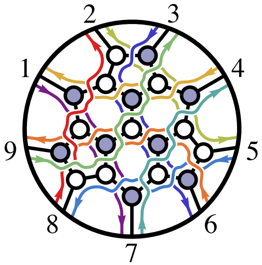
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries: the *Yangian*
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5+\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4+\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3+\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

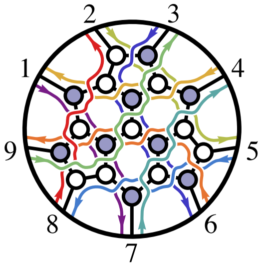
On-Shell Physics: planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **planar**
- physical symmetries: the *Yangian*
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5{+}\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4{+}\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3{+}\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

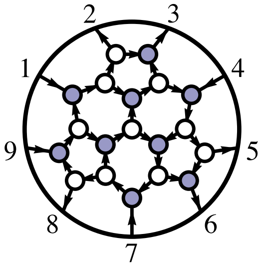
On-Shell Physics: planar $\mathcal{N}<4$

- on-shell diagrams
  - bi-colored, **directed** , **planar**
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)\times \mathcal{J}^{\mathcal{N}-4}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \;\equiv\; \begin{pmatrix} 1 & \alpha_8 & \alpha_5{+}\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4{+}\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3{+}\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \;\equiv\; \left(\frac{d\alpha_1}{\alpha_1}\wedge\cdots\wedge\frac{d\alpha_{14}}{\alpha_{14}}\right)\times(1)^{\mathcal{N}-4}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C\ \delta(C,p,h)$$

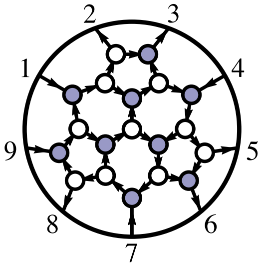
On-Shell Physics: planar $\mathcal{N}<4$

- on-shell diagrams
  - bi-colored, **directed** , **planar**
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big)\times\mathcal{J}^{\mathcal{N}-4}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \equiv \begin{pmatrix} 1 & \alpha_8 & \alpha_5{+}\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4{+}\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0 \\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3{+}\alpha_6\alpha_{12} \\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1}\wedge\cdots\wedge\frac{d\alpha_{14}}{\alpha_{14}}\right)\times(1)^{\mathcal{N}-4}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C,p,h)$$

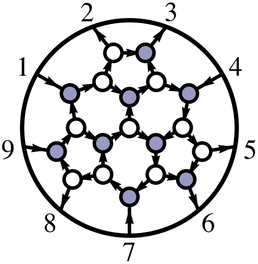
On-Shell Physics: planar $\mathcal{N}<4$

- on-shell diagrams
  - bi-colored, **directed** , **planar**
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
  - positroid variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big)\times\mathcal{J}^{\mathcal{N}-4}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C \;\equiv\; \begin{pmatrix} 1 & \alpha_8 & \alpha_5{+}\alpha_8\alpha_{14} & \alpha_5\alpha_{11} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_4{+}\alpha_{10}\alpha_{13} & \alpha_4\alpha_7 & 0 & 0\\ \alpha_3\alpha_9 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_3{+}\alpha_6\alpha_{12}\\ \alpha_9 & 0 & \alpha_1 & \alpha_1\alpha_{11} & 0 & \alpha_1\alpha_2 & \alpha_1\alpha_2\alpha_7 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1}\wedge\cdots\wedge\frac{d\alpha_{14}}{\alpha_{14}}\right)\times(1)^{\mathcal{N}-4}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

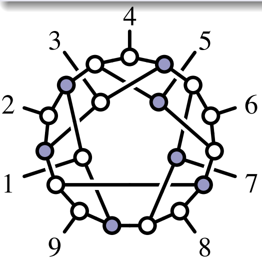
On-Shell Physics: non-planar $\mathcal{N}\!=\!4$

- on-shell diagrams
  - bi-colored, **un**directed, **non**-planar
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C\in G(k,n)$, volume-form $\Omega_C\}$
  - cluster variety   , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C^\perp \equiv \begin{pmatrix} \alpha_1 & 1 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 1 & 0 & \alpha_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & 1 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_7 & \alpha_8 \\ 0 & 0 & 0 & 0 & 0 & \alpha_9 & 1 & \alpha_{10} & 0 \\ \alpha_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_{12} \\ 0 & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

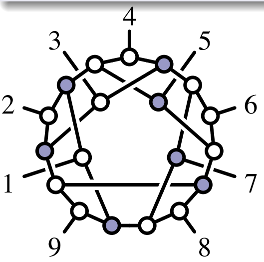
On-Shell Physics: non-planar $\mathcal{N}=4$

- on-shell diagrams
  - bi-colored, **un**directed, **non**-planar
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - cluster variety , $\left(\prod_i \frac{d\alpha_i}{\alpha_i}\right)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C^\perp \equiv \begin{pmatrix} \alpha_1 & 1 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 1 & 0 & \alpha_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & 1 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_7 & \alpha_8 \\ 0 & 0 & 0 & 0 & 0 & \alpha_9 & 1 & \alpha_{10} & 0 \\ \alpha_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_{12} \\ 0 & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

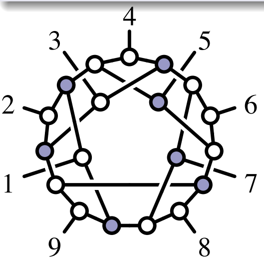
On-Shell Physics: non-planar $\mathcal{N}\!=\!4$

- on-shell diagrams
  - bi-colored, **un**directed, **non**-planar
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Longleftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - cluster variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big)$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C^\perp \equiv \begin{pmatrix} \alpha_1 & 1 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 1 & 0 & \alpha_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & 1 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_7 & \alpha_8 \\ 0 & 0 & 0 & 0 & 0 & \alpha_9 & 1 & \alpha_{10} & 0 \\ \alpha_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_{12} \\ 0 & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}} \right)$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

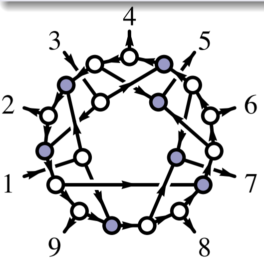
On-Shell Physics: non-planar $\mathcal{N} < 4$

- on-shell diagrams
  - bi-colored, **directed** , **non**-planar
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - cluster variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big)\times\mathcal{J}^{\mathcal{N}-4}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C^\perp \equiv \begin{pmatrix} \alpha_1 & 1 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 1 & 0 & \alpha_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & 1 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_7 & \alpha_8 \\ 0 & 0 & 0 & 0 & 0 & \alpha_9 & 1 & \alpha_{10} & 0 \\ \alpha_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_{12} \\ 0 & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \left(\frac{d\alpha_1}{\alpha_1}\wedge\cdots\wedge\frac{d\alpha_{14}}{\alpha_{14}}\right)\times\big(1+\alpha_2\alpha_4\alpha_{13}(\alpha_8+\alpha_7\alpha_{12})\big)^{\mathcal{N}-4}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

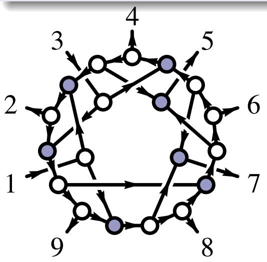
On-Shell Physics: non-planar $\mathcal{N} < 4$

- on-shell diagrams
  - bi-colored, **directed** , **non**-planar
- physical symmetries: **?**
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
  - cluster variety , $\big(\prod_i \frac{d\alpha_i}{\alpha_i}\big) \times \mathcal{J}^{\mathcal{N}-4}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations



$$C^\perp \equiv \begin{pmatrix} \alpha_1 & 1 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 1 & 0 & \alpha_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & 1 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_7 & \alpha_8 \\ 0 & 0 & 0 & 0 & 0 & \alpha_9 & 1 & \alpha_{10} & 0 \\ \alpha_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha_{12} \\ 0 & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega_C \equiv \Big(\frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_{14}}{\alpha_{14}}\Big) \times \big(1 + \alpha_2\alpha_4\alpha_{13}(\alpha_8 + \alpha_7\alpha_{12})\big)^{\mathcal{N}-4}$$

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_{\Gamma} \equiv \prod_{i} \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_{v} \mathcal{A}_v \equiv \int \Omega_C \ \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \;\delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k, n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i\Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C,p,h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist?

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i,q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \ \delta(C,p,h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)
- what (functional) relations do they satisfy?

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Bigg(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Bigg) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams

- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k, n)$, volume-form $\Omega_C\}$

- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)
- what (functional) relations do they satisfy?
- what are their (infinite-dimensional) symmetries?

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams

- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$

- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)
- what (functional) relations do they satisfy?
- what are their (infinite-dimensional) symmetries?
  - do these extend to entire *amplitudes*?

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\text{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)
- what (functional) relations do they satisfy?
- what are their (infinite-dimensional) symmetries?
  - do these extend to entire *amplitudes*?
- do loop-level recursion relations exist?

On-Shell Physics/Grassmannian Geometry Correspondence

$$f_\Gamma \equiv \prod_i \Big(\sum_{h_i, q_i} \int d^3\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v \equiv \int \Omega_C \; \delta(C, p, h)$$

On-Shell Physics

- on-shell diagrams
- physical symmetries
  - trivial symmetries (identities)

$\Leftrightarrow$

Grassmannian Geometry

- $\{$strata $C \in G(k,n)$, volume-form $\Omega_C\}$
- volume-preserving diffeomorphisms
  - cluster coordinate mutations

Important Open Questions (for math *and* physics)

- how many functions exist? (how to name them?)
- what (functional) relations do they satisfy?
- what are their (infinite-dimensional) symmetries?
  - do these extend to entire *amplitudes*?
- do loop-level recursion relations exist?

Organization and Outline

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

Supercomputer Computations in Quantum Chromodynamics

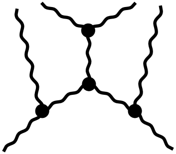
Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

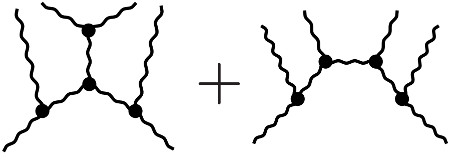
- 220 Feynman diagrams



Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

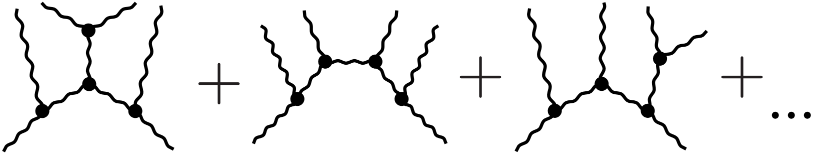
- 220 Feynman diagrams



Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

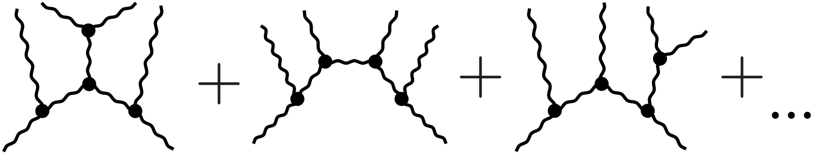
- 220 Feynman diagrams



Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

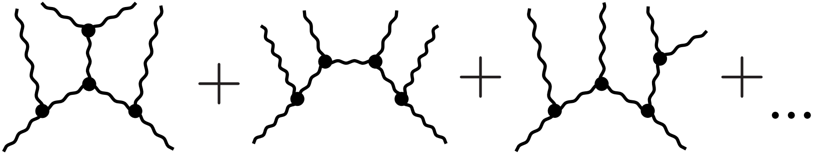
- 220 Feynman diagrams, thousands of terms

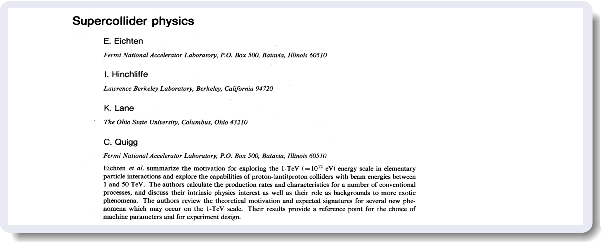


Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms





Supercollider physics

E. Eichten

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

I. Hinchliffe

Lawrence Berkeley Laboratory, Berkeley, California 94720

K. Lane

The Ohio State University, Columbus, Ohio 43210

C. Quigg

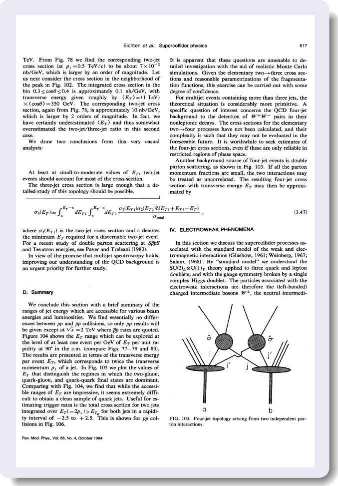
Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

Eichten *et al.* summarize the motivation for exploring the 1-TeV ($=10^{12}$ eV) energy scale in elementary particle interactions and explore the capabilities of proton-(anti)proton colliders with beam energies between 1 and 50 TeV. The authors calculate the production rates and characteristics for a number of conventional processes, and discuss their intrinsic physics interest as well as their role as backgrounds to more exotic phenomena. The authors review the theoretical motivation and expected signatures for several new phenomena which may occur on the 1-TeV scale. Their results provide a reference point for the choice of machine parameters and for experiment design.

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms



Eichten *et al.*: Supercollider physics 617

TeV. From Fig. 78 we find the corresponding two-jet cross section (at $p_\perp = 0.5$ TeV/c) to be about 7×10^{-2} nb/GeV, which is larger by an order of magnitude. Let us next consider the cross section in the neighborhood of the peak in Fig. 102. The integrated cross section in the bin $0.3 \leq \cos\theta \leq 0.4$ is approximately 0.1 nb/GeV, with transverse energy given roughly by $\langle E_T \rangle \approx (1\ \text{TeV}) \times \langle \cos\theta \rangle = 350$ GeV. The corresponding two-jet cross section, again from Fig. 78, is approximately 10 nb/GeV, which is larger by 2 orders of magnitude. In fact, we have certainly underestimated $\langle E_T \rangle$ and thus somewhat overestimated the two-jet/three-jet ratio in this second case.

We draw two conclusions from this very casual analysis:

At least at small-to-moderate values of E_T, two-jet events should account for most of the cross section.

The three-jet cross section is large enough that a detailed study of this topology should be possible.

It is apparent that these questions are amenable to detailed investigation with the aid of realistic Monte Carlo simulations. Given the elementary two→three cross sections and reasonable parametrizations of the fragmentation functions, this exercise can be carried out with some degree of confidence.

For multijet events containing more than three jets, the theoretical situation is considerably more primitive. A specific question of interest concerns the QCD four-jet background to the detection of W^+W^- pairs in their nonleptonic decays. The cross sections for the elementary two→four processes have not been calculated, and their complexity is such that they may not be evaluated in the foreseeable future. It is worthwhile to seek estimates of the four-jet cross sections, even if these are only reliable in restricted regions of phase space.

Another background source of four-jet events is double parton scattering, as shown in Fig. 103. If all the parton momentum fractions are small, the two interactions may be treated as uncorrelated. The resulting four-jet cross section with transverse energy E_T may then be approximated by

$$\sigma_4(E_T) \simeq \int_\varepsilon^{E_T-\varepsilon} dE_{T1} \int_\varepsilon^{E_T-\varepsilon} dE_{T2} \, \frac{\sigma_2(E_{T1})\sigma_2(E_{T2})\delta(E_{T1}+E_{T2}-E_T)}{\sigma_{\text{total}}} , \tag{3.47}$$

where $\sigma_2(E_{T1})$ is the two-jet cross section and ε denotes the minimum E_T required for a discernable two-jet event. For a recent study of double parton scattering at $Sp\bar{p}S$ and Tevatron energies, see Paver and Treleani (1983).

In view of the promise that multijet spectroscopy holds, improving our understanding of the QCD background is an urgent priority for further study.

D. Summary

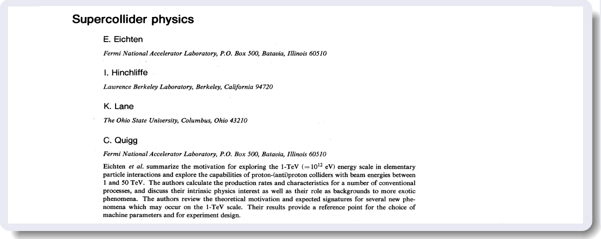
We conclude this section with a brief summary of the ranges of jet energy which are accessible for various beam energies and luminosities. We find essentially no differences between pp and $\bar{p}p$ collisions, so only pp results will be given except at $\sqrt{s} = 2$ TeV where $\bar{p}p$ rates are quoted. Figure 104 shows the E_T range which can be explored at the level of at least one event per GeV of E_T per unit rapidity at 90° in the c.m. (compare Figs. 77—79 and 83). The results are presented in terms of the transverse energy per event E_T, which corresponds to twice the transverse momentum $p_\perp$ of a jet. In Fig. 105 we plot the values of E_T that distinguish the regimes in which the two-gluon, quark-gluon, and quark-quark final states are dominant. Comparing with Fig. 104, we find that while the accessible ranges of E_T are impressive, it seems extremely difficult to obtain a clean sample of quark jets. Useful for estimating trigger rates is the total cross section for two jets integrated over $E_T (= 2p_\perp) > E_{T_0}$ for both jets in a rapidity interval of -2.5 to $+2.5$. This is shown for pp collisions in Fig. 106.

Rev. Mod. Phys., Vol. 56, No. 4, October 1984

IV. ELECTROWEAK PHENOMENA

In this section we discuss the supercollider processes associated with the standard model of the weak and electromagnetic interactions (Glashow, 1961; Weinberg, 1967; Salam, 1968). By "standard model" we understand the $SU(2)_L \otimes U(1)_Y$ theory applied to three quark and lepton doublets, and with the gauge symmetry broken by a single complex Higgs doublet. The particles associated with the electroweak interactions are therefore the (left-handed) charged intermediate bosons $W^\pm$, the neutral intermedi-

FIG. 103. Four-jet topology arising from two independent parton interactions.



Supercollider physics

E. Eichten

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

I. Hinchliffe

Lawrence Berkeley Laboratory, Berkeley, California 94720

K. Lane

The Ohio State University, Columbus, Ohio 43210

C. Quigg

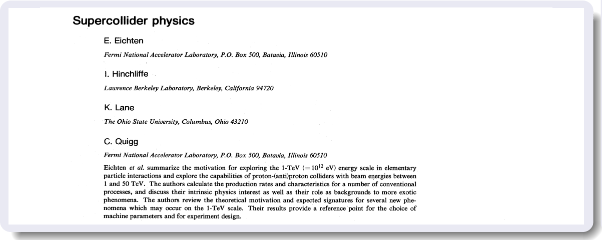
Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

Eichten *et al.* summarize the motivation for exploring the 1-TeV ($= 10^{12}$ eV) energy scale in elementary particle interactions and explore the capabilities of proton-(anti)proton colliders with beam energies between 1 and 50 TeV. The authors calculate the production rates and characteristics for a number of conventional processes, and discuss their intrinsic physics interest as well as their role as backgrounds to more exotic phenomena. The authors review the theoretical motivation and expected signatures for several new phenomena which may occur on the 1-TeV scale. Their results provide a reference point for the choice of machine parameters and for experiment design.

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms



Supercollider physics

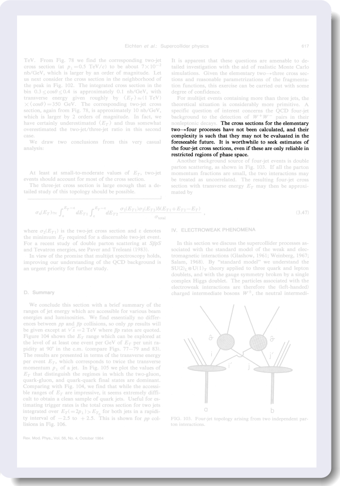
E. Eichten
Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

I. Hinchliffe
Lawrence Berkeley Laboratory, Berkeley, California 94720

K. Lane
The Ohio State University, Columbus, Ohio 43210

C. Quigg
Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510

Eichten *et al.* summarize the motivation for exploring the 1-TeV ($=10^{12}$ eV) energy scale in elementary particle interactions and explore the capabilities of proton-(anti)proton colliders with beam energies between 1 and 50 TeV. The authors calculate the production rates and characteristics for a number of conventional processes, and discuss their intrinsic physics interest as well as their role as backgrounds to more exotic phenomena. The authors review the theoretical motivation and expected signatures for several new phenomena which may occur on the 1-TeV scale. Their results provide a reference point for the choice of machine parameters and for experiment design.



Eichten *et al.*: Supercollider physics 617

TeV. From Fig. 78 we find the corresponding two-jet cross section (at $p_\perp = 0.5$ TeV/c) to be about 7×10^{-2} nb/GeV, which is larger by an order of magnitude. Let us next consider the cross section in the neighborhood of the peak in Fig. 102. The integrated cross section in the bin $0.3 \le \cos\theta \le 0.4$ is approximately 0.1 nb/GeV, with transverse energy given roughly by $\langle E_T\rangle \approx (1\text{ TeV}) \times \langle\cos\theta\rangle = 350$ GeV. The corresponding two-jet cross section, again from Fig. 78, is approximately 10 nb/GeV, which is larger by 2 orders of magnitude. In fact, we have certainly underestimated $\langle E_T\rangle$ and thus somewhat overestimated the two-jet/three-jet ratio in this second case.

We draw two conclusions from this very casual analysis:

At least at small-to-moderate values of E_T, two-jet events should account for most of the cross section.

The three-jet cross section is large enough that a detailed study of this topology should be possible.

It is apparent that these questions are amenable to detailed investigation with the aid of realistic Monte Carlo simulations. Given the elementary two→three cross sections and reasonable parametrizations of the fragmentation functions, this exercise can be carried out with some degree of confidence.

For multijet events containing more than three jets, the theoretical situation is considerably more primitive. A specific question of interest concerns the QCD four-jet background to the detection of W^+W^- pairs in their nonleptonic decays. **The cross sections for the elementary two→four processes have not been calculated, and their complexity is such that they may not be evaluated in the foreseeable future. It is worthwhile to seek estimates of the four-jet cross sections, even if these are only reliable in restricted regions of phase space.**

Another background source of four-jet events is double parton scattering, as shown in Fig. 103. If all the parton momentum fractions are small, the two interactions may be treated as uncorrelated. The resulting four-jet cross section with transverse energy E_T may then be approximated by

$$\sigma_4(E_T) \simeq \int_\varepsilon^{E_T-\varepsilon} dE_{T1} \int_\varepsilon^{E_T-\varepsilon} dE_{T2} \frac{\sigma_2(E_{T1})\sigma_2(E_{T2})\delta(E_{T1}+E_{T2}-E_T)}{\sigma_{\text{total}}}, \tag{3.47}$$

where $\sigma_2(E_{T1})$ is the two-jet cross section and ε denotes the minimum E_T required for a discernable two-jet event. For a recent study of double parton scattering at *S$\bar{p}p$S* and Tevatron energies, see Paver and Treleani (1983).

In view of the promise that multijet spectroscopy holds, improving our understanding of the QCD background is an urgent priority for further study.

D. Summary

We conclude this section with a brief summary of the ranges of jet energy which are accessible for various beam energies and luminosities. We find essentially no differences between pp and $\bar{p}p$ collisions, so only pp results will be given except at $\sqrt{s} = 2$ TeV where $\bar{p}p$ rates are quoted. Figure 104 shows the E_T range which can be explored at the level of at least one event per GeV of E_T per unit rapidity at 90° in the c.m. (compare Figs. 77—79 and 83). The results are presented in terms of the transverse energy per event E_T, which corresponds to twice the transverse momentum $p_\perp$ of a jet. In Fig. 105 we plot the values of E_T that distinguish the regimes in which the two-gluon, quark-gluon, and quark-quark final states are dominant. Comparing with Fig. 104, we find that while the accessible ranges of E_T are impressive, it seems extremely difficult to obtain a clean sample of quark jets. Useful for estimating trigger rates is the total cross section for two jets integrated over $E_T(=2p_\perp) > E_{T_0}$ for both jets in a rapidity interval of -2.5 to $+2.5$. This is shown for pp collisions in Fig. 106.

IV. ELECTROWEAK PHENOMENA

In this section we discuss the supercollider processes associated with the standard model of the weak and electromagnetic interactions (Glashow, 1961; Weinberg, 1967; Salam, 1968). By "standard model" we understand the $SU(2)_L \otimes U(1)_Y$ theory applied to three quark and lepton doublets, and with the gauge symmetry broken by a single complex Higgs doublet. The particles associated with the electroweak interactions are therefore the (left-handed) charged intermediate bosons $W^\pm$, the neutral intermedi-

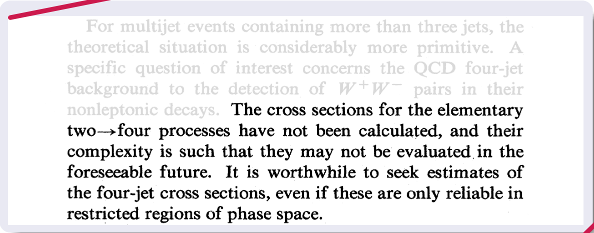
FIG. 103. Four-jet topology arising from two independent parton interactions.

Rev. Mod. Phys., Vol. 56, No. 4, October 1984

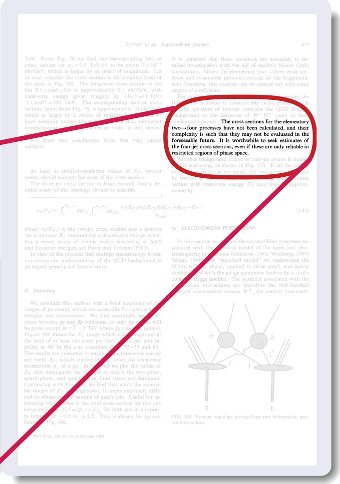
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms



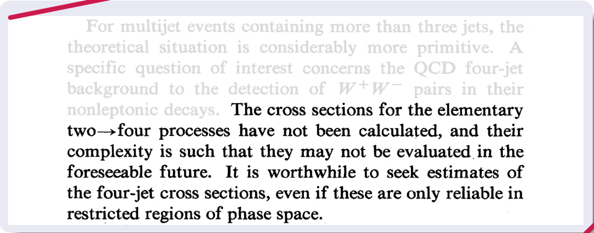
For multijet events containing more than three jets, the theoretical situation is considerably more primitive. A specific question of interest concerns the QCD four-jet background to the detection of W^+W^- pairs in their nonleptonic decays. The cross sections for the elementary two→four processes have not been calculated, and their complexity is such that they may not be evaluated in the foreseeable future. It is worthwhile to seek estimates of the four-jet cross sections, even if these are only reliable in restricted regions of phase space.



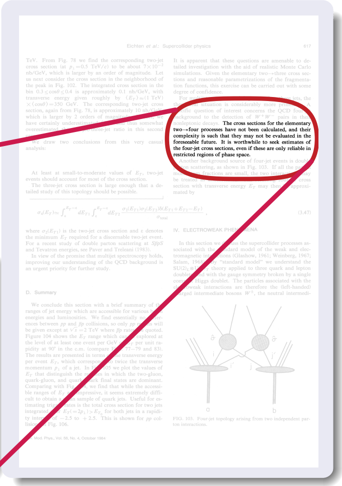
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms



For multijet events containing more than three jets, the theoretical situation is considerably more primitive. A specific question of interest concerns the QCD four-jet background to the detection of W^+W^- pairs in their nonleptonic decays. The cross sections for the elementary two→four processes have not been calculated, and their complexity is such that they may not be evaluated in the foreseeable future. It is worthwhile to seek estimates of the four-jet cross sections, even if these are only reliable in restricted regions of phase space.



Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available

Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers

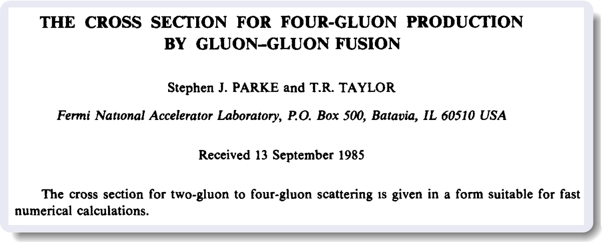
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers



THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.

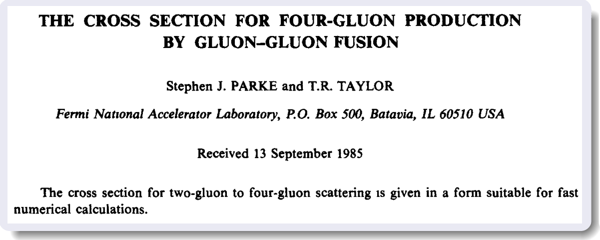
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.

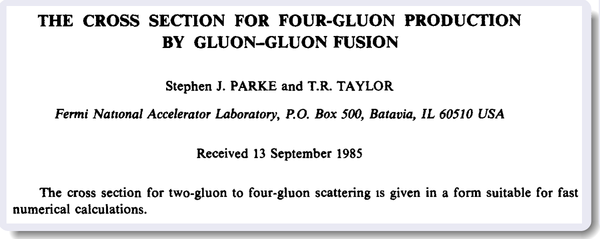
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



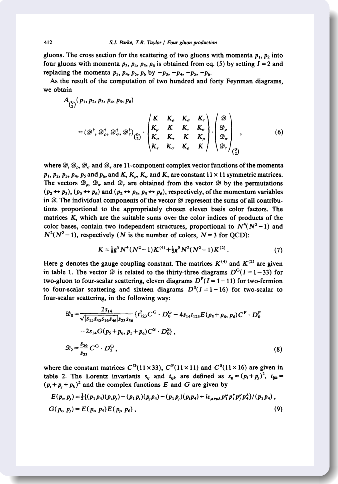
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



412 *S.J. Parke, T.R. Taylor / Four gluon production*

gluons. The cross section for the scattering of two gluons with momenta p_1, p_2 into four gluons with momenta p_3, p_4, p_5, p_6 is obtained from eq. (5) by setting $I=2$ and replacing the momenta p_3, p_4, p_5, p_6 by $-p_3, -p_4, -p_5, -p_6$.

As the result of the computation of two hundred and forty Feynman diagrams, we obtain

$$A_{\binom{0}{2}}(p_1, p_2, p_3, p_4, p_5, p_6)$$

$$=(\mathscr{D}^\dagger, \mathscr{D}^\dagger_\rho, \mathscr{D}^\dagger_\sigma, \mathscr{D}^\dagger_\tau)_{\binom{0}{2}} \cdot \begin{pmatrix} K & K_\rho & K_\sigma & K_\tau \\ K_\rho & K & K_\tau & K_\sigma \\ K_\sigma & K_\tau & K & K_\rho \\ K_\tau & K_\sigma & K_\rho & K \end{pmatrix} \cdot \begin{pmatrix} \mathscr{D} \\ \mathscr{D}_\rho \\ \mathscr{D}_\sigma \\ \mathscr{D}_\tau \end{pmatrix}_{\binom{0}{2}}, \qquad (6)$$

where $\mathscr{D}$, $\mathscr{D}_\rho$, $\mathscr{D}_\sigma$ and $\mathscr{D}_\tau$ are 11-component complex vector functions of the momenta p_1, p_2, p_3, p_4, p_5 and p_6, and K, K_ρ, K_σ and K_τ are constant 11×11 symmetric matrices. The vectors $\mathscr{D}_\rho$, $\mathscr{D}_\sigma$ and $\mathscr{D}_\tau$ are obtained from the vector $\mathscr{D}$ by the permutations $(p_2 \leftrightarrow p_3)$, $(p_5 \leftrightarrow p_6)$ and $(p_2 \leftrightarrow p_3, p_5 \leftrightarrow p_6)$, respectively, of the momentum variables in $\mathscr{D}$. The individual components of the vector $\mathscr{D}$ represent the sums of all contributions proportional to the appropriately chosen eleven basis color factors. The matrices K, which are the suitable sums over the color indices of products of the color bases, contain two independent structures, proportional to $N^4(N^2-1)$ and $N^2(N^2-1)$, respectively (N is the number of colors, $N=3$ for QCD):

$$K = \tfrac{1}{8}g^8 N^4(N^2-1)K^{(4)} + \tfrac{1}{2}g^8 N^2(N^2-1)K^{(2)}. \qquad (7)$$

Here g denotes the gauge coupling constant. The matrices $K^{(4)}$ and $K^{(2)}$ are given in table 1. The vector $\mathscr{D}$ is related to the thirty-three diagrams $D^{\mathrm{G}}(I=1-33)$ for two-gluon to four-scalar scattering, eleven diagrams $D^{\mathrm{F}}(I=1-11)$ for two-fermion to four-scalar scattering and sixteen diagrams $D^{\mathrm{S}}(I=1-16)$ for two-scalar to four-scalar scattering, in the following way:

$$\mathscr{D}_0 = \frac{2s_{14}}{\sqrt{|s_{15}s_{45}s_{16}s_{46}|}\,s_{23}s_{56}} \{t_{123}^2 C^{\mathrm{G}} \cdot D_0^{\mathrm{G}} - 4s_{14}t_{123}E(p_5+p_6, p_6)C^{\mathrm{F}} \cdot D_0^{\mathrm{F}}$$

$$-2s_{14}G(p_5+p_6, p_5+p_6)C^{\mathrm{S}} \cdot D_0^{\mathrm{S}}\},$$

$$\mathscr{D}_2 = \frac{s_{56}}{s_{23}} C^{\mathrm{G}} \cdot D_2^{\mathrm{G}}, \qquad (8)$$

where the constant matrices $C^{\mathrm{G}}(11\times 33)$, $C^{\mathrm{F}}(11\times 11)$ and $C^{\mathrm{S}}(11\times 16)$ are given in table 2. The Lorentz invariants s_{ij} and t_{ijk} are defined as $s_{ij}=(p_i+p_j)^2$, $t_{ijk}=(p_i+p_j+p_k)^2$ and the complex functions E and G are given by

$$E(p_i, p_j) = \tfrac{1}{2}\{(p_1p_4)(p_ip_j) - (p_1p_i)(p_jp_4) - (p_1p_j)(p_ip_4) + i\varepsilon_{\mu\nu\rho\lambda}p_1^\mu p_i^\nu p_j^\rho p_4^\lambda\}/(p_1p_4),$$

$$G(p_i, p_j) = E(p_i, p_5)E(p_j, p_6), \qquad (9)$$

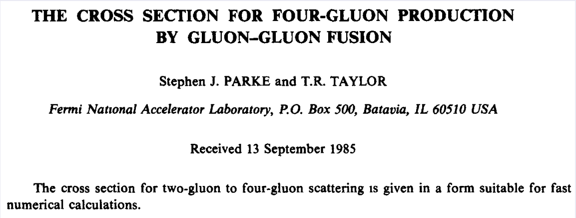
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



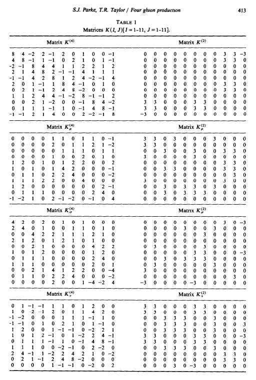
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



S.J. Parke, T.R. Taylor / Four gluon production 413

TABLE 1
Matrices $K(I, J)[I = 1\text{-}11, J = 1\text{-}11]$.

Matrix $K^{(4)}$											Matrix $K^{(2)}$										
8	4	−2	2	−1	2	0	1	0	0	−1	0	0	0	0	0	0	0	0	3	3	−3
4	8	−1	1	−1	0	2	1	0	1	−1	0	0	0	0	0	0	0	0	3	3	0
−2	−1	8	4	4	1	1	2	2	1	2	0	0	0	0	0	0	0	0	0	0	0
2	1	4	8	2	−1	−1	4	1	1	1	0	0	0	0	0	0	0	0	0	0	0
−1	−1	4	2	8	1	2	4	−2	−1	4	0	0	0	0	0	0	0	0	0	0	0
2	0	1	−1	1	8	4	−1	0	1	0	0	0	0	0	0	0	0	0	3	3	0
0	2	1	−1	2	4	8	−2	0	0	0	0	0	0	0	0	0	0	0	3	3	0
1	1	2	4	4	−1	−2	8	−1	−1	2	0	0	0	0	0	0	0	0	0	0	0
0	0	2	1	−2	0	0	−1	8	4	−2	3	3	0	0	0	3	3	0	0	0	0
0	1	1	1	−1	1	0	−1	4	8	−1	3	3	0	0	0	3	3	0	0	0	0
−1	−1	2	1	4	0	0	2	−2	−1	8	−3	0	0	0	0	0	0	0	0	0	0
Matrix $K_{\rho}^{(4)}$											Matrix $K_{\rho}^{(2)}$										
0	0	0	0	1	1	0	1	1	0	−1	3	3	0	3	0	0	0	3	0	0	0
0	0	0	0	2	0	1	1	2	1	−2	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	1	1	0	0	3	0	0	3	0	0	3	3	0
0	0	0	0	1	0	0	2	0	1	0	3	0	0	0	0	3	0	0	0	0	0
1	2	0	1	0	1	2	2	0	0	2	0	0	0	0	0	0	0	0	3	3	0
1	0	1	0	1	4	2	0	0	0	−1	0	0	3	3	0	0	0	0	3	3	0
0	1	1	0	2	2	4	0	0	0	−2	0	0	0	0	0	0	0	0	0	3	0
1	1	1	2	2	0	0	4	0	0	0	3	0	0	0	0	0	0	0	3	0	0
1	2	0	0	0	0	0	0	0	2	−1	0	0	3	0	3	3	0	3	0	0	0
0	1	1	1	0	0	0	0	2	4	0	0	0	3	0	3	3	3	0	0	0	0
−1	−2	1	0	2	−1	−2	0	−1	0	4	0	0	0	0	0	0	0	0	0	0	0
Matrix $K_{\sigma}^{(4)}$											Matrix $K_{\sigma}^{(2)}$										
4	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	−3
2	4	0	1	0	0	1	1	0	1	0	0	0	0	0	3	0	0	3	0	0	0
0	0	4	2	2	1	1	1	2	1	0	0	0	0	0	0	0	3	0	0	0	0
2	1	2	0	1	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	1	0	0	0	0	4	2	2	0	3	0	0	0	0	3	0	0	0	0
1	0	1	2	0	0	0	0	1	2	0	0	0	0	0	0	3	3	0	0	0	−3
0	1	1	1	0	0	0	0	2	4	0	0	0	3	0	3	3	3	0	0	0	0
1	1	1	0	0	0	0	0	2	0	1	0	3	0	0	0	0	0	3	0	0	0
0	0	2	1	4	1	2	2	0	0	−4	3	0	0	0	0	0	0	0	0	0	0
0	1	1	0	2	2	4	0	0	0	−2	0	0	0	0	0	0	0	0	0	3	0
0	0	0	0	2	0	0	1	−4	−2	4	−3	0	0	0	0	−3	0	0	0	0	0
Matrix $K_{\tau}^{(4)}$											Matrix $K_{\tau}^{(2)}$										
0	1	−1	−1	1	1	0	1	2	0	0	3	3	0	0	0	3	3	0	0	0	0
1	0	−2	−1	2	0	1	1	4	2	0	3	3	0	0	0	3	3	0	0	0	0
−1	−2	0	0	0	1	1	1	−1	1	0	0	0	3	3	3	0	0	3	0	0	0
−1	−1	0	1	0	2	1	0	1	−1	0	0	0	3	3	3	0	0	3	0	0	3
1	2	0	0	1	−1	−1	0	−2	2	1	0	0	3	3	3	0	0	3	0	0	0
1	0	1	2	−1	0	1	−2	2	4	−1	3	3	0	0	0	3	3	0	0	0	−3
0	1	1	1	−1	1	0	−1	4	8	−1	3	3	0	0	0	3	3	0	0	0	0
1	1	1	0	0	−2	−1	0	2	−2	0	0	0	3	3	3	0	0	3	0	0	0
2	4	−1	1	−2	2	4	2	1	0	−2	0	0	0	0	0	0	0	0	3	3	0
0	2	1	−1	2	4	8	−2	0	0	0	0	0	0	0	0	0	0	0	3	3	0
0	0	0	0	1	−1	−1	0	−2	0	2	0	0	0	3	0	−3	0	0	0	0	0

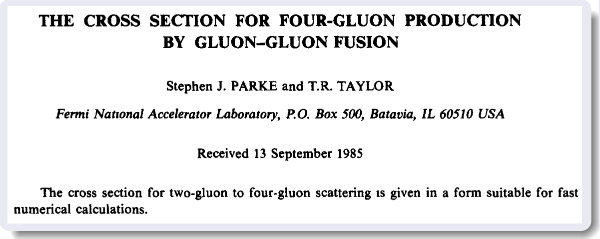
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



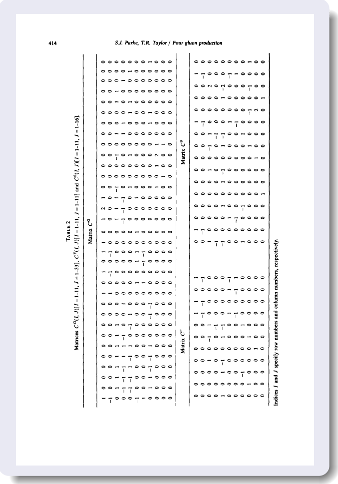
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



414 S.J. Parke, T.R. Taylor / Four gluon production

TABLE 2

Matrices $C^G(I,J)[I=1\text{-}11, J=1\text{-}33]$, $C^F(I,J)[I=1\text{-}11, J=1\text{-}11]$ and $C^S(I,J)[I=1\text{-}11, J=1\text{-}16]$.

Matrix C^G

Matrix C^F

Matrix C^S

Indices I and J specify row numbers and column numbers, respectively.

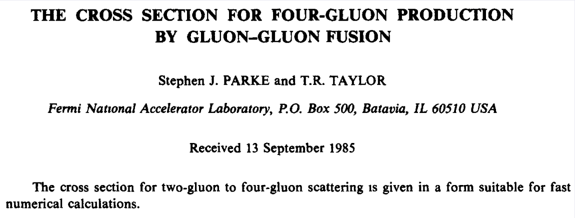
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



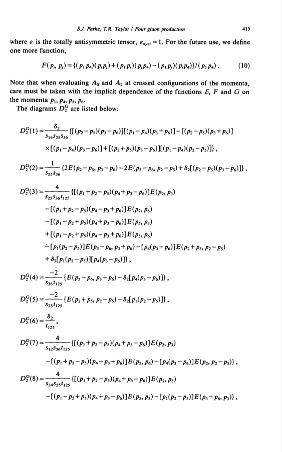
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



S.J. Parke, T.R. Taylor / Four gluon production 415

where ε is the totally antisymmetric tensor, $\varepsilon_{xyzt}=1$. For the future use, we define one more function,

$$F(p_i, p_j)=\{(p_1p_4)(p_ip_j)+(p_1p_i)(p_jp_4)-(p_1p_j)(p_ip_4)\}/(p_1p_4)\,. \qquad (10)$$

Note that when evaluating A_0 and A_2 at crossed configurations of the momenta, care must be taken with the implicit dependence of the functions E, F and G on the momenta p_1, p_4, p_5, p_6.

The diagrams D_2^{G} are listed below:

$$D_2^{\mathrm{G}}(1)=\frac{\delta_2}{s_{14}s_{25}s_{36}}\{[(p_2-p_5)(p_3-p_6)][(p_1-p_4)(p_3+p_6)]-[(p_2-p_5)(p_3+p_6)]$$
$$\times[(p_1-p_4)(p_3-p_6)]+[(p_2+p_5)(p_3-p_6)][(p_1-p_4)(p_2-p_5)]\}\,,$$

$$D_2^{\mathrm{G}}(2)=\frac{1}{s_{25}s_{36}}\{2E(p_2-p_5, p_3-p_6)-2E(p_3-p_6, p_2-p_5)+\delta_2[(p_2-p_5)(p_3-p_6)]\}\,,$$

$$D_2^{\mathrm{G}}(3)=\frac{4}{s_{25}s_{36}t_{125}}\{[(p_1+p_2-p_5)(p_4+p_3-p_6)]E(p_2, p_3)$$
$$-[(p_1+p_2-p_5)(p_4-p_3+p_6)]E(p_2, p_6)$$
$$-[(p_1-p_2+p_5)(p_4+p_3-p_6)]E(p_5, p_3)$$
$$+[(p_1-p_2+p_5)(p_4-p_3+p_6)]E(p_5, p_6)$$
$$-[p_1(p_2-p_5)]E(p_3-p_6, p_3+p_6)-[p_4(p_3-p_6)]E(p_2+p_5, p_2-p_5)$$
$$+\delta_2[p_1(p_2-p_5)][p_4(p_3-p_6)]\}\,,$$

$$D_2^{\mathrm{G}}(4)=\frac{-2}{s_{36}t_{125}}\{E(p_3-p_6, p_3+p_6)-\delta_2[p_4(p_3-p_6)]\}\,,$$

$$D_2^{\mathrm{G}}(5)=\frac{-2}{s_{25}t_{125}}\{E(p_2+p_5, p_2-p_5)-\delta_2[p_1(p_2-p_5)]\}\,,$$

$$D_2^{\mathrm{G}}(6)=\frac{\delta_2}{t_{125}}\,,$$

$$D_2^{\mathrm{G}}(7)=\frac{4}{s_{12}s_{36}t_{125}}\{[(p_1+p_2-p_5)(p_4+p_3-p_6)]E(p_2, p_3)$$
$$-[(p_1+p_2-p_5)(p_4-p_3+p_6)]E(p_2, p_6)-[p_4(p_3-p_6)]E(p_2, p_2-p_5)\}\,,$$

$$D_2^{\mathrm{G}}(8)=\frac{4}{s_{34}s_{25}t_{125}}\{[(p_1+p_2-p_5)(p_4+p_3-p_6)]E(p_2, p_3)$$
$$-[(p_1-p_2+p_5)(p_4+p_3-p_6)]E(p_5, p_3)-[p_1(p_2-p_5)]E(p_3-p_6, p_3)\}\,,$$

Supercomputer Computations in Quantum Chromodynamics

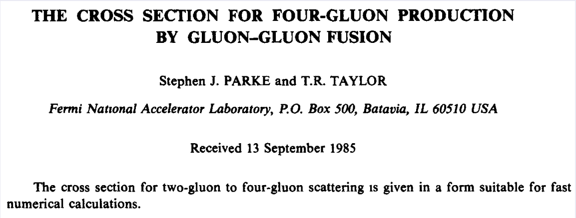
Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.

Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



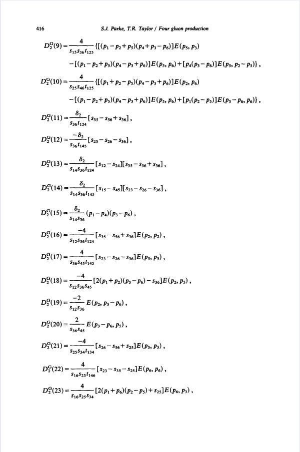
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION
BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



416 *S.J. Parke, T.R. Taylor / Four gluon production*

$$D_2^G(9) = \frac{4}{s_{15}s_{36}t_{125}}\{[(p_1 - p_2 + p_5)(p_4 + p_3 - p_6)]E(p_5, p_3)$$
$$-[(p_1 - p_2 + p_5)(p_4 - p_3 + p_6)]E(p_5, p_6) + [p_4(p_3 - p_6)]E(p_5, p_2 - p_5)\},$$

$$D_2^G(10) = \frac{4}{s_{25}s_{46}t_{125}}\{[(p_1 + p_2 - p_5)(p_4 - p_3 + p_6)]E(p_2, p_6)$$
$$-[(p_1 - p_2 + p_5)(p_4 - p_3 + p_6)]E(p_5, p_6) + [p_1(p_2 - p_5)]E(p_3 - p_6, p_6)\},$$

$$D_2^G(11) = \frac{\delta_2}{s_{36}t_{124}}[s_{35} - s_{56} + s_{36}],$$

$$D_2^G(12) = \frac{-\delta_2}{s_{36}t_{145}}[s_{23} - s_{26} - s_{36}],$$

$$D_2^G(13) = \frac{\delta_2}{s_{14}s_{36}t_{124}}[s_{12} - s_{24}][s_{35} - s_{56} + s_{36}],$$

$$D_2^G(14) = \frac{\delta_2}{s_{14}s_{36}t_{145}}[s_{15} - s_{45}][s_{23} - s_{26} - s_{36}],$$

$$D_2^G(15) = \frac{\delta_2}{s_{14}s_{36}}(p_1 - p_4)(p_3 - p_6),$$

$$D_2^G(16) = \frac{-4}{s_{12}s_{36}t_{124}}[s_{35} - s_{56} + s_{36}]E(p_2, p_2),$$

$$D_2^G(17) = \frac{4}{s_{36}s_{45}t_{145}}[s_{23} - s_{26} - s_{36}]E(p_5, p_5),$$

$$D_2^G(18) = \frac{-4}{s_{12}s_{36}s_{45}}[2(p_1 + p_2)(p_3 - p_6) - s_{36}]E(p_2, p_5),$$

$$D_2^G(19) = \frac{-2}{s_{12}s_{36}}E(p_2, p_3 - p_6),$$

$$D_2^G(20) = \frac{2}{s_{36}s_{45}}E(p_3 - p_6, p_5),$$

$$D_2^G(21) = \frac{-4}{s_{25}s_{34}t_{134}}[s_{26} - s_{56} + s_{25}]E(p_3, p_3),$$

$$D_2^G(22) = \frac{4}{s_{16}s_{25}t_{146}}[s_{23} - s_{35} - s_{25}]E(p_6, p_6),$$

$$D_2^G(23) = \frac{4}{s_{16}s_{25}s_{34}}[2(p_1 + p_6)(p_2 - p_5) + s_{25}]E(p_6, p_3),$$

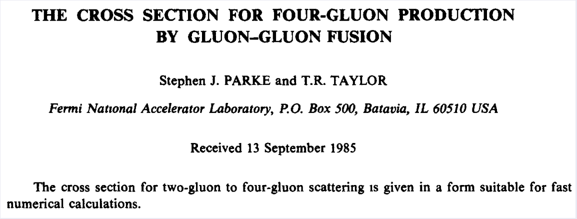
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



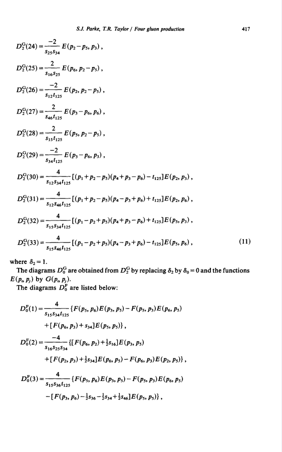
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



S.J. Parke, T.R. Taylor / Four gluon production 417

$$D_2^{\mathrm{G}}(24) = \frac{-2}{s_{25}s_{34}} E(p_2 - p_5, p_3)\,,$$

$$D_2^{\mathrm{G}}(25) = \frac{2}{s_{16}s_{25}} E(p_6, p_2 - p_5)\,,$$

$$D_2^{\mathrm{G}}(26) = \frac{-2}{s_{12}t_{125}} E(p_2, p_2 - p_5)\,,$$

$$D_2^{\mathrm{G}}(27) = \frac{2}{s_{46}t_{125}} E(p_3 - p_6, p_6)\,,$$

$$D_2^{\mathrm{G}}(28) = \frac{2}{s_{15}t_{125}} E(p_5, p_2 - p_5)\,,$$

$$D_2^{\mathrm{G}}(29) = \frac{-2}{s_{34}t_{125}} E(p_3 - p_6, p_3)\,,$$

$$D_2^{\mathrm{G}}(30) = \frac{4}{s_{12}s_{34}t_{125}} [(p_1 + p_2 - p_5)(p_4 + p_3 - p_6) - t_{125}] E(p_2, p_3)\,,$$

$$D_2^{\mathrm{G}}(31) = \frac{4}{s_{12}s_{46}t_{125}} [(p_1 + p_2 - p_5)(p_4 - p_3 + p_6) + t_{125}] E(p_2, p_6)\,,$$

$$D_2^{\mathrm{G}}(32) = \frac{4}{s_{15}s_{34}t_{125}} [(p_1 - p_2 + p_5)(p_4 + p_3 - p_6) + t_{125}] E(p_5, p_3)\,,$$

$$D_2^{\mathrm{G}}(33) = \frac{4}{s_{15}s_{46}t_{125}} [(p_1 - p_2 + p_5)(p_4 - p_3 + p_6) - t_{125}] E(p_5, p_6)\,, \tag{11}$$

where $\delta_2 = 1$.

The diagrams D_0^{G} are obtained from D_2^{G} by replacing δ_2 by $\delta_0 = 0$ and the functions $E(p_i, p_j)$ by $G(p_i, p_j)$.

The diagrams D_0^{F} are listed below:

$$D_0^{\mathrm{F}}(1) = \frac{4}{s_{15}s_{34}t_{125}} \{F(p_5, p_6)E(p_3, p_5) - F(p_5, p_3)E(p_6, p_5)$$
$$+ [F(p_6, p_3) + s_{34}]E(p_5, p_5)\}\,,$$

$$D_0^{\mathrm{F}}(2) = \frac{-4}{s_{16}s_{25}s_{34}} \{[F(p_6, p_2) + \tfrac{1}{2}s_{16}]E(p_3, p_5)$$
$$+ [F(p_2, p_3) + \tfrac{1}{2}s_{34}]E(p_6, p_5) - F(p_6, p_3)E(p_2, p_5)\}\,,$$

$$D_0^{\mathrm{F}}(3) = \frac{4}{s_{15}s_{36}t_{125}} \{F(p_5, p_6)E(p_3, p_5) - F(p_5, p_3)E(p_6, p_5)$$
$$- [F(p_3, p_6) - \tfrac{1}{2}s_{36} - \tfrac{1}{2}s_{34} + \tfrac{1}{2}s_{46}]E(p_5, p_5)\}\,,$$

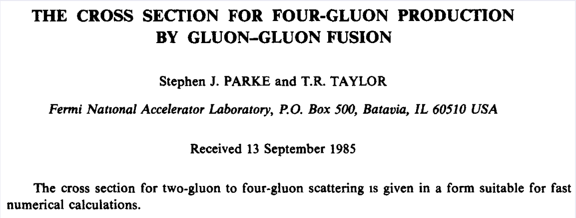
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$. Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



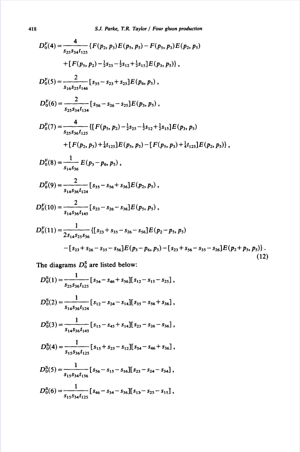
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



418 *S.J. Parke, T.R. Taylor / Four gluon production*

$$D_0^{\mathrm{F}}(4) = \frac{4}{s_{25}s_{34}t_{125}}\{F(p_2, p_3)E(p_5, p_5) - F(p_5, p_3)E(p_2, p_5)$$
$$+[F(p_5, p_2) - \tfrac{1}{2}s_{25} - \tfrac{1}{2}s_{12} + \tfrac{1}{2}s_{15}]E(p_3, p_5)\},$$

$$D_0^{\mathrm{F}}(5) = \frac{2}{s_{16}s_{25}t_{146}}[s_{35} - s_{23} + s_{25}]E(p_6, p_5),$$

$$D_0^{\mathrm{F}}(6) = \frac{2}{s_{25}s_{34}t_{134}}[s_{56} - s_{26} - s_{25}]E(p_3, p_5),$$

$$D_0^{\mathrm{F}}(7) = \frac{4}{s_{25}s_{36}t_{125}}\{[F(p_5, p_2) - \tfrac{1}{2}s_{25} - \tfrac{1}{2}s_{12} + \tfrac{1}{2}s_{15}]E(p_3, p_5)$$
$$+[F(p_2, p_3) + \tfrac{1}{4}t_{125}]E(p_5, p_5) - [F(p_5, p_3) + \tfrac{1}{4}t_{125}]E(p_2, p_5)\},$$

$$D_0^{\mathrm{F}}(8) = \frac{1}{s_{14}s_{36}}E(p_3 - p_6, p_5),$$

$$D_0^{\mathrm{F}}(9) = \frac{2}{s_{14}s_{36}t_{124}}[s_{35} - s_{56} + s_{36}]E(p_2, p_5),$$

$$D_0^{\mathrm{F}}(10) = \frac{2}{s_{14}s_{36}t_{145}}[s_{23} - s_{26} - s_{36}]E(p_5, p_5),$$

$$D_0^{\mathrm{F}}(11) = \frac{1}{2s_{14}s_{25}s_{36}}\{[s_{23} + s_{35} - s_{26} - s_{56}]E(p_2 - p_5, p_5)$$
$$-[s_{23} + s_{26} - s_{35} - s_{56}]E(p_3 - p_6, p_5) - [s_{23} + s_{56} - s_{35} - s_{26}]E(p_2 + p_5, p_5)\}. \qquad (12)$$

The diagrams D_0^{S} are listed below:

$$D_0^{\mathrm{S}}(1) = \frac{1}{s_{25}s_{36}t_{125}}[s_{34} - s_{46} + s_{36}][s_{12} - s_{15} - s_{25}],$$

$$D_0^{\mathrm{S}}(2) = \frac{1}{s_{14}s_{36}t_{124}}[s_{12} - s_{24} - s_{14}][s_{35} - s_{56} + s_{36}],$$

$$D_0^{\mathrm{S}}(3) = \frac{1}{s_{14}s_{36}t_{145}}[s_{15} - s_{45} + s_{14}][s_{23} - s_{26} - s_{36}],$$

$$D_0^{\mathrm{S}}(4) = \frac{1}{s_{15}s_{36}t_{125}}[s_{15} + s_{25} - s_{12}][s_{34} - s_{46} + s_{36}],$$

$$D_0^{\mathrm{S}}(5) = \frac{1}{s_{15}s_{34}t_{156}}[s_{56} - s_{15} - s_{16}][s_{23} - s_{24} - s_{34}],$$

$$D_0^{\mathrm{S}}(6) = \frac{1}{s_{15}s_{34}t_{125}}[s_{46} - s_{34} - s_{36}][s_{12} - s_{25} - s_{15}],$$

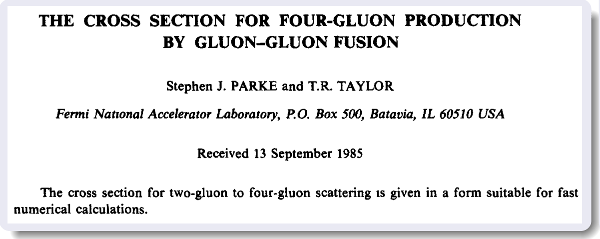
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



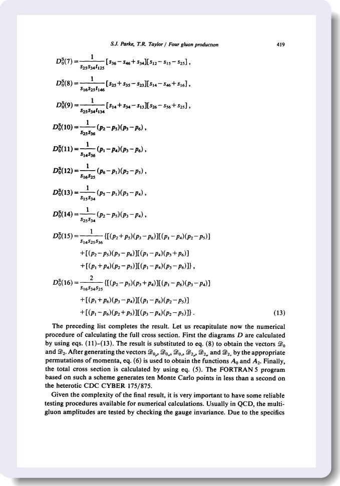
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



$$D_0^S(7)=\frac{1}{s_{25}s_{34}t_{125}}[s_{36}-s_{46}+s_{34}][s_{12}-s_{15}-s_{25}]\,,$$

$$D_0^S(8)=\frac{1}{s_{16}s_{25}t_{146}}[s_{25}+s_{35}-s_{23}][s_{14}-s_{46}+s_{16}]\,,$$

$$D_0^S(9)=\frac{1}{s_{25}s_{34}t_{134}}[s_{14}+s_{34}-s_{13}][s_{26}-s_{56}+s_{25}]\,,$$

$$D_0^S(10)=\frac{1}{s_{25}s_{36}}(p_2-p_5)(p_3-p_6)\,,$$

$$D_0^S(11)=\frac{1}{s_{14}s_{36}}(p_1-p_4)(p_3-p_6)\,,$$

$$D_0^S(12)=\frac{1}{s_{16}s_{25}}(p_6-p_1)(p_2-p_5)\,,$$

$$D_0^S(13)=\frac{1}{s_{15}s_{34}}(p_5-p_1)(p_3-p_4)\,,$$

$$D_0^S(14)=\frac{1}{s_{25}s_{34}}(p_2-p_5)(p_3-p_4)\,,$$

$$\begin{aligned}D_0^S(15)=\frac{1}{s_{14}s_{25}s_{36}}&\{[(p_2+p_5)(p_3-p_6)][(p_1-p_4)(p_2-p_5)]\\&+[(p_2-p_5)(p_3-p_6)][(p_1-p_4)(p_3+p_6)]\\&+[(p_1+p_4)(p_2-p_5)][(p_1-p_4)(p_3-p_6)]\}\,,\\D_0^S(16)=\frac{2}{s_{16}s_{34}s_{25}}&\{[(p_2-p_5)(p_3+p_4)][(p_1-p_6)(p_3-p_4)]\\&+[(p_1+p_6)(p_3-p_4)][(p_1-p_6)(p_2-p_5)]\\&+[(p_1-p_6)(p_2+p_5)][(p_3-p_4)(p_2-p_5)]\}\,.\end{aligned}\tag{13}$$

The preceding list completes the result. Let us recapitulate now the numerical procedure of calculating the full cross section. First the diagrams D are calculated by using eqs. (11)-(13). The result is substituted to eq. (8) to obtain the vectors $\mathcal{D}_0$ and $\mathcal{D}_2$. After generating the vectors $\mathcal{D}_{0_\rho}$, $\mathcal{D}_{0_\sigma}$, $\mathcal{D}_{0_\tau}$, $\mathcal{D}_{2_\rho}$, $\mathcal{D}_{2_\sigma}$ and $\mathcal{D}_{2_\tau}$ by the appropriate permutations of momenta, eq. (6) is used to obtain the functions A_0 and A_2. Finally, the total cross section is calculated by using eq. (5). The FORTRAN 5 program based on such a scheme generates ten Monte Carlo points in less than a second on the heterotic CDC CYBER 175/875.

Given the complexity of the final result, it is very important to have some reliable testing procedures available for numerical calculations. Usually in QCD, the multigluon amplitudes are tested by checking the gauge invariance. Due to the specifics

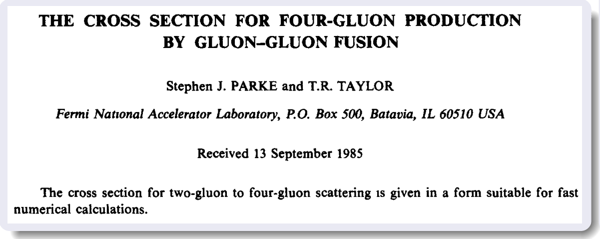
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



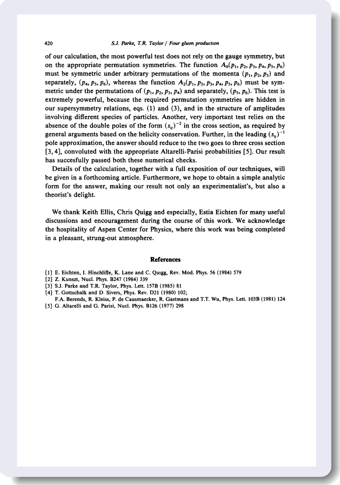
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



of our calculation, the most powerful test does not rely on the gauge symmetry, but on the appropriate permutation symmetries. The function $A_0(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under arbitrary permutations of the momenta (p_1, p_2, p_3) and separately, (p_4, p_5, p_6), whereas the function $A_2(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under the permutations of (p_1, p_2, p_3, p_4) and separately, (p_5, p_6). This test is extremely powerful, because the required permutation symmetries are hidden in our supersymmetry relations, eqs. (1) and (3), and in the structure of amplitudes involving different species of particles. Another, very important test relies on the absence of the double poles of the form $(s_{ij})^{-2}$ in the cross section, as required by general arguments based on the helicity conservation. Further, in the leading $(s_{ij})^{-1}$ pole approximation, the answer should reduce to the two goes to three cross section [3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed in a pleasant, strung-out atmosphere.

References

[1] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phys. 56 (1984) 579
[2] Z. Kunszt, Nucl. Phys. B247 (1984) 339
[3] S.J. Parke and T.R. Taylor, Phys. Lett. 157B (1985) 81
[4] T. Gottschalk and D. Sivers, Phys. Rev. D21 (1980) 102;
F.A. Berends, R. Kleiss, P. de Causmaecker, R. Gastmans and T.T. Wu, Phys. Lett. 103B (1981) 124
[5] G. Altarelli and G. Parisi, Nucl. Phys. B126 (1977) 298

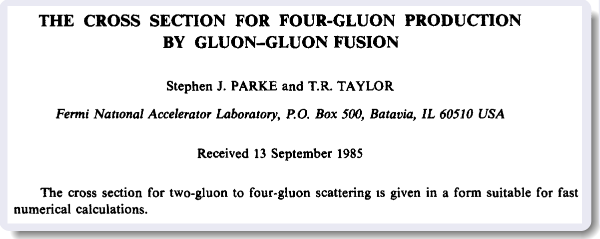
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



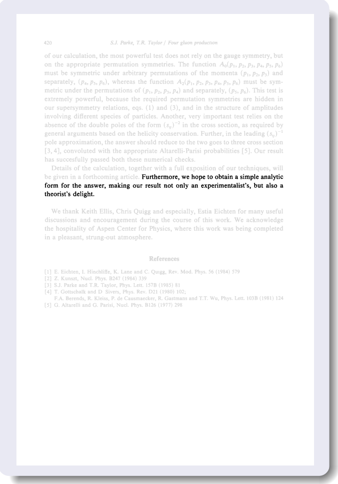
THE CROSS SECTION FOR FOUR-GLUON PRODUCTION BY GLUON–GLUON FUSION

Stephen J. PARKE and T.R. TAYLOR

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510 USA

Received 13 September 1985

The cross section for two-gluon to four-gluon scattering is given in a form suitable for fast numerical calculations.



420 *S.J. Parke, T.R. Taylor / Four gluon production*

of our calculation, the most powerful test does not rely on the gauge symmetry, but on the appropriate permutation symmetries. The function $A_0(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under arbitrary permutations of the momenta (p_1, p_2, p_3) and separately, (p_4, p_5, p_6), whereas the function $A_2(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under the permutations of (p_1, p_2, p_3, p_4) and separately, (p_5, p_6). This test is extremely powerful, because the required permutation symmetries are hidden in our supersymmetry relations, eqs. (1) and (3), and in the structure of amplitudes involving different species of particles. Another, very important test relies on the absence of the double poles of the form $(s_{ij})^{-2}$ in the cross section, as required by general arguments based on the helicity conservation. Further, in the leading $(s_{ij})^{-1}$ pole approximation, the answer should reduce to the two goes to three cross section [3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed in a pleasant, strung-out atmosphere.

References

[1] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phys. 56 (1984) 579
[2] Z. Kunszt, Nucl. Phys. B247 (1984) 339
[3] S.J. Parke and T.R. Taylor, Phys. Lett. 157B (1985) 81
[4] T. Gottschalk and D. Sivers, Phys. Rev. D21 (1980) 102;
F.A. Berends, R. Kleiss, P. de Causmaecker, R. Gastmans and T.T. Wu, Phys. Lett. 103B (1981) 124
[5] G. Altarelli and G. Parisi, Nucl. Phys. B126 (1977) 298

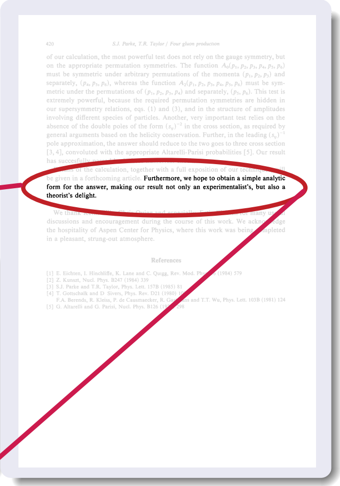
Supercomputer Computations in Quantum Chromodynamics

Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.
Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



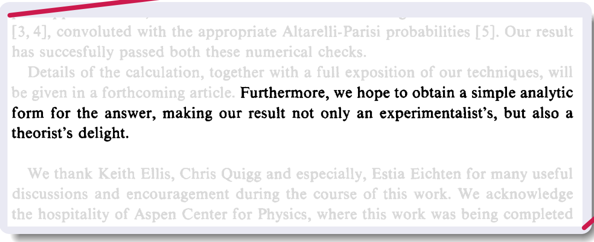
of our calculation, the most powerful test does not rely on the gauge symmetry, but on the appropriate permutation symmetries. The function $A_0(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under arbitrary permutations of the momenta (p_1, p_2, p_3) and separately, (p_4, p_5, p_6), whereas the function $A_2(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under the permutations of (p_1, p_2, p_3, p_4) and separately, (p_5, p_6). This test is extremely powerful, because the required permutation symmetries are hidden in our supersymmetry relations, eqs. (1) and (3), and in the structure of amplitudes involving different species of particles. Another, very important test relies on the absence of the double poles of the form $(s_{ij})^{-2}$ in the cross section, as required by general arguments based on the helicity conservation. Further, in the leading $(s_{ij})^{-1}$ pole approximation, the answer should reduce to the two goes to three cross section [3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed in a pleasant, strung-out atmosphere.

References

[1] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phys. [illegible]6 (1984) 579
[2] Z. Kunszt, Nucl. Phys. B247 (1984) 339
[3] S.J. Parke and T.R. Taylor, Phys. Lett. 157B (1985) 81
[4] T. Gottschalk and D. Sivers, Phys. Rev. D21 (1980) 10[illegible]
F.A. Berends, R. Kleiss, P. de Causmaecker, R. Gastmans and T.T. Wu, Phys. Lett. 103B (1981) 124
[5] G. Altarelli and G. Parisi, Nucl. Phys. B126 (19[illegible]) 298



[3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed

Supercomputer Computations in Quantum Chromodynamics

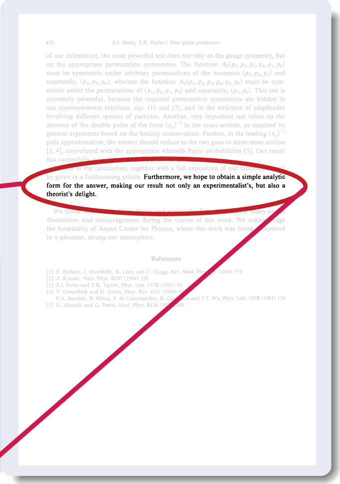
Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.

Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages


420 *S.J. Parke, T.R. Taylor / Four gluon production*

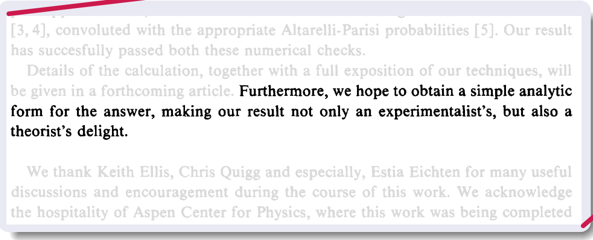
of our calculation, the most powerful test does not rely on the gauge symmetry, but on the appropriate permutation symmetries. The function $A_0(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under arbitrary permutations of the momenta (p_1, p_2, p_3) and separately, (p_4, p_5, p_6), whereas the function $A_2(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under the permutations of (p_1, p_2, p_3, p_4) and separately, (p_5, p_6). This test is extremely powerful, because the required permutation symmetries are hidden in our supersymmetry relations, eqs. (1) and (3), and in the structure of amplitudes involving different species of particles. Another, very important test relies on the absence of the double poles of the form $(s_{ij})^{-2}$ in the cross section, as required by general arguments based on the helicity conservation. Further, in the leading $(s_{ij})^{-1}$ pole approximation, the answer should reduce to the two goes to three cross section [3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed in a pleasant, strung-out atmosphere.

References

[1] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phy[illegible] (1984) 579
[2] Z. Kunszt, Nucl. Phys. B247 (1984) 339
[3] S.J. Parke and T.R. Taylor, Phys. Lett. 157B (1985) 81
[4] T. Gottschalk and D. Sivers, Phys. Rev. D21 (1980) 10[illegible]
F.A. Berends, R. Kleiss, P. de Causmaecker, R. Gas[illegible]ans and T.T. Wu, Phys. Lett. 103B (1981) 124
[5] G. Altarelli and G. Parisi, Nucl. Phys. B126 (19[illegible]) 298



[3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed

Supercomputer Computations in Quantum Chromodynamics

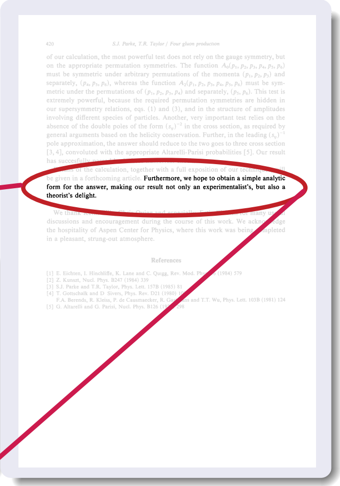
Consider the amplitude for two gluons to collide and produce four: $gg \to gggg$.

Before modern computers, this would have been computationally intractable

- 220 Feynman diagrams, thousands of terms

In 1985, Parke and Taylor took up the challenge

- using every theoretical tool available
- and the world's best supercomputers
- final formula fit into 8 pages



420 *S.J. Parke, T.R. Taylor / Four gluon production*

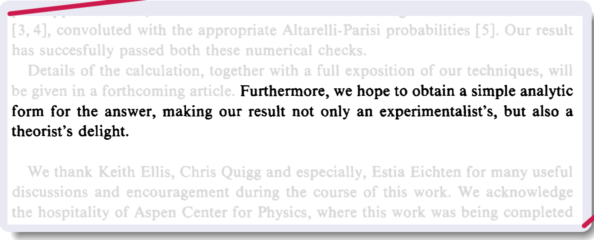
of our calculation, the most powerful test does not rely on the gauge symmetry, but on the appropriate permutation symmetries. The function $A_0(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under arbitrary permutations of the momenta (p_1, p_2, p_3) and separately, (p_4, p_5, p_6), whereas the function $A_2(p_1, p_2, p_3, p_4, p_5, p_6)$ must be symmetric under the permutations of (p_1, p_2, p_3, p_4) and separately, (p_5, p_6). This test is extremely powerful, because the required permutation symmetries are hidden in our supersymmetry relations, eqs. (1) and (3), and in the structure of amplitudes involving different species of particles. Another, very important test relies on the absence of the double poles of the form $(s_{ij})^{-2}$ in the cross section, as required by general arguments based on the helicity conservation. Further, in the leading $(s_{ij})^{-1}$ pole approximation, the answer should reduce to the two goes to three cross section [3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed in a pleasant, strung-out atmosphere.

References

[1] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phys. [illegible] (1984) 579
[2] Z. Kunszt, Nucl. Phys. B247 (1984) 339
[3] S.J. Parke and T.R. Taylor, Phys. Lett. 157B (1985) 81
[4] T. Gottschalk and D. Sivers, Phys. Rev. D21 (1980) [illegible]
F.A. Berends, R. Kleiss, P. de Causmaecker, R. [illegible] and T.T. Wu, Phys. Lett. 103B (1981) 124
[5] G. Altarelli and G. Parisi, Nucl. Phys. B126 ([illegible]) 298



[3, 4], convoluted with the appropriate Altarelli-Parisi probabilities [5]. Our result has succesfully passed both these numerical checks.

Details of the calculation, together with a full exposition of our techniques, will be given in a forthcoming article. **Furthermore, we hope to obtain a simple analytic form for the answer, making our result not only an experimentalist's, but also a theorist's delight.**

We thank Keith Ellis, Chris Quigg and especially, Estia Eichten for many useful discussions and encouragement during the course of this work. We acknowledge the hospitality of Aspen Center for Physics, where this work was being completed

The Discovery of Incredible, Unanticipated Simplicity

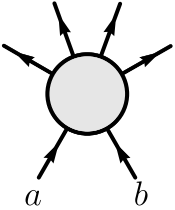
They soon **guessed** a simplified form of the amplitude

The Discovery of Incredible, Unanticipated Simplicity

They soon **guessed** a simplified form of the amplitude (checked numerically):

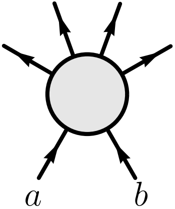
The Discovery of Incredible, Unanticipated Simplicity

They soon **guessed** a simplified form of the amplitude (checked numerically):



The Discovery of Incredible, Unanticipated Simplicity

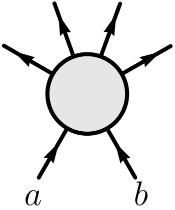
They soon **guessed** a simplified form of the amplitude (checked numerically):



$$= \frac{\langle a\, b\rangle^4}{\langle 1\, 2\rangle\langle 2\, 3\rangle\langle 3\, 4\rangle\langle 4\, 5\rangle\langle 5\, 6\rangle\langle 6\, 1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

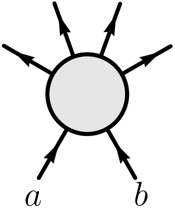
They soon **guessed** a simplified form of the amplitude (checked numerically):



$$= \frac{\langle a\, b\rangle^4}{\langle 1\, 2\rangle\langle 2\, 3\rangle\langle 3\, 4\rangle\langle 4\, 5\rangle\langle 5\, 6\rangle\langle 6\, 1\rangle}\, \delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

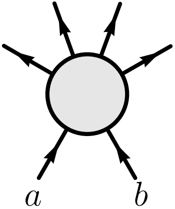
They soon **guessed** a simplified form of the amplitude (checked numerically):



$$= \frac{\langle a\, b\rangle^4}{\langle 1\, 2\rangle\langle 2\, 3\rangle\langle 3\, 4\rangle\langle 4\, 5\rangle\langle 5\, 6\rangle\langle 6\, 1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

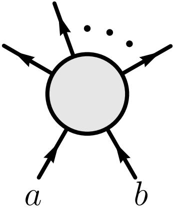
They soon **guessed** a simplified form of the amplitude (checked numerically):
—which naturally suggested the amplitude for **all** multiplicity!



$$= \frac{\langle a\, b\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,4\rangle\langle 4\,5\rangle\langle 5\,6\rangle\langle 6\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

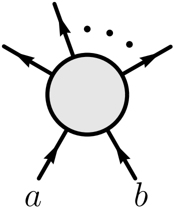
They soon **guessed** a simplified form of the amplitude (checked numerically):
—which naturally suggested the amplitude for **all** multiplicity!



$$= \frac{\langle a\, b\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,4\rangle\langle 4\,5\rangle\,\cdots\,\langle n\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

They soon **guessed** a simplified form of the amplitude (checked numerically):
—which naturally suggested the amplitude for **all** multiplicity!

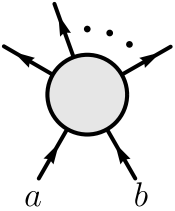


$$= \frac{\langle a\, b\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,4\rangle\langle 4\,5\rangle\,\cdots\,\langle n\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

The Discovery of Incredible, Unanticipated Simplicity

They soon **guessed** a simplified form of the amplitude (checked numerically):

—which naturally suggested the amplitude for **all** multiplicity!



$$= \frac{\langle a\,b\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,4\rangle\langle 4\,5\rangle\,\cdots\,\langle n\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

On What *Data* Does a Scattering Amplitude Depend?

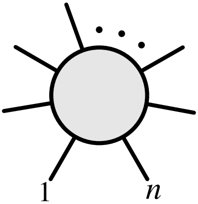
A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$$\mathcal{A}_n$$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

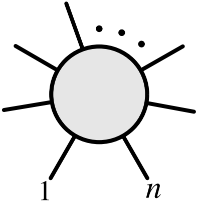
$$\mathcal{A}_n \equiv$$



On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

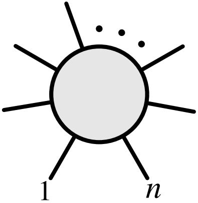
$\mathcal{A}_n \equiv$



On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$

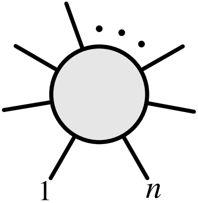


Physical data for the a^{th} particle: $|a\rangle$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$



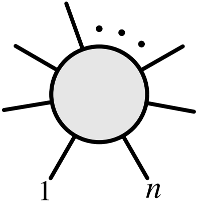
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$

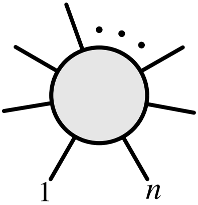


Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum
- m_a mass

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

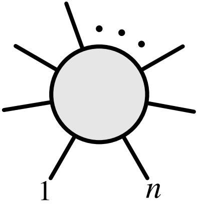
Physical data for the a^{th} particle: $|a\rangle$

- p_a^{μ} momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- m_a mass

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$



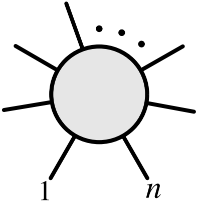
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$



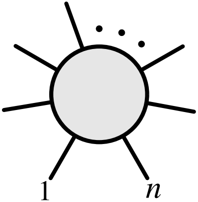
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$

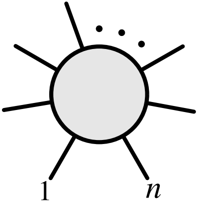


Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a \in \{\sigma_a, \ldots, -\sigma_a\}$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

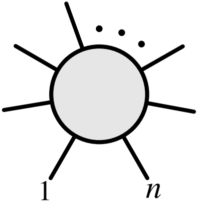
$\mathcal{A}_n \equiv$ 

Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a \in \{\sigma_a, \ldots, -\sigma_a\}$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

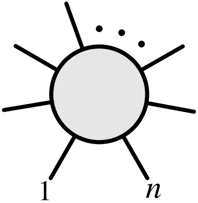
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$

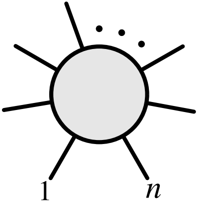


Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

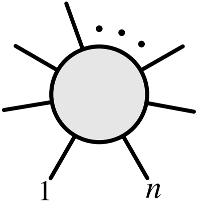
$\mathcal{A}_n \equiv$ 

Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

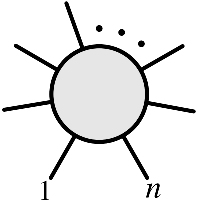
$\mathcal{A}_n \equiv$ 

Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

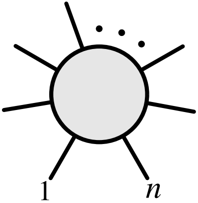
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, ...)

Although a Lagrangian formalism requires that we use polarization tensors,

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

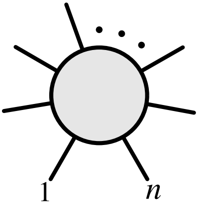
Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$


Physical data for the a^{th} particle: $|a\rangle$

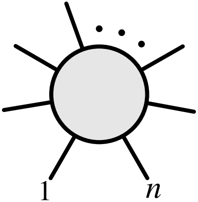
- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$



Physical data for the a^{th} particle: $|a\rangle$

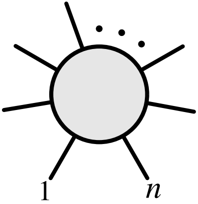
- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

$$\epsilon_a^\mu \sim \epsilon_a^\mu + \alpha(p_a) p_a^\mu$$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

Physical data for the a^{th} particle: $|a\rangle$

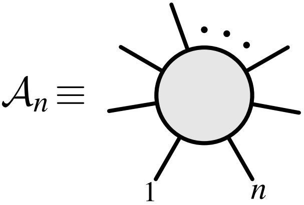
- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, ...)

Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

$$\epsilon_a^\mu \sim \epsilon_a^\mu + \alpha(p_a) p_a^\mu$$

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.



Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

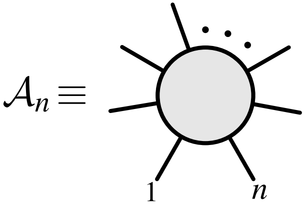
Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

$$\epsilon_a^\mu \sim \epsilon_a^\mu + \alpha(p_a) p_a^\mu$$

Such *unphysical baggage* is almost certainly responsible for the incredible obfuscation of simplicity in the traditional approach to quantum field theory.

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.



Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

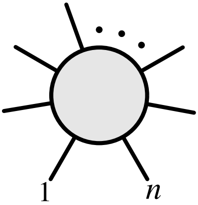
Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

$$\epsilon_a^\mu \sim \epsilon_a^\mu + \alpha(p_a)p_a^\mu$$

Such *unphysical baggage* is almost certainly responsible for the incredible obfuscation of simplicity in the traditional approach to quantum field theory.

On What *Data* Does a Scattering Amplitude Depend?

A scattering amplitude, $\mathcal{A}_n$, can be a generally complicated(?) function of all the *physically observable data* describing each of the particles involved.

$\mathcal{A}_n \equiv$ 

Physical data for the a^{th} particle: $|a\rangle$

- p_a^μ momentum, *on-shell*: $p_a^2 - m_a^2 = 0$
- σ_a spin, helicity $h_a = \pm\sigma_a \quad (m_a = 0)$
- q_a all the *non-kinematical* quantum numbers of a (color, flavor, . . .)

Although a Lagrangian formalism requires that we use polarization tensors, it is *impossible* to continuously define polarizations for each helicity state without introducing *unobservable* (**gauge**) redundancy—*e.g.* for $\sigma_a = 1$:

$$\epsilon_a^\mu \sim \epsilon_a^\mu + \alpha(p_a) p_a^\mu$$

Such *unphysical baggage* is almost certainly responsible for the incredible obfuscation of simplicity in the traditional approach to quantum field theory.

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

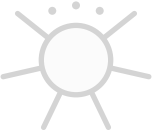
We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

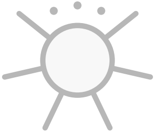
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



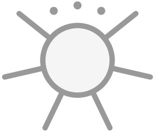
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



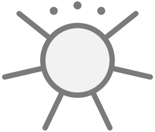
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



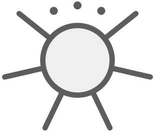
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



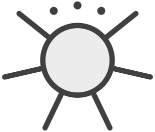
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



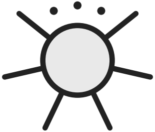
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



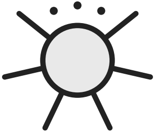
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



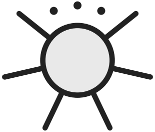
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



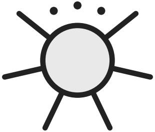
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



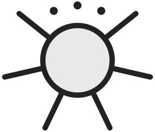
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



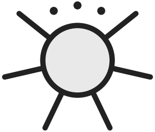
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



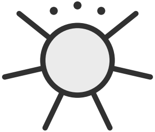
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



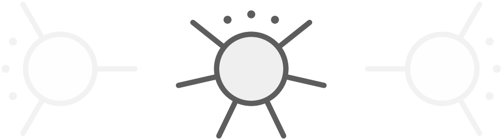
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



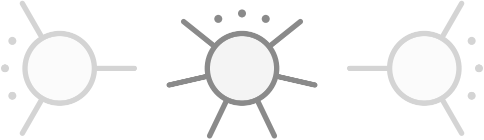
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



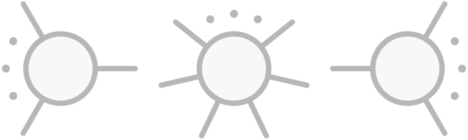
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



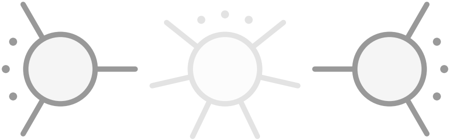
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



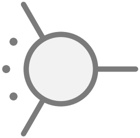
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



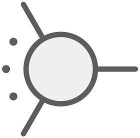
Broadening the Class of Physically Meaningful Functions

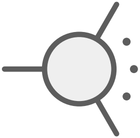
We are interested in the class of functions involving **only** observable quantities



Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities





Broadening the Class of Physically Meaningful Functions

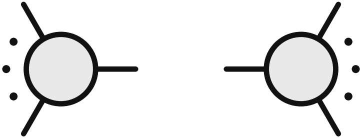
We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

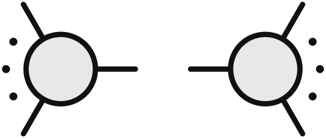


Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

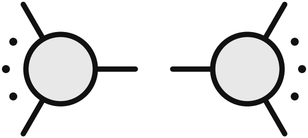
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



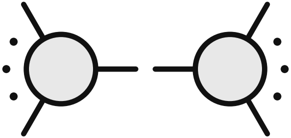
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



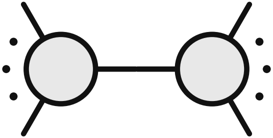
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



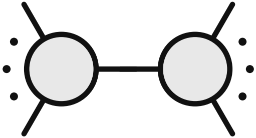
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



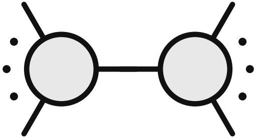
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



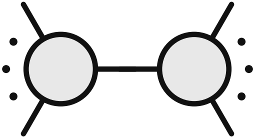
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



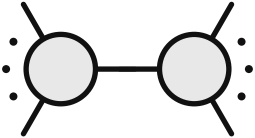
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



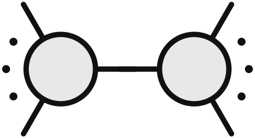
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



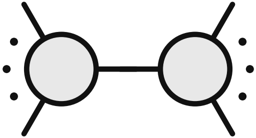
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



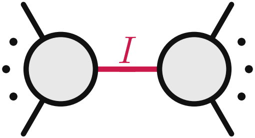
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



Broadening the Class of Physically Meaningful Functions

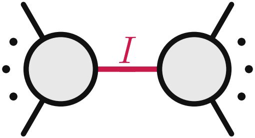
We are interested in the class of functions involving **only** observable quantities



Internal Particles:

Broadening the Class of Physically Meaningful Functions

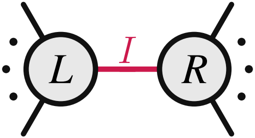
We are interested in the class of functions involving **only** observable quantities



Internal Particles: locality dictates that we multiply each amplitude,

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

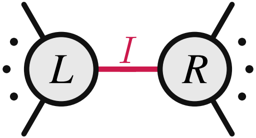


Internal Particles: locality dictates that we multiply each amplitude,

$$\mathcal{A}_L(\ldots, I)\times\mathcal{A}_R(I,\ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

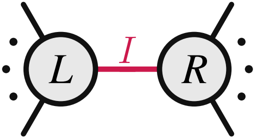


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states

$$\mathcal{A}_L(\ldots,I)\times\mathcal{A}_R(I,\ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

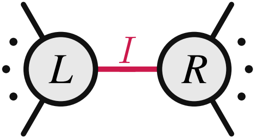


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space ("LIPS") for each particle I,

$$\mathcal{A}_L(\ldots, I) \times \mathcal{A}_R(I, \ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

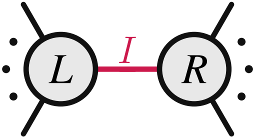


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space ("LIPS") for each particle I,

$$\int d^{d-1}\mathrm{LIPS}_I \; \mathcal{A}_L(\ldots,I)\times\mathcal{A}_R(I,\ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

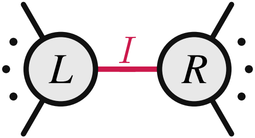


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space ("LIPS") for each particle I, and summing over the possible states

$$\int d^{d-1}\text{LIPS}_I \; \mathcal{A}_L(\ldots, I) \times \mathcal{A}_R(I, \ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

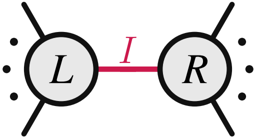


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space (“LIPS”) for each particle I, and summing over the possible states

$$\sum_{\text{states } I} \int d^{d-1}\text{LIPS}_I \; \mathcal{A}_L(\ldots, I) \times \mathcal{A}_R(I, \ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

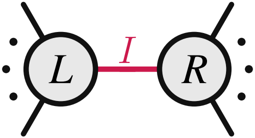


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space ("LIPS") for each particle I, and summing over the possible states (helicities, masses, colours, etc.).

$$\sum_{\text{states } I} \int d^{d-1}\text{LIPS}_I \; \mathcal{A}_L(\ldots, I) \times \mathcal{A}_R(I, \ldots)$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

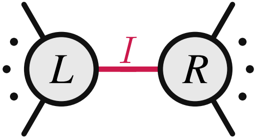


Internal Particles: locality dictates that we multiply each amplitude, and unitarity dictates that we marginalize over unobserved states—integrating over the Lorentz-invariant phase space (“LIPS”) for each particle I, and summing over the possible states (helicities, masses, colours, etc.).

$$\sum_{\text{states } I} \int d^{d-1}\text{LIPS}_{I}\; \mathcal{A}_L(\ldots,I)\times\mathcal{A}_R(I,\ldots)$$

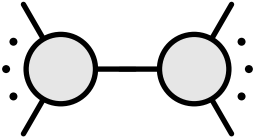
Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



Broadening the Class of Physically Meaningful Functions

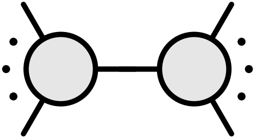
We are interested in the class of functions involving **only** observable quantities



On-Shell Functions:

Broadening the Class of Physically Meaningful Functions

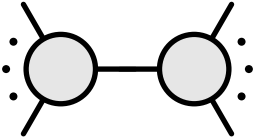
We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

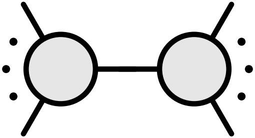


On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i\in I}\Bigg(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\text{LIPS}_i\Bigg)\prod_v \mathcal{A}_v$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities

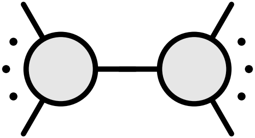


On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i \in I} \Bigg(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\mathrm{LIPS}_i \Bigg) \prod_v \mathcal{A}_v$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



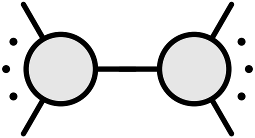
On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i\in I}\Bigg(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Bigg)\prod_v \mathcal{A}_v$$

Counting Constraints:

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

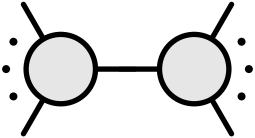
$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

Counting Constraints:

n_δ

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

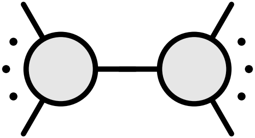
$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

Counting Constraints:

$$n_\delta \equiv d\times n_V$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

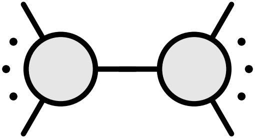
$$f_\Gamma \equiv \prod_{i \in I} \Big(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v$$

Counting Constraints:

$$n_\delta \equiv d \times n_V - (d-1) \times n_I$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

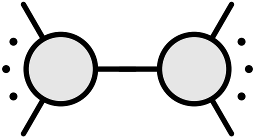
$$f_\Gamma \equiv \prod_{i \in I} \Big(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d \times n_V - (d-1) \times n_I - d$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

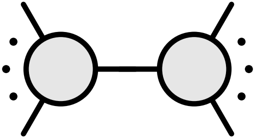
$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d\times n_V-(d{-}1)\times n_I-d \;=\text{number of excess }\delta\text{-functions}$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i \in I} \Bigg(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\mathrm{LIPS}_i \Bigg) \prod_v \mathcal{A}_v$$

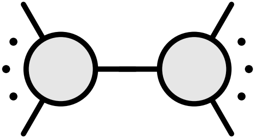
Counting Constraints:

$$\widehat{n}_\delta \equiv d \times n_V - (d{-}1) \times n_I - d \;=\; \text{number of excess } \delta\text{-functions}$$

$$(= \text{minus number of remaining integrations})$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

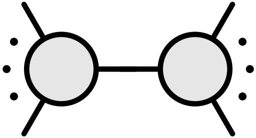
Counting Constraints:

$$\widehat{n}_\delta \equiv d\times n_V-(d{-}1)\times n_I-d \;=\text{number of excess }\delta\text{-functions}$$

$$(=\text{minus number of remaining integrations})$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i \in I} \Bigg(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\text{LIPS}_i \Bigg) \prod_v \mathcal{A}_v$$

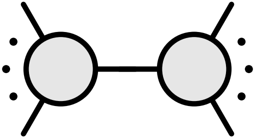
Counting Constraints:

$$\widehat{n}_\delta \equiv d \times n_V - (d{-}1) \times n_I - d \; = \text{number of excess } \delta\text{-functions}$$

$$(= \text{minus number of remaining integrations})$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

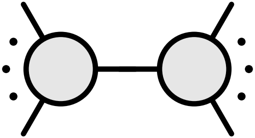
$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d\times n_V - (d{-}1)\times n_I - d \;=0$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

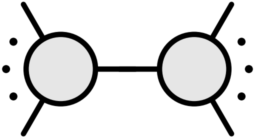
$$f_\Gamma \equiv \prod_{i\in I}\Bigg(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Bigg)\prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d\times n_V - (d\text{–}1)\times n_I - d \;=0 \quad\Longrightarrow\quad \text{ordinary (rational) function}$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i \in I$, forming a graph Γ called an "on-shell diagram".

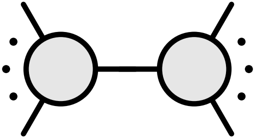
$$f_\Gamma \equiv \prod_{i \in I} \Big(\sum_{\substack{h_i, q_i, \\ m_i, \cdots}} \int d^{d-1}\mathrm{LIPS}_i \Big) \prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d \times n_V - (d{-}1) \times n_I - d \begin{array}{lcl} > 0 & \Rightarrow & (\widehat{n}_\delta) \text{ kinematical constraints} \\ = 0 & \Rightarrow & \text{ordinary (rational) function} \\ < 0 & \Rightarrow & (-\widehat{n}_\delta) \text{ non-trivial integrations} \end{array}$$

Broadening the Class of Physically Meaningful Functions

We are interested in the class of functions involving **only** observable quantities



On-Shell Functions: networks of amplitudes, $\mathcal{A}_v$, connected by any number of internal particles, $i\in I$, forming a graph Γ called an "on-shell diagram".

$$f_\Gamma \equiv \prod_{i\in I}\Big(\sum_{\substack{h_i,q_i,\\ m_i,\cdots}}\int d^{d-1}\mathrm{LIPS}_i\Big)\prod_v \mathcal{A}_v$$

Counting Constraints:

$$\widehat{n}_\delta \equiv d\times n_V-(d{-}1)\times n_I-d \begin{array}{lcl} >0 & \Rightarrow & (\widehat{n}_\delta)\text{ kinematical constraints}\\ =0 & \Rightarrow & \text{ordinary (rational) function}\\ <0 & \Rightarrow & (-\widehat{n}_\delta)\text{ non-trivial integrations}\end{array}$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

p_a^{μ}

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}}$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}}$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu}\mapsto p_a^{\alpha\dot{\alpha}}\equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}}=\begin{pmatrix}p_a^0+p_a^3 & p_a^1-ip_a^2\\ p_a^1+ip_a^2 & p_a^0-p_a^3\end{pmatrix}$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = m_a^2$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 =$ 0, for massless particles.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = 0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot{\alpha}}$ as an outer product of 2-vectors.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot\alpha} \equiv p_a^\mu \sigma_\mu^{\alpha\dot\alpha} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot\alpha}$$

- Notice that $\det\left(p_a^{\alpha\dot\alpha}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = 0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot\alpha}$ as an outer product of 2-vectors.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu}\mapsto p_a^{\alpha\dot{\alpha}}\equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}}=\begin{pmatrix}p_a^0+p_a^3 & p_a^1-ip_a^2\\ p_a^1+ip_a^2 & p_a^0-p_a^3\end{pmatrix}\equiv\lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right)=\left(p_a^0\right)^2-\left(p_a^1\right)^2-\left(p_a^2\right)^2-\left(p_a^3\right)^2=0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot{\alpha}}$ as an outer product of 2-vectors.

- When p_a is *real* $\left(p_a\in\mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}}=(p_a^{\alpha\dot{\alpha}})^{\dagger}$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot\alpha} \equiv p_a^\mu \sigma_\mu^{\alpha\dot\alpha} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot\alpha}$$

- Notice that $\det\left(p_a^{\alpha\dot\alpha}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = 0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot\alpha}$ as an outer product of 2-vectors.

- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot\alpha} = (p_a^{\alpha\dot\alpha})^\dagger$, which implies that $\left(\lambda_a^\alpha\right)^* = \pm\widetilde{\lambda}_a^{\dot\alpha}$.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = 0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot{\alpha}}$ as an outer product of 2-vectors.

- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}} = (p_a^{\alpha\dot{\alpha}})^\dagger$, which implies that $\left(\lambda_a^\alpha\right)^* = \pm \widetilde{\lambda}_a^{\dot{\alpha}}$. (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}$$

- Notice that $\det\left(p_a^{\alpha\dot{\alpha}}\right) = \left(p_a^0\right)^2 - \left(p_a^1\right)^2 - \left(p_a^2\right)^2 - \left(p_a^3\right)^2 = 0$, for massless particles. This can be made *manifest* by writing $p_a^{\alpha\dot{\alpha}}$ as an outer product of 2-vectors.
- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}} = (p_a^{\alpha\dot{\alpha}})^\dagger$, which implies that $\left(\lambda_a^\alpha\right)^* = \pm \widetilde{\lambda}_a^{\dot{\alpha}}$. (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$

- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}} = (p_a^{\alpha\dot{\alpha}})^\dagger$, which implies that $\left(\lambda_a^\alpha\right)^* = \pm \widetilde{\lambda}_a^{\dot{\alpha}}$.
  (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**.

- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}} = (p_a^{\alpha\dot{\alpha}})^{\dagger}$, which implies that $\left(\lambda_a^{\alpha}\right)^{*} = \pm\widetilde{\lambda}_a^{\dot{\alpha}}$. (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu}\mapsto p_a^{\alpha\dot{\alpha}}\equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}}=\begin{pmatrix}p_a^0+p_a^3 & p_a^1-ip_a^2\\ p_a^1+ip_a^2 & p_a^0-p_a^3\end{pmatrix}\equiv\lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a,\widetilde{\lambda}_a)\mapsto(t_a\lambda_a,t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

- When p_a is *real* $\left(p_a\in\mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}}=(p_a^{\alpha\dot{\alpha}})^{\dagger}$, which implies that $\left(\lambda_a^{\alpha}\right)^{*}=\pm\widetilde{\lambda}_a^{\dot{\alpha}}$. (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- When p_a is *real* $\left(p_a \in \mathbb{R}^{3,1}\right)$, $p_a^{\alpha\dot{\alpha}} = (p_a^{\alpha\dot{\alpha}})^\dagger$, which implies that $\left(\lambda_a^\alpha\right)^* = \pm \widetilde{\lambda}_a^{\dot{\alpha}}$. (but allowing for complex momenta, λ_a and $\widetilde{\lambda}_a$ become independent.)

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a}|a\rangle^{h_a}$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a}|a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively.

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a}|a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \qquad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b)$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0+p_a^3 & p_a^1-ip_a^2 \\ p_a^1+ip_a^2 & p_a^0-p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}} \Leftrightarrow \text{“} a\rangle[a \text{”}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \qquad\qquad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b)$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - i p_a^2 \\ p_a^1 + i p_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}} \Leftrightarrow \text{``} a\rangle[a \text{''}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \equiv \langle a\, b \rangle \quad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b) \equiv [a\, b]$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot\alpha} \equiv p_a^\mu \sigma_\mu^{\alpha\dot\alpha} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot\alpha} \Leftrightarrow \text{“} a\rangle[a \text{”}$$

- $p_a^{\alpha\dot\alpha}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\epsilon_{\alpha\beta} \lambda_a^\alpha \lambda_b^\beta \equiv \langle a\, b\rangle \qquad \epsilon_{\dot\alpha\dot\beta} \widetilde{\lambda}_a^{\dot\alpha} \widetilde{\lambda}_b^{\dot\beta} \equiv [a\, b]$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^\mu \sigma_\mu^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0+p_a^3 & p_a^1-ip_a^2 \\ p_a^1+ip_a^2 & p_a^0-p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}} \Leftrightarrow \text{“}a\rangle[a\text{”}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \equiv \langle a\, b\rangle \quad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b) \equiv [a\, b]$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^\mu \mapsto p_a^{\alpha\dot\alpha} \equiv p_a^\mu \sigma_\mu^{\alpha\dot\alpha} = \begin{pmatrix} p_a^0 + p_a^3 & p_a^1 - ip_a^2 \\ p_a^1 + ip_a^2 & p_a^0 - p_a^3 \end{pmatrix} \equiv \lambda_a^\alpha \widetilde{\lambda}_a^{\dot\alpha} \Leftrightarrow \text{“} a\rangle [a \text{”}$$

- $p_a^{\alpha\dot\alpha}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a \lambda_a, t_a^{-1} \widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a} |a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \equiv \langle a\, b\rangle \quad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b) \equiv [a\, b]$$

Making *Masslessness* Manifest: Spinor-Helicity Variables

To avoid *constraining* each particle's momentum to be null, van der Waerden introduced (in 1929!) **spinor-helicity** variables to make this always trivial.

$$p_a^{\mu} \mapsto p_a^{\alpha\dot{\alpha}} \equiv p_a^{\mu}\sigma_{\mu}^{\alpha\dot{\alpha}} = \begin{pmatrix} p_a^0+p_a^3 & p_a^1-ip_a^2 \\ p_a^1+ip_a^2 & p_a^0-p_a^3 \end{pmatrix} \equiv \lambda_a^{\alpha}\widetilde{\lambda}_a^{\dot{\alpha}} \Leftrightarrow \text{“} a\rangle[a \text{”}$$

- $p_a^{\alpha\dot{\alpha}}$ is unchanged by $(\lambda_a, \widetilde{\lambda}_a) \mapsto (t_a\lambda_a, t_a^{-1}\widetilde{\lambda}_a)$—the action of the **little group**. Under little group transformations, wave functions transform according to:

$$|a\rangle^{h_a} \mapsto t_a^{-2h_a}|a\rangle^{h_a}$$

- The (local) Lorentz group, $SL(2)_L \times SL(2)_R$, acts on λ_a and $\widetilde{\lambda}_a$, respectively. Therefore, Lorentz-invariants must be constructed using the determinants:

$$\det(\lambda_a, \lambda_b) \equiv \langle a\, b\rangle \quad \det(\widetilde{\lambda}_a, \widetilde{\lambda}_b) \equiv [a\, b]$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\begin{pmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1 & \cdots & \lambda_n^1\\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \cdots & \lambda_n^2\end{pmatrix} \qquad \begin{pmatrix}\widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} & \cdots & \widetilde{\lambda}_n^{\dot{1}}\\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} & \cdots & \widetilde{\lambda}_n^{\dot{2}}\end{pmatrix}$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 & \cdots & \lambda_n^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \ldots & \lambda_n^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} & \cdots & \widetilde{\lambda}_n^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} & \ldots & \widetilde{\lambda}_n^{\dot{2}} \end{pmatrix}$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 & \cdots & \lambda_n^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \ldots & \lambda_n^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} & \cdots & \widetilde{\lambda}_n^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} & \ldots & \widetilde{\lambda}_n^{\dot{2}} \end{pmatrix}$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\begin{pmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1 & \ldots & \lambda_n^1\\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \ldots & \lambda_n^2\end{pmatrix}\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\qquad \widetilde{\lambda}\equiv\begin{pmatrix}\widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} & \ldots & \widetilde{\lambda}_n^{\dot{1}}\\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} & \ldots & \widetilde{\lambda}_n^{\dot{2}}\end{pmatrix}\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{pmatrix}\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix} \qquad \widetilde{\lambda}\equiv\begin{pmatrix}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{pmatrix}\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\left(\lambda_1\;\;\lambda_2\;\;\lambda_3\;\;\cdots\;\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\;\widetilde{\lambda}_2\;\;\widetilde{\lambda}_3\;\;\cdots\;\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\\widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$,

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\left(\begin{array}{ccccc}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{array}\right)\equiv\left(\begin{array}{c}\lambda^1\\ \lambda^2\end{array}\right)\qquad\widetilde{\lambda}\equiv\left(\begin{array}{ccccc}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{array}\right)\equiv\left(\begin{array}{c}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{array}\right)$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\left(\begin{array}{ccccc}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{array}\right)\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix} \qquad \widetilde{\lambda}\equiv\left(\begin{array}{ccccc}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{array}\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

$$\lambda\equiv\left(\lambda_1\;\lambda_2\;\lambda_3\;\cdots\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\widetilde{\lambda}_2\;\widetilde{\lambda}_3\;\cdots\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\\widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The **"two–plane"** λ:

the *span* of 2 vectors $\lambda^\alpha\in\mathbb{C}^n$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

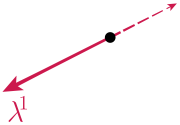
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The **"two–plane"** λ:

the *span* of 2 vectors $\lambda^\alpha \in \mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

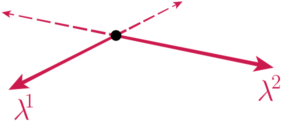
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The **"two–plane"** λ:

the *span* of 2 vectors $\lambda^\alpha \in \mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

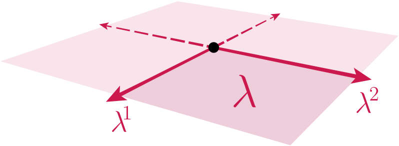
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The "two–plane" λ:

the *span* of 2 vectors $\lambda^\alpha \in \mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

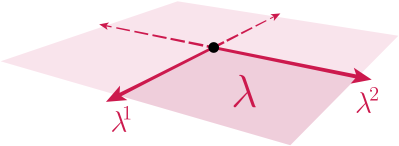
$$\lambda\equiv\left(\lambda_1\;\;\lambda_2\;\;\lambda_3\;\cdots\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\;\widetilde{\lambda}_2\;\;\widetilde{\lambda}_3\;\cdots\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\\widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The **"two–plane"** λ:

the *span* of 2 vectors $\lambda^\alpha\in\mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

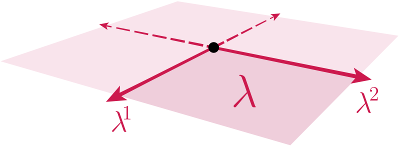
$$\lambda\equiv\left(\lambda_1\;\;\lambda_2\;\;\lambda_3\;\cdots\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\;\widetilde{\lambda}_2\;\;\widetilde{\lambda}_3\;\cdots\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\\widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The **“two–plane”** λ:

the *span* of 2 vectors $\lambda^\alpha\in\mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

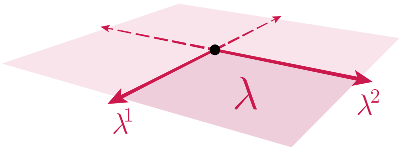
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot 1} \\ \widetilde{\lambda}^{\dot 2} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot\alpha}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

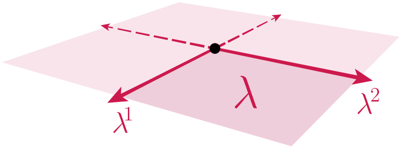
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

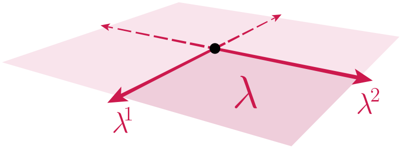
$$\lambda\equiv\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{pmatrix}\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\begin{pmatrix}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{pmatrix}\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

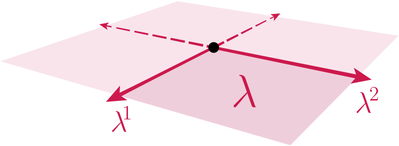
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:
  (taking all the momenta to be incoming)

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

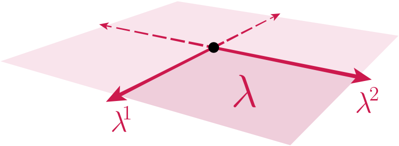
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:
  (taking all the momenta to be incoming)
  $\delta^4\left(\sum_a p_a^\mu\right)$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

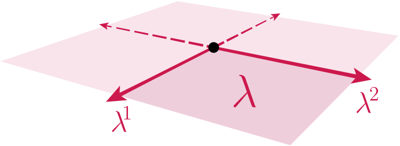
$$\lambda\equiv\left(\lambda_1\;\;\lambda_2\;\;\lambda_3\;\cdots\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\;\widetilde{\lambda}_2\;\;\widetilde{\lambda}_3\;\cdots\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:
  (taking all the momenta to be incoming)

$$\delta^4\left(\sum_a p_a^\mu\right)=\delta^{2\times2}\left(\sum_a p_a^{\alpha\dot{\alpha}}\right)$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

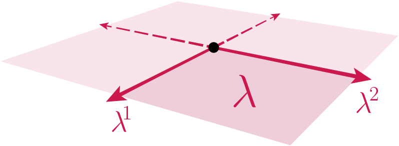
$$\lambda\equiv\left(\lambda_1\;\;\lambda_2\;\;\lambda_3\;\cdots\;\lambda_n\right)\equiv\begin{pmatrix}\lambda^1\\\lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\left(\widetilde{\lambda}_1\;\;\widetilde{\lambda}_2\;\;\widetilde{\lambda}_3\;\cdots\;\widetilde{\lambda}_n\right)\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\\widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:
  (taking all the momenta to be incoming)

$$\delta^4\left(\sum_a p_a^\mu\right)=\delta^{2\times2}\left(\sum_a\lambda_a^\alpha\widetilde{\lambda}_a^{\dot{\alpha}}\right)$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

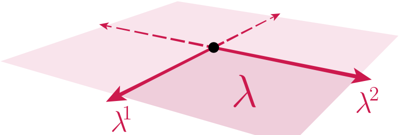
$$\lambda \equiv \left(\lambda_1\ \ \lambda_2\ \ \lambda_3\ \cdots\ \lambda_n\right) \equiv \begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix} \qquad \widetilde{\lambda} \equiv \left(\widetilde{\lambda}_1\ \ \widetilde{\lambda}_2\ \ \widetilde{\lambda}_3\ \cdots\ \widetilde{\lambda}_n\right) \equiv \begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**:
  (taking all the momenta to be incoming)

$$\delta^4\left(\sum_a p_a^\mu\right) = \delta^{2\times2}\left(\sum_a \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}\right) \equiv \delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

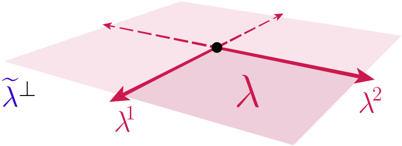
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda} \subset \lambda^\perp$ and $\lambda \subset \widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

$$\delta^4\left(\sum_a p_a^\mu\right) = \delta^{2\times 2}\left(\sum_a \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}\right) \equiv \delta^{2\times 2}(\lambda \cdot \widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

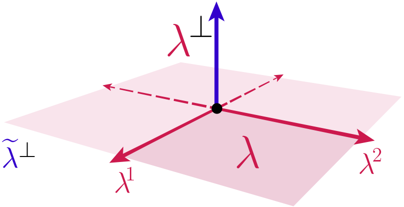
$$\lambda\equiv\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{pmatrix}\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\begin{pmatrix}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{pmatrix}\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\in\mathbb{C}^2$ for a column, $\lambda^\alpha\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot\alpha}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda}\subset\lambda^\perp$ and $\lambda\subset\widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

$$\delta^4\Big(\sum_a p_a^\mu\Big)=\delta^{2\times2}\Big(\sum_a\lambda_a^\alpha\widetilde{\lambda}_a^{\dot\alpha}\Big)\equiv\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

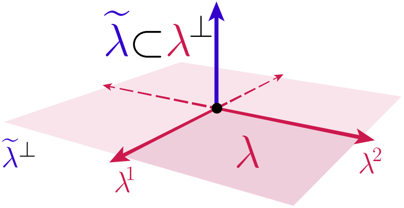
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda} \subset \lambda^\perp$ and $\lambda \subset \widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

$$\delta^4\Big(\sum_a p_a^\mu\Big) = \delta^{2\times 2}\Big(\sum_a \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}\Big) \equiv \delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

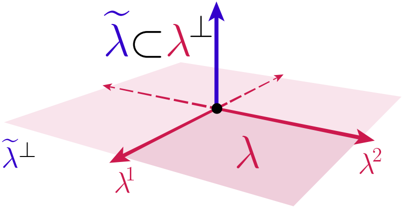
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda} \subset \lambda^\perp$ and $\lambda \subset \widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

$$\delta^4\left(\sum_a p_a^\mu\right) = \delta^{2\times 2}\left(\sum_a \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}\right) \equiv \delta^{2\times 2}(\lambda \cdot \widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

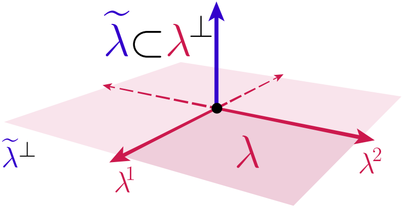
$$\lambda\equiv\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n\end{pmatrix}\equiv\begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\qquad\widetilde{\lambda}\equiv\begin{pmatrix}\widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n\end{pmatrix}\equiv\begin{pmatrix}\widetilde{\lambda}^{\dot{1}}\\ \widetilde{\lambda}^{\dot{2}}\end{pmatrix}$$

writing $\lambda_a\!\in\mathbb{C}^2$ for a column, $\lambda^\alpha\!\in\mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda}\subset\lambda^\perp$ and $\lambda\subset\widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

$$\delta^4\Big(\sum_a p_a^\mu\Big)=\delta^{2\times2}\Big(\sum_a\lambda_a^\alpha\widetilde{\lambda}_a^{\dot{\alpha}}\Big)\equiv\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

The *Grassmannian* Geometry of Kinematical Constraints

Thus, all the kinematical data can be described by a pair of $(2\times n)$ matrices:

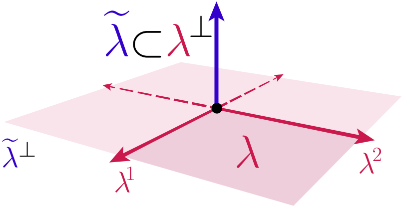
$$\lambda \equiv \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \end{pmatrix} \equiv \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \qquad \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1 & \widetilde{\lambda}_2 & \widetilde{\lambda}_3 & \cdots & \widetilde{\lambda}_n \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\lambda}^{\dot{1}} \\ \widetilde{\lambda}^{\dot{2}} \end{pmatrix}$$

writing $\lambda_a \in \mathbb{C}^2$ for a column, $\lambda^\alpha \in \mathbb{C}^n$ for a row.

- Because Lorentz transformations mix the rows of each matrix, λ^α, $\widetilde{\lambda}^{\dot{\alpha}}$, and the little group allows for rescaling, the **invariant** content of the data is:

The *Grassmannian* $G(k,n)$:

the *span* of k vectors in $\mathbb{C}^n$



- **Momentum conservation**: $\widetilde{\lambda} \subset \lambda^\perp$ and $\lambda \subset \widetilde{\lambda}^\perp$ (taking all the momenta to be incoming)

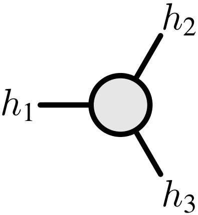
$$\delta^4\Big(\sum_a p_a^\mu\Big) = \delta^{2\times2}\Big(\sum_a \lambda_a^\alpha \widetilde{\lambda}_a^{\dot{\alpha}}\Big) \equiv \delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).

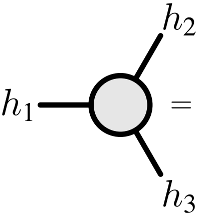
Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



Building Blocks: the S-Matrix for Three Massless Particles

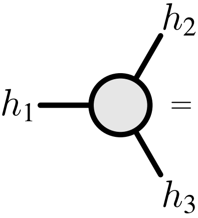
Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



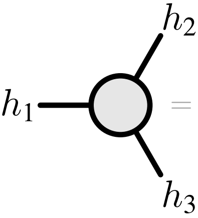
$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



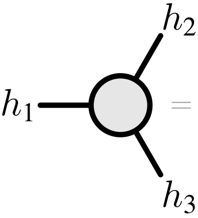
$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

Building Blocks: the S-Matrix for Three Massless Particles

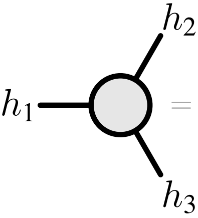
Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \Rightarrow \begin{cases} \lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix} \\ \\ \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix} \end{cases}$$

Building Blocks: the S-Matrix for Three Massless Particles

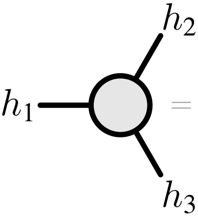
Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \Rightarrow \begin{cases} \lambda^{\perp} \equiv \big(\langle 23\rangle\,\langle 31\rangle\,\langle 12\rangle\big) \\ \lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix} \\ \\ \\ \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix} \\ \widetilde{\lambda}^{\perp} \equiv \big([23]\,[31]\,[12]\big) \end{cases}$$

Building Blocks: the S-Matrix for Three Massless Particles

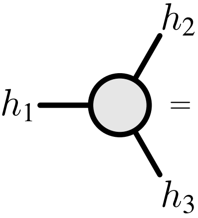
Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \Rightarrow \begin{cases} \lambda^{\perp} \equiv \left(\langle23\rangle\,\langle31\rangle\,\langle12\rangle\right) \supset \widetilde{\lambda} \\ \lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix} \\ \\ \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix} \\ \widetilde{\lambda}^{\perp} \equiv \left([23]\,[31]\,[12]\right) \supset \lambda \end{cases}$$

Building Blocks: the S-Matrix for Three Massless Particles

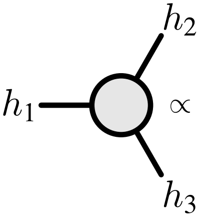
Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= f(\lambda_1\widetilde{\lambda}_1, \lambda_2\widetilde{\lambda}_2, \lambda_3\widetilde{\lambda}_3)\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \Rightarrow \begin{cases} \lambda^{\perp} \equiv \left(\langle 23\rangle\,\langle 31\rangle\,\langle 12\rangle\right) \supset \widetilde{\lambda} \\ \lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix} \\ \\ \text{or} \\ \\ \widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix} \\ \widetilde{\lambda}^{\perp} \equiv \left([23]\,[31]\,[12]\right) \supset \lambda \end{cases}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} f(\lambda_1, \lambda_2, \lambda_3) \\ \\ f(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3) \end{cases}$$

$$\lambda^{\perp} \equiv \big(\langle 23\rangle\,\langle 31\rangle\,\langle 12\rangle\big) \supset \widetilde{\lambda}$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

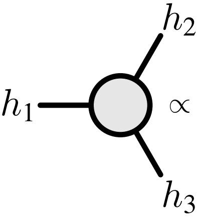
or

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp} \equiv \big([23]\,[31]\,[12]\big) \supset \lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} f(\lambda_1, \lambda_2, \lambda_3) \\ \\ f(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3) \end{cases}$$

$$\lambda^{\perp} \equiv \left(\langle 23 \rangle \, \langle 31 \rangle \, \langle 12 \rangle \right) \supset \widetilde{\lambda}$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

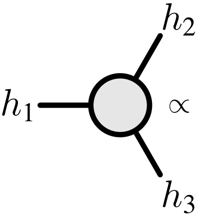
or

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp} \equiv \left([23] \, [31] \, [12] \right) \supset \lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} f(\lambda_1,\lambda_2,\lambda_3) \\ \\ f(\widetilde{\lambda}_1,\widetilde{\lambda}_2,\widetilde{\lambda}_3) \end{cases}$$

$$\lambda^{\perp} \equiv \big(\langle 23\rangle\,\langle 31\rangle\,\langle 12\rangle\big) \supset \widetilde{\lambda}$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

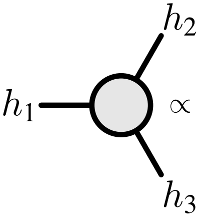
or

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp} \equiv \big([23]\,[31]\,[12]\big) \supset \lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1} \\ \\ [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2} \end{cases}$$

$$\lambda^{\perp} \equiv \left(\langle 23\rangle\ \langle 31\rangle\ \langle 12\rangle\right) \supset \widetilde{\lambda}$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

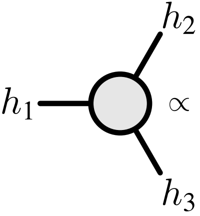
or

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp} \equiv \left([23]\ [31]\ [12]\right) \supset \lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1} \\ \\ [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2} \end{cases}$$

$$\lambda^{\perp}\equiv\big(\langle 23\rangle\;\langle 31\rangle\;\langle 12\rangle\big)\supset\widetilde{\lambda}$$

$$\lambda\;\equiv\begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

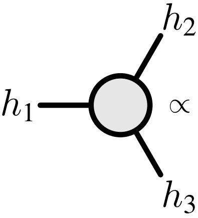
or

$$\widetilde{\lambda}\;\equiv\begin{pmatrix} \widetilde{\lambda}_1^{\dot 1} & \widetilde{\lambda}_2^{\dot 1} & \widetilde{\lambda}_3^{\dot 1} \\ \widetilde{\lambda}_1^{\dot 2} & \widetilde{\lambda}_2^{\dot 2} & \widetilde{\lambda}_3^{\dot 2} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp}\equiv\big([23]\;[31]\;[12]\big)\supset\lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1} \\ \qquad \xrightarrow[\langle a\,b\rangle\to\mathcal{O}(\epsilon)]{} \mathcal{O}\big(\epsilon^{-(h_1+h_2+h_3)}\big) \\ \\ [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2} \\ \qquad \xrightarrow[[a\,b]\to\mathcal{O}(\epsilon)]{} \mathcal{O}\big(\epsilon^{(h_1+h_2+h_3)}\big) \end{cases}$$

$$\lambda^{\perp} \equiv \big(\langle 23\rangle\;\langle 31\rangle\;\langle 12\rangle\big) \supset \widetilde{\lambda}$$

$$\lambda \equiv \begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

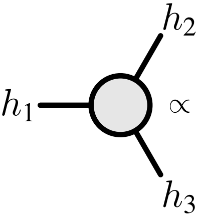
or

$$\widetilde{\lambda} \equiv \begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp} \equiv \big([23]\;[31]\;[12]\big) \supset \lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \begin{cases} \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1} \\ \qquad \xrightarrow[\langle a\,b\rangle\to\mathcal{O}(\epsilon)]{} \mathcal{O}\Big(\epsilon^{-(h_1+h_2+h_3)}\Big) \\ \\ [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2} \\ \qquad \xrightarrow[[a\,b]\to\mathcal{O}(\epsilon)]{} \mathcal{O}\Big(\epsilon^{(h_1+h_2+h_3)}\Big) \end{cases}$$

$$\lambda^{\perp}\equiv\big(\langle 23\rangle\ \langle 31\rangle\ \langle 12\rangle\big)\supset\widetilde{\lambda}$$

$$\lambda\ \equiv\begin{pmatrix} \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}$$

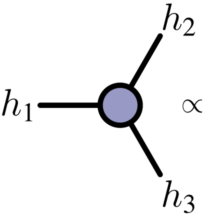
or

$$\widetilde{\lambda}\ \equiv\begin{pmatrix} \widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}} \\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}} \end{pmatrix}$$

$$\widetilde{\lambda}^{\perp}\equiv\big([23]\ [31]\ [12]\big)\supset\lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



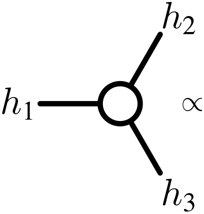
$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$

$$h_1+h_2+h_3\leq 0$$

$$\lambda^{\perp}\equiv\left(\langle 23\rangle\ \langle 31\rangle\ \langle 12\rangle\right)\supset\widetilde{\lambda}$$

$$\lambda\ \equiv\begin{pmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1\\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2\end{pmatrix}$$

or



$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$

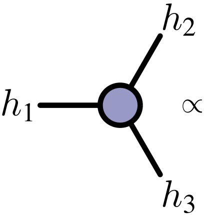
$$h_1+h_2+h_3\geq 0$$

$$\widetilde{\lambda}\ \equiv\begin{pmatrix}\widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}}\\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}}\end{pmatrix}$$

$$\widetilde{\lambda}^{\perp}\equiv\left([23]\ [31]\ [12]\right)\supset\lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



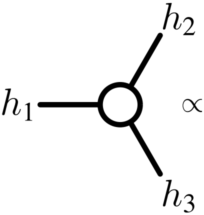
$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$

$$h_1+h_2+h_3\leq 0$$

$$\lambda^{\perp}\equiv\big(\langle 23\rangle\;\langle 31\rangle\;\langle 12\rangle\big)\supset\widetilde{\lambda}$$

$$\lambda\;\equiv\begin{pmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1\\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2\end{pmatrix}$$

or



$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$

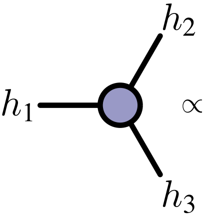
$$h_1+h_2+h_3\geq 0$$

$$\widetilde{\lambda}\;\equiv\begin{pmatrix}\widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}}\\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}}\end{pmatrix}$$

$$\widetilde{\lambda}^{\perp}\equiv\big([23]\;[31]\;[12]\big)\supset\lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



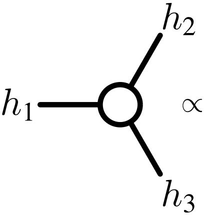
$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$

$$h_1+h_2+h_3\leq 0$$

$$\lambda^{\perp}\equiv\big(\langle 23\rangle\;\langle 31\rangle\;\langle 12\rangle\big)\supset\widetilde{\lambda}$$

$$\lambda\;\equiv\begin{pmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1\\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2\end{pmatrix}$$

or



$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$

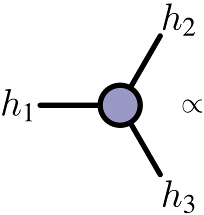
$$h_1+h_2+h_3\geq 0$$

$$\widetilde{\lambda}\;\equiv\begin{pmatrix}\widetilde{\lambda}_1^{\dot{1}} & \widetilde{\lambda}_2^{\dot{1}} & \widetilde{\lambda}_3^{\dot{1}}\\ \widetilde{\lambda}_1^{\dot{2}} & \widetilde{\lambda}_2^{\dot{2}} & \widetilde{\lambda}_3^{\dot{2}}\end{pmatrix}$$

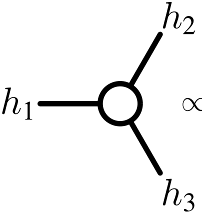
$$\widetilde{\lambda}^{\perp}\equiv\big([23]\;[31]\;[12]\big)\supset\lambda$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



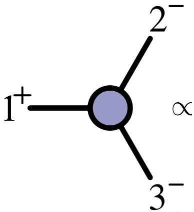
$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$
$$h_1+h_2+h_3 \leq 0$$



$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$
$$h_1+h_2+h_3 \geq 0$$

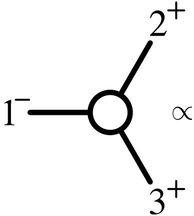
Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$

$$h_1+h_2+h_3 \leq 0$$

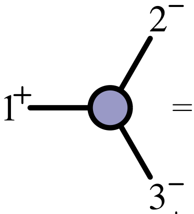


$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$

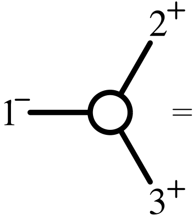
$$h_1+h_2+h_3 \geq 0$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



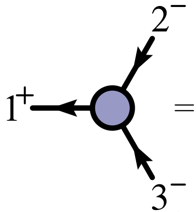
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\;\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$



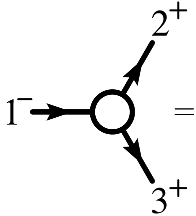
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\;\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



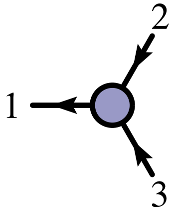
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$



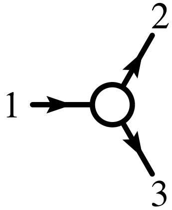
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



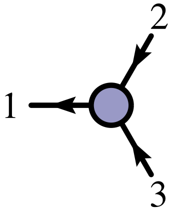
$$= \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$



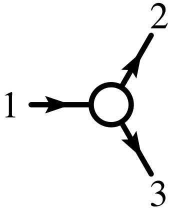
$$= \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



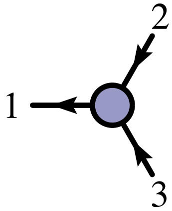
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(+,-,-)$$



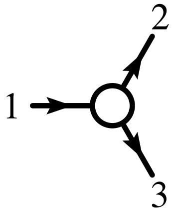
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(-,+,+)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



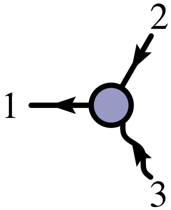
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(+,-,-)$$



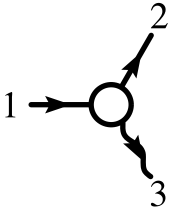
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(-,+,+)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



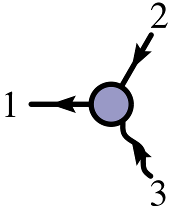
$$= \frac{\langle 3\,1\rangle\langle 2\,3\rangle^3}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\big(+\tfrac{1}{2},-\tfrac{1}{2},-\big)$$



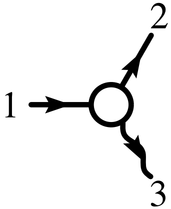
$$= \frac{[3\,1][2\,3]^3}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\big(-\tfrac{1}{2},+\tfrac{1}{2},+\big)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



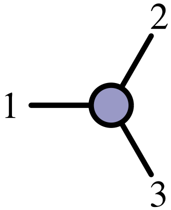
$$= \frac{\langle 3\,1\rangle\langle 2\,3\rangle^3}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\big(+\tfrac{1}{2},-\tfrac{1}{2},-\big)$$



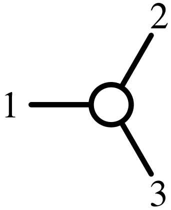
$$= \frac{[3\,1][2\,3]^3}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\big(-\tfrac{1}{2},+\tfrac{1}{2},+\big)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



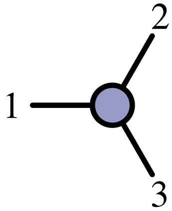
$$= \quad \frac{\delta^{2\times4}\big(\lambda\!\cdot\!\widetilde{\eta}\big)}{\langle1\,2\rangle\langle2\,3\rangle\langle3\,1\rangle}\,\delta^{2\times2}\big(\lambda\!\cdot\!\widetilde{\lambda}\big)\equiv\mathcal{A}_3^{(2)}$$



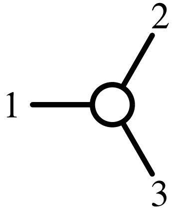
$$= \quad \frac{\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\!\cdot\!\widetilde{\eta}\big)}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}\big(\lambda\!\cdot\!\widetilde{\lambda}\big)\equiv\mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



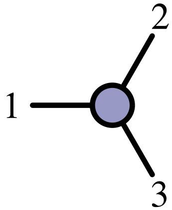
$$= \quad \frac{\delta^{2\times4}\big(\lambda\!\cdot\!\widetilde{\eta}\big)}{\langle1\,2\rangle\langle2\,3\rangle\langle3\,1\rangle}\,\delta^{2\times2}\big(\lambda\!\cdot\!\widetilde{\lambda}\big)\equiv\mathcal{A}_3^{(2)}$$



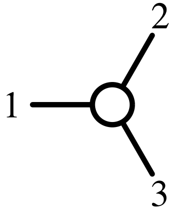
$$= \quad \frac{\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\!\cdot\!\widetilde{\eta}\big)}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}\big(\lambda\!\cdot\!\widetilde{\lambda}\big)\equiv\mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



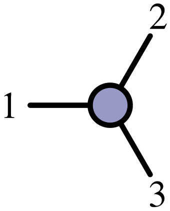
$$= \frac{\delta^{2\times4}(\lambda\cdot\widetilde{\eta})}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(2)}$$



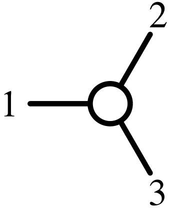
$$= \frac{\delta^{1\times4}(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta})}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



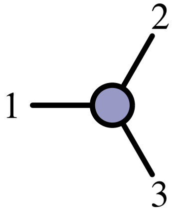
$$= \quad \frac{\delta^{2\times4}(\lambda\cdot\widetilde{\eta})}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(2)}$$



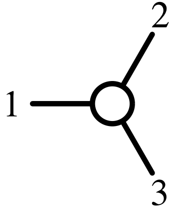
$$= \quad \frac{\delta^{1\times4}(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta})}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= \quad \frac{\delta^{2\times4}\big(\lambda\cdot\widetilde{\eta}\big)}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}\big(\lambda\cdot\widetilde{\lambda}\big) \equiv \mathcal{A}_3^{(2)}$$



$$= \quad \frac{\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}\big(\lambda\cdot\widetilde{\lambda}\big) \equiv \mathcal{A}_3^{(1)}$$

The *Simplest* Quantum Field Theory: $\mathcal{N}\!=\!4$ superYang-Mills

$$\widetilde{Q}\;|a\rangle^{h_a}=|a\rangle^{h_a\text{-}1/2}$$

The *Simplest* Quantum Field Theory: $\mathcal{N}=4$ superYang-Mills

$$\widetilde{Q}_I|a\rangle^{h_a}=|a\rangle_I^{h_a\text{-}1/2}$$

$$|a\rangle\equiv e^{\widetilde{Q}_I\widetilde{\eta}_a^I}|a\rangle^{+\sigma}$$

The *Simplest* Quantum Field Theory: $\mathcal{N}=4$ superYang-Mills

$$\widetilde{Q}_I|a\rangle^{h_a}=|a\rangle_I^{h_a-1/2}$$

$$|a\rangle\equiv e^{\widetilde{Q}_I\widetilde{\eta}_a^I}|a\rangle^{+\sigma}$$

$$|a\rangle\equiv|a\rangle^{+1}+\widetilde{\eta}_a^I|a\rangle_I^{+1/2}+\frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J|a\rangle_{IJ}^0+\frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K|a\rangle_{IJK}^{-1/2}+\frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L|a\rangle_{IJKL}^{-1}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J |a\rangle_{IJ}^{0} + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J|a\rangle_{IJ}^{0} + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K|a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L|a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \qquad \widetilde{\lambda}^{\perp} = \left(\frac{c_1 \quad\;\; c_2 \quad\;\; c_3}{[23]\;\,[31]\;\,[12]}\right)$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J |a\rangle_{IJ}^0 + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \qquad \widetilde{\lambda}^{\perp} = \left(\frac{c_1}{[23]}\;\frac{c_2}{[31]}\;\frac{c_3}{[12]}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\widetilde{\eta}_{a_1}^I\cdots\widetilde{\eta}_{a_k}^I\right)$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I \widetilde{\eta}_a^J |a\rangle_{IJ}^0 + \frac{1}{3!}\widetilde{\eta}_a^I \widetilde{\eta}_a^J \widetilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I \widetilde{\eta}_a^J \widetilde{\eta}_a^K \widetilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \widetilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}} \left(\sum_{1\leq a_1<\cdots<a_k\leq n} (a_1 \cdots a_k)\; \widetilde{\eta}_{a_1}^I \cdots \widetilde{\eta}_{a_k}^I \right)$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J |a\rangle_{IJ}^{0} + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \widetilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}} \left(\sum_{1\leq a_1<\cdots<a_k\leq n} (a_1\cdots a_k)\ \widetilde{\eta}_{a_1}^I\cdots\widetilde{\eta}_{a_k}^I \right)$$

$$\begin{aligned}
\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big) = \quad & \big((1)\,\widetilde{\eta}_1^1\big)\ \big((1)\,\widetilde{\eta}_1^2\big)\ \big((1)\,\widetilde{\eta}_1^3\big)\ \big((1)\,\widetilde{\eta}_1^4\big) + \ldots \quad \text{(3 terms)} \\
+ & \big((1)\,\widetilde{\eta}_1^1\big)\ \big((1)\,\widetilde{\eta}_1^2\big)\ \big((1)\,\widetilde{\eta}_1^3\big)\ \big((2)\,\widetilde{\eta}_2^4\big) + \ldots \text{(24 terms)} \\
+ & \big((1)\,\widetilde{\eta}_1^1\big)\ \big((1)\,\widetilde{\eta}_1^2\big)\ \big((2)\,\widetilde{\eta}_2^3\big)\ \big((3)\,\widetilde{\eta}_3^4\big) + \ldots \text{(36 terms)} \\
+ & \big((1)\,\widetilde{\eta}_1^1\big)\ \big((1)\,\widetilde{\eta}_1^2\big)\ \big((2)\,\widetilde{\eta}_2^3\big)\ \big((2)\,\widetilde{\eta}_2^4\big) + \ldots \text{(18 terms)}
\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \tilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\tilde{\eta}_a^I\tilde{\eta}_a^J|a\rangle_{IJ}^{0} + \frac{1}{3!}\tilde{\eta}_a^I\tilde{\eta}_a^J\tilde{\eta}_a^K|a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\tilde{\eta}_a^I\tilde{\eta}_a^J\tilde{\eta}_a^K\tilde{\eta}_a^L|a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times4}\big(\tilde{\lambda}^{\perp}\cdot\tilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \tilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\tilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\tilde{\eta}_{a_1}^I\cdots\tilde{\eta}_{a_k}^I\right)$$

$$\begin{aligned}\delta^{1\times4}\big(\tilde{\lambda}^{\perp}\cdot\tilde{\eta}\big) = \;& \big([23]\tilde{\eta}_1^1\big)\big([23]\tilde{\eta}_1^2\big)\big([23]\tilde{\eta}_1^3\big)\big([23]\tilde{\eta}_1^4\big) + \ldots \quad \text{(3 terms)} \\ +\;& \big([23]\tilde{\eta}_1^1\big)\big([23]\tilde{\eta}_1^2\big)\big([23]\tilde{\eta}_1^3\big)\big([31]\tilde{\eta}_2^4\big) + \ldots \; \text{(24 terms)} \\ +\;& \big([23]\tilde{\eta}_1^1\big)\big([23]\tilde{\eta}_1^2\big)\big([31]\tilde{\eta}_2^3\big)\big([12]\tilde{\eta}_3^4\big) + \ldots \; \text{(36 terms)} \\ +\;& \big([23]\tilde{\eta}_1^1\big)\big([23]\tilde{\eta}_1^2\big)\big([31]\tilde{\eta}_2^3\big)\big([31]\tilde{\eta}_2^4\big) + \ldots \; \text{(18 terms)}\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}^I_a |a\rangle^{+1/2}_I + \frac{1}{2!}\widetilde{\eta}^I_a\widetilde{\eta}^J_a |a\rangle^{0}_{IJ} + \frac{1}{3!}\widetilde{\eta}^I_a\widetilde{\eta}^J_a\widetilde{\eta}^K_a |a\rangle^{-1/2}_{IJK} + \frac{1}{4!}\widetilde{\eta}^I_a\widetilde{\eta}^J_a\widetilde{\eta}^K_a\widetilde{\eta}^L_a |a\rangle^{-1}_{IJKL}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \widetilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\widetilde{\eta}^I_{a_1}\cdots\widetilde{\eta}^I_{a_k}\right)$$

$$\begin{aligned}
\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big) = \;& \big([23]\widetilde{\eta}^1_1\big)\,\big([23]\widetilde{\eta}^2_1\big)\,\big([23]\widetilde{\eta}^3_1\big)\,\big([23]\widetilde{\eta}^4_1\big) + \ldots \quad \text{(3 terms)}\\
+\;& \big([23]\widetilde{\eta}^1_1\big)\,\big([23]\widetilde{\eta}^2_1\big)\,\big([23]\widetilde{\eta}^3_1\big)\,\big([31]\widetilde{\eta}^4_2\big) + \ldots \;\text{(24 terms)}\\
+\;& \big([23]\widetilde{\eta}^1_1\big)\,\big([23]\widetilde{\eta}^2_1\big)\,\big([31]\widetilde{\eta}^3_2\big)\,\big([12]\widetilde{\eta}^4_3\big) + \ldots \;\text{(36 terms)}\\
+\;& \big([23]\widetilde{\eta}^1_1\big)\,\big([23]\widetilde{\eta}^2_1\big)\,\big([31]\widetilde{\eta}^3_2\big)\,\big([31]\widetilde{\eta}^4_2\big) + \ldots \;\text{(18 terms)}
\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \tilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\tilde{\eta}_a^I \tilde{\eta}_a^J |a\rangle_{IJ}^{0} + \frac{1}{3!}\tilde{\eta}_a^I \tilde{\eta}_a^J \tilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\tilde{\eta}_a^I \tilde{\eta}_a^J \tilde{\eta}_a^K \tilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\tilde{\lambda}^{\perp}\cdot\tilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \tilde{\lambda}^{\perp} = \big(\overset{c_1}{[23]}\ \overset{c_2}{[31]}\ \overset{c_3}{[12]}\big)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\tilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}} \left(\sum_{1\leq a_1<\cdots<a_k\leq n} (a_1\cdots a_k)\ \tilde{\eta}_{a_1}^I \cdots \tilde{\eta}_{a_k}^I \right)$$

$$\begin{aligned}
\delta^{1\times 4}\big(\tilde{\lambda}^{\perp}\cdot\tilde{\eta}\big) = &\ \ \big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_1^3\,\tilde{\eta}_1^4\big)\times[23]^4 && +\ldots\ \ (3\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_1^3\,\tilde{\eta}_2^4\big)\times[23]^3[31] && +\ldots\ (24\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_2^3\,\tilde{\eta}_3^4\big)\times[23]^2[31][12] && +\ldots\ (36\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_2^3\,\tilde{\eta}_2^4\big)\times[23]^2[31]^2 && +\ldots\ (18\text{ terms})
\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle\equiv|a\rangle^{+1}+\widetilde{\eta}_a^I|a\rangle_I^{+1/2}+\frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J|a\rangle_{IJ}^{0}+\frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K|a\rangle_{IJK}^{-1/2}+\frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L|a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)}\propto\frac{\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]}\quad\text{with}\quad C=\widetilde{\lambda}^{\perp}=\left(\begin{array}{ccc} c_1 & c_2 & c_3\\ \hline [23] & [31] & [12]\end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big)\equiv\prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\widetilde{\eta}_{a_1}^I\cdots\widetilde{\eta}_{a_k}^I\right)$$

$$\begin{aligned}\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)=\;&\big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_1^4\big)\times[23]^4 &&+\ldots\;\;(3\text{ terms})\\ &+\big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_2^4\big)\times[23]^3[31] &&+\ldots\;(24\text{ terms})\\ &+\big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_3^4\big)\times[23]^2[31][12] &&+\ldots\;(36\text{ terms})\\ &+\big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_2^4\big)\times[23]^2[31]^2 &&+\ldots\;(18\text{ terms})\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J |a\rangle_{IJ}^{0} + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \widetilde{\lambda}^{\perp} = \Big(\overset{c_1}{[23]}\ \overset{c_2}{[31]}\ \overset{c_3}{[12]}\Big)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\ \widetilde{\eta}_{a_1}^I\cdots\widetilde{\eta}_{a_k}^I\right)$$

$$\begin{aligned}
\mathcal{A}_3^{(1)} = \quad & \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_1^4\big)\mathcal{A}\big(|1\rangle_{1234}^{-1},|2\rangle^{+1},|3\rangle^{+1}\big) + \ldots \quad \text{(3 terms)}\\
+ & \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{123}^{-1/2},|2\rangle_4^{+1/2},|3\rangle^{+1}\big) + \ldots \quad \text{(24 terms)}\\
+ & \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_3^4\big)\mathcal{A}\big(|1\rangle_{12}^{0},|2\rangle_3^{+1/2},|3\rangle_4^{+1/2}\big) + \ldots \quad \text{(36 terms)}\\
+ & \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{12}^{0},|2\rangle_{34}^{0},|3\rangle^{+1}\big) + \ldots \quad \text{(18 terms)}
\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \tilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\tilde{\eta}_a^I\tilde{\eta}_a^J |a\rangle_{IJ}^{0} + \frac{1}{3!}\tilde{\eta}_a^I\tilde{\eta}_a^J\tilde{\eta}_a^K |a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\tilde{\eta}_a^I\tilde{\eta}_a^J\tilde{\eta}_a^K\tilde{\eta}_a^L |a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\tilde{\lambda}^{\perp}\cdot\tilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \tilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\tilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\tilde{\eta}_{a_1}^I\cdots\tilde{\eta}_{a_k}^I\right)$$

$$\begin{aligned}
\mathcal{A}_3^{(1)} = \quad & \big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_1^3\,\tilde{\eta}_1^4\big)\mathcal{A}\big(|1\rangle_{1234}^{-1},|2\rangle^{+1},|3\rangle^{+1}\big)+\ldots \quad (3\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_1^3\,\tilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{123}^{-1/2},|2\rangle_4^{+1/2},|3\rangle^{+1}\big)+\ldots\;(24\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_2^3\,\tilde{\eta}_3^4\big)\mathcal{A}\big(|1\rangle_{12}^{0},|2\rangle_3^{+1/2},|3\rangle_4^{+1/2}\big)+\ldots\;(36\text{ terms})\\
&+\big(\tilde{\eta}_1^1\,\tilde{\eta}_1^2\,\tilde{\eta}_2^3\,\tilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{12}^{0},|2\rangle_{34}^{0},|3\rangle^{+1}\big)+\ldots\;(18\text{ terms})
\end{aligned}$$

Extracting Components from Super-Amplitudes

$$|a\rangle \equiv |a\rangle^{+1} + \widetilde{\eta}_a^I |a\rangle_I^{+1/2} + \frac{1}{2!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J|a\rangle_{IJ}^0 + \frac{1}{3!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K|a\rangle_{IJK}^{-1/2} + \frac{1}{4!}\widetilde{\eta}_a^I\widetilde{\eta}_a^J\widetilde{\eta}_a^K\widetilde{\eta}_a^L|a\rangle_{IJKL}^{-1}$$

$$\mathcal{A}_3^{(1)} \propto \frac{\delta^{1\times 4}\big(\widetilde{\lambda}^{\perp}\!\cdot\widetilde{\eta}\big)}{[12]\,[23]\,[31]} \quad \text{with} \quad C = \widetilde{\lambda}^{\perp} = \left(\begin{array}{ccc} c_1 & c_2 & c_3 \\ \hline [23] & [31] & [12] \end{array}\right)$$

$$\delta^{k\times\mathcal{N}}\big(C\cdot\widetilde{\eta}\big) \equiv \prod_{I=1}^{\mathcal{N}}\left(\sum_{1\leq a_1<\cdots<a_k\leq n}(a_1\cdots a_k)\;\widetilde{\eta}_{a_1}^I\cdots\widetilde{\eta}_{a_k}^I\right)$$

$$\begin{aligned}
\mathcal{A}_3^{(1)} = \;& \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_1^4\big)\mathcal{A}\big(|1\rangle_{1234}^{-1},|2\rangle^{+1}\;,|3\rangle^{+1}\;\big)+\ldots \quad \text{(3 terms)}\\
+& \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_1^3\,\widetilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{123}^{-1/2},|2\rangle_4^{+1/2},|3\rangle^{+1}\;\big)+\ldots \quad \text{(24 terms)}\\
+& \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_3^4\big)\mathcal{A}\big(|1\rangle_{12}^{0}\;,|2\rangle_3^{+1/2},|3\rangle_4^{+1/2}\big)+\ldots \quad \text{(36 terms)}\\
+& \big(\widetilde{\eta}_1^1\,\widetilde{\eta}_1^2\,\widetilde{\eta}_2^3\,\widetilde{\eta}_2^4\big)\mathcal{A}\big(|1\rangle_{12}^{0}\;,|2\rangle_{34}^{0}\;,|3\rangle^{+1}\;\big)+\ldots \quad \text{(18 terms)}
\end{aligned}$$

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory.

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big)$$

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big) = f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma}\delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big) = f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma}\delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$

Bose statistics requires that $\mathcal{A}$ be symmetric under the exchange $2 \leftrightarrow 3$;

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big) = f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma}\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

Bose statistics requires that $\mathcal{A}$ be symmetric under the exchange $2 \leftrightarrow 3$;

  - even-spin: f^{q_1,q_2,q_3} must be **totally symmetric**

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big) = f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma}\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

Bose statistics requires that $\mathcal{A}$ be symmetric under the exchange $2\leftrightarrow 3$;

- even-spin: f^{q_1,q_2,q_3} must be **totally symmetric**
- odd spin: f^{q_1,q_2,q_3} must be **totally antisymmetric**

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}] = [\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma \in \mathbb{Z}$:

$$\mathcal{A}\big(1^{+\sigma}_{q_1}, 2^{-\sigma}_{q_2}, 3^{-\sigma}_{q_3}\big) = f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma} \delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$

Bose statistics requires that $\mathcal{A}$ be symmetric under the exchange $2 \leftrightarrow 3$;

- even-spin: f^{q_1,q_2,q_3} must be **totally symmetric**
- odd spin: f^{q_1,q_2,q_3} must be **totally antisymmetric**

Coupling Constant Constraints: Scaling and Spin/Statistics

The coupling constants f^{q_1,q_2,q_3} are quantum-number-dependent **constants** which define the theory. Because *all* the kinematical dependence is fixed, *these couplings cannot 'run'*.

- Dimensional analysis shows that the mass-dimension of the coupling is:

$$[f^{q_1,q_2,q_3}]=[\text{mass}]^{1-|h_1+h_2+h_3|}$$

- Consider a theory involving only particles with **integer spin** $\sigma\in\mathbb{Z}$:

$$\mathcal{A}\left(1^{+\sigma}_{q_1},2^{-\sigma}_{q_2},3^{-\sigma}_{q_3}\right)=f^{q_1,q_2,q_3}\left(\frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\right)^{\sigma}\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})$$

Bose statistics requires that $\mathcal{A}$ be symmetric under the exchange $2\leftrightarrow 3$;

  - even-spin: f^{q_1,q_2,q_3} must be **totally symmetric**
  - odd spin: f^{q_1,q_2,q_3} must be **totally antisymmetric**

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Channeling Some Consequences of Factorization

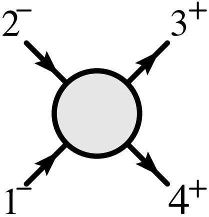
In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

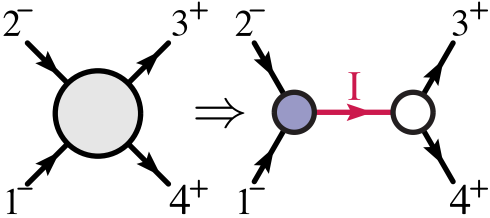
Consider the behavior of any local, unitarity theory in a **factorization limit**:



Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

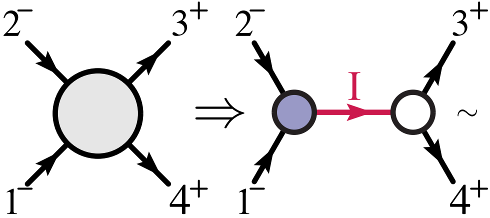
Consider the behavior of any local, unitarity theory in a **factorization limit**:



Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:

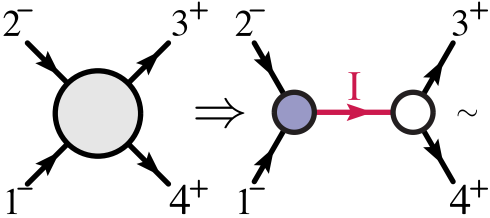


$$\sim f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}\left(\frac{\langle 12\rangle^3}{\langle 2\,\mathrm{I}\rangle\langle \mathrm{I}\,1\rangle}\frac{[34]^3}{[\mathrm{I}\,3][4\,\mathrm{I}]}\right)^{\sigma}$$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:

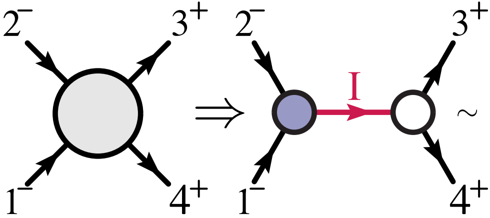


$$\sim\; f^{q_1,q_2,\bullet} f^{\bullet,q_3,q_4}\left(\frac{\langle 12\rangle^3}{\langle 2\,\mathrm{I}\rangle\langle \mathrm{I}\,1\rangle}\frac{[34]^3}{[\mathrm{I}\,3][4\,\mathrm{I}]}\right)^{\sigma}$$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:

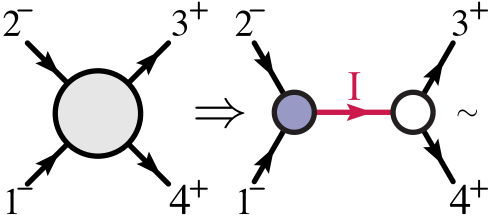


$$\sim \; f^{q_1,q_2,\bullet} f^{\bullet,q_3,q_4} \; \frac{\big(\langle 12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



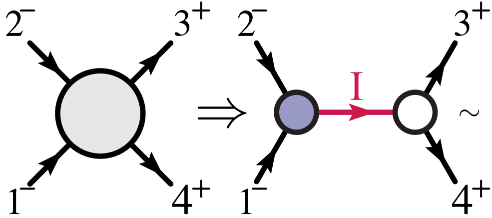
$$\sim f^{q_1,q_2,\bullet} f^{\bullet,q_3,q_4} \, \frac{\left(\langle 12\rangle[34]\right)^{2\sigma}}{u^{\sigma}}$$

with $u \equiv (p_2+p_4)^2$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



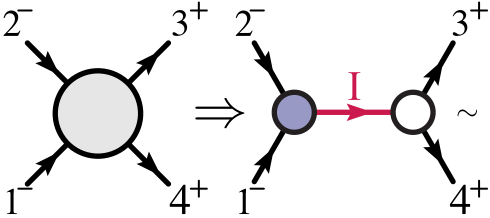
$$\sim\; f^{q_1,q_2,\bullet}\,f^{\bullet,q_3,q_4}\;\frac{\big(\langle 12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

with $u\equiv(p_2+p_4)^2$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



$$\sim\; f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}\;\frac{\big(\langle12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

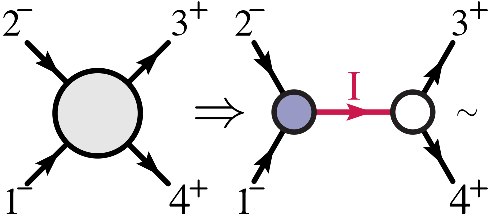
with $u \equiv (p_2+p_4)^2$

- **Homework**:

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



$$\sim f^{q_1,q_2,\bullet}\,f^{\bullet,q_3,q_4}\;\frac{\big(\langle 12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

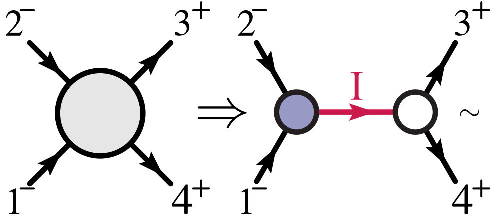
with $u \equiv (p_2+p_4)^2$

- **Homework**: use the result, together with the analogous u- and t-channels to determine the form of $\mathcal{A}_4$

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



$$\sim\; f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}\;\frac{\left(\langle 12\rangle[34]\right)^{2\sigma}}{u^{\sigma}}$$

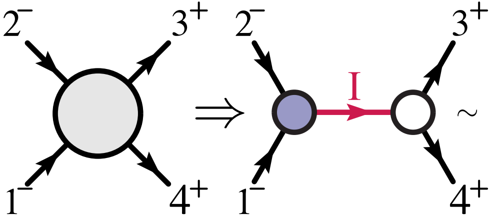
with $u\equiv(p_2+p_4)^2$

- **Homework**: use the result, together with the analogous *u*- and *t*-channels to determine the form of $\mathcal{A}_4$ and show that if $\sigma>2$ all factorizations vanish.

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



$$\sim\; f^{q_1,q_2,\bullet}\, f^{\bullet,q_3,q_4}\; \frac{\big(\langle 12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

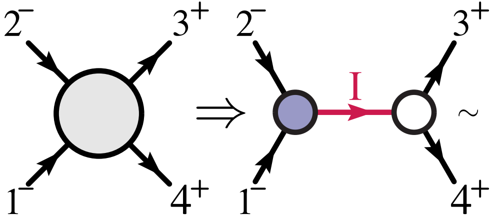
with $u \equiv (p_2+p_4)^2$

- **Homework**: use the result, together with the analogous u- and t-channels to determine the form of $\mathcal{A}_4$ and show that if $\sigma > 2$ all factorizations vanish. This is *Wienberg's theorem*

Channeling Some Consequences of Factorization

In [arXiv:0705.4305], Benincasa and Cachazo described how elementary considerations of locality and unitarity **strongly** restricts the choice of coupling constants, and hence possible quantum field theories.

Consider the behavior of any local, unitarity theory in a **factorization limit**:



$$\sim\; f^{q_1,q_2,\bullet}\,f^{\bullet,q_3,q_4}\;\frac{\big(\langle 12\rangle[34]\big)^{2\sigma}}{u^{\sigma}}$$

with $u\equiv(p_2+p_4)^2$

- **Homework**: use the result, together with the analogous u- and t-channels to determine the form of $\mathcal{A}_4$ and show that if $\sigma>2$ all factorizations vanish. This is *Wienberg's theorem*—proving that long-range physics requires $\sigma\leq2$.

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

Quantum Consistency Conditions from Cauchy's Theorem

Using Cauchy's theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}+f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4}+f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4}=0.$$

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4} + f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4} + f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4} = 0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!

Quantum Consistency Conditions from Cauchy's Theorem

Using Cauchy's theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4} + f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4} + f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4} = 0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!
- $\sigma = 2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma=1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}+f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4}+f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4}=0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!
- $\sigma=2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors—*there is at most one graviton*!

Quantum Consistency Conditions from Cauchy's Theorem

Using Cauchy's theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4} + f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4} + f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4} = 0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!
- $\sigma = 2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors—*there is at most one graviton*!
  - the coupling strength of any spin-2 particle to itself must be the same as its coupling to any other field

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4}+f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4}+f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4}=0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!
- $\sigma = 2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors—*there is at most one graviton*!
  - the coupling strength of any spin-2 particle to itself must be the same as its coupling to any other field—*the equivalence principle*!

Quantum Consistency Conditions from Cauchy's Theorem

Using Cauchy's theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4} + f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4} + f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4} = 0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!
- $\sigma = 2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors—*there is at most one graviton*!
  - the coupling strength of any spin-2 particle to itself must be the same as its coupling to any other field—*the equivalence principle*!

Quantum Consistency Conditions from Cauchy’s Theorem

Using Cauchy’s theorem to relate the three factorization channels to each other, Benincasa and Cachazo prove in [arXiv:0705.4305] following:

- $\sigma = 1$: the coupling constants satisfy a Jacobi identity!

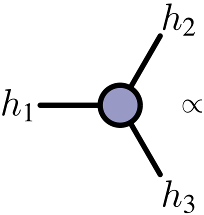
$$f^{q_1,q_2,\bullet}f^{\bullet,q_3,q_4} + f^{q_2,q_3,\bullet}f^{\bullet,q_1,q_4} + f^{q_3,q_1,\bullet}f^{\bullet,q_2,q_4} = 0.$$

  - *whatever* quantum numbers distinguish mutually interacting spin-1 particles, they form the **adjoint representation** of a Lie algebra!

- $\sigma = 2$: multiple spin-2 particles can always be decomposed into mutually non-interacting sectors—*there is at most one graviton*!
  - the coupling strength of any spin-2 particle to itself must be the same as its coupling to any other field—*the equivalence principle*!

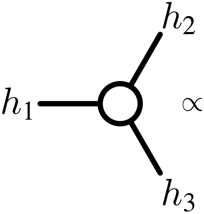
Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$\propto \quad \langle 12\rangle^{h_3-h_1-h_2}\langle 23\rangle^{h_1-h_2-h_3}\langle 31\rangle^{h_2-h_3-h_1}$$

$$h_1+h_2+h_3 \leq 0$$

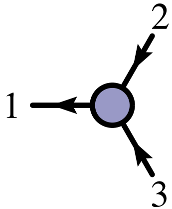


$$\propto \quad [12]^{h_1+h_2-h_3}[23]^{h_2+h_3-h_1}[31]^{h_3+h_1-h_2}$$

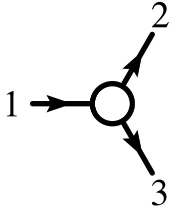
$$h_1+h_2+h_3 \geq 0$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



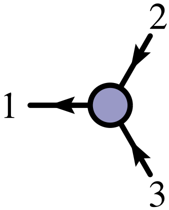
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\;\delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$



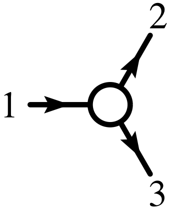
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\;\delta^{2\times 2}\big(\lambda\cdot\widetilde{\lambda}\big)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



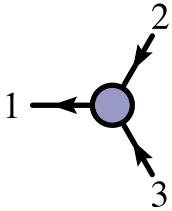
$$= \quad \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(+,-,-)$$



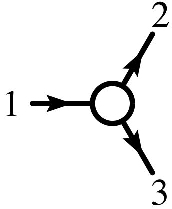
$$= \quad \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(-,+,+)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



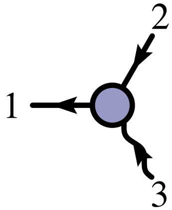
$$= \frac{\langle 2\,3\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(+,-,-)$$



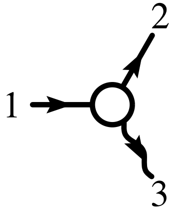
$$= \frac{[2\,3]^4}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3(-,+,+)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



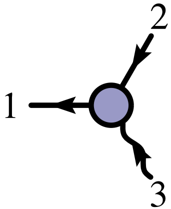
$$= \frac{\langle 3\,1\rangle\langle 2\,3\rangle^{3}}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\left(+\tfrac{1}{2},-\tfrac{1}{2},-\right)$$



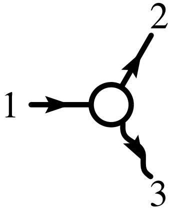
$$= \frac{[3\,1][2\,3]^{3}}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\left(-\tfrac{1}{2},+\tfrac{1}{2},+\right)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



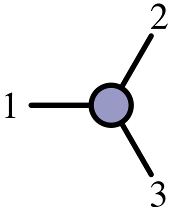
$$= \frac{\langle 3\,1\rangle\langle 2\,3\rangle^3}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\left(+\tfrac{1}{2},-\tfrac{1}{2},-\right)$$



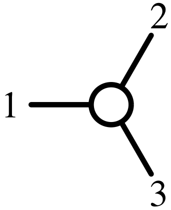
$$= \frac{[3\,1][2\,3]^3}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3\left(-\tfrac{1}{2},+\tfrac{1}{2},+\right)$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



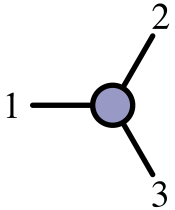
$$= \quad \frac{\delta^{2\times4}\left(\lambda\cdot\widetilde{\eta}\right)}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}\left(\lambda\cdot\widetilde{\lambda}\right)\equiv\mathcal{A}_3^{(2)}$$



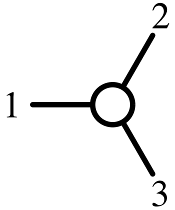
$$= \quad \frac{\delta^{1\times4}\left(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\right)}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}\left(\lambda\cdot\widetilde{\lambda}\right)\equiv\mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



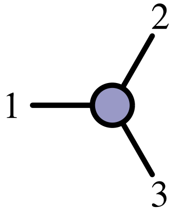
$$= \frac{\delta^{2\times4}\big(\lambda\cdot\widetilde{\eta}\big)}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}\big(\lambda\cdot\widetilde{\lambda}\big) \equiv \mathcal{A}_3^{(2)}$$



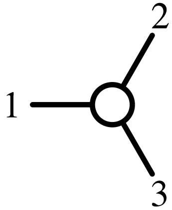
$$= \frac{\delta^{1\times4}\big(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta}\big)}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}\big(\lambda\cdot\widetilde{\lambda}\big) \equiv \mathcal{A}_3^{(1)}$$

Building Blocks: the S-Matrix for Three Massless Particles

Momentum conservation and Poincaré-invariance **uniquely** fix the kinematical dependence of the amplitude for three massless particles (to all loop orders!).



$$= \quad \frac{\delta^{2\times4}(\lambda\cdot\widetilde{\eta})}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,1\rangle}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(2)}$$



$$= \quad \frac{\delta^{1\times4}(\widetilde{\lambda}^{\perp}\cdot\widetilde{\eta})}{[1\,2]\,[2\,3]\,[3\,1]}\,\delta^{2\times2}(\lambda\cdot\widetilde{\lambda}) \equiv \mathcal{A}_3^{(1)}$$

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined
to all orders of perturbation theory

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

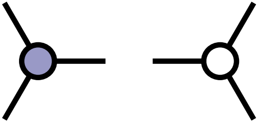
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

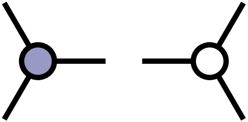
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



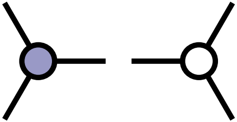
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



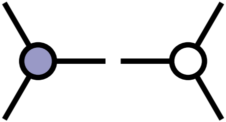
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



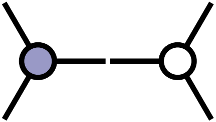
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



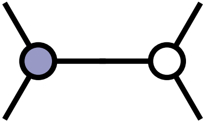
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



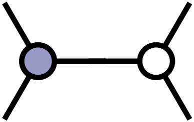
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



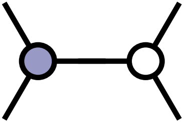
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



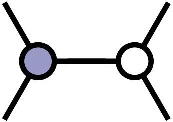
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



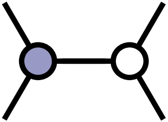
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



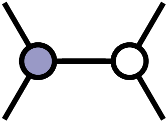
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



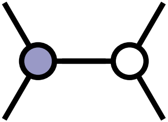
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



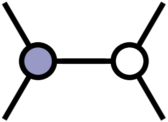
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



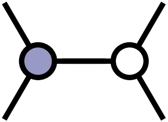
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



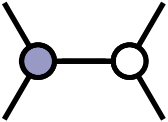
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



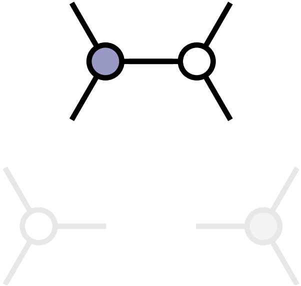
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



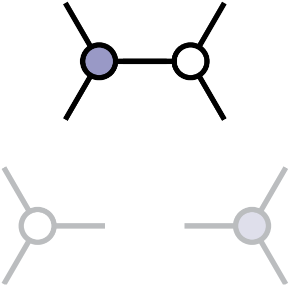
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



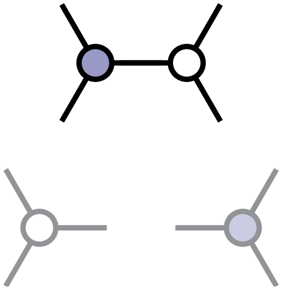
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



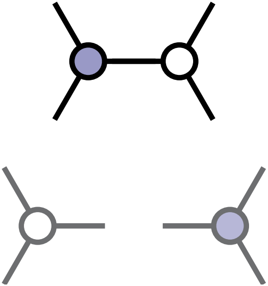
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



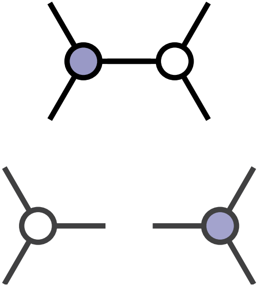
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



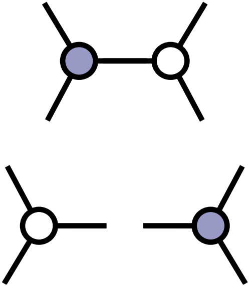
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



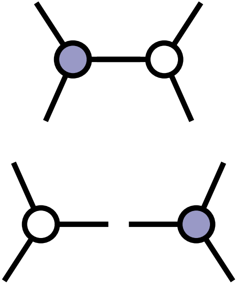
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



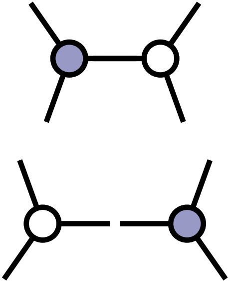
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



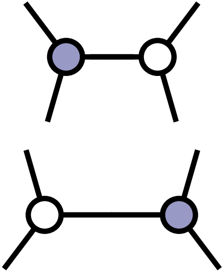
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



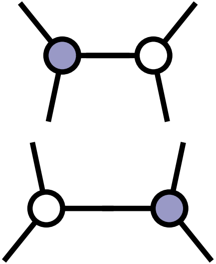
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



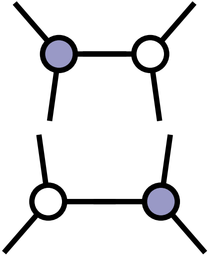
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



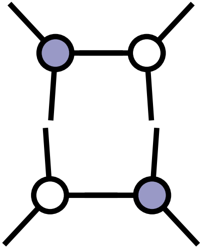
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



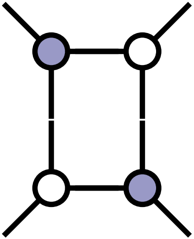
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



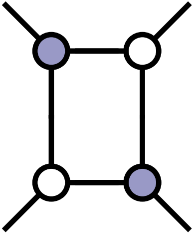
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



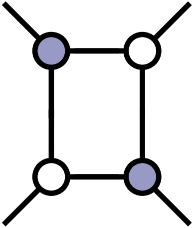
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



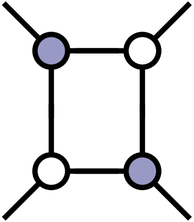
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



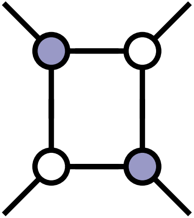
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



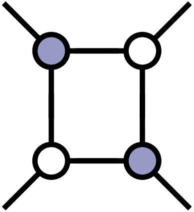
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



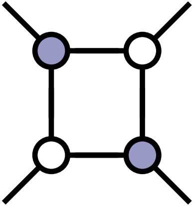
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



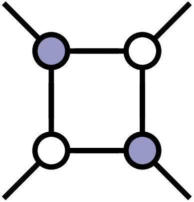
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



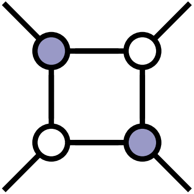
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



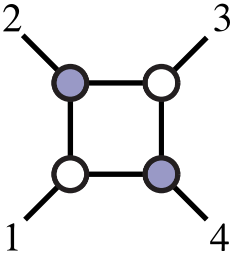
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



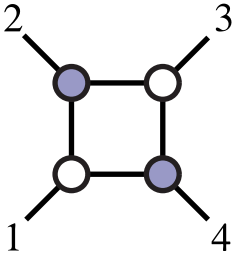
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



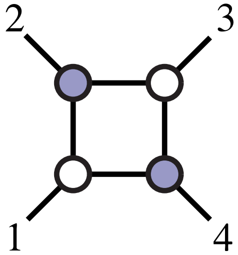
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



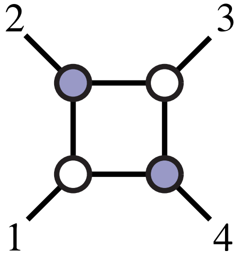
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



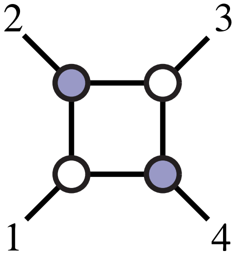
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



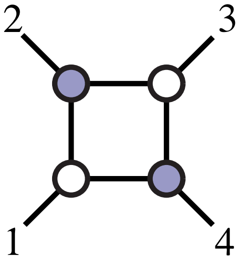
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



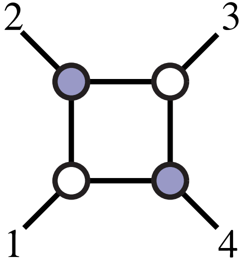
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



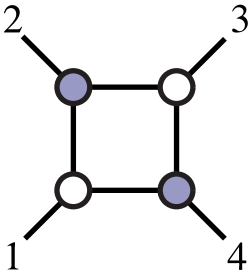
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



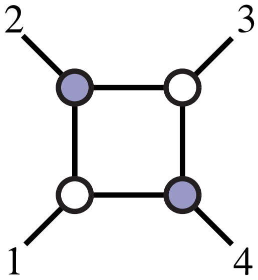
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



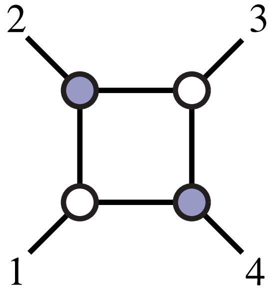
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



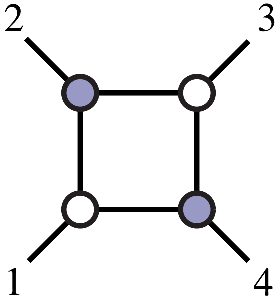
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



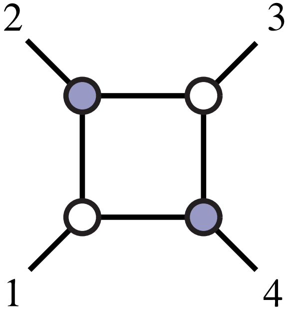
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



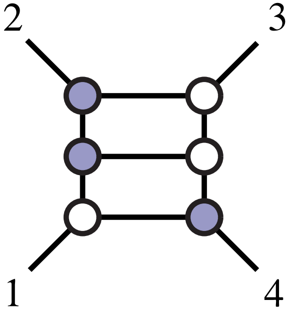
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



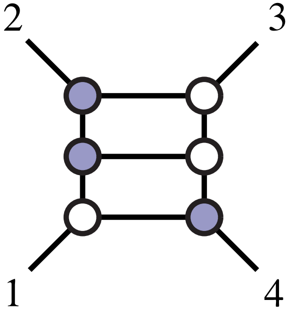
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



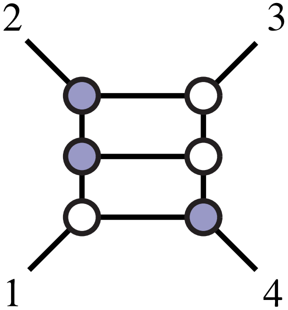
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



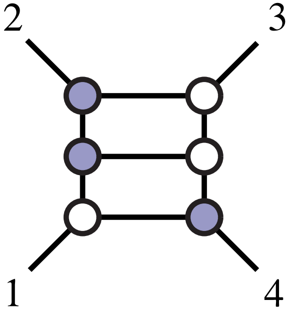
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



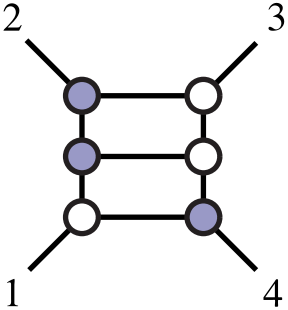
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



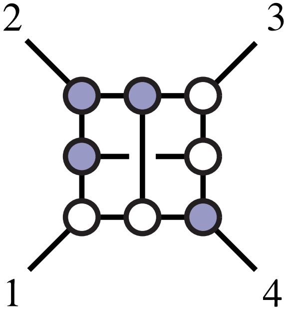
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



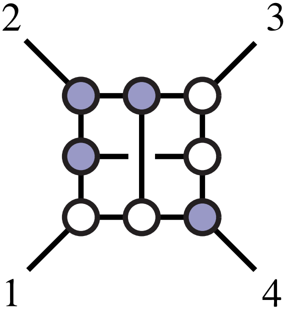
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



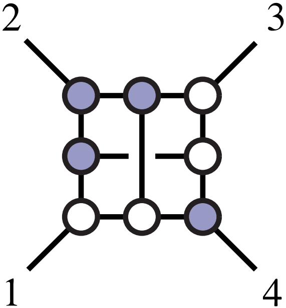
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



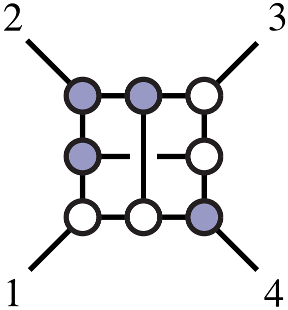
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



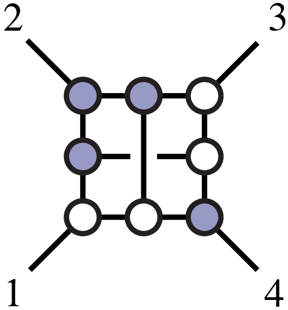
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



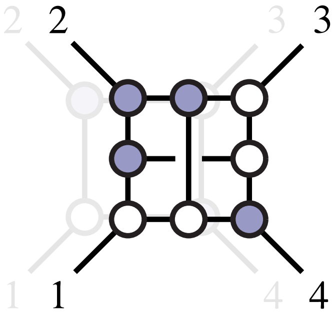
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



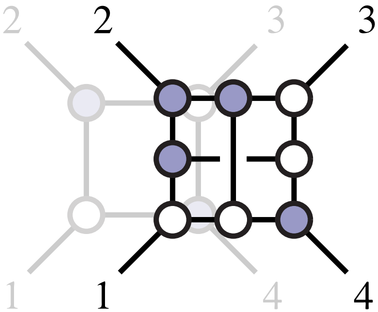
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



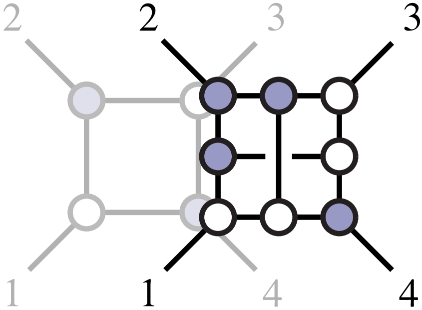
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



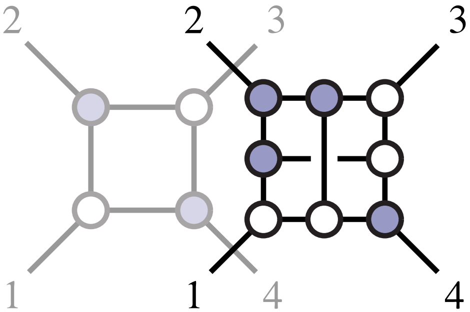
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



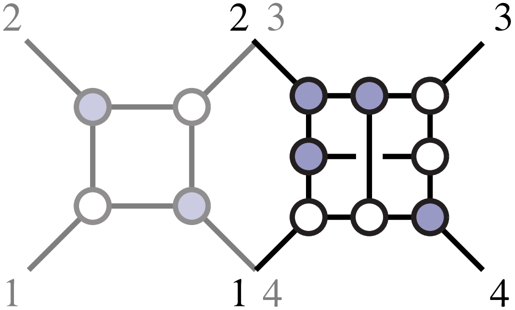
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



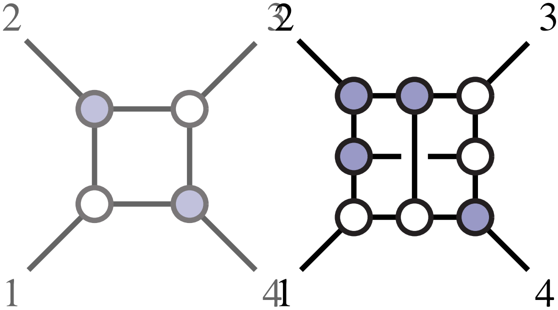
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



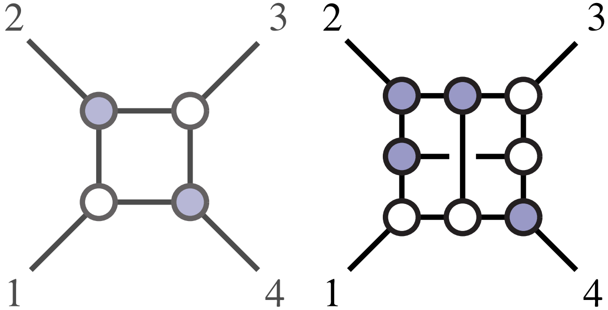
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



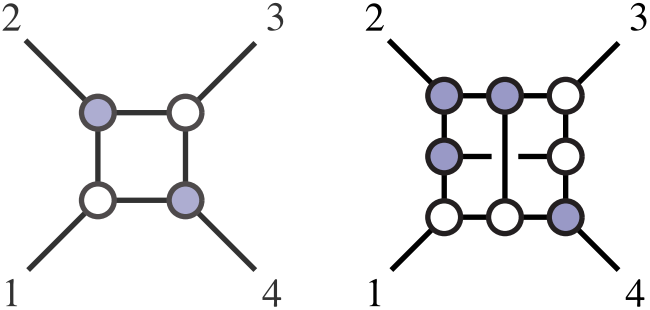
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



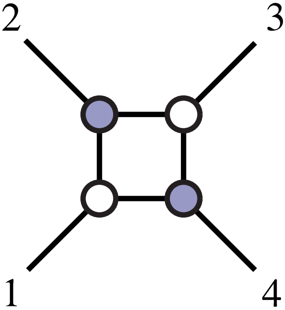
Amalgamating Diagrams from Three-Particle Amplitudes

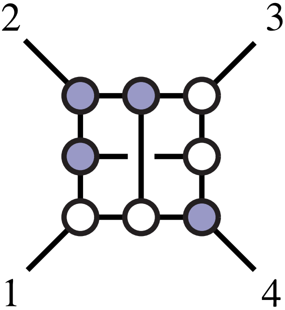
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



Amalgamating Diagrams from Three-Particle Amplitudes

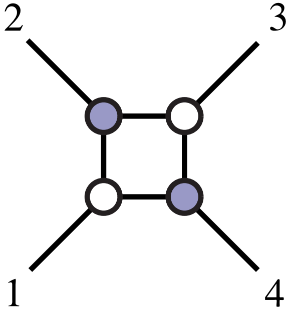
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

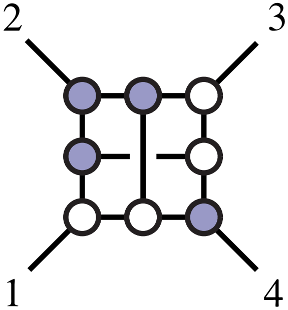




Amalgamating Diagrams from Three-Particle Amplitudes

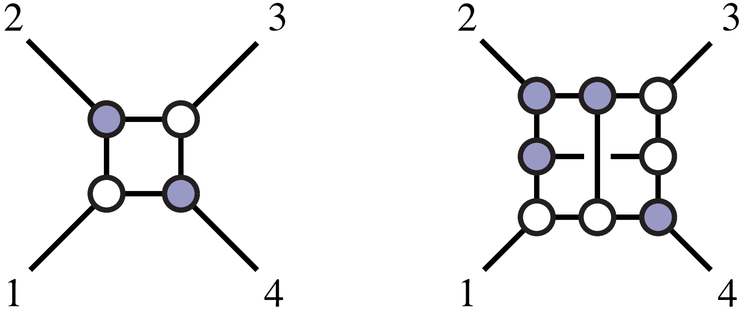
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:





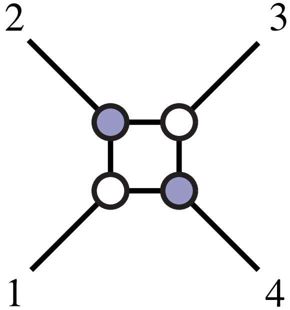
Amalgamating Diagrams from Three-Particle Amplitudes

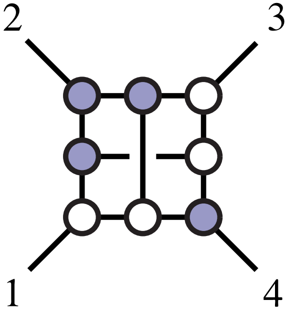
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



Amalgamating Diagrams from Three-Particle Amplitudes

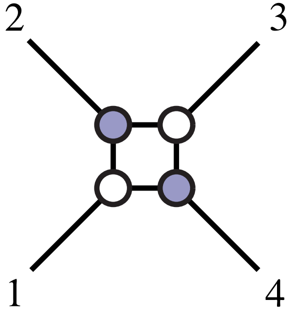
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

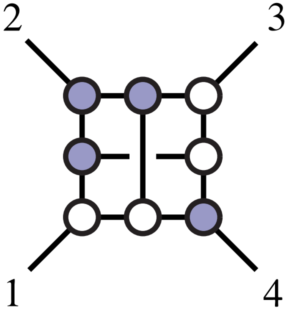




Amalgamating Diagrams from Three-Particle Amplitudes

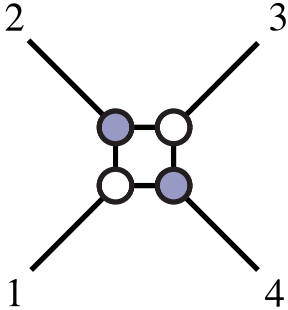
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

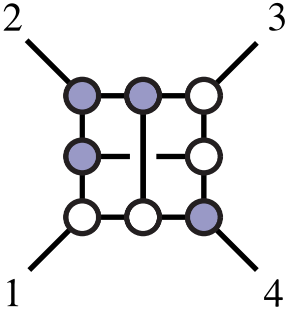




Amalgamating Diagrams from Three-Particle Amplitudes

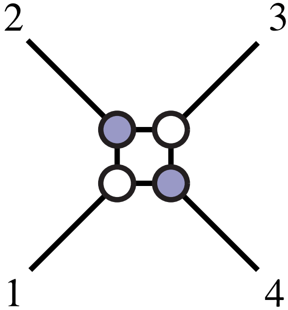
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

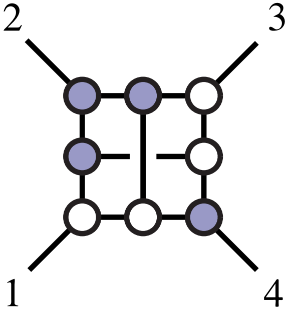




Amalgamating Diagrams from Three-Particle Amplitudes

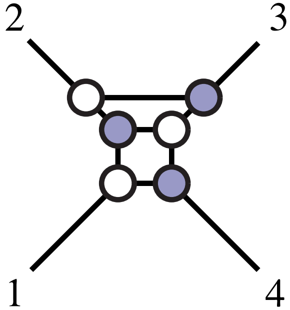
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

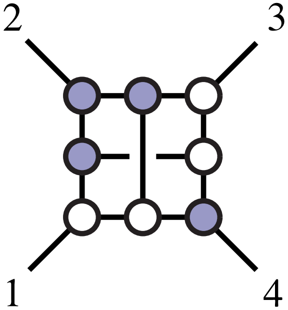




Amalgamating Diagrams from Three-Particle Amplitudes

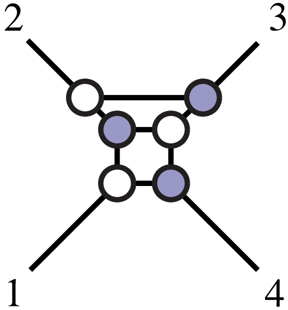
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

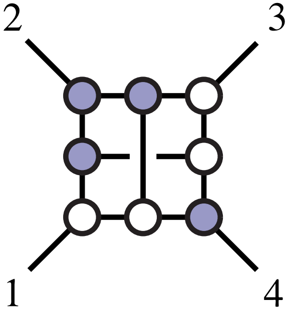




Amalgamating Diagrams from Three-Particle Amplitudes

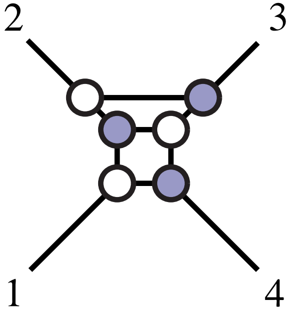
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

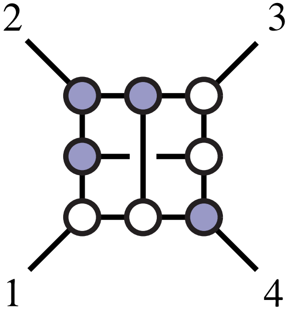




Amalgamating Diagrams from Three-Particle Amplitudes

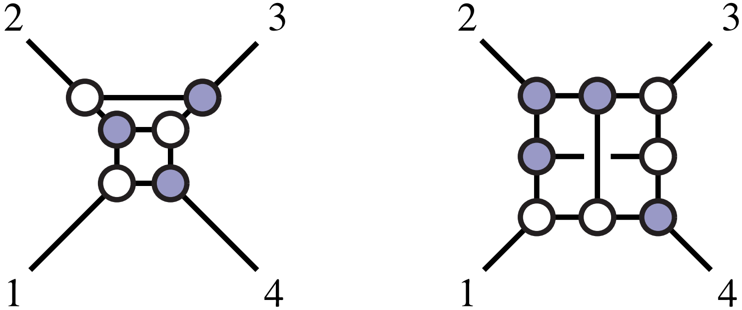
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:





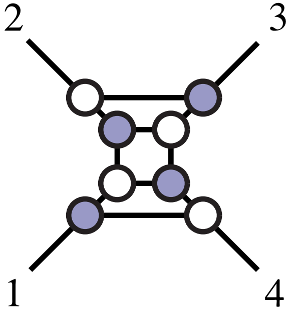
Amalgamating Diagrams from Three-Particle Amplitudes

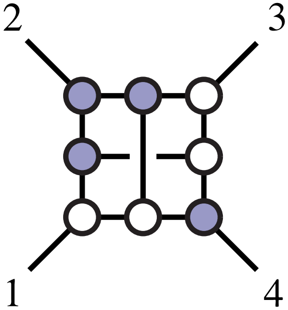
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



Amalgamating Diagrams from Three-Particle Amplitudes

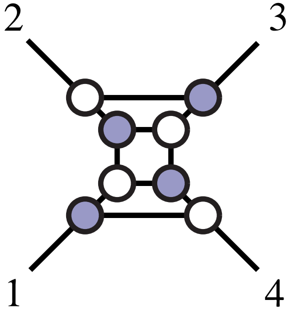
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

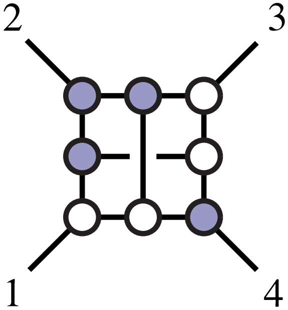




Amalgamating Diagrams from Three-Particle Amplitudes

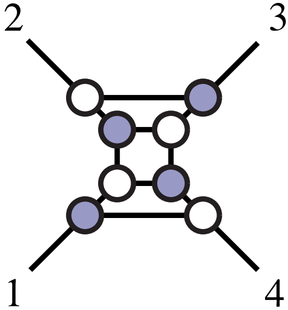
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

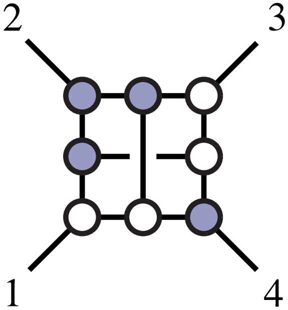




Amalgamating Diagrams from Three-Particle Amplitudes

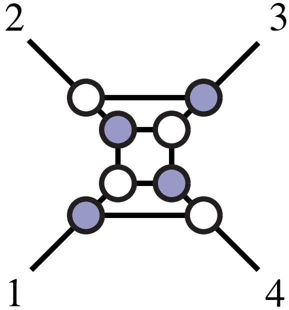
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

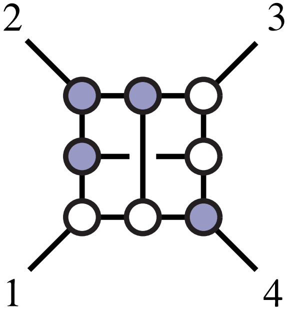




Amalgamating Diagrams from Three-Particle Amplitudes

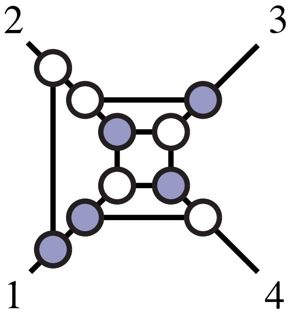
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

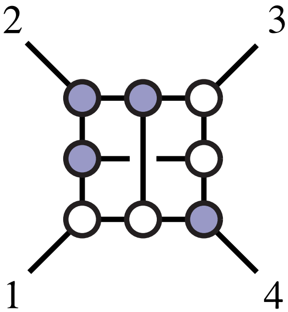




Amalgamating Diagrams from Three-Particle Amplitudes

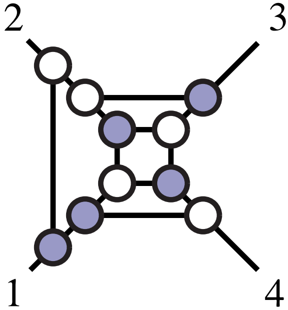
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

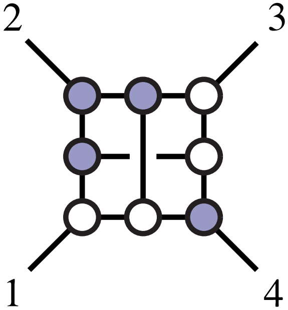




Amalgamating Diagrams from Three-Particle Amplitudes

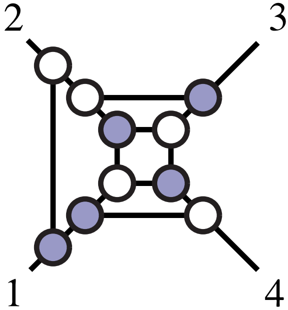
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

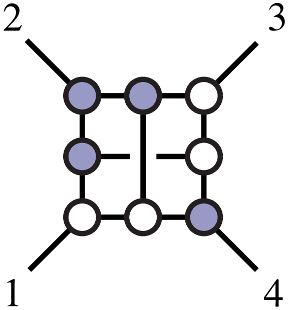




Amalgamating Diagrams from Three-Particle Amplitudes

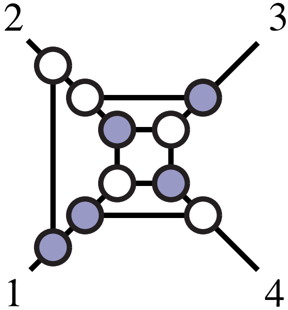
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

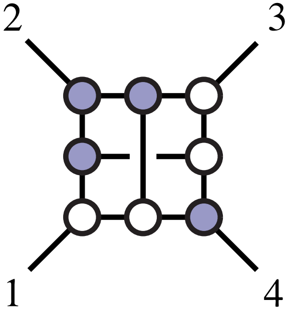




Amalgamating Diagrams from Three-Particle Amplitudes

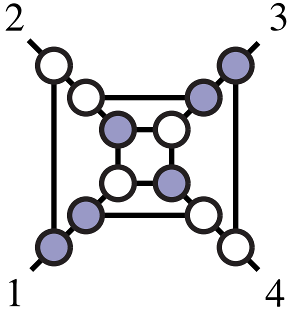
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

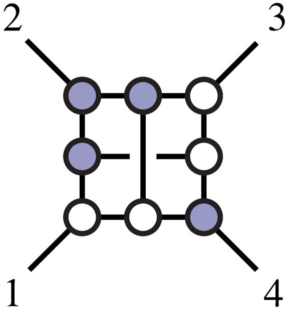




Amalgamating Diagrams from Three-Particle Amplitudes

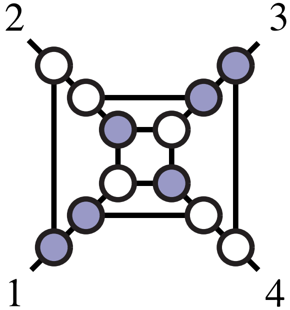
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

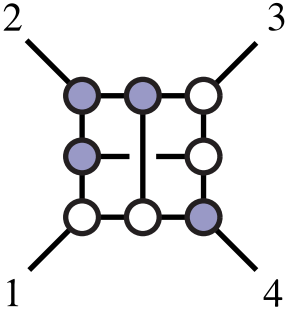




Amalgamating Diagrams from Three-Particle Amplitudes

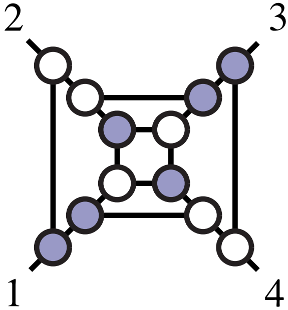
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

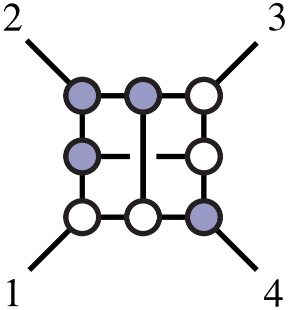




Amalgamating Diagrams from Three-Particle Amplitudes

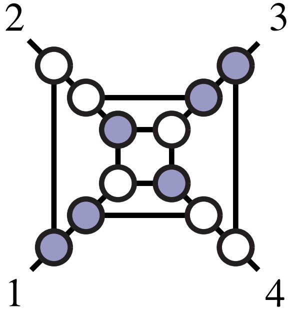
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

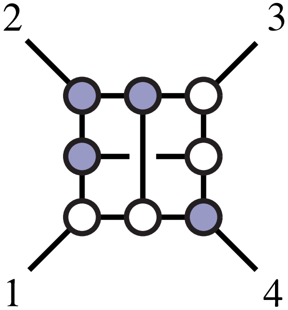




Amalgamating Diagrams from Three-Particle Amplitudes

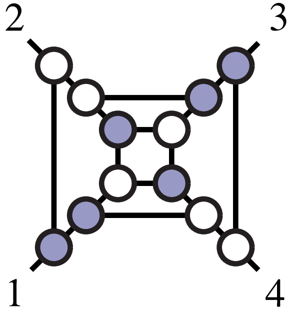
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

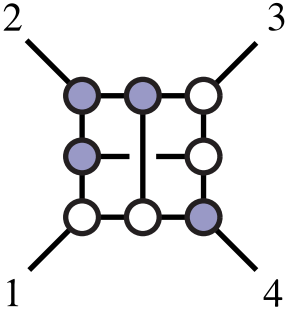




Amalgamating Diagrams from Three-Particle Amplitudes

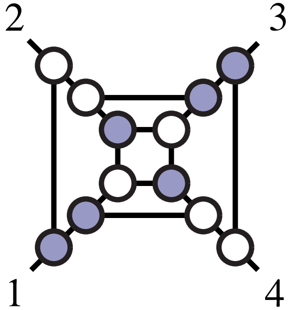
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

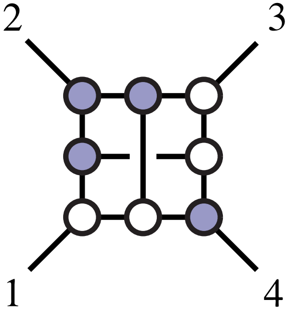




Amalgamating Diagrams from Three-Particle Amplitudes

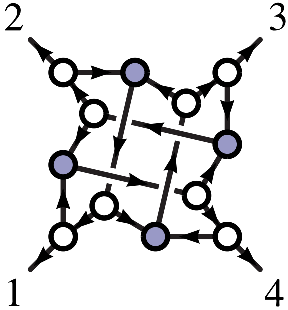
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

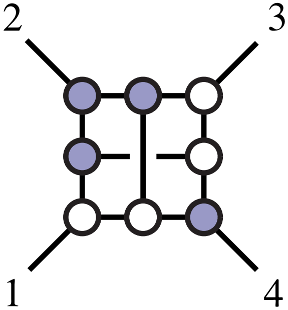




Amalgamating Diagrams from Three-Particle Amplitudes

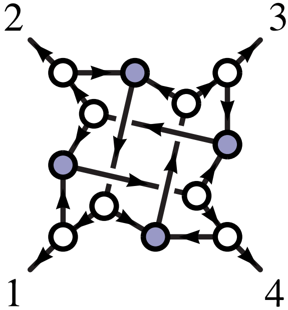
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

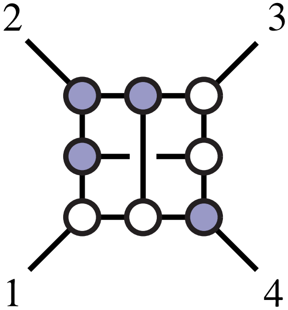




Amalgamating Diagrams from Three-Particle Amplitudes

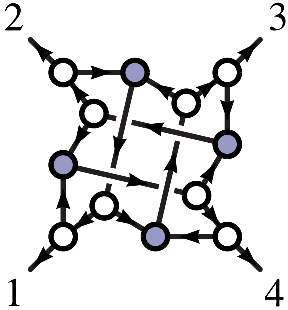
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

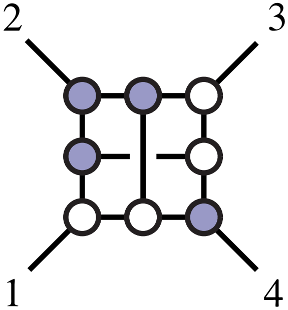




Amalgamating Diagrams from Three-Particle Amplitudes

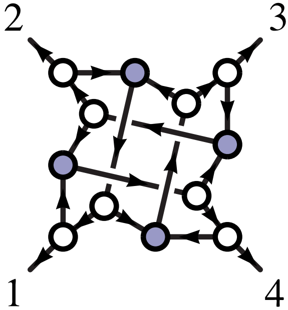
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

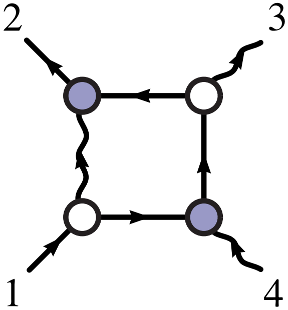




Amalgamating Diagrams from Three-Particle Amplitudes

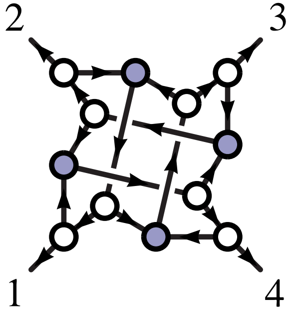
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

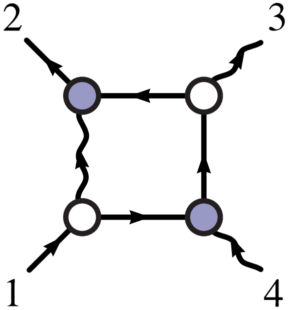




Amalgamating Diagrams from Three-Particle Amplitudes

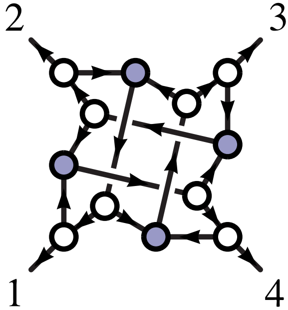
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

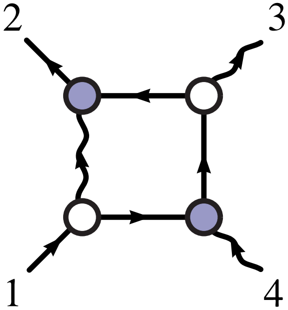




Amalgamating Diagrams from Three-Particle Amplitudes

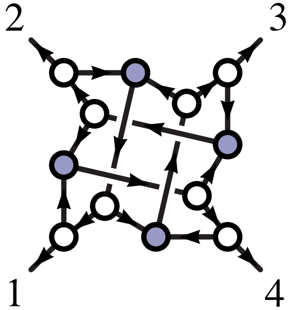
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

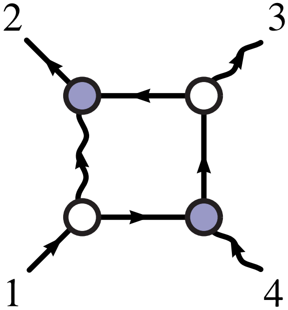




Amalgamating Diagrams from Three-Particle Amplitudes

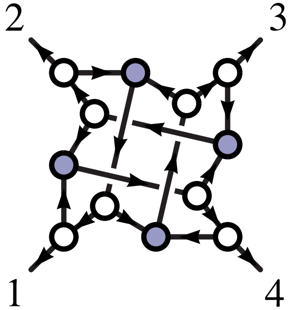
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

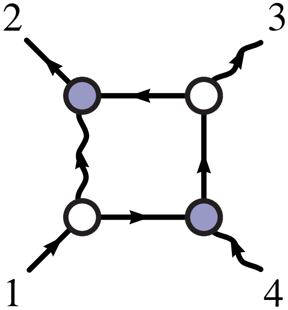




Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:





Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

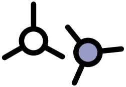
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:

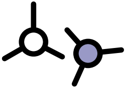
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



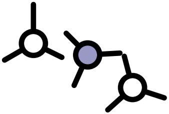
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



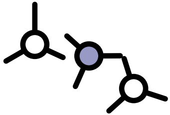
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



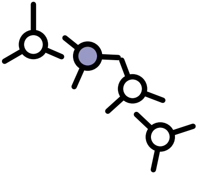
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



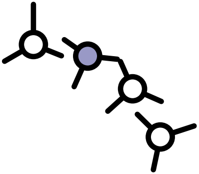
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



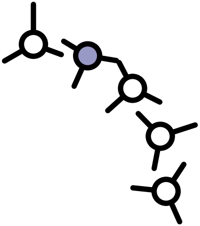
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



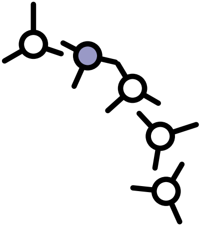
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



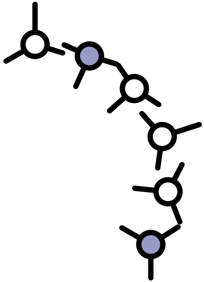
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



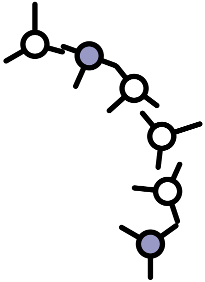
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



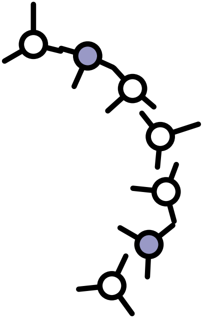
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



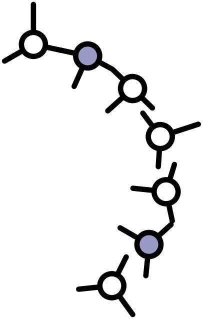
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



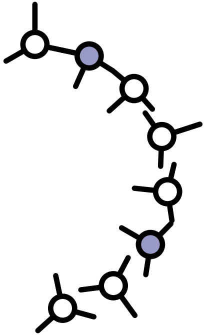
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



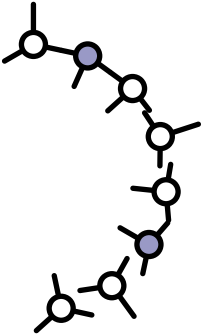
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



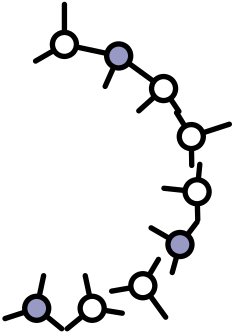
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



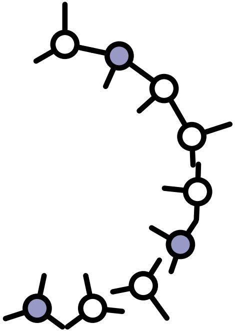
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



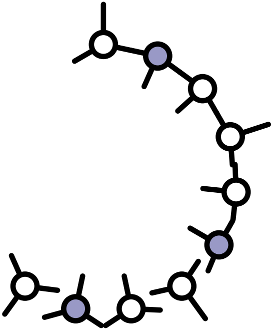
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



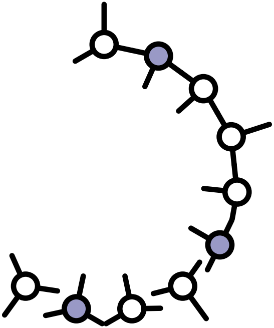
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



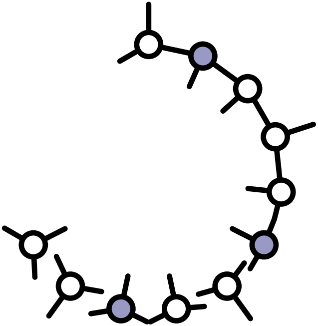
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



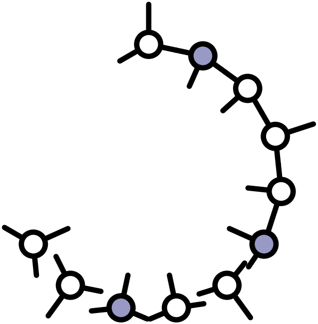
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



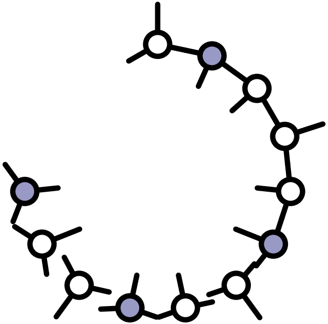
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



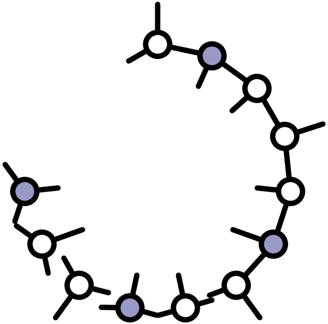
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



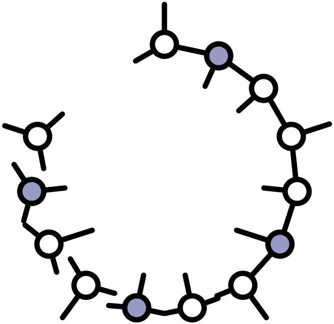
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



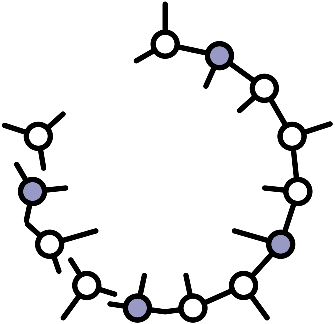
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



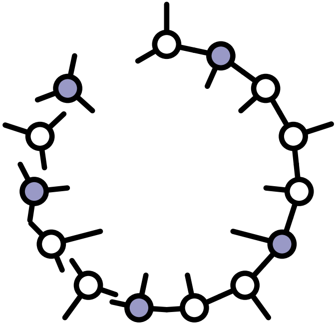
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



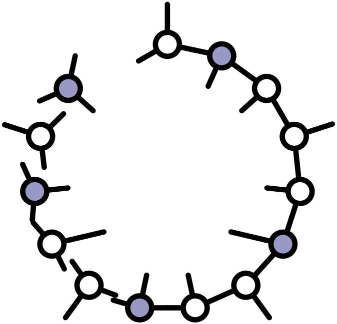
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



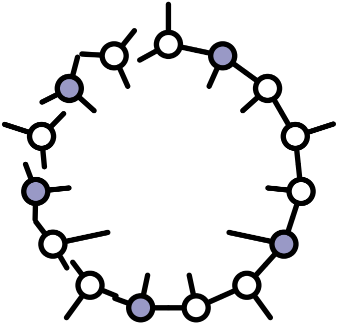
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



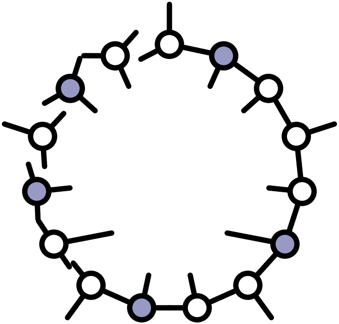
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



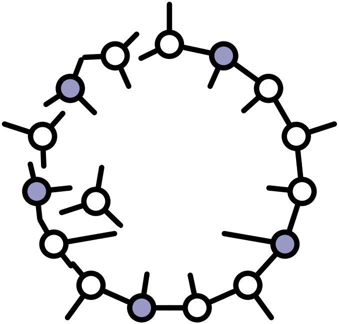
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



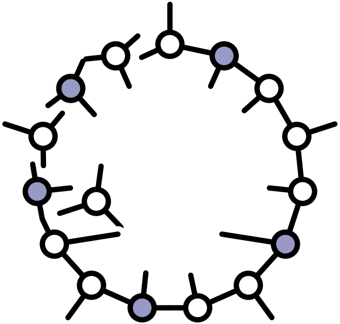
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



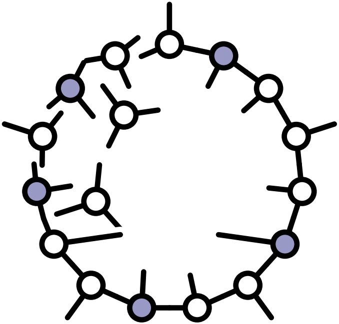
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



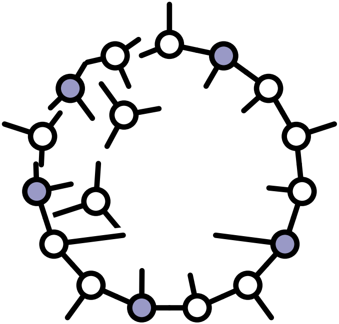
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



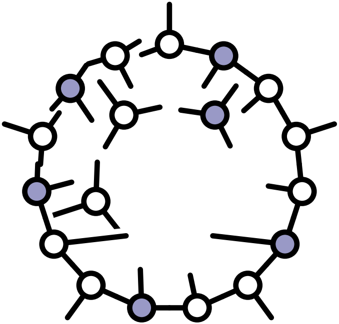
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



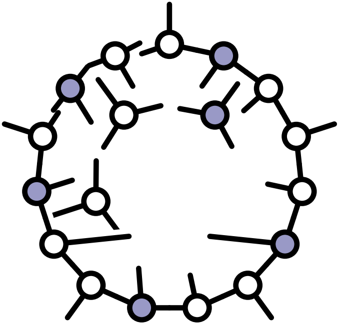
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



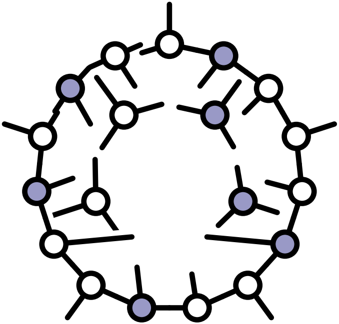
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



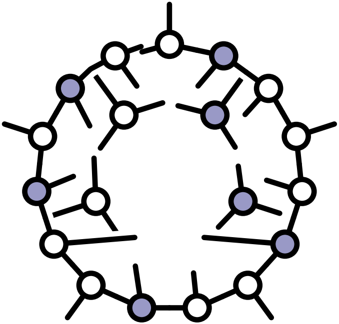
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



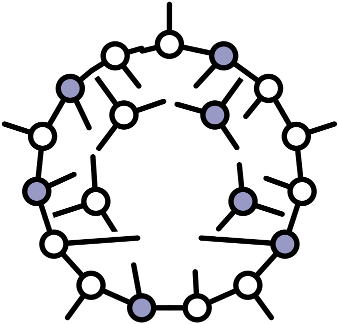
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



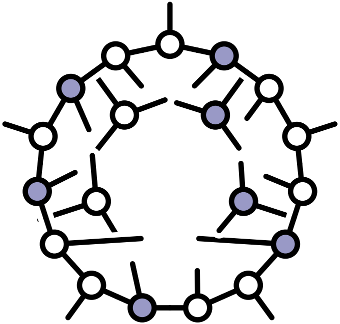
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



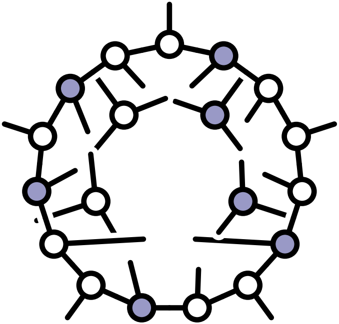
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



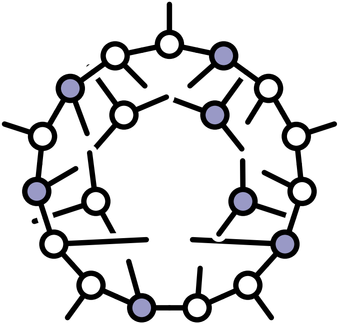
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



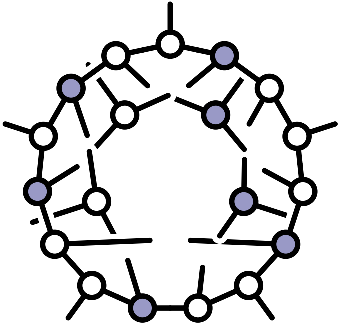
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



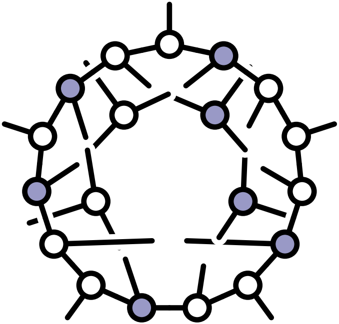
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



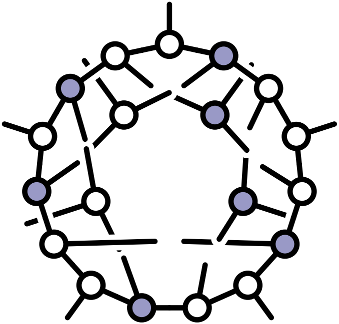
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



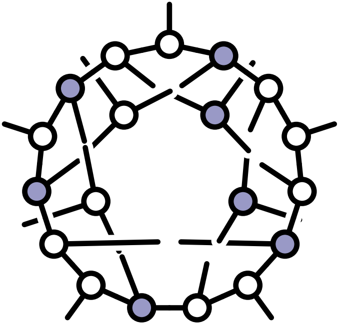
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



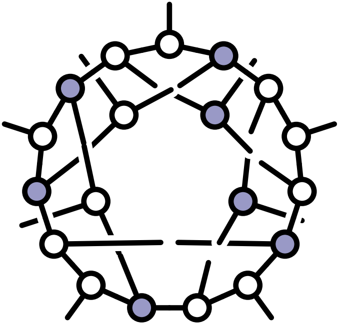
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



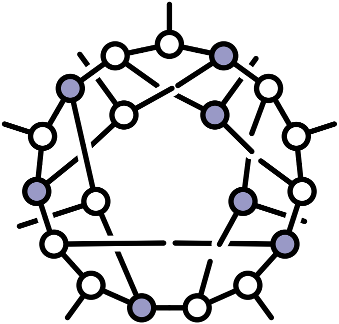
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



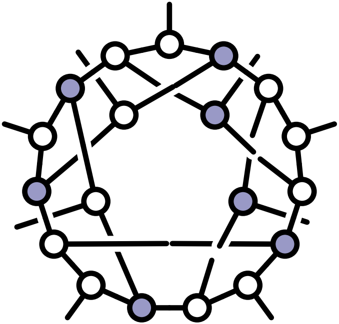
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



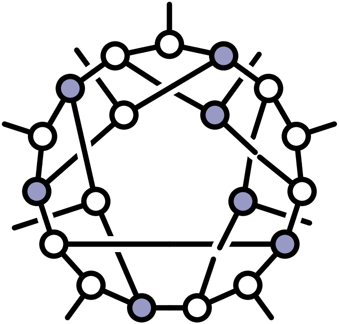
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



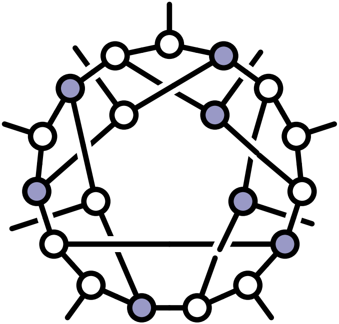
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



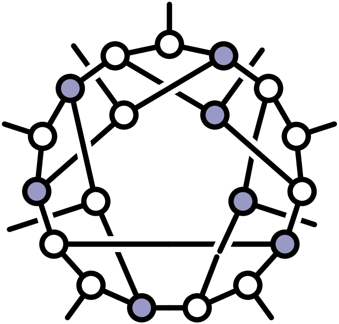
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



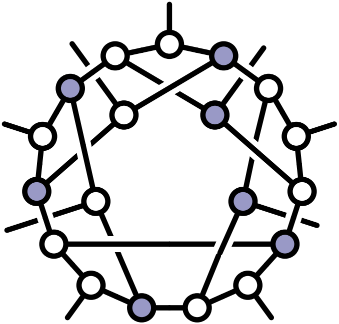
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



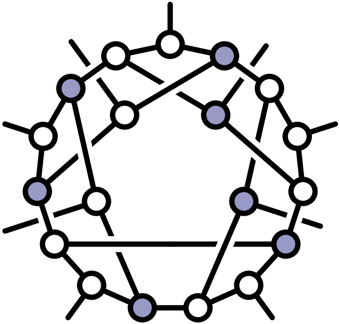
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



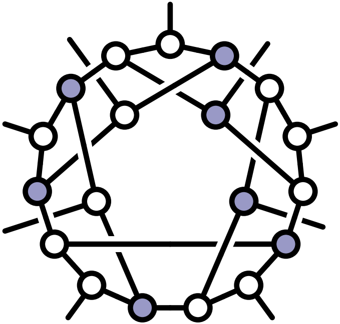
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



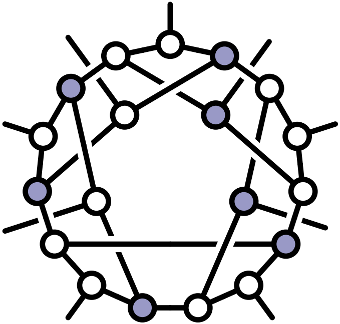
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



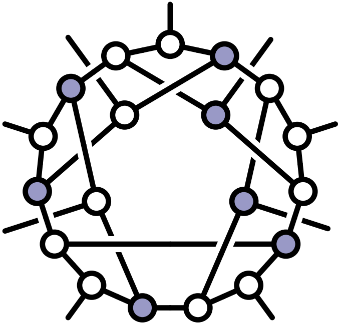
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



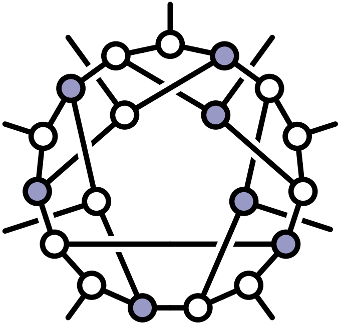
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



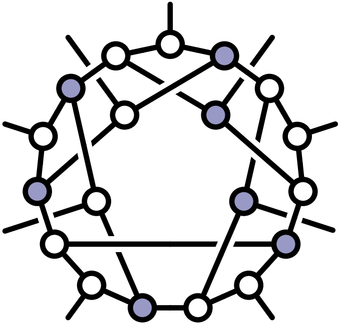
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



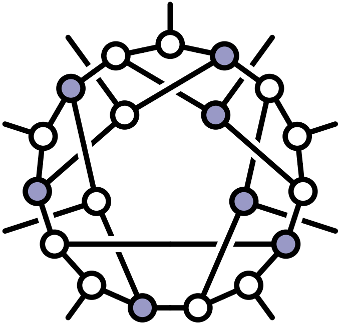
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



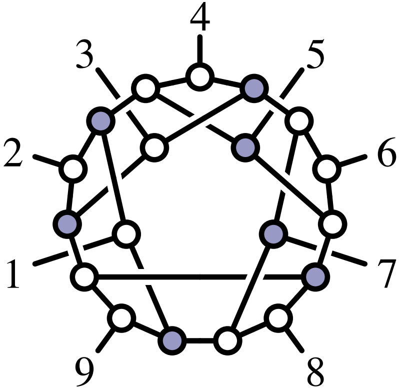
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



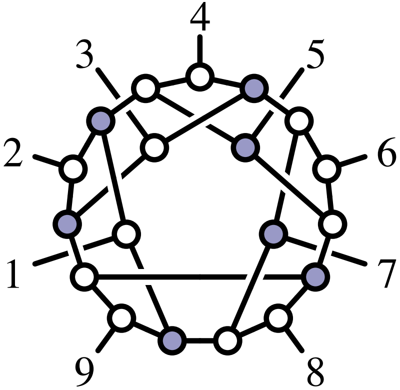
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



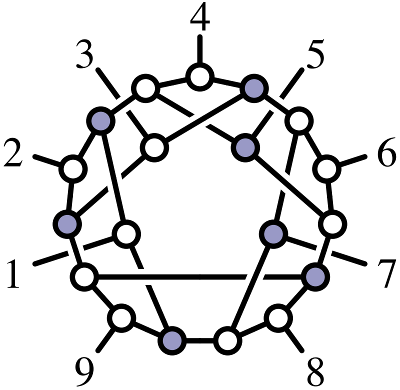
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



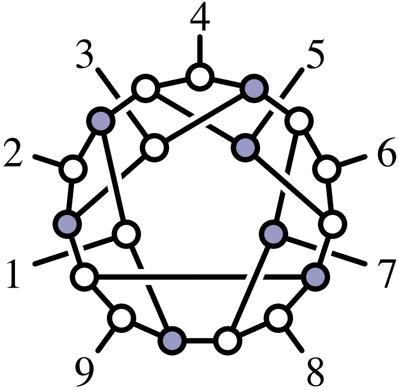
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



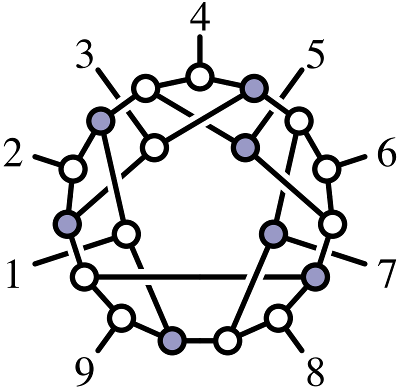
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



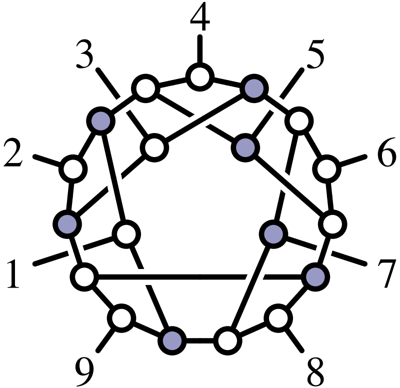
Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



Amalgamating Diagrams from Three-Particle Amplitudes

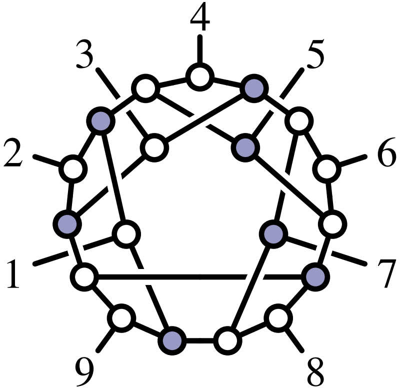
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



$$= \frac{(\langle 91\rangle\langle 23\rangle\langle 46\rangle - \langle 16\rangle\langle 34\rangle\langle 29\rangle)^2 \quad \delta^{2\times 4}(\lambda\cdot\widetilde{\eta})\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})}{\langle 12\rangle\langle 23\rangle\langle 34\rangle\langle 45\rangle\langle 56\rangle\langle 67\rangle\langle 78\rangle\langle 81\rangle\langle 14\rangle\langle 42\rangle\langle 29\rangle\langle 96\rangle\langle 63\rangle\langle 39\rangle\langle 91\rangle}$$

Amalgamating Diagrams from Three-Particle Amplitudes

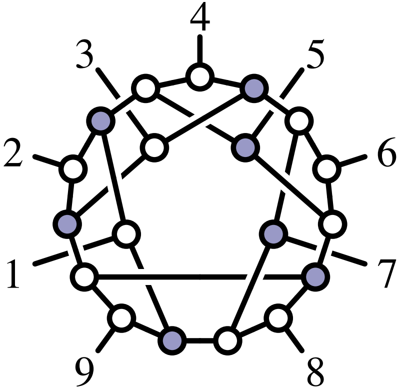
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



$$= \frac{(\langle 91\rangle\langle 23\rangle\langle 46\rangle - \langle 16\rangle\langle 34\rangle\langle 29\rangle)^2 \quad \delta^{2\times4}(\lambda\cdot\widetilde{\eta})\delta^{2\times2}(\lambda\cdot\widetilde{\lambda})}{\langle 12\rangle\langle 23\rangle\langle 34\rangle\langle 45\rangle\langle 56\rangle\langle 67\rangle\langle 78\rangle\langle 81\rangle\langle 14\rangle\langle 42\rangle\langle 29\rangle\langle 96\rangle\langle 63\rangle\langle 39\rangle\langle 91\rangle}$$

Amalgamating Diagrams from Three-Particle Amplitudes

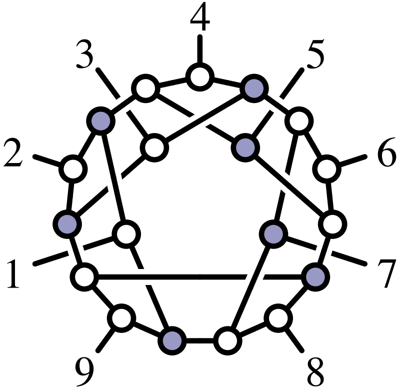
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



$$= \frac{(\langle 91\rangle\langle 23\rangle\langle 46\rangle - \langle 16\rangle\langle 34\rangle\langle 29\rangle)^2 \quad \delta^{2\times 4}(\lambda\cdot\widetilde{\eta})\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})}{\langle 12\rangle\langle 23\rangle\langle 34\rangle\langle 45\rangle\langle 56\rangle\langle 67\rangle\langle 78\rangle\langle 81\rangle\langle 14\rangle\langle 42\rangle\langle 29\rangle\langle 96\rangle\langle 63\rangle\langle 39\rangle\langle 91\rangle}$$

Amalgamating Diagrams from Three-Particle Amplitudes

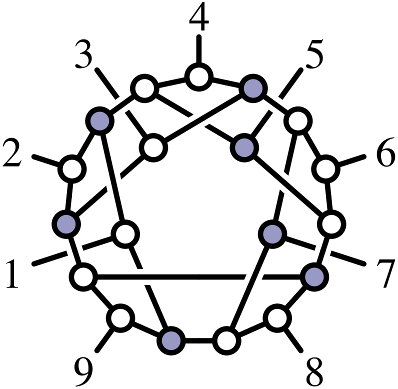
On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



$$= \frac{(\langle 91\rangle\langle 23\rangle\langle 46\rangle - \langle 16\rangle\langle 34\rangle\langle 29\rangle)^2 \quad \delta^{2\times 4}(\lambda\!\cdot\!\widetilde{\eta})\delta^{2\times 2}(\lambda\!\cdot\!\widetilde{\lambda})}{\langle 12\rangle\langle 23\rangle\langle 34\rangle\langle 45\rangle\langle 56\rangle\langle 67\rangle\langle 78\rangle\langle 81\rangle\langle 14\rangle\langle 42\rangle\langle 29\rangle\langle 96\rangle\langle 63\rangle\langle 39\rangle\langle 91\rangle}$$

Amalgamating Diagrams from Three-Particle Amplitudes

On-shell diagrams built out of only **three-particle amplitudes** are well-defined to all orders of perturbation theory, generating a large class of functions:



$$= \frac{(\langle 91\rangle\langle 23\rangle\langle 46\rangle - \langle 16\rangle\langle 34\rangle\langle 29\rangle)^2 \quad \delta^{2\times 4}(\lambda\cdot\widetilde{\eta})\delta^{2\times 2}(\lambda\cdot\widetilde{\lambda})}{\langle 12\rangle\langle 23\rangle\langle 34\rangle\langle 45\rangle\langle 56\rangle\langle 67\rangle\langle 78\rangle\langle 81\rangle\langle 14\rangle\langle 42\rangle\langle 29\rangle\langle 96\rangle\langle 63\rangle\langle 39\rangle\langle 91\rangle}$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

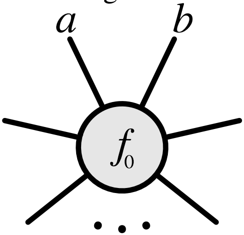
Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

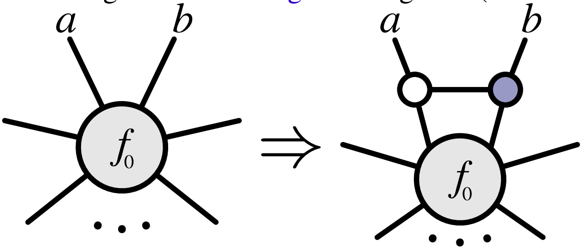
Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



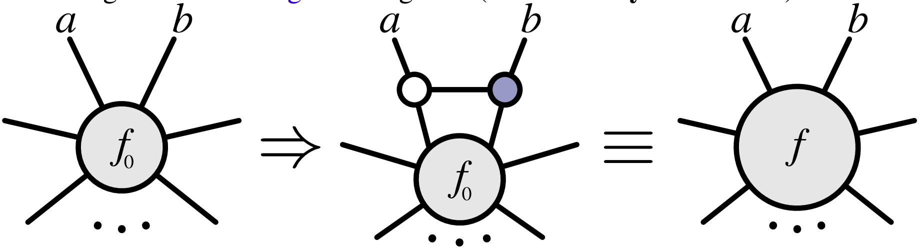
Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



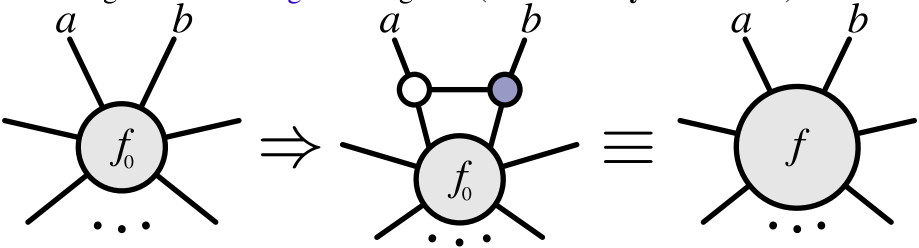
Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Building-Up On-Shell Diagrams with “BCFW” Bridges

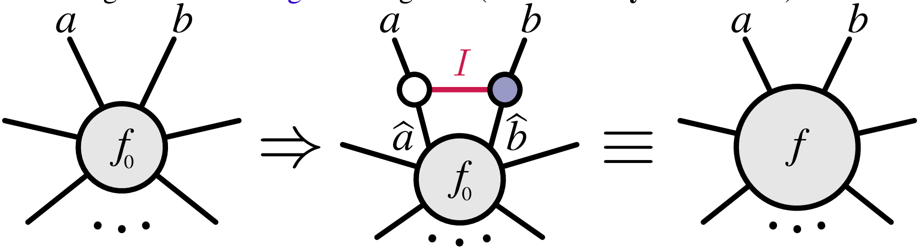
Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

Building-Up On-Shell Diagrams with “BCFW” Bridges

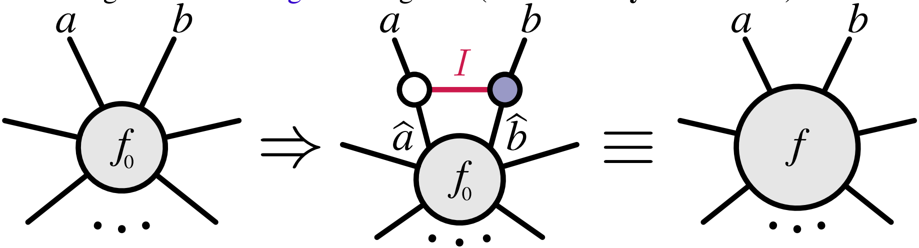
Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

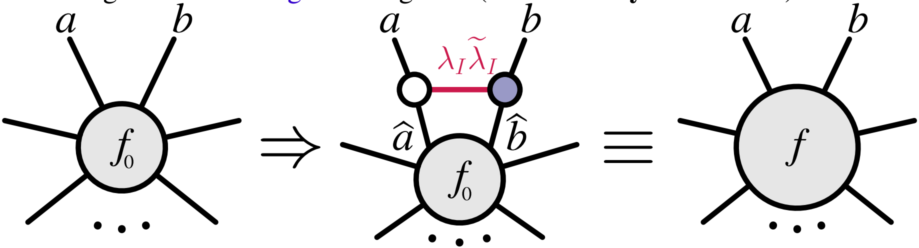


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\hat{a}}\widetilde{\lambda}_{\hat{a}} = \lambda_a\widetilde{\lambda}_a - \lambda_I\widetilde{\lambda}_I \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\hat{b}}\widetilde{\lambda}_{\hat{b}} = \lambda_b\widetilde{\lambda}_b + \lambda_I\widetilde{\lambda}_I\,,$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

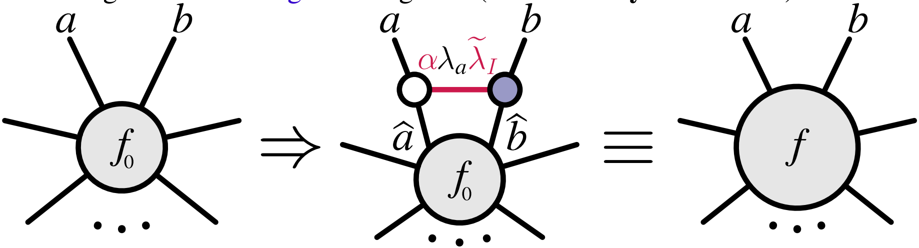


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\hat{a}}\widetilde{\lambda}_{\hat{a}} = \lambda_a\widetilde{\lambda}_a - \lambda_I\widetilde{\lambda}_I \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\hat{b}}\widetilde{\lambda}_{\hat{b}} = \lambda_b\widetilde{\lambda}_b + \lambda_I\widetilde{\lambda}_I\,,$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

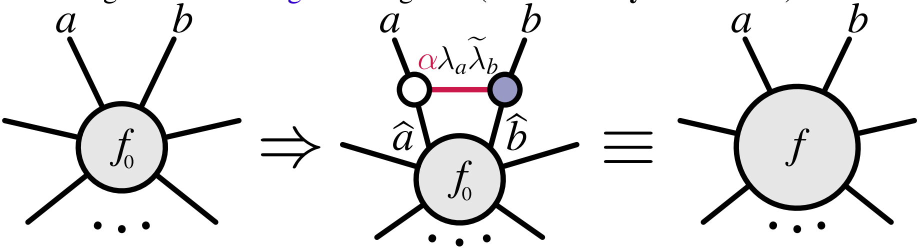


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\hat{a}}\widetilde{\lambda}_{\hat{a}} = \lambda_a\widetilde{\lambda}_a - \alpha\lambda_a\widetilde{\lambda}_I \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\hat{b}}\widetilde{\lambda}_{\hat{b}} = \lambda_b\widetilde{\lambda}_b + \alpha\lambda_a\widetilde{\lambda}_I\,,$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

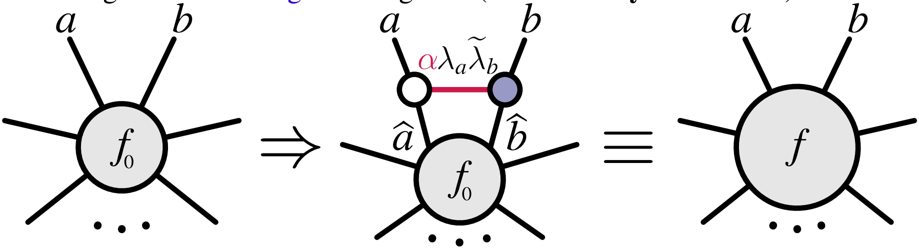


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\hat{a}}\widetilde{\lambda}_{\hat{a}} = \lambda_a\widetilde{\lambda}_a - \alpha\lambda_a\widetilde{\lambda}_b \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\hat{b}}\widetilde{\lambda}_{\hat{b}} = \lambda_b\widetilde{\lambda}_b + \alpha\lambda_a\widetilde{\lambda}_b,$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

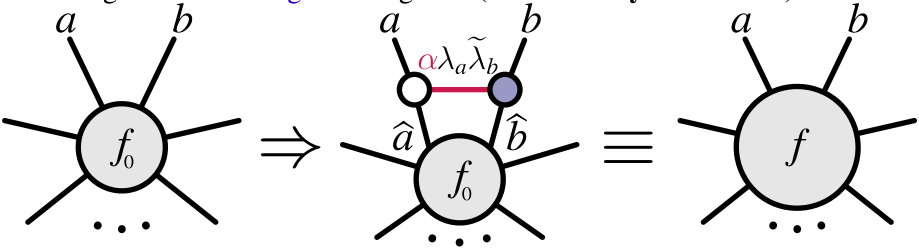


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\hat{a}}\widetilde{\lambda}_{\hat{a}} = \lambda_a\left(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\right) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\hat{b}}\widetilde{\lambda}_{\hat{b}} = \lambda_b\widetilde{\lambda}_b + \alpha\lambda_a\widetilde{\lambda}_b,$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):

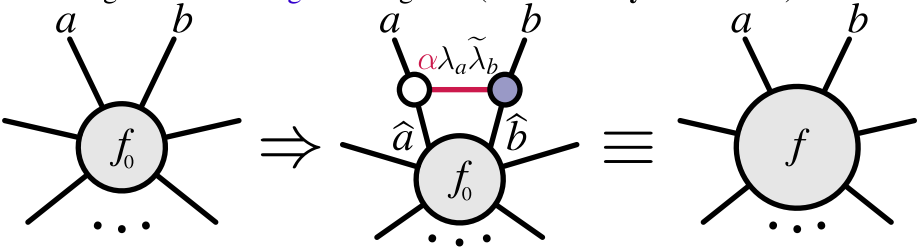


Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\widehat{a}}\widetilde{\lambda}_{\widehat{a}} = \lambda_a\big(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\big) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\widehat{b}}\widetilde{\lambda}_{\widehat{b}} = \big(\lambda_b + \alpha\lambda_a\big)\widetilde{\lambda}_b,$$

Building-Up On-Shell Diagrams with "BCFW" Bridges

Very complex on-shell diagrams can be constructed by successively adding "BCFW" bridges to diagrams (an **extremely** useful tool!):



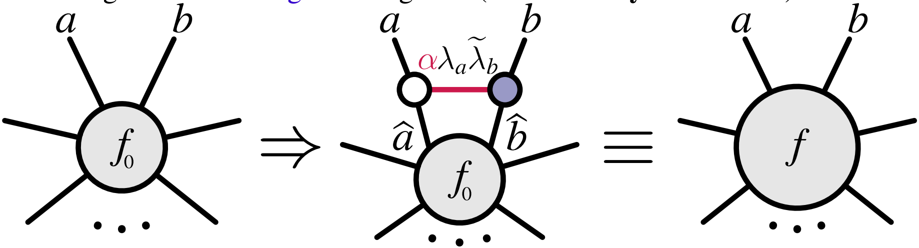
Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\widehat{a}}\widetilde{\lambda}_{\widehat{a}} = \lambda_a\big(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\big) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\widehat{b}}\widetilde{\lambda}_{\widehat{b}} = \big(\lambda_b + \alpha\lambda_a\big)\widetilde{\lambda}_b,$$

introducing a new parameter α, in terms of which we may write:

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

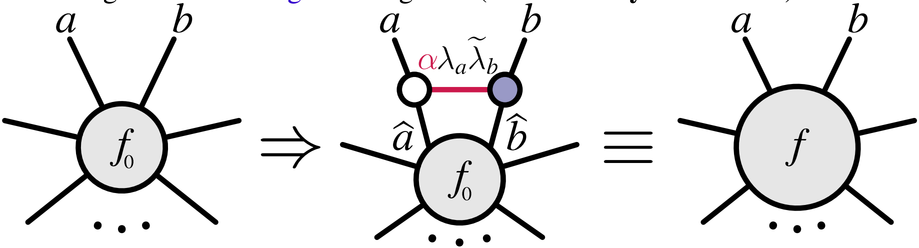
$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\widehat{a}}\widetilde{\lambda}_{\widehat{a}} = \lambda_a\big(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\big) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\widehat{b}}\widetilde{\lambda}_{\widehat{b}} = \big(\lambda_b + \alpha\lambda_a\big)\widetilde{\lambda}_b,$$

introducing a new parameter α, in terms of which we may write:

$$\boxed{f(\ldots,a,b,\ldots) = \frac{d\alpha}{\alpha} f_0(\ldots,\widehat{a},\widehat{b},\ldots)}$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

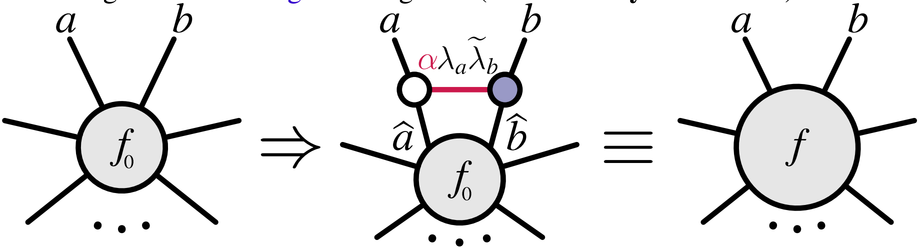
$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\widehat{a}}\widetilde{\lambda}_{\widehat{a}} = \lambda_a\big(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\big) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\widehat{b}}\widetilde{\lambda}_{\widehat{b}} = \big(\lambda_b + \alpha\lambda_a\big)\widetilde{\lambda}_b,$$

introducing a new parameter α, in terms of which we may write:

$$f(\ldots,a,b,\ldots) = \frac{d\alpha}{\alpha} f_0(\ldots,\widehat{a},\widehat{b},\ldots)$$

Building-Up On-Shell Diagrams with “BCFW” Bridges

Very complex on-shell diagrams can be constructed by successively adding “BCFW” bridges to diagrams (an **extremely** useful tool!):



Adding the bridge has the effect of shifting the momenta p_a and p_b flowing into the diagram f_0 according to:

$$\lambda_a\widetilde{\lambda}_a \mapsto \lambda_{\widehat{a}}\widetilde{\lambda}_{\widehat{a}} = \lambda_a\big(\widetilde{\lambda}_a - \alpha\widetilde{\lambda}_b\big) \quad \text{and} \quad \lambda_b\widetilde{\lambda}_b \mapsto \lambda_{\widehat{b}}\widetilde{\lambda}_{\widehat{b}} = \big(\lambda_b + \alpha\lambda_a\big)\widetilde{\lambda}_b,$$

introducing a new parameter α, in terms of which we may write:

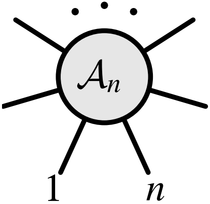
$$\boxed{f(\ldots,a,b,\ldots) = \frac{d\alpha}{\alpha} f_0(\ldots,\widehat{a},\widehat{b},\ldots)}$$

The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full *n*-particle scattering amplitude

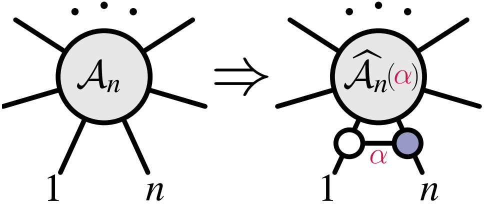
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude:



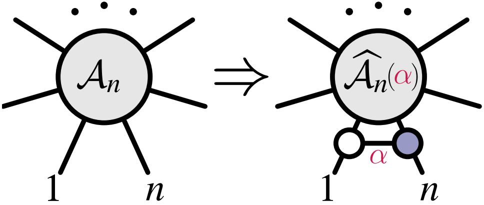
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude:



The Analytic Bootstrap: All-Loop Recursion Relations

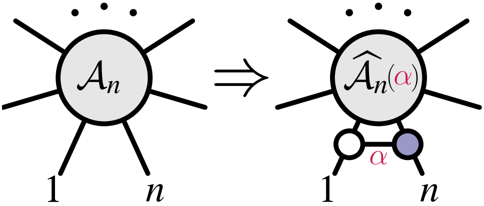
Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:



The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

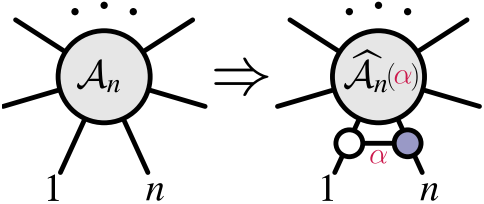


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n=\widehat{\mathcal{A}}_n(\alpha\to0)\propto\oint\limits_{\alpha=0}\frac{d\alpha}{\alpha}\,\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:

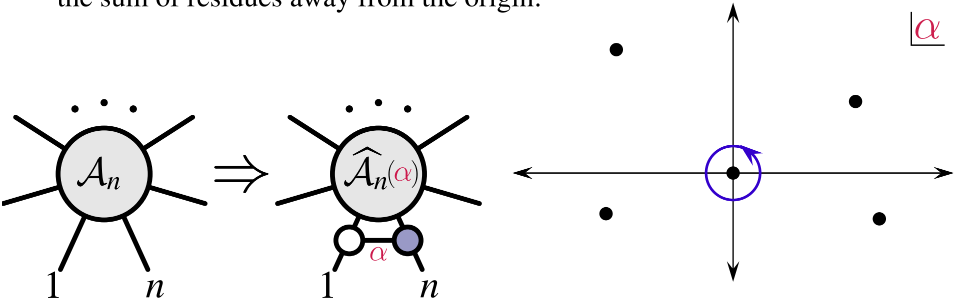


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:

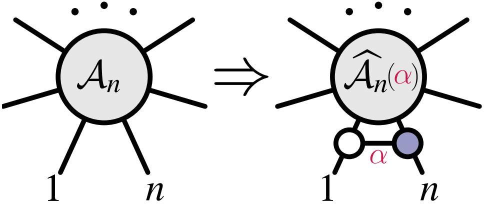


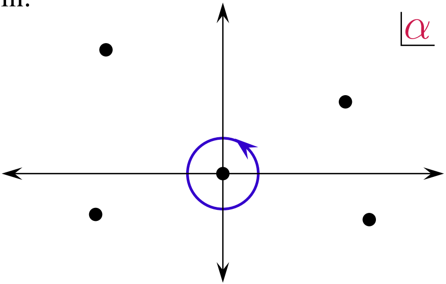
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude
the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha\!=\!0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\!\to\!0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha\!=\!0$ for (minus) the sum of residues away from the origin:



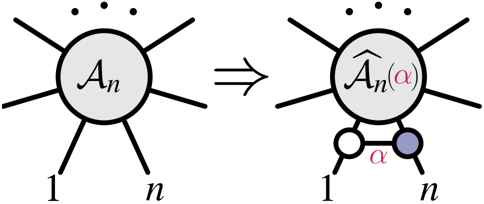


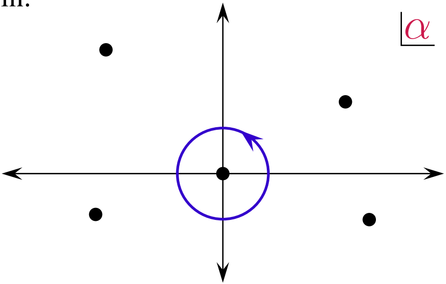
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



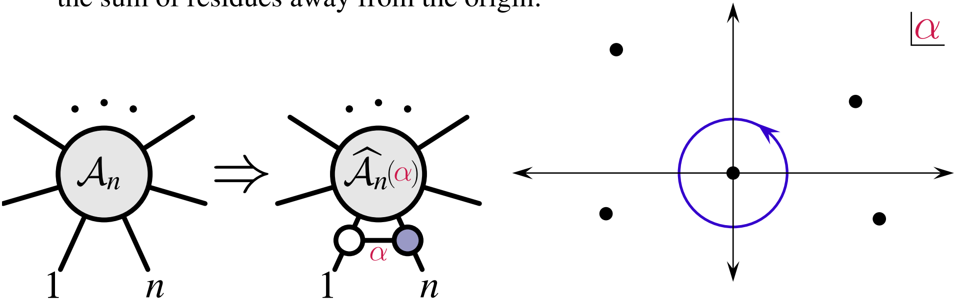


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:

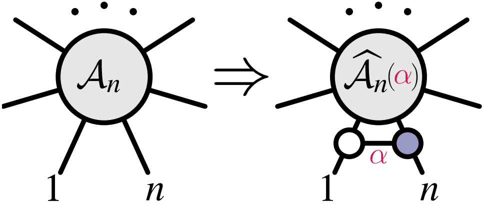


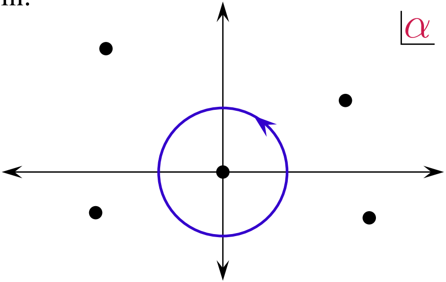
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



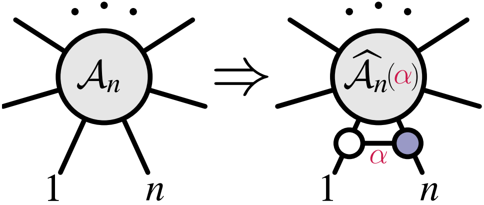


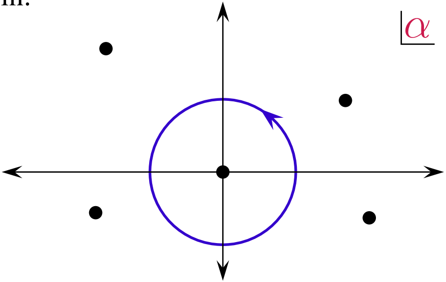
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



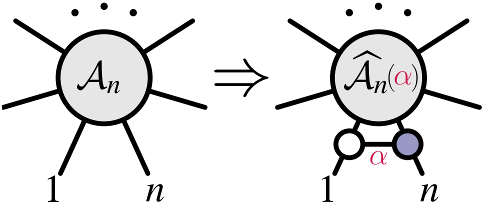


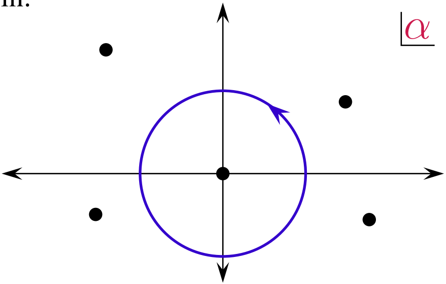
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude
the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



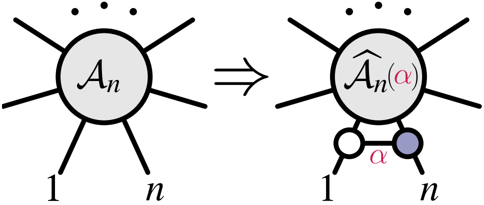


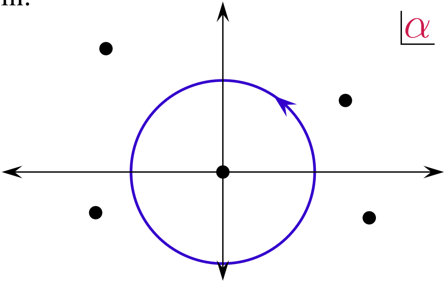
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



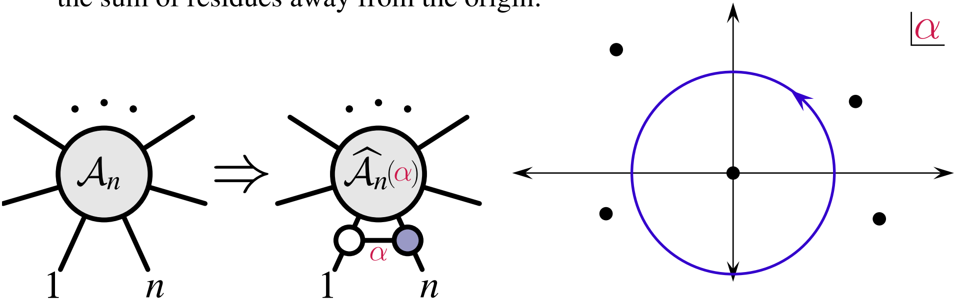


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:

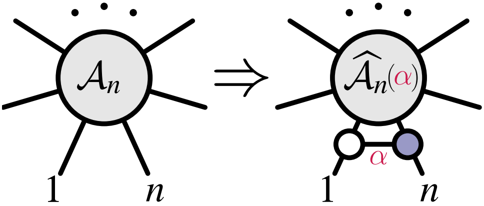


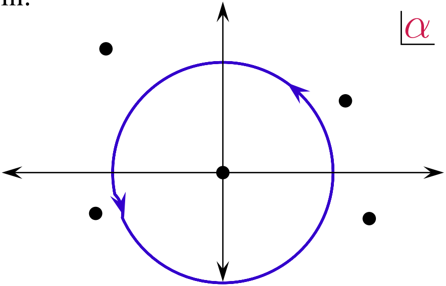
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to0) \propto \oint\limits_{\alpha=0}\frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



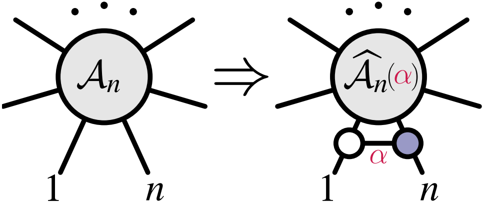


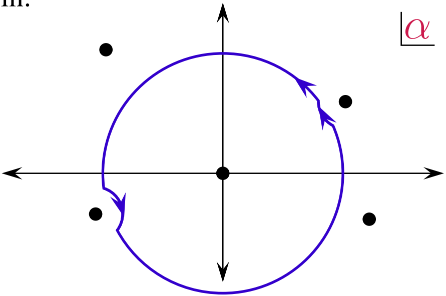
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude
 the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus)
 the sum of residues away from the origin:



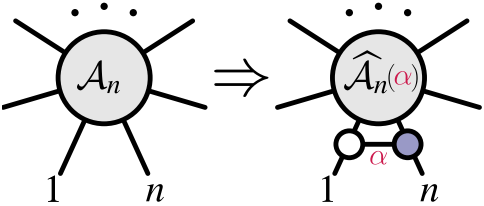


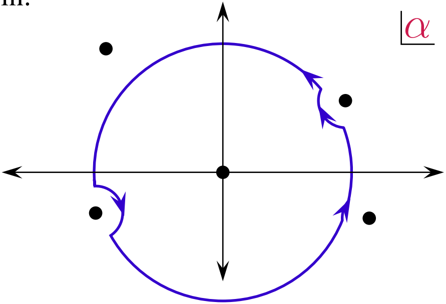
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



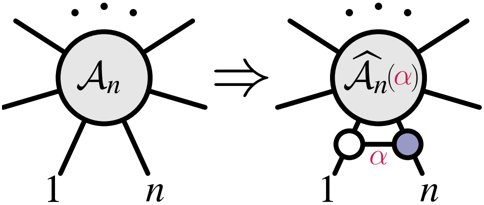


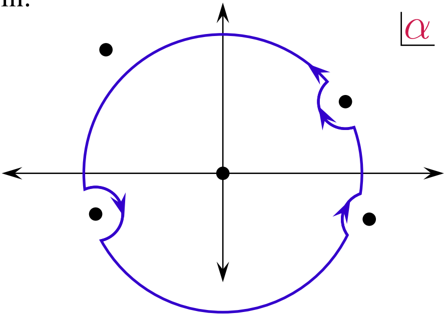
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha\!=\!0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\!\to\!0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha\!=\!0$ for (minus) the sum of residues away from the origin:



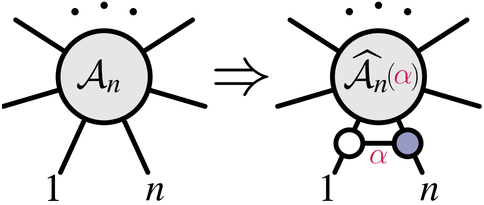


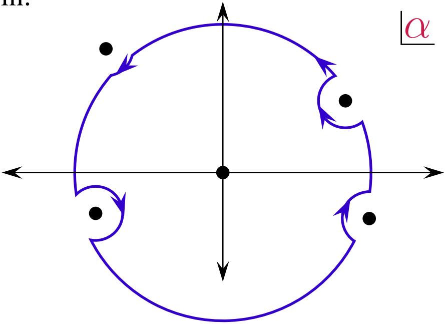
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



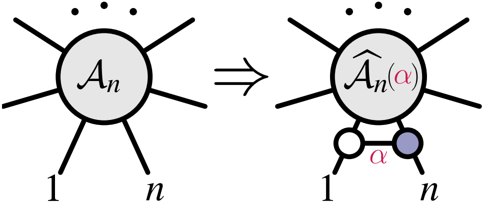


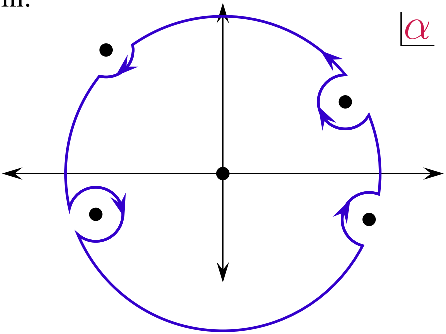
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



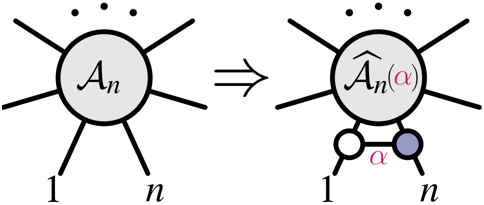


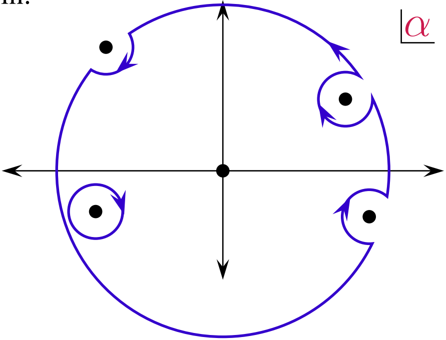
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



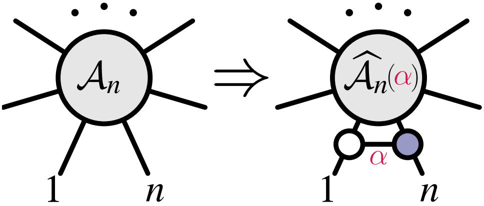


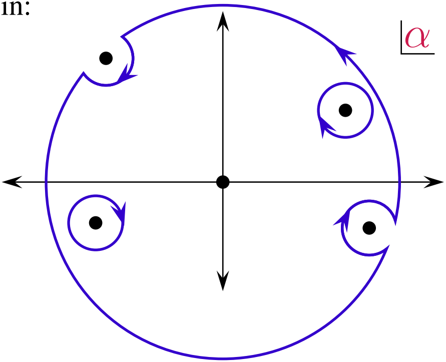
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



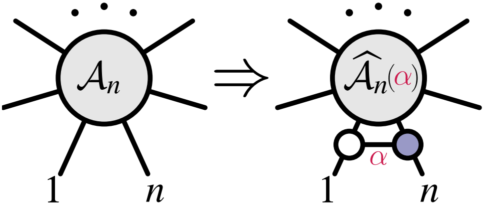


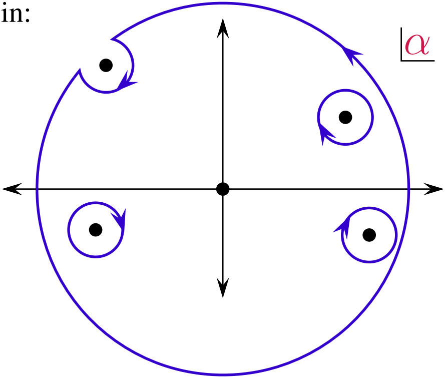
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



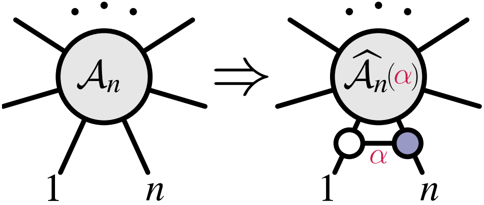


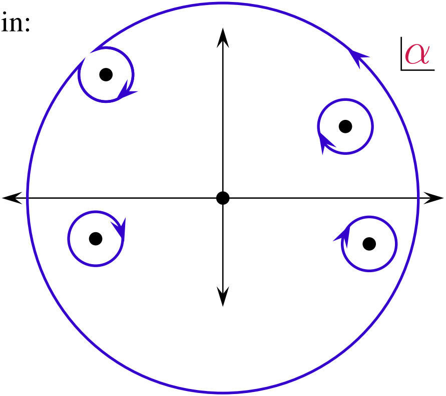
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



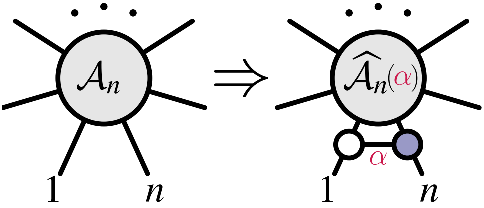


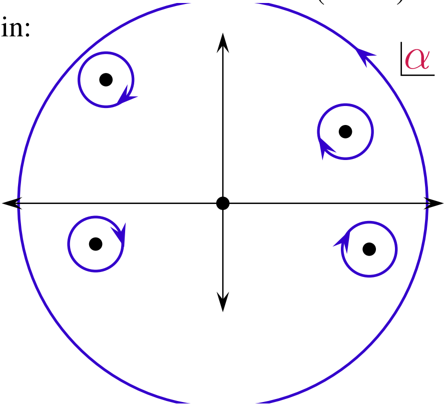
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to0) \propto \oint_{\alpha=0}\frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



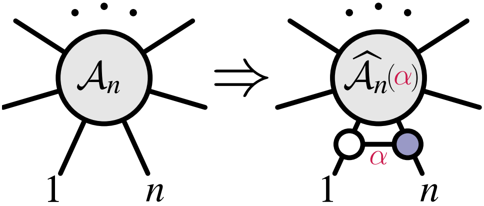


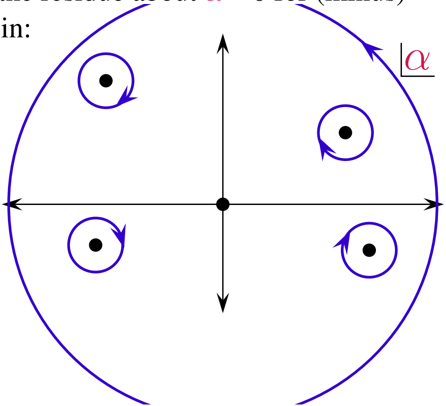
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n=\widehat{\mathcal{A}}_n(\alpha\to0)\propto\oint\limits_{\alpha=0}\frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



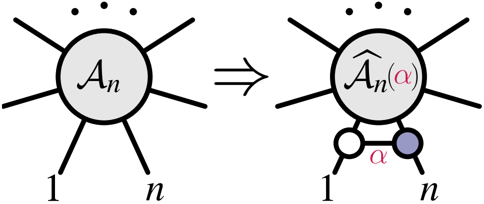


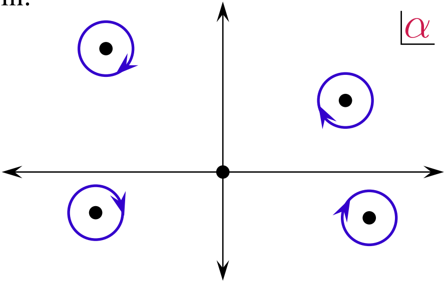
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin:



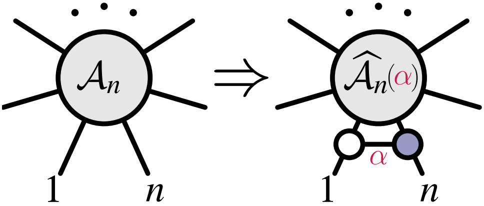


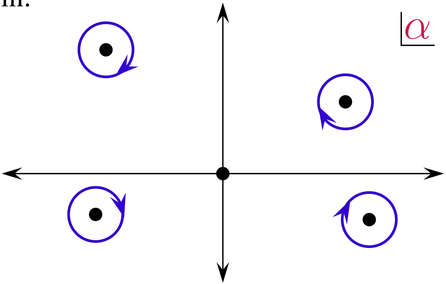
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin:



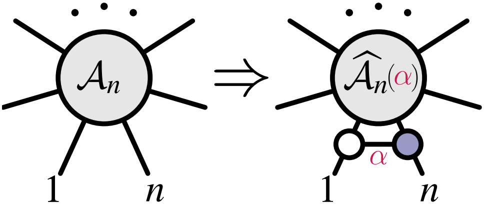


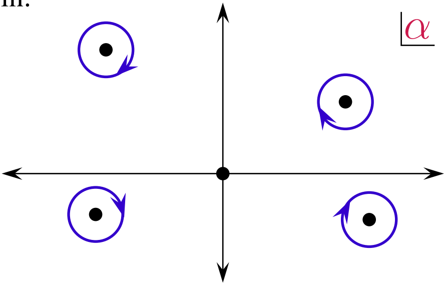
The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha\!=\!0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\!\to\!0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha\!=\!0$ for (minus) the sum of residues away from the origin:



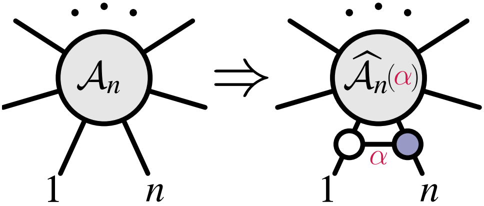


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha\!=\!0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\!\to\!0) \propto \oint\limits_{\alpha=0}\frac{d\alpha}{\alpha}\;\widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha\!=\!0$ for (minus) the sum of residues away from the origin—these come in two types:

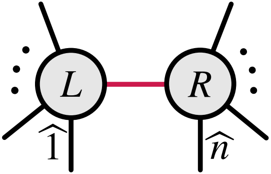


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin—these come in two types: factorization-channels

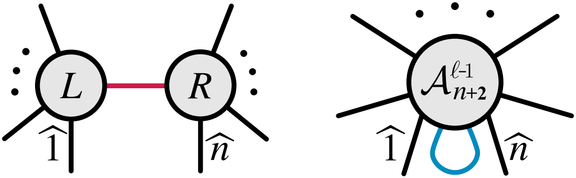


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full *n*-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\ \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin—these come in two types: factorization-channels and forward-limits

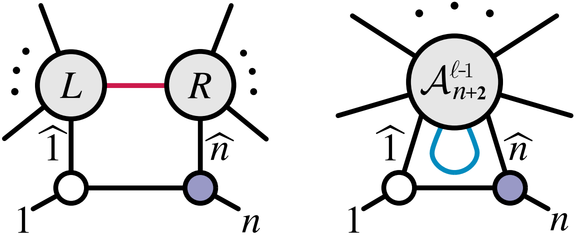


The Analytic Bootstrap: All-Loop Recursion Relations

Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha=0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha\to 0) \propto \oint\limits_{\alpha=0} \frac{d\alpha}{\alpha}\; \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha=0$ for (minus) the sum of residues away from the origin—these come in two types: factorization-channels and forward-limits

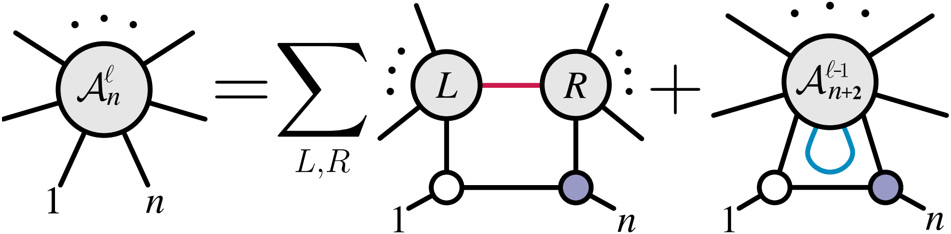


The Analytic Bootstrap: All-Loop Recursion Relations

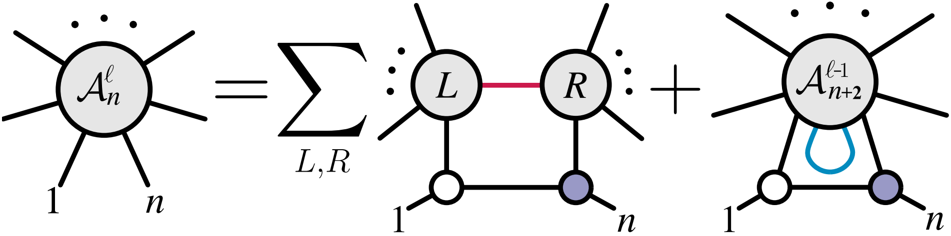
Consider adding a BCFW bridge to the full n-particle scattering amplitude the undeformed amplitude $\mathcal{A}_n$ is recovered as the **residue** about $\alpha = 0$:

$$\mathcal{A}_n = \widehat{\mathcal{A}}_n(\alpha \to 0) \propto \oint_{\alpha=0} \frac{d\alpha}{\alpha}\, \widehat{\mathcal{A}}_n(\alpha)$$

We can use **Cauchy's theorem** to trade the residue about $\alpha = 0$ for (minus) the sum of residues away from the origin—these come in two types: factorization-channels and forward-limits

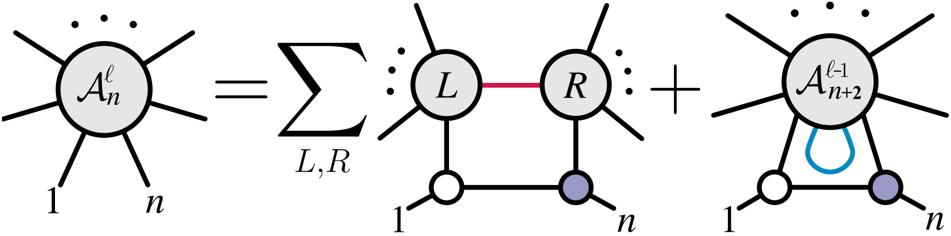


The Analytic Bootstrap: All-Loop Recursion Relations



The Analytic Bootstrap: All-Loop Recursion Relations

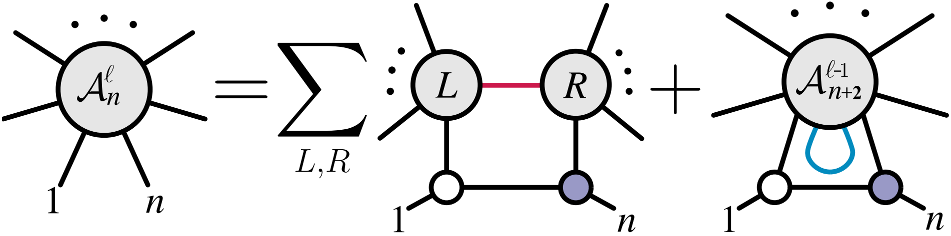
Diagrams are characterized by 'm'—the number of "minus-helicity" gluons:



The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by ‘m’—the number of “minus-helicity” gluons:

$$m \equiv 2n_B + n_W - n_I.$$



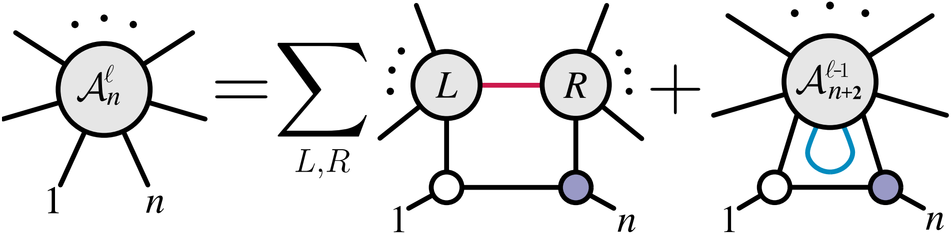
The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by 'm'—the number of "minus-helicity" gluons:

$$m \equiv 2n_B + n_W - n_I.$$

For the bridge terms, we have:

$$m_L + m_R = m + 1.$$



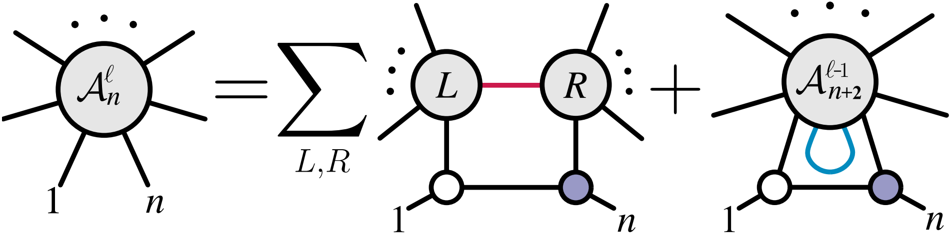
The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by ‘m’—the number of “minus-helicity” gluons:

$$m \equiv 2n_B + n_W - n_I.$$

For the bridge terms, we have:

$$m_L + m_R = m + 1.$$



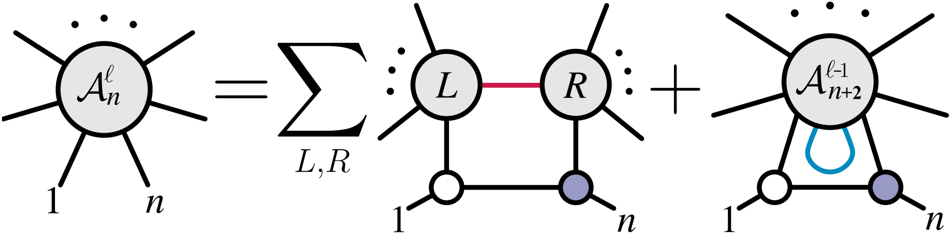
The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by 'm'—the number of "minus-helicity" gluons:

$$m \equiv 2n_B + n_W - n_I.$$

For the bridge terms, we have:

$$m_L + m_R = m + 1.$$



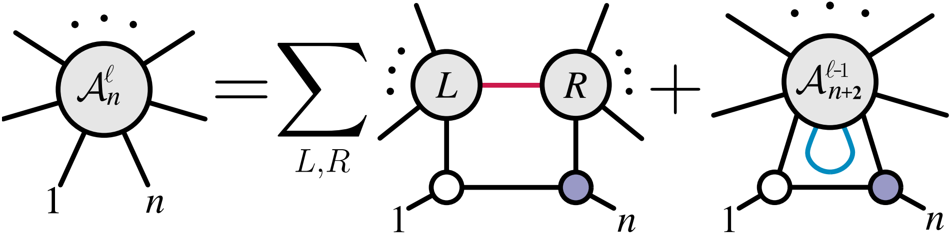
The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by 'm'—the number of "minus-helicity" gluons:

$$m \equiv 2n_B + n_W - n_I.$$

For the bridge terms, we have:

$$m_L + m_R = m + 1.$$



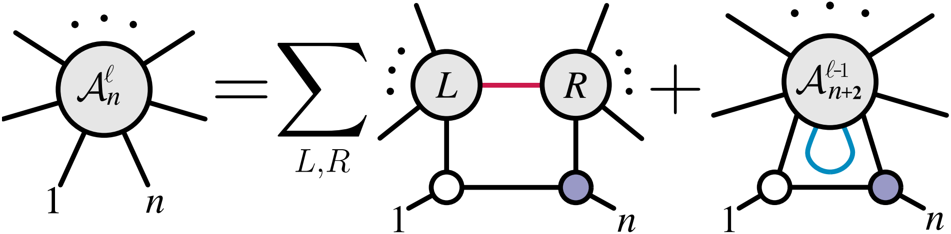
The Analytic Bootstrap: All-Loop Recursion Relations

Diagrams are characterized by 'm'—the number of "minus-helicity" gluons:

$$m \equiv 2n_B + n_W - n_I.$$

For the bridge terms, we have:

$$m_L + m_R = m + 1.$$



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$!

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)}\bigotimes\mathcal{A}_3^{(1)}$:

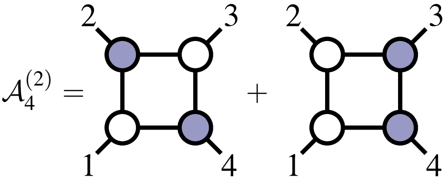
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)}\bigotimes\mathcal{A}_3^{(1)}$:

$$\mathcal{A}_4^{(2)} =$$

Exempli Gratia: On-Shell Representations of Amplitudes

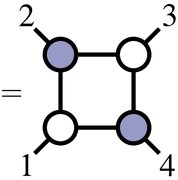
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:



Exempli Gratia: On-Shell Representations of Amplitudes

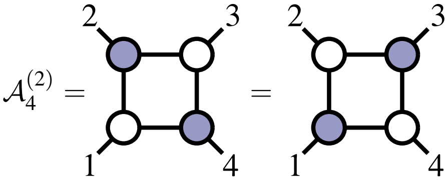
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:

$$\mathcal{A}_4^{(2)} =$$



Exempli Gratia: On-Shell Representations of Amplitudes

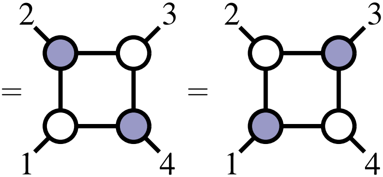
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)}\bigotimes\mathcal{A}_3^{(1)}$:



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:

$\mathcal{A}_4^{(2)} =$



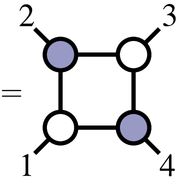
$\mathcal{A}_5^{(2)} =$

Exempli Gratia: On-Shell Representations of Amplitudes

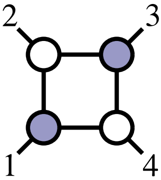
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$!

The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:

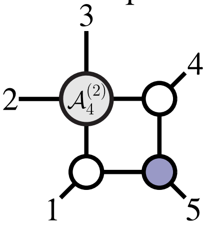
$\mathcal{A}_4^{(2)} =$



$=$

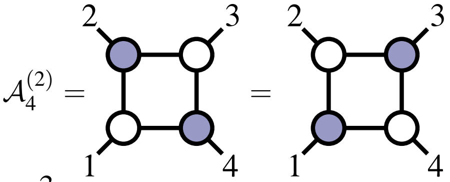


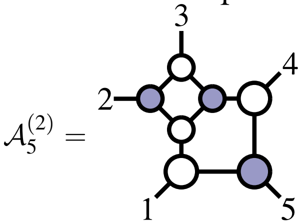
$\mathcal{A}_5^{(2)} =$



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \otimes \mathcal{A}_3^{(1)}$:

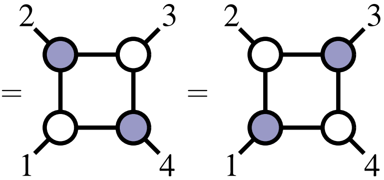




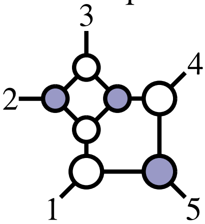
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)}\otimes\mathcal{A}_3^{(1)}$:

$\mathcal{A}_4^{(2)} =$



$\mathcal{A}_5^{(2)} =$

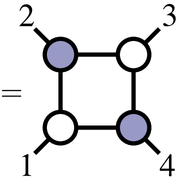


$\mathcal{A}_6^{(2)} =$

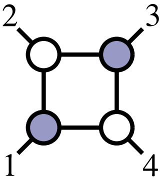
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:

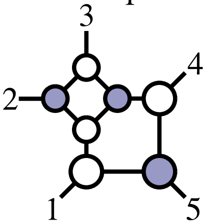
$\mathcal{A}_4^{(2)} =$



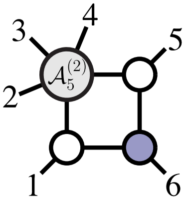
$=$



$\mathcal{A}_5^{(2)} =$



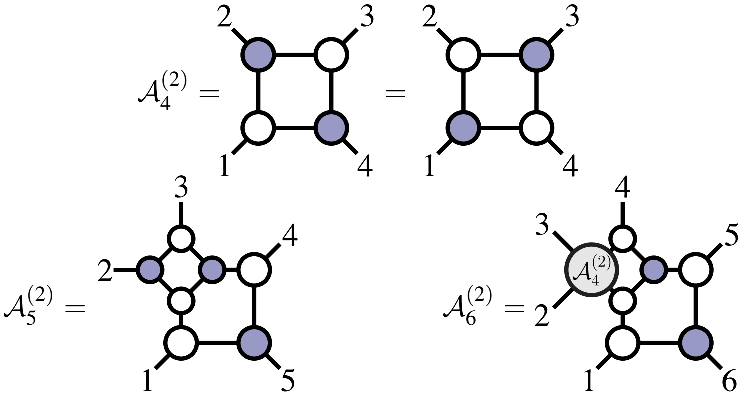
$\mathcal{A}_6^{(2)} =$



Exempli Gratia: On-Shell Representations of Amplitudes

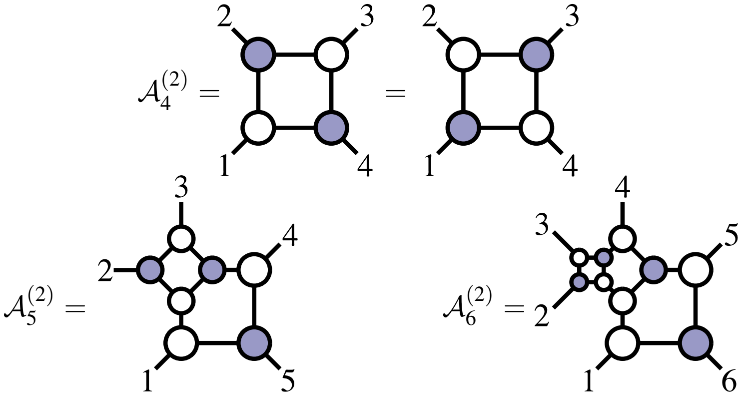
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$!

The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \otimes \mathcal{A}_3^{(1)}$:



Exempli Gratia: On-Shell Representations of Amplitudes

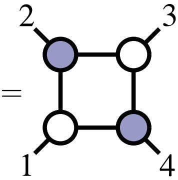
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:



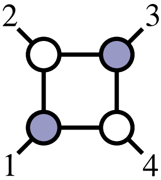
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! The **only** (non-vanishing) contribution to $\mathcal{A}_n^{(2)}$ is $\mathcal{A}_{n-1}^{(2)} \bigotimes \mathcal{A}_3^{(1)}$:

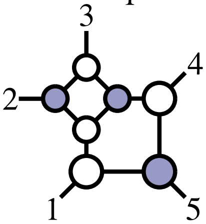
$\mathcal{A}_4^{(2)} =$



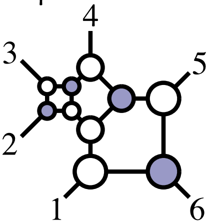
$=$



$\mathcal{A}_5^{(2)} =$



$\mathcal{A}_6^{(2)} =$



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

Exempli Gratia: On-Shell Representations of Amplitudes

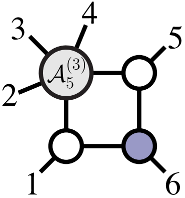
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

$$\mathcal{A}_6^{(3)} =$$

Exempli Gratia: On-Shell Representations of Amplitudes

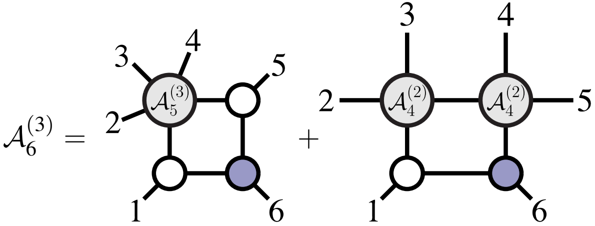
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

$\mathcal{A}_6^{(3)} =$



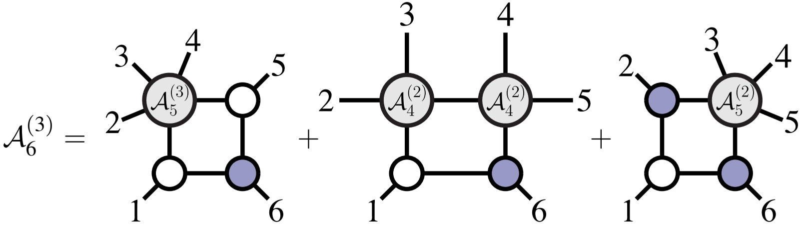
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



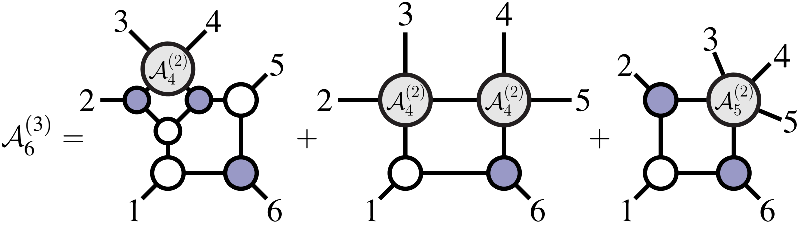
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



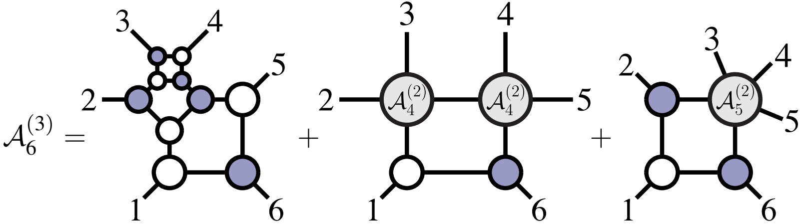
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



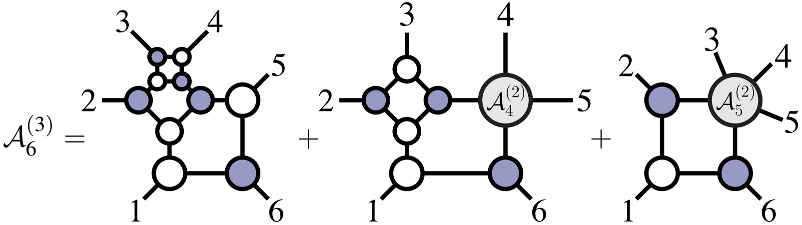
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



Exempli Gratia: On-Shell Representations of Amplitudes

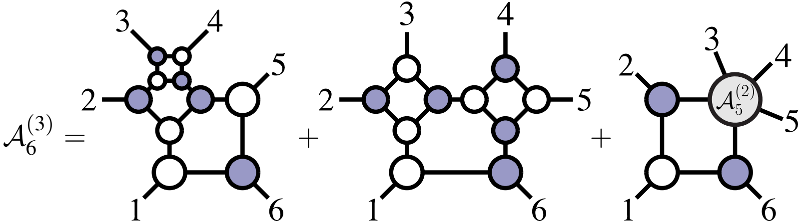
The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$!

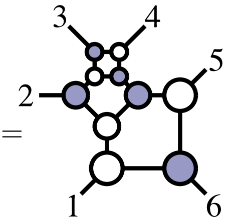
And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



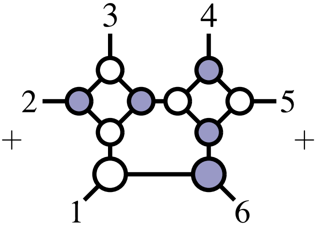
Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m\!=\!2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

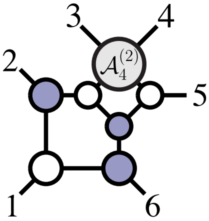
$\mathcal{A}_6^{(3)} =$



$+$



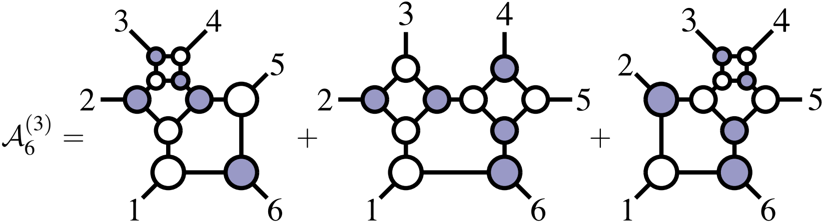
$+$



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m\!=\!2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

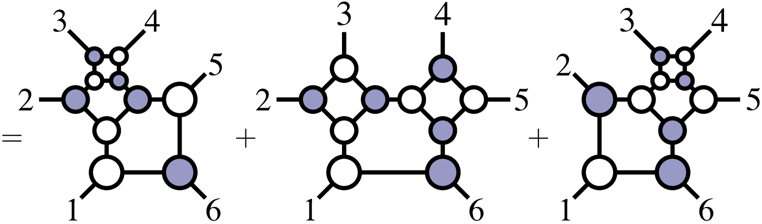
$\mathcal{A}_6^{(3)} =$



Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

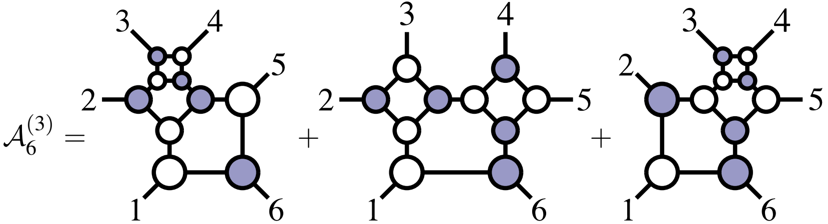
$\mathcal{A}_6^{(3)} =$



Observations regarding recursed representations of scattering amplitudes:

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

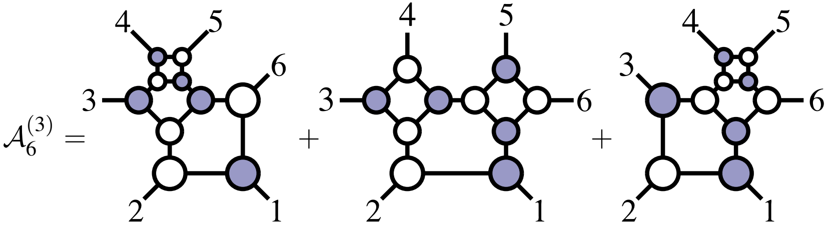


Observations regarding recursed representations of scattering amplitudes:

- varying recursion ‘schema’ can generate *many* ‘BCFW formulae’

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



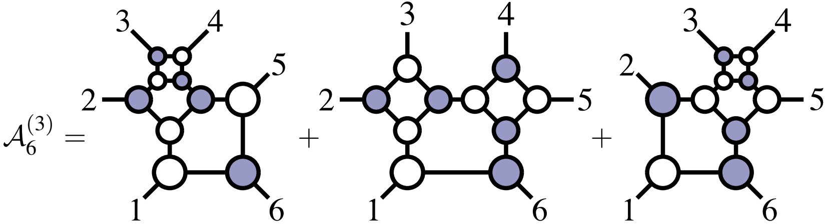
Observations regarding recursed representations of scattering amplitudes:

- varying recursion ‘schema’ can generate *many* ‘BCFW formulae’

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$!

And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

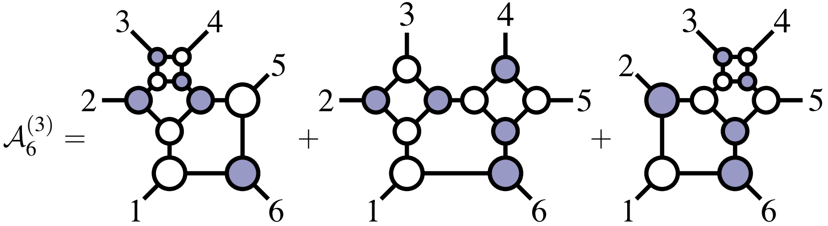


Observations regarding recursed representations of scattering amplitudes:

- varying recursion 'schema' can generate *many* 'BCFW formulae'

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



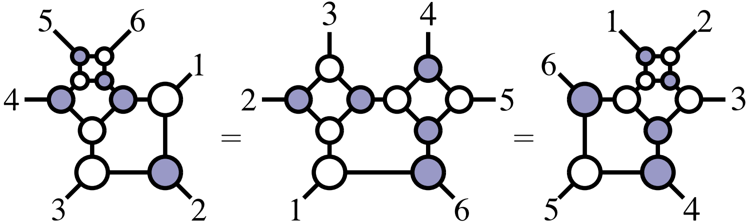
Observations regarding recursed representations of scattering amplitudes:

- varying recursion 'schema' can generate *many* 'BCFW formulae'
- on-shell diagrams can often be related in surprising ways

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$!

And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

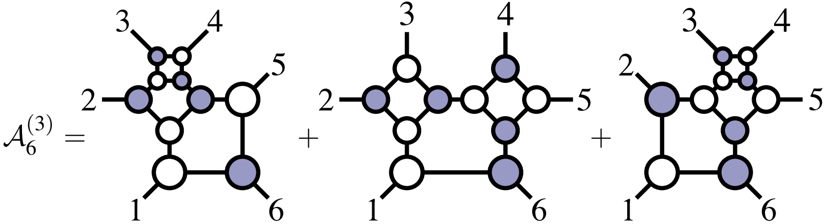


Observations regarding recursed representations of scattering amplitudes:

- varying recursion 'schema' can generate *many* 'BCFW formulae'
- on-shell diagrams can often be related in surprising ways

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



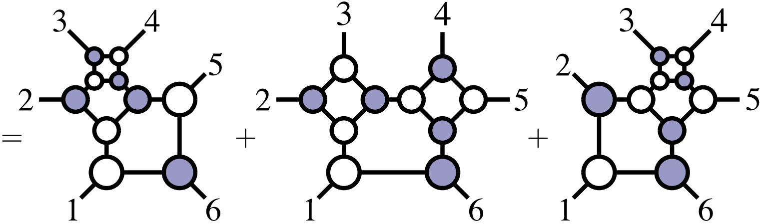
Observations regarding recursed representations of scattering amplitudes:

- varying recursion ‘schema’ can generate *many* ‘BCFW formulae’
- on-shell diagrams can often be related in surprising ways

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m = 2$, $\mathcal{A}_n^{(2)}$! And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:

$\mathcal{A}_6^{(3)} =$



Observations regarding recursed representations of scattering amplitudes:

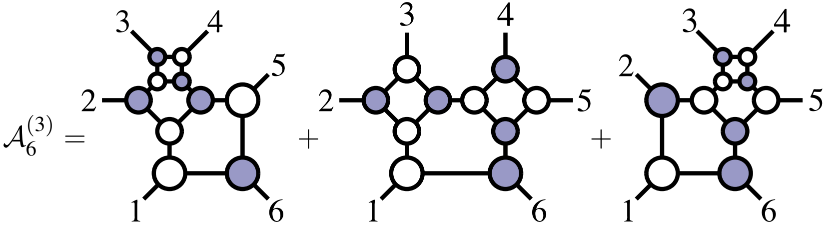
- varying recursion 'schema' can generate *many* 'BCFW formulae'
- on-shell diagrams can often be related in surprising ways

Is there any way to invariantly characterize the on-shell functions associated with on-shell diagrams?

Exempli Gratia: On-Shell Representations of Amplitudes

The BCFW recursion relations realize an incredible fantasy: they **directly** produces the **Parke-Taylor** formula for all amplitudes with $m=2$, $\mathcal{A}_n^{(2)}$!

And it generates **very concise** formulae for all other amplitudes—*e.g.* $\mathcal{A}_6^{(3)}$:



Observations regarding recursed representations of scattering amplitudes:

- varying recursion ‘schema’ can generate *many* ‘BCFW formulae’
- on-shell diagrams can often be related in surprising ways

Is there any way to invariantly characterize the on-shell functions associated with on-shell diagrams?

On-Shell Recursion of Loop-Amplitude Integrands

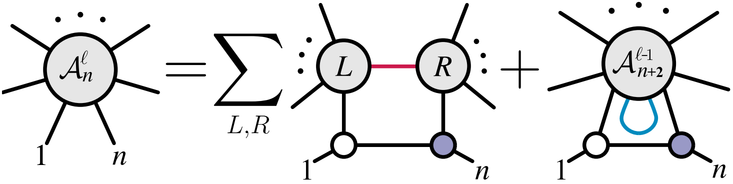
Let's look at an example of how loop amplitudes are represented by recursion.

On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion.

On-Shell Recursion of Loop-Amplitude Integrands

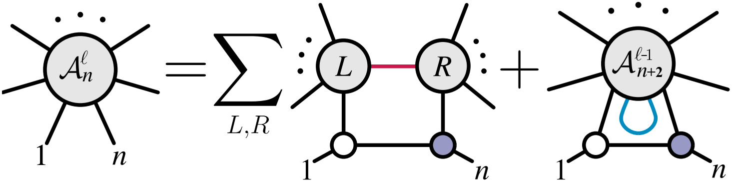
Let’s look at an example of how loop amplitudes are represented by recursion.



On-Shell Recursion of Loop-Amplitude Integrands

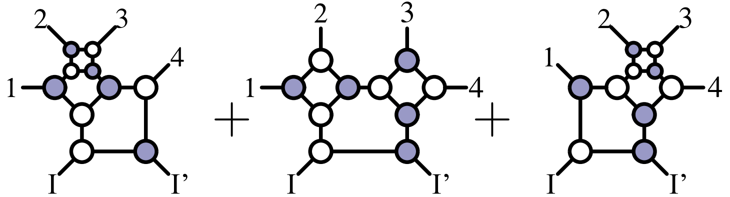
Let's look at an example of how loop amplitudes are represented by recursion.

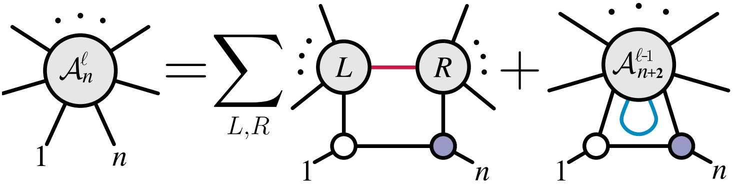
For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:



On-Shell Recursion of Loop-Amplitude Integrands

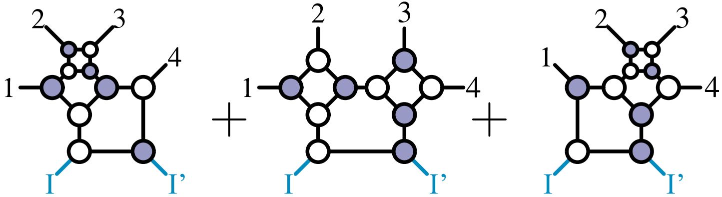
Let's look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:

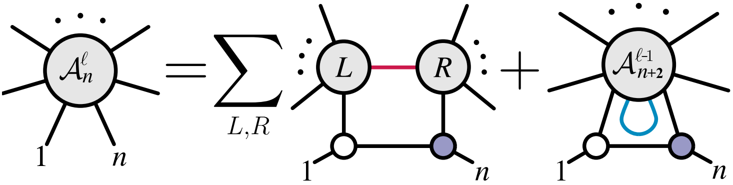




On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:

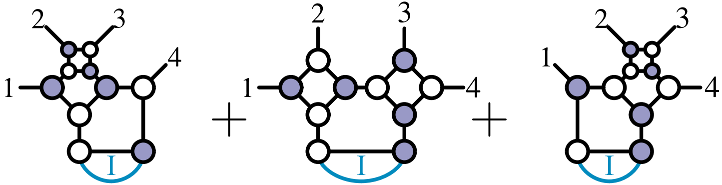


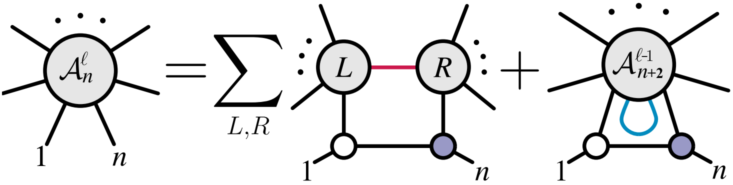


On-Shell Recursion of Loop-Amplitude Integrands

Let’s look at an example of how loop amplitudes are represented by recursion.

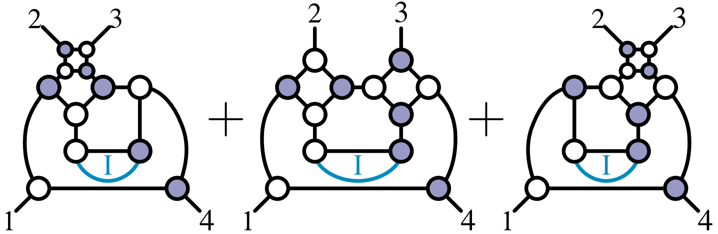
For $\mathcal{A}_4^{(2),1}$, the only terms come from the ‘forward limit’ of the tree $\mathcal{A}_6^{(3),0}$:

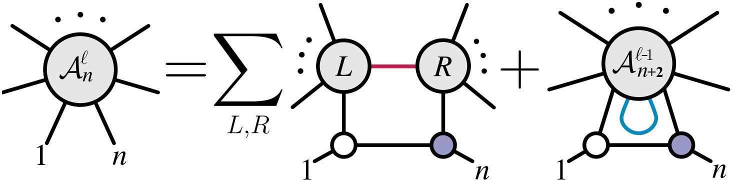




On-Shell Recursion of Loop-Amplitude Integrands

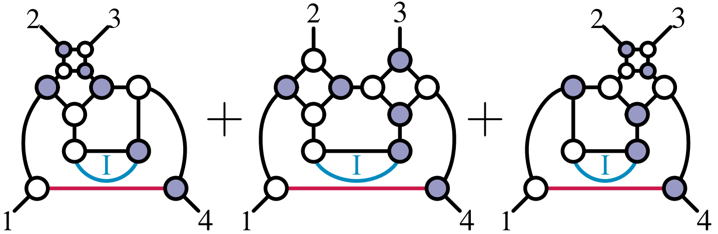
Let’s look at an example of how loop amplitudes are represented by recursion.
For $\mathcal{A}_4^{(2),1}$, the only terms come from the ‘forward limit’ of the tree $\mathcal{A}_6^{(3),0}$:

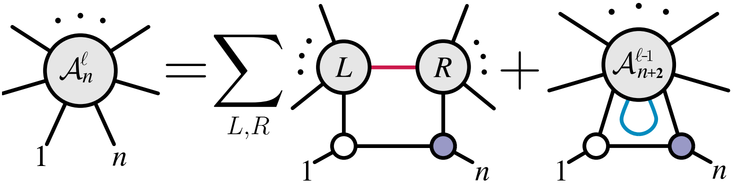




On-Shell Recursion of Loop-Amplitude Integrands

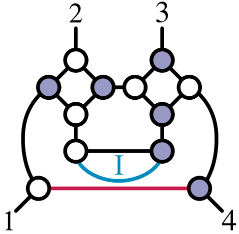
Let's look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:

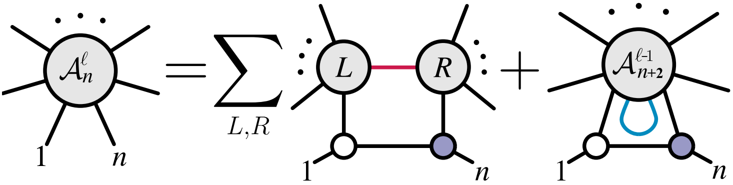




On-Shell Recursion of Loop-Amplitude Integrands

Let’s look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the ‘forward limit’ of the tree $\mathcal{A}_6^{(3),0}$:

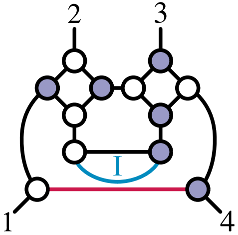


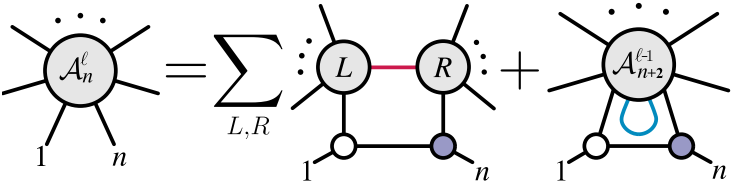


On-Shell Recursion of Loop-Amplitude Integrands

Let’s look at an example of how loop amplitudes are represented by recursion.

For $\mathcal{A}_4^{(2),1}$, the only terms come from the ‘forward limit’ of the tree $\mathcal{A}_6^{(3),0}$:

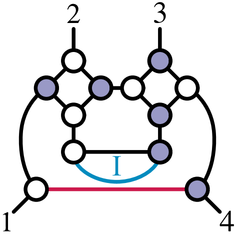




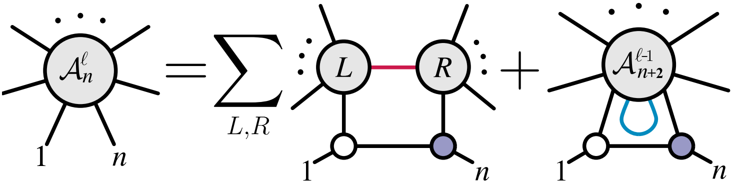
On-Shell Recursion of Loop-Amplitude Integrands

Let’s look at an example of how loop amplitudes are represented by recursion.

For $\mathcal{A}_4^{(2),1}$, the only terms come from the ‘forward limit’ of the tree $\mathcal{A}_6^{(3),0}$:

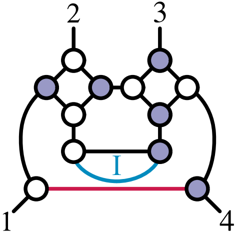


$$\int_{\ell\in\mathbb{R}^{3,1}} d^4\ell$$

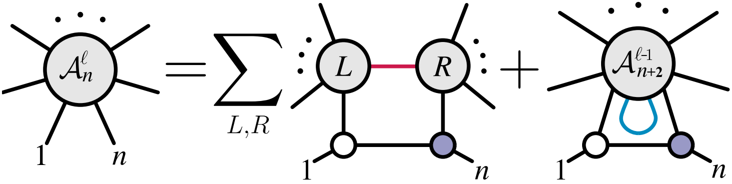


On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:

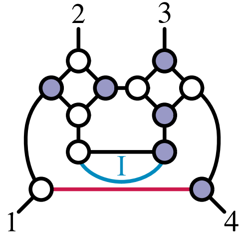


$$\int\limits_{\ell\in\mathbb{R}^{3,1}} d^4\ell \quad \Leftrightarrow$$

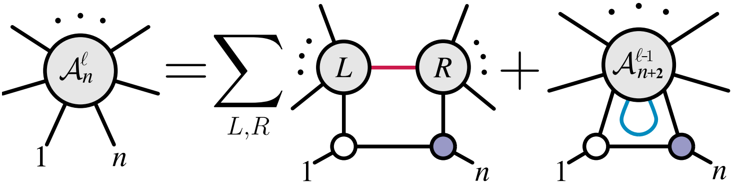


On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion. For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:



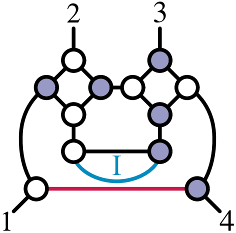
$$\int_{\ell\in\mathbb{R}^{3,1}} d^4\ell \quad\Leftrightarrow\quad \int_{\ell\equiv(\lambda_{\text{I}}\widetilde{\lambda}_{\text{I}}+\alpha\lambda_1\widetilde{\lambda}_4)\in\mathbb{R}^{3,1}} \frac{d^2\lambda_{\text{I}}\,d^2\widetilde{\lambda}_{\text{I}}}{\text{vol}(GL_1)}\,d\alpha\,\langle \text{I}1\rangle\,[n\text{I}]$$



On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion.

For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:



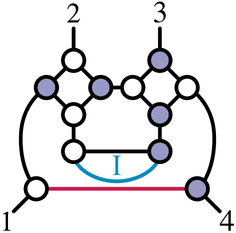
$$\int\limits_{\ell\in\mathbb{R}^{3,1}} d^4\ell \quad \Leftrightarrow \quad \int\limits_{\ell\equiv(\lambda_{\mathrm{I}}\widetilde{\lambda}_{\mathrm{I}}+\alpha\lambda_1\widetilde{\lambda}_4)\in\mathbb{R}^{3,1}} \frac{d^2\lambda_{\mathrm{I}}d^2\widetilde{\lambda}_{\mathrm{I}}}{\mathrm{vol}(GL_1)} d\alpha \langle \mathrm{I}1\rangle\,[n\mathrm{I}]$$

$$\mathcal{A}_4^{(2),0}\times\int\limits_{\ell\in\mathbb{R}^{3,1}} d\log\left(\frac{\ell^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell+p_1)^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell+p_1+p_2)^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell-p_4)^2}{(\ell-\ell^*)^2}\right)$$

On-Shell Recursion of Loop-Amplitude Integrands

Let's look at an example of how loop amplitudes are represented by recursion.

For $\mathcal{A}_4^{(2),1}$, the only terms come from the 'forward limit' of the tree $\mathcal{A}_6^{(3),0}$:



$$\int\limits_{\ell\in\mathbb{R}^{3,1}} d^4\ell \quad \Leftrightarrow \quad \int\limits_{\ell\equiv(\lambda_{\mathrm{I}}\widetilde{\lambda}_{\mathrm{I}}+\alpha\lambda_1\widetilde{\lambda}_4)\in\mathbb{R}^{3,1}} \frac{d^2\lambda_{\mathrm{I}}d^2\widetilde{\lambda}_{\mathrm{I}}}{\mathrm{vol}(GL_1)} d\alpha\langle\mathrm{I}1\rangle\,[n\mathrm{I}]$$

$$\mathcal{A}_4^{(2),0}\times\int\limits_{\ell\in\mathbb{R}^{3,1}} d\log\left(\frac{\ell^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell+p_1)^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell+p_1+p_2)^2}{(\ell-\ell^*)^2}\right) d\log\left(\frac{(\ell-p_4)^2}{(\ell-\ell^*)^2}\right)$$

$$=\mathcal{A}_4^{(2),0}\times\int\limits_{\ell\in\mathbb{R}^{3,1}} d^4\ell\,\frac{(p_1+p_2)^2(p_3+p_4)^2}{\ell^2(\ell+p_1)^2(\ell+p_1+p_2)^2(\ell-p_4)^2}$$

Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

Combinatorial Characterization of On-Shell Diagrams

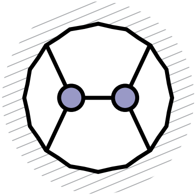
On-shell diagrams can be altered without changing their associated functions

- chains of equivalent three-particle vertices can be arbitrarily connected

Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

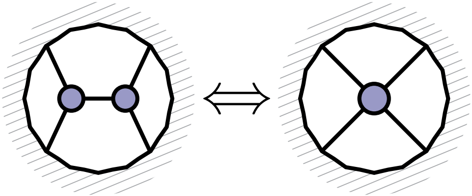
- chains of equivalent three-particle vertices can be arbitrarily connected



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

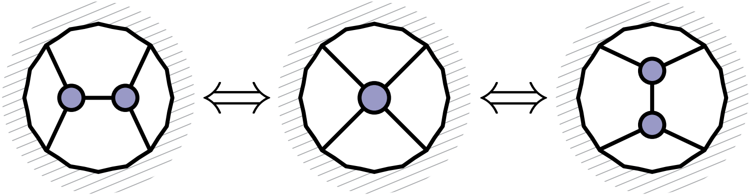
- chains of equivalent three-particle vertices can be arbitrarily connected



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

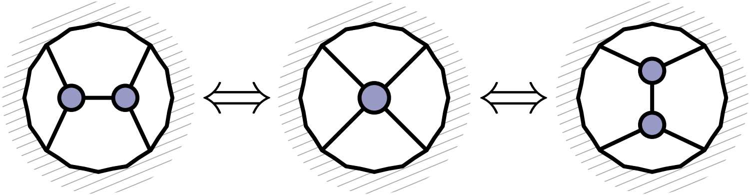
- chains of equivalent three-particle vertices can be arbitrarily connected



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

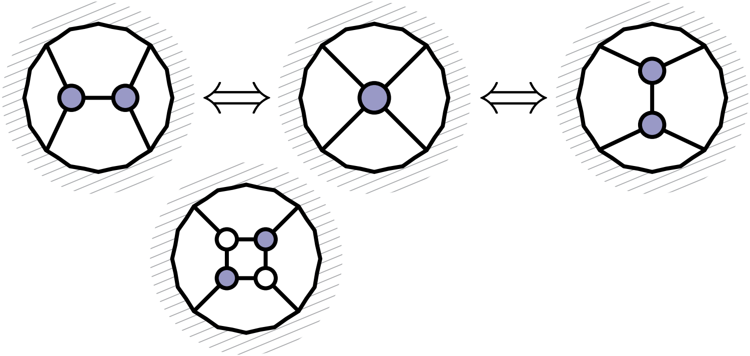
- chains of equivalent three-particle vertices can be arbitrarily connected
- any four-particle 'square' can be drawn in its two equivalent ways



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

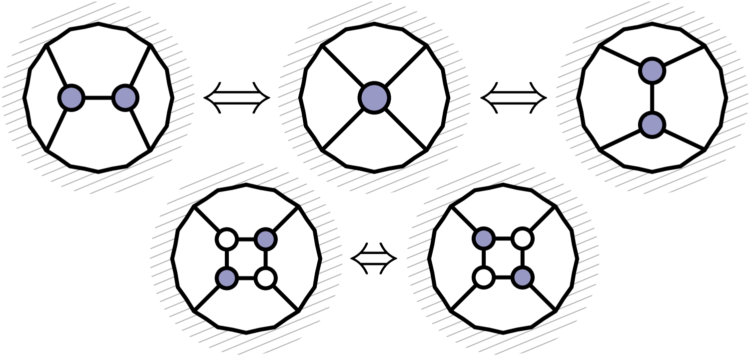
- chains of equivalent three-particle vertices can be arbitrarily connected
- any four-particle ‘square’ can be drawn in its two equivalent ways



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

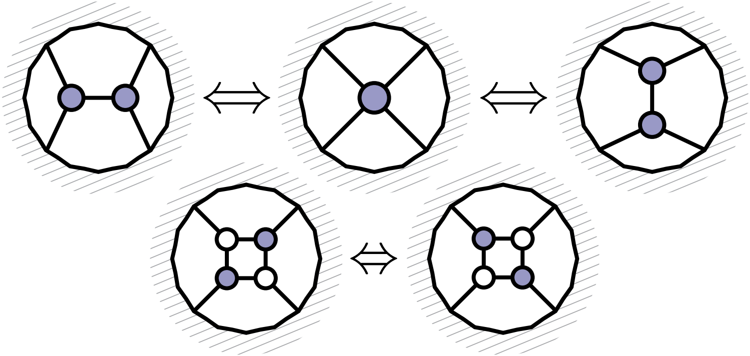
- chains of equivalent three-particle vertices can be arbitrarily connected
- any four-particle ‘square’ can be drawn in its two equivalent ways



Combinatorial Characterization of On-Shell Diagrams

On-shell diagrams can be altered without changing their associated functions

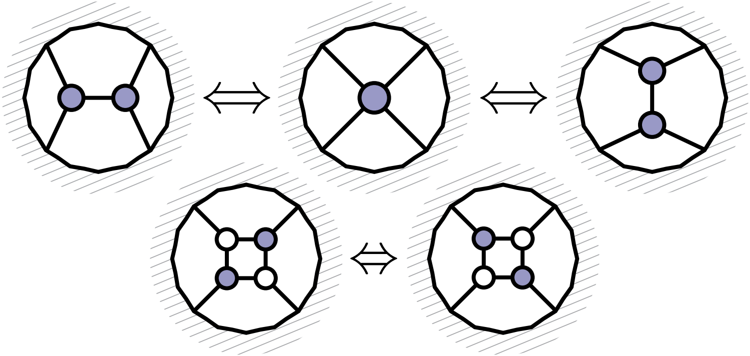
- chains of equivalent three-particle vertices can be arbitrarily connected
- any four-particle 'square' can be drawn in its two equivalent ways



Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

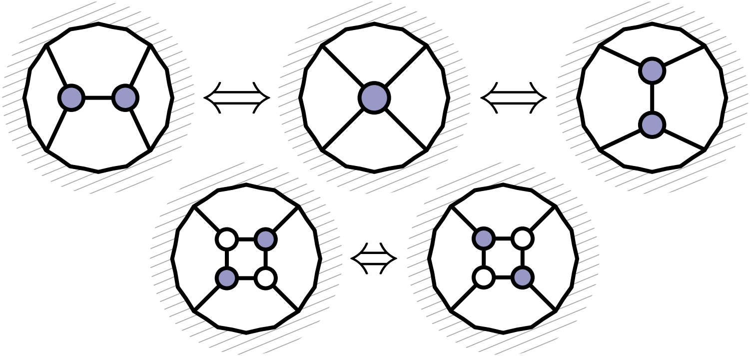


Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’:

Starting from any leg a, turn:

- *left* at each white vertex;

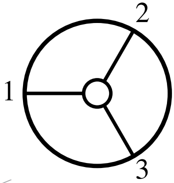


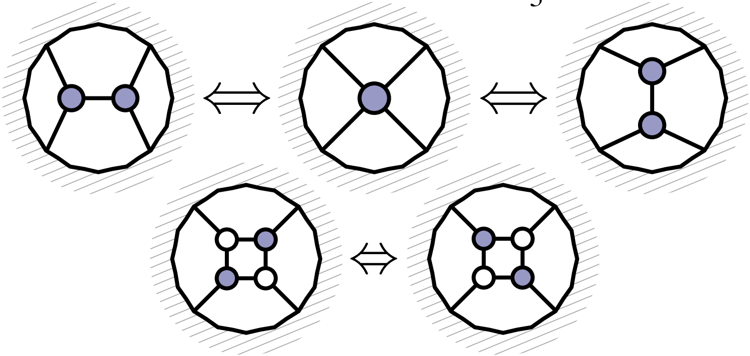
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’:

Starting from any leg a, turn:

- *left* at each white vertex;



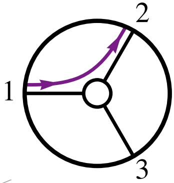


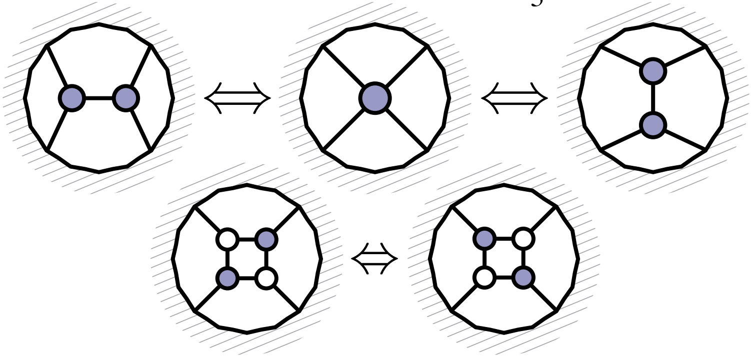
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;



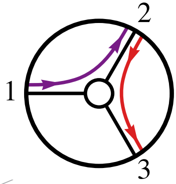


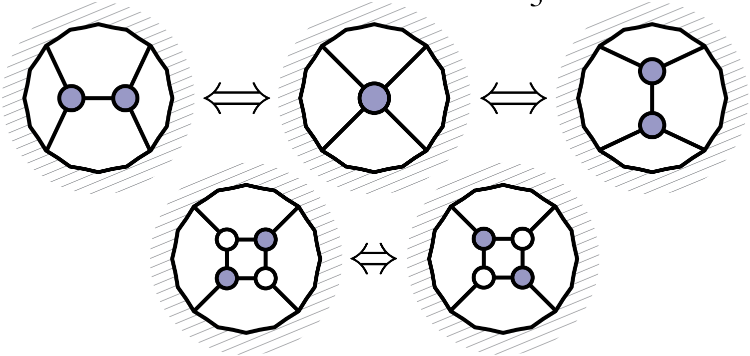
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg *a*, turn:

- *left* at each white vertex;



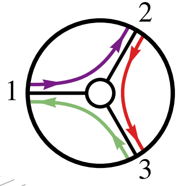


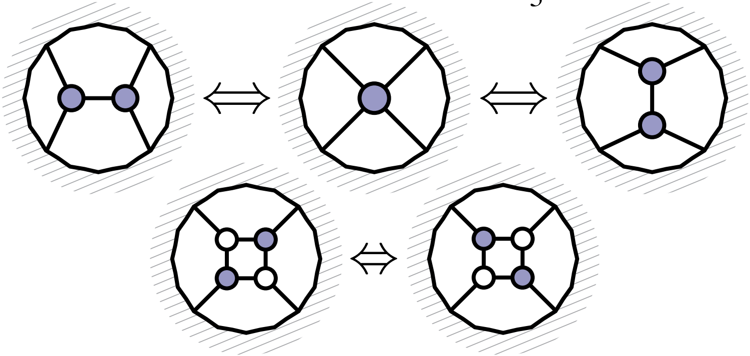
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;



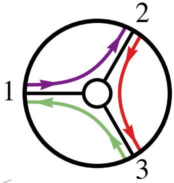


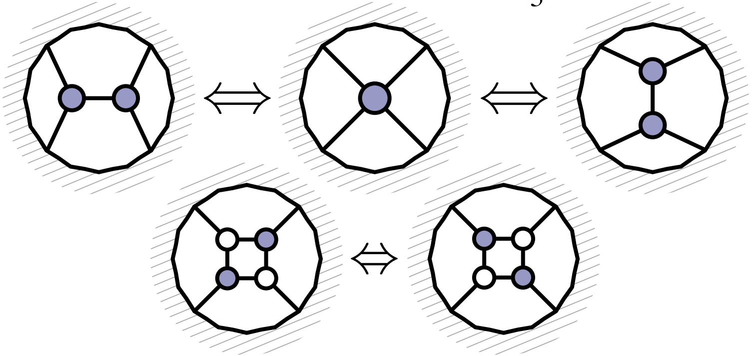
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.



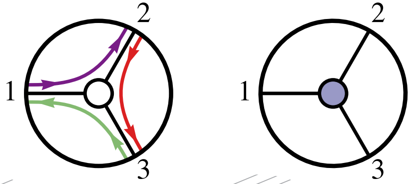


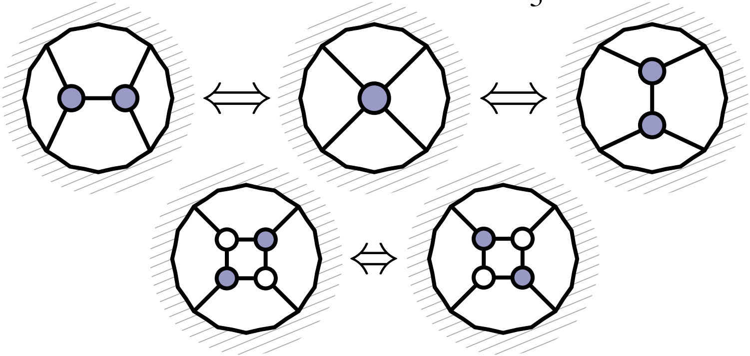
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’:

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.



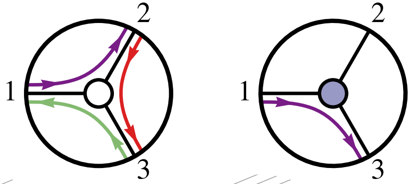


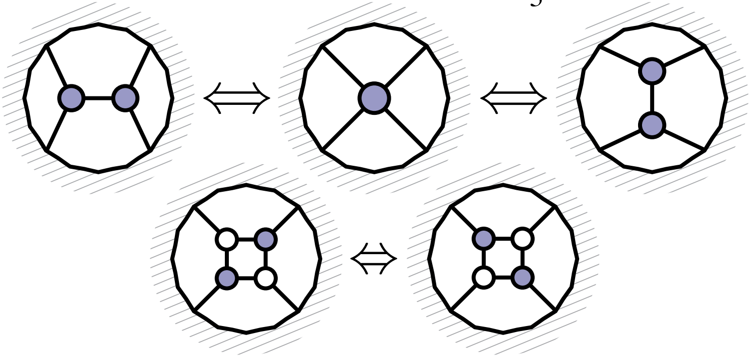
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’:

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.



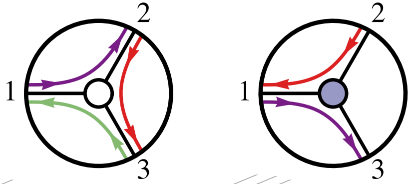


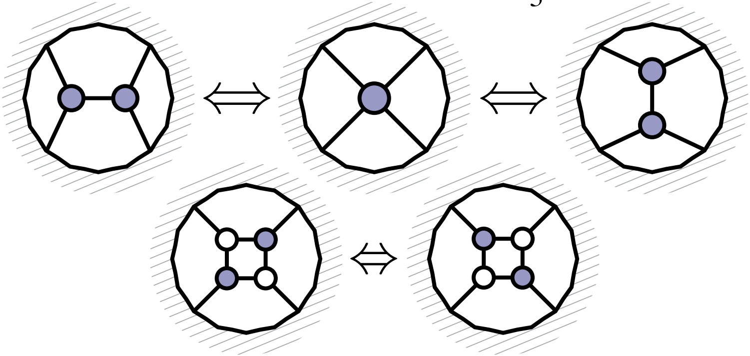
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.



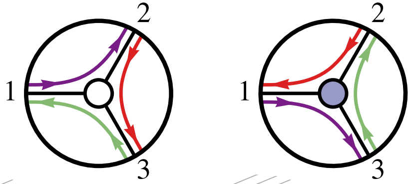


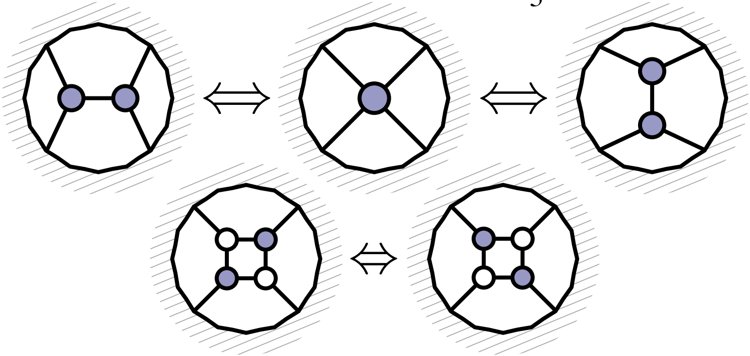
Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg *a*, turn:

- *left* at each white vertex;
- *right* at each blue vertex.





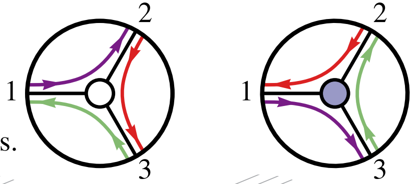
Combinatorial Characterization of On-Shell Diagrams

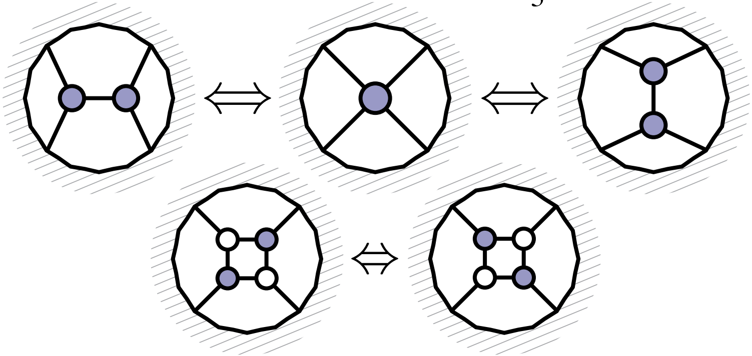
These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





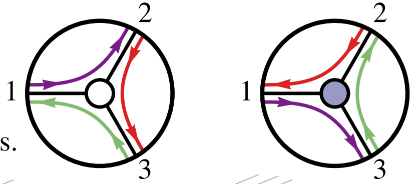
Combinatorial Characterization of On-Shell Diagrams

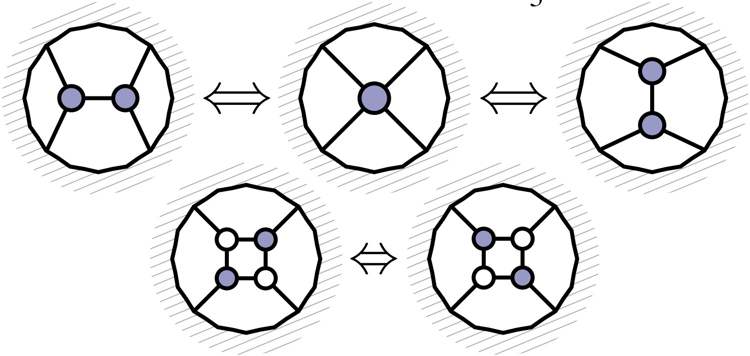
These moves leave invariant a permutation defined by 'left-right paths':

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





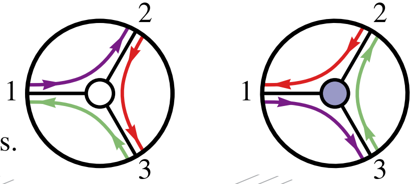
Combinatorial Characterization of On-Shell Diagrams

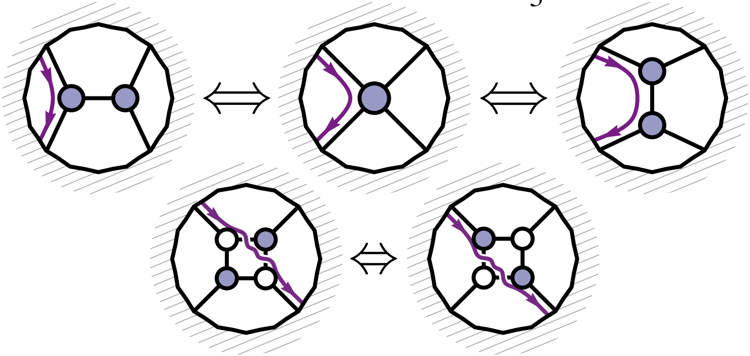
These moves leave invariant a permutation defined by 'left-right paths'

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





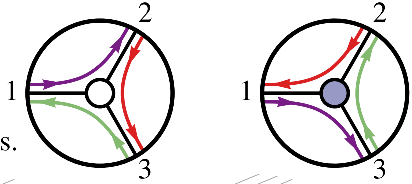
Combinatorial Characterization of On-Shell Diagrams

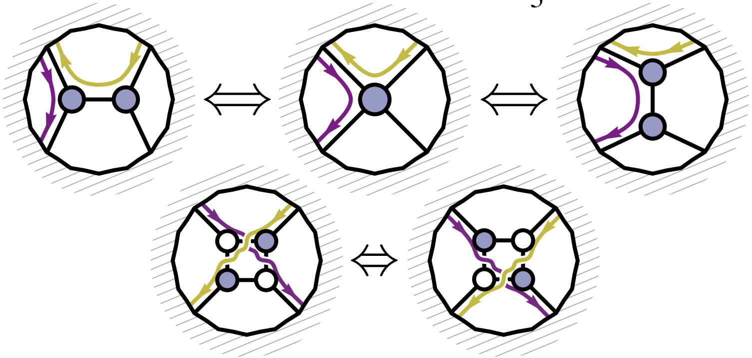
These moves leave invariant a permutation defined by 'left-right paths'

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





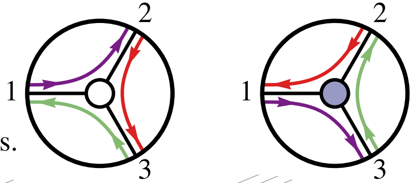
Combinatorial Characterization of On-Shell Diagrams

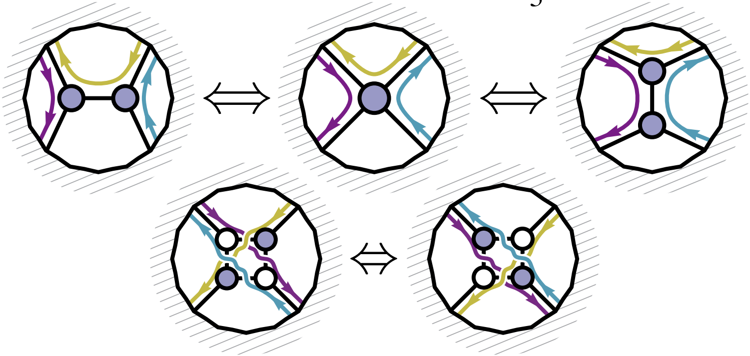
These moves leave invariant a permutation defined by ‘left-right paths’

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





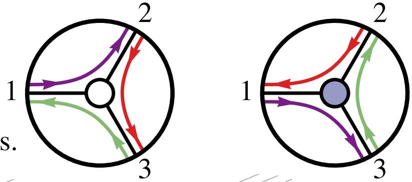
Combinatorial Characterization of On-Shell Diagrams

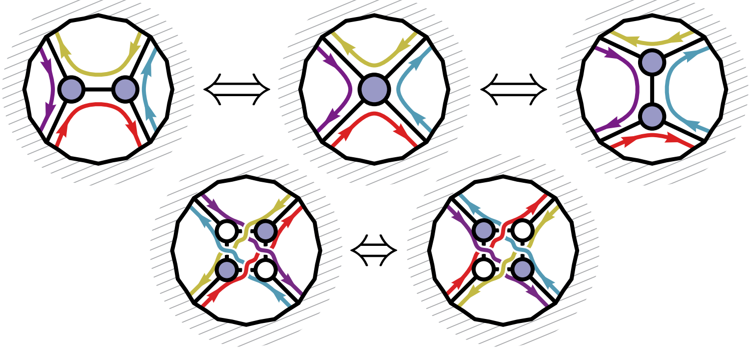
These moves leave invariant a permutation defined by ‘left-right paths’

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.





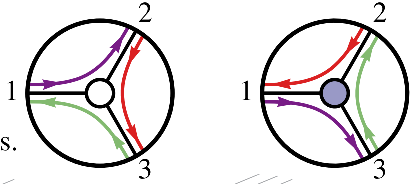
Combinatorial Characterization of On-Shell Diagrams

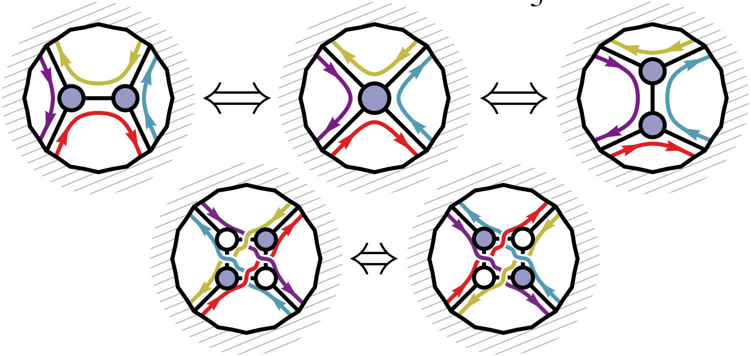
These moves leave invariant a permutation defined by 'left-right paths'

Starting from any leg a, turn:

- *left* at each white vertex;
- *right* at each blue vertex.

Let $\sigma(a)$ denote where path terminates.

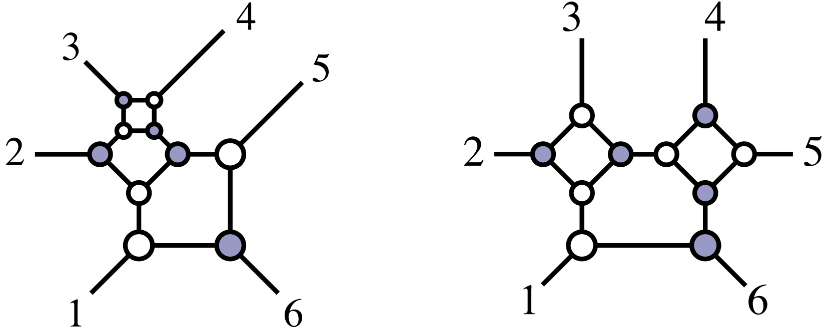




Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

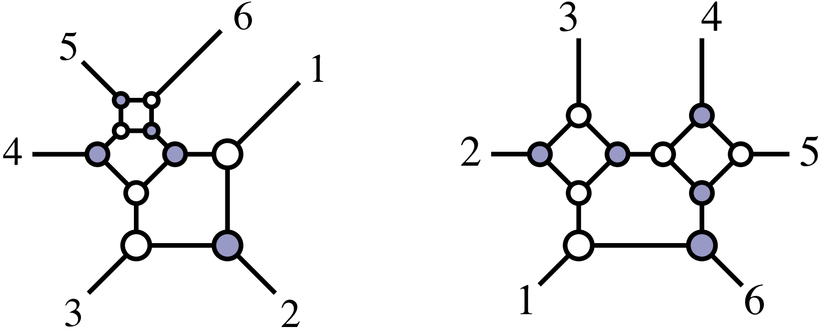
Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

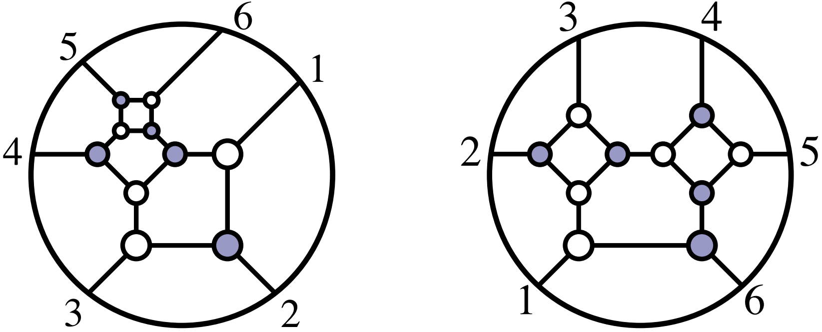
Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

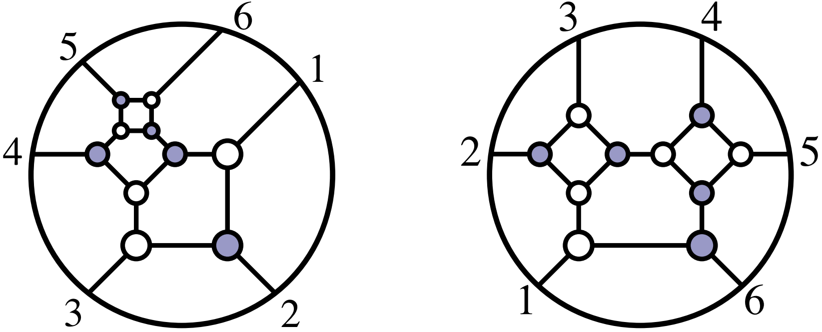
Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



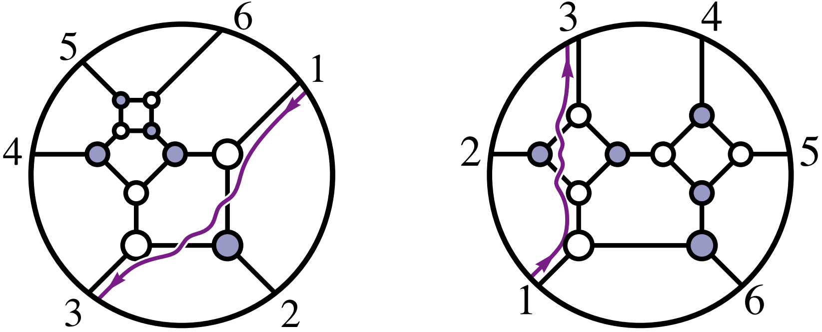
left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ & & & & & \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



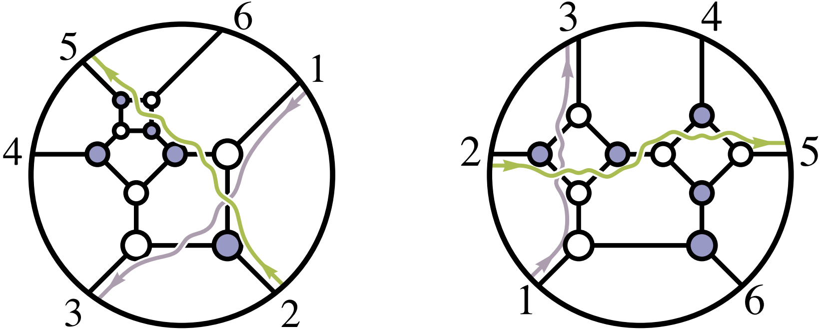
left-right permutation σ

$$\sigma : \begin{pmatrix} \overset{1}{\underset{3}{\downarrow}} & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



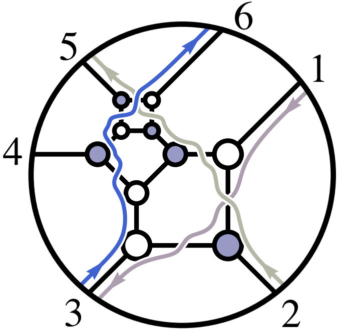
left-right permutation σ

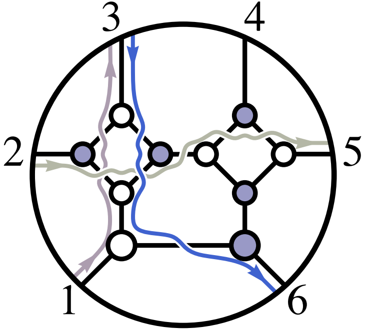
$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & & & & \\ 3 & 5 & & & & \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:





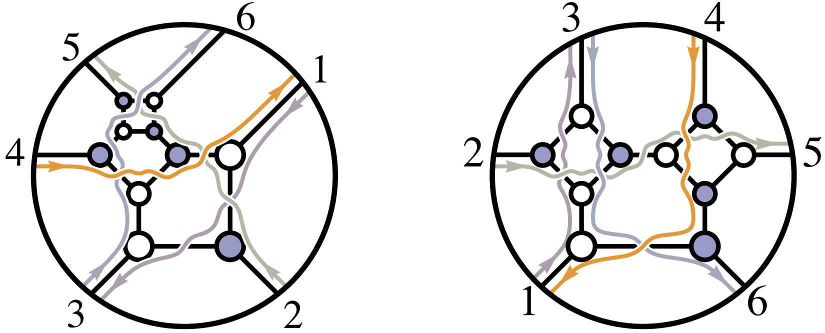
left-right permutation σ

$$\sigma:\begin{pmatrix}1&2&3&4&5&6\\ \downarrow&\downarrow&\downarrow&&&\\ 3&5&6&&&\end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



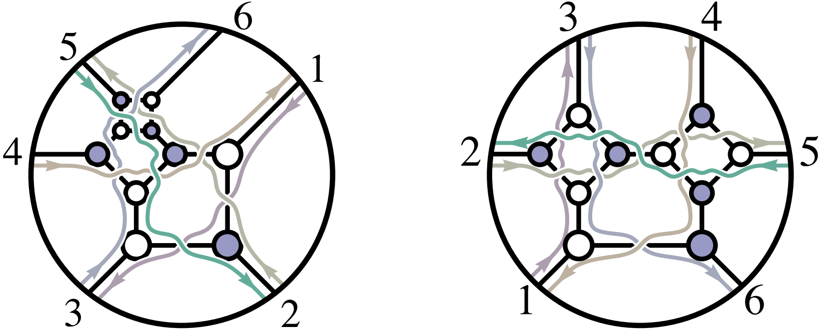
left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \\ 3 & 5 & 6 & \mathbf{1} & & \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by ‘left-right paths’.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



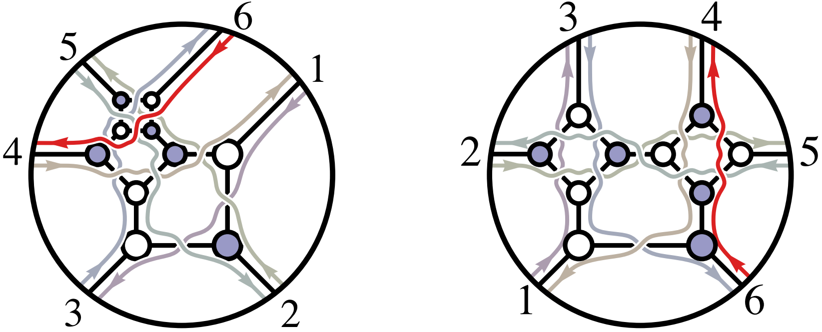
left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ 3 & 5 & 6 & \mathbf{1} & \mathbf{2} & \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



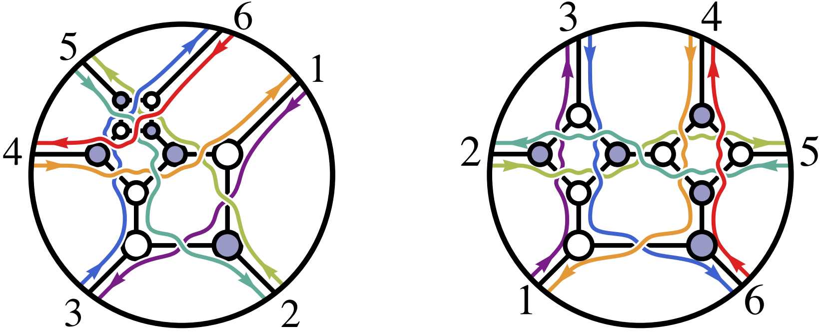
left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 5 & 6 & \mathbf{1} & \mathbf{2} & \mathbf{4} \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



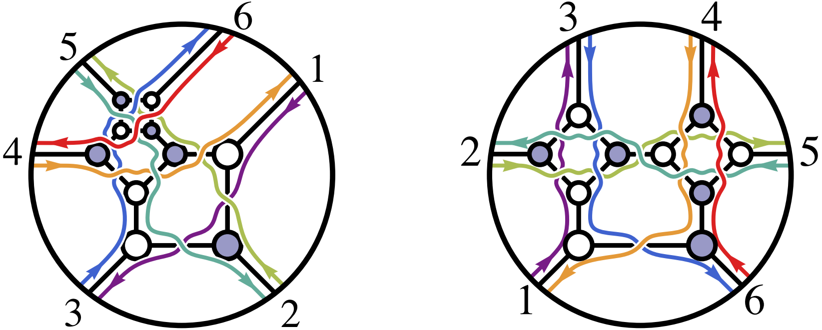
left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 5 & 6 & \mathbf{1} & \mathbf{2} & \mathbf{4} \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



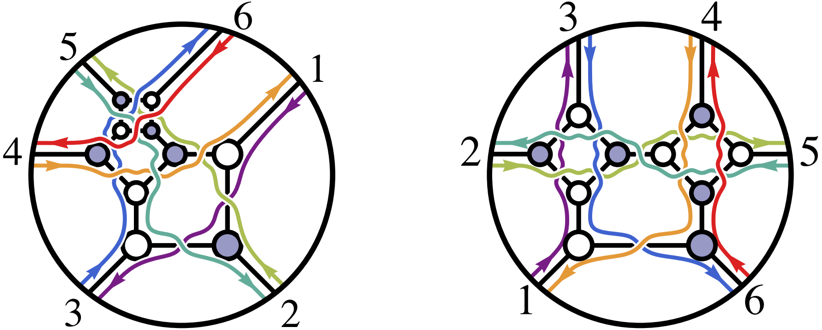
left-right permutation σ

$$\sigma:\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 5 & 6 & \mathbf{1} & \mathbf{2} & \mathbf{4} \end{array}\right)$$

Combinatorial Characterization of On-Shell Diagrams

These moves leave invariant a permutation defined by 'left-right paths'.

Recall that different contributions to $\mathcal{A}_6^{(3)}$ were related by rotation:



left-right permutation σ

$$\sigma : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 5 & 6 & 7 & 8 & 10 \end{pmatrix}$$

Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant.

Combinatorial Characterization of On-Shell Diagrams

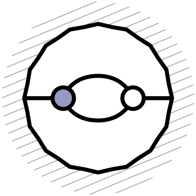
Notice that the merge and square moves leave the number of ‘faces’ of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by ‘reduction’

Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of ‘faces’ of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by ‘reduction’—also known as ‘bubble deletion’:

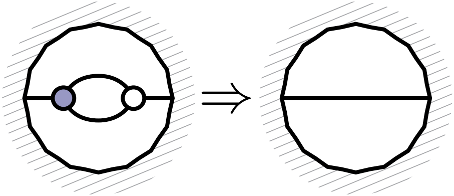
Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':



Combinatorial Characterization of On-Shell Diagrams

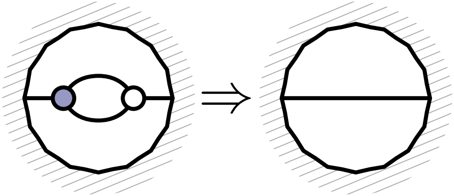
Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':



Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':

Bubble-deletion does not, however, relate 'identical' on-shell diagrams:

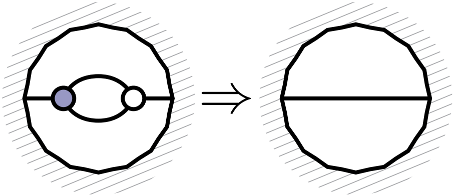


Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':

Bubble-deletion does not, however, relate 'identical' on-shell diagrams:

- it leaves behind an overall factor of $d\alpha/\alpha$ in the on-shell function

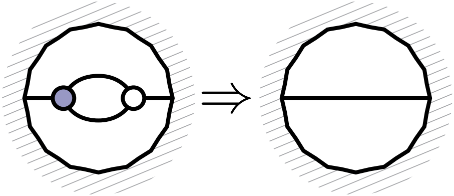


Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':

Bubble-deletion does not, however, relate 'identical' on-shell diagrams:

- it leaves behind an overall factor of $d\alpha/\alpha$ in the on-shell function
- and it alters the corresponding left-right path permutation

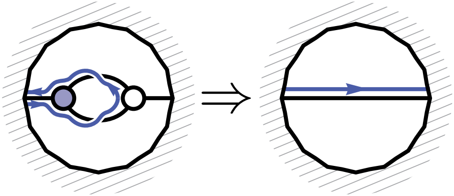


Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of ‘faces’ of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by ‘reduction’—also known as ‘bubble deletion’:

Bubble-deletion does not, however, relate ‘identical’ on-shell diagrams:

- it leaves behind an overall factor of $d\alpha/\alpha$ in the on-shell function
- and it alters the corresponding left-right path permutation

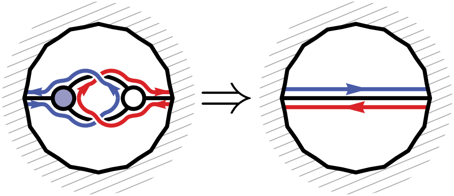


Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':

Bubble-deletion does not, however, relate 'identical' on-shell diagrams:

- it leaves behind an overall factor of $d\alpha/\alpha$ in the on-shell function
- and it alters the corresponding left-right path permutation



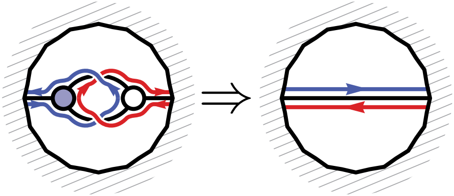
Combinatorial Characterization of On-Shell Diagrams

Notice that the merge and square moves leave the number of 'faces' of an on-shell diagram invariant. Diagrams with different numbers of faces can be related by 'reduction'—also known as 'bubble deletion':

Bubble-deletion does not, however, relate 'identical' on-shell diagrams:

- it leaves behind an overall factor of $d\alpha/\alpha$ in the on-shell function
- and it alters the corresponding left-right path permutation

Such factors of $d\alpha/\alpha$ arising from bubble deletion encode loop integrands!



Canonical Coordinates for Computing On-Shell Functions

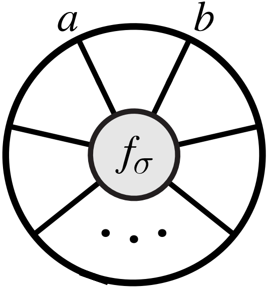
Recall that attaching ‘BCFW bridges’ can lead to very rich on-shell diagrams.

Canonical Coordinates for Computing On-Shell Functions

Recall that attaching ‘BCFW bridges’ can lead to very rich on-shell diagrams.
Conveniently, adding a BCFW bridge acts very nicely on permutations:

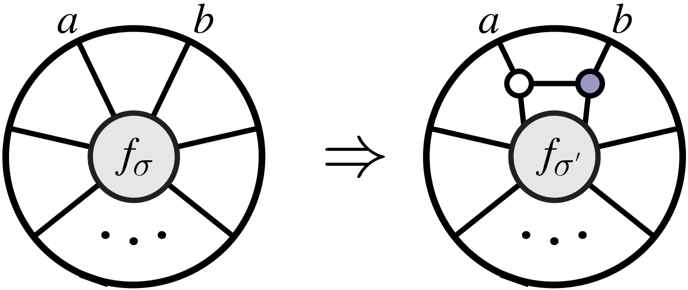
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams. Conveniently, adding a BCFW bridge acts very nicely on permutations:



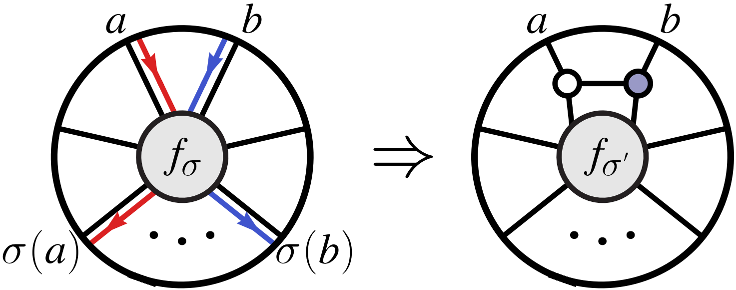
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams. Conveniently, adding a BCFW bridge acts very nicely on permutations:



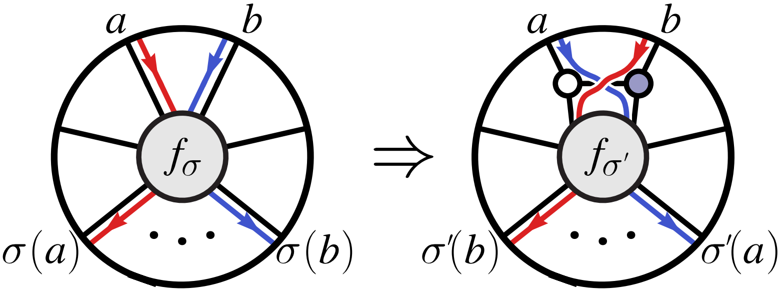
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching ‘BCFW bridges’ can lead to very rich on-shell diagrams. Conveniently, adding a BCFW bridge acts very nicely on permutations:



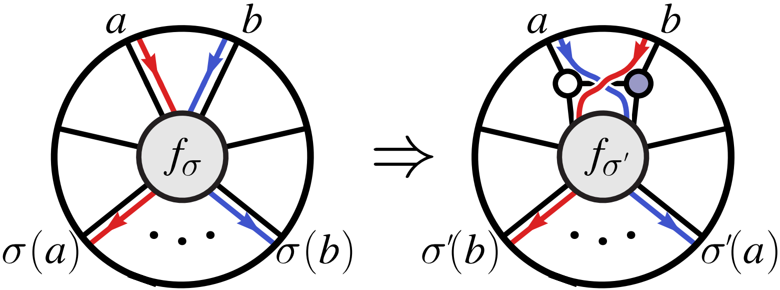
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams.
Conveniently, adding a BCFW bridge acts very nicely on permutations:



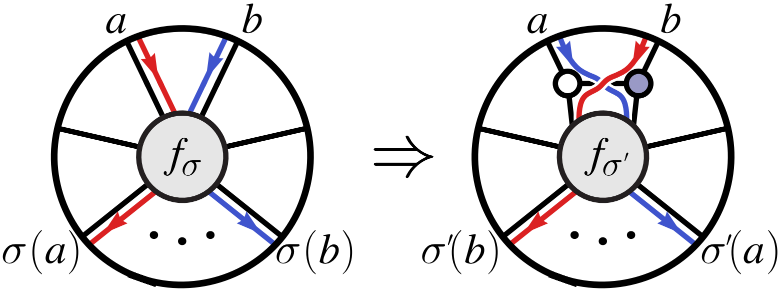
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching ‘BCFW bridges’ can lead to very rich on-shell diagrams.
Conveniently, adding a BCFW bridge acts very nicely on permutations:
it merely transposes the images of σ!



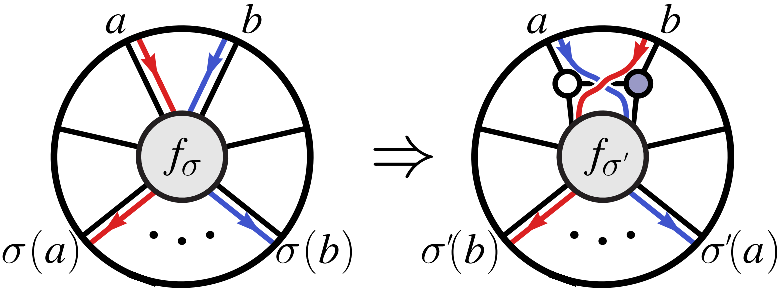
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams.
Conveniently, adding a BCFW bridge acts very nicely on permutations:
it merely transposes the images of σ!



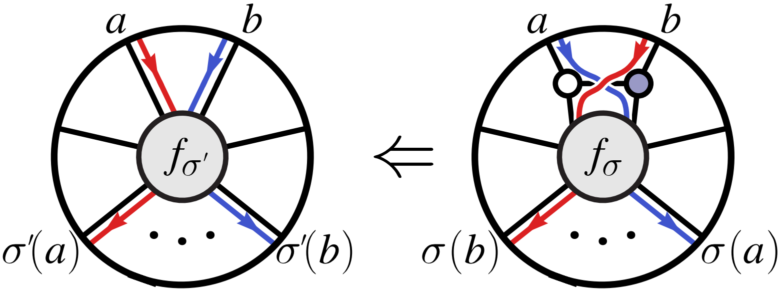
Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams.
Read the other way,



Canonical Coordinates for Computing On-Shell Functions

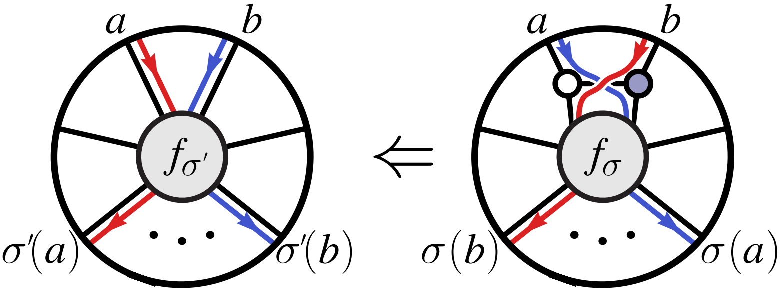
Recall that attaching ‘BCFW bridges’ can lead to very rich on-shell diagrams.
Read the other way,



Canonical Coordinates for Computing On-Shell Functions

Recall that attaching 'BCFW bridges' can lead to very rich on-shell diagrams.

Read the other way, we can 'peel-off' bridges and thereby decompose a permutation into transpositions according to $\sigma = (a\, b) \circ \sigma'$



Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions

'Bridge' Decomposition

$$\sigma:\begin{pmatrix}1&2&3&4&5&6\\\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow\\3&5&6&7&8&10\end{pmatrix}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions

‘Bridge’ Decomposition

$$\begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ f_\sigma & \{3 & 5 & 6 & 7 & 8 & 10\} \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

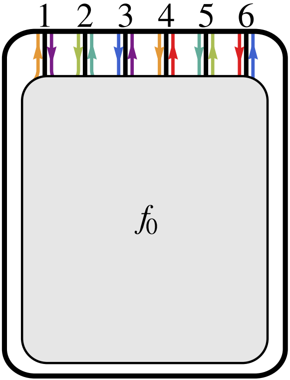
There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

‘Bridge’ Decomposition

$$\begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ f_\sigma & \{3 & 5 & 6 & 7 & 8 & 10\} \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

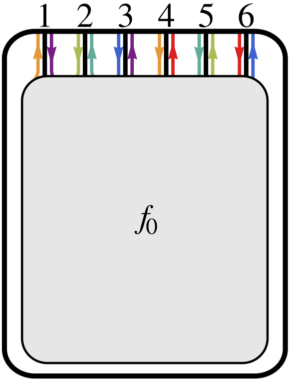


'Bridge' Decomposition

$$\begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\ f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



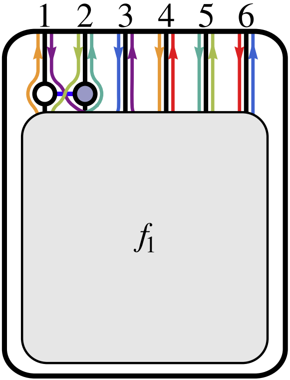
'Bridge' Decomposition

$$\begin{array}{c c c c c c c c}
 & 1 & 2 & 3 & 4 & 5 & 6 & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\
f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & (1\,2)
\end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1} f_1$$



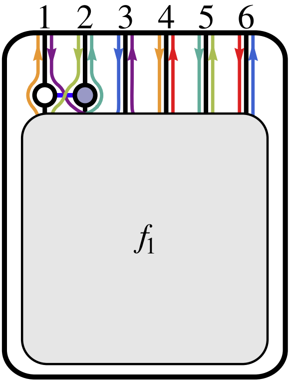
'Bridge' Decomposition

$$\begin{array}{lcccccccl} & 1 & 2 & 3 & 4 & 5 & 6 & & \tau \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\ f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & & \\ & & & & & & & & (1\,2) \\ f_1 & \{5 & 3 & 6 & 7 & 8 & 10\} & & \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1} f_1$$



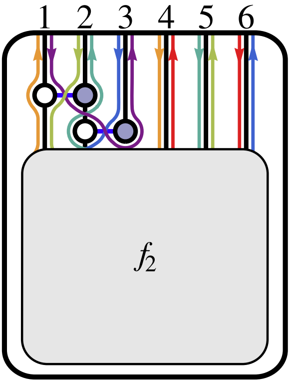
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2} f_2$$



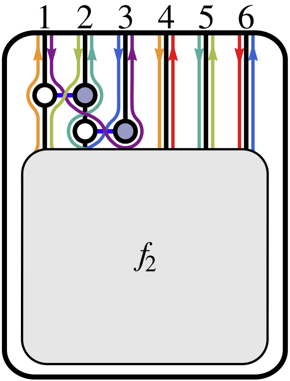
‘Bridge’ Decomposition

$$\begin{array}{lccccccl}
 & 1 & 2 & 3 & 4 & 5 & 6 & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\
f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & \\
 & & & & & & & (1\,2) \\
f_1 & \{5 & 3 & 6 & 7 & 8 & 10\} & \\
 & & & & & & & (2\,3) \\
f_2 & \{5 & 6 & 3 & 7 & 8 & 10\} &
\end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2} f_2$$



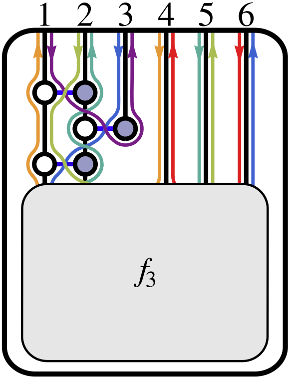
‘Bridge’ Decomposition

$$\begin{array}{lcccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \tau \\
f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & & (1\,2) \\
f_1 & \{5 & 3 & 6 & 7 & 8 & 10\} & & (2\,3) \\
f_2 & \{5 & 6 & 3 & 7 & 8 & 10\} & & (1\,2)
\end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3} f_3$$



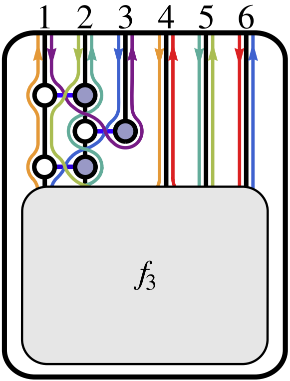
'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}f_3$$



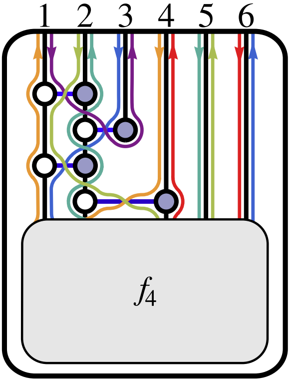
‘Bridge’ Decomposition

	1	2	3	4	5	6	
	↓	↓	↓	↓	↓	↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}f_4$$



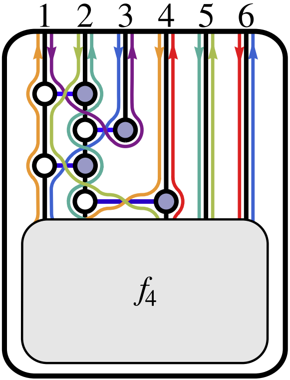
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4} f_4$$



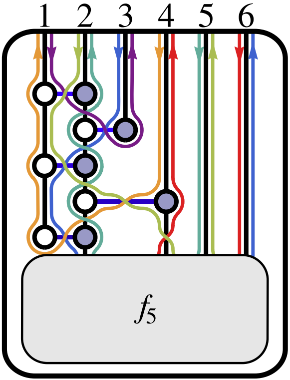
‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}f_5$$



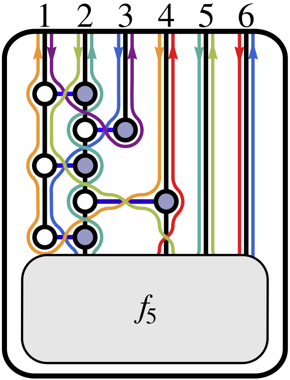
'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}f_5$$



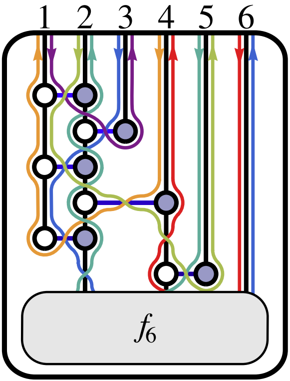
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}f_6$$



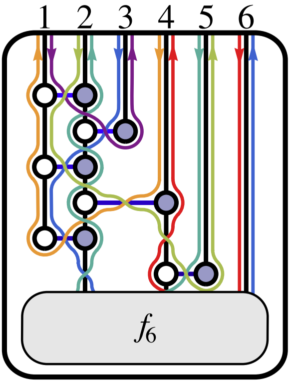
'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}f_6$$



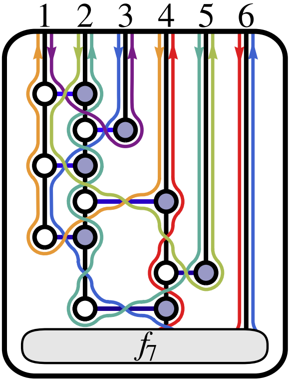
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}f_7$$



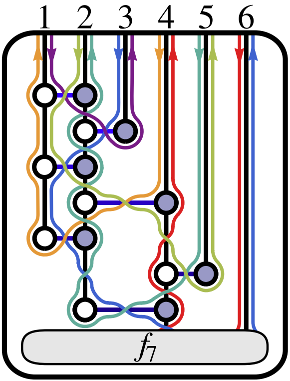
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}f_7$$



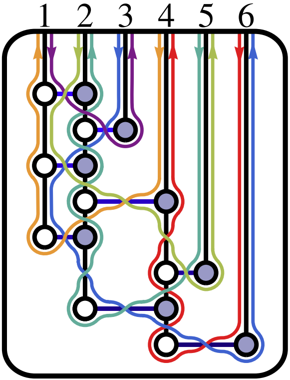
'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



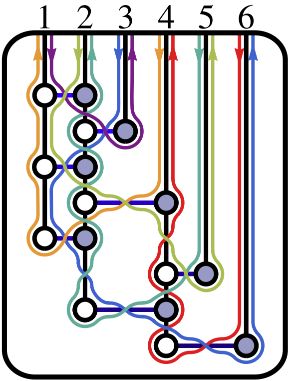
‘Bridge’ Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$



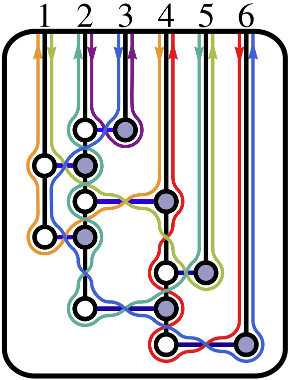
'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$



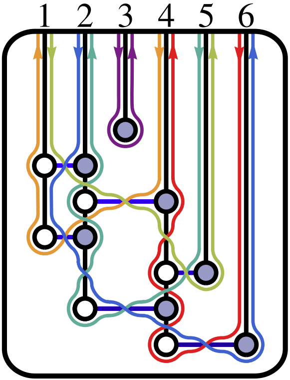
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$



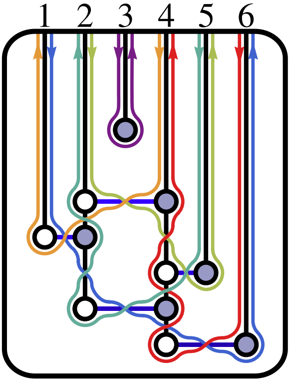
‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$



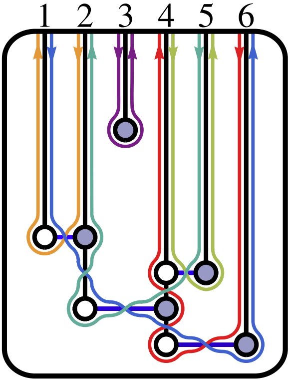
‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



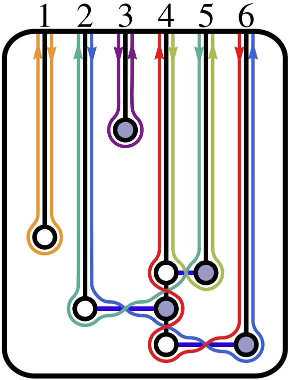
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



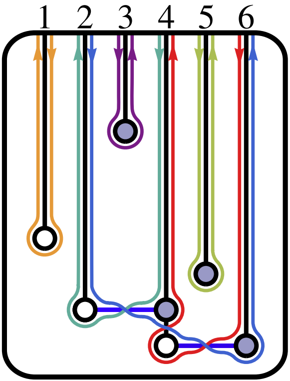
‘Bridge’ Decomposition

	1	2	3	4	5	6	
	↓	↓	↓	↓	↓	↓	τ
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



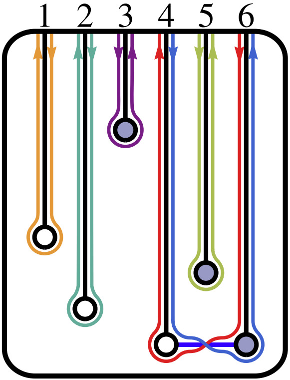
‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



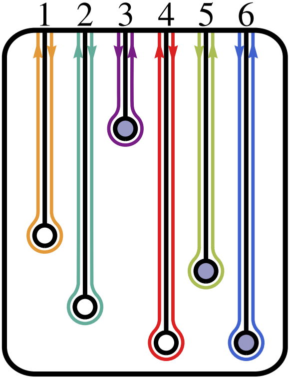
'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_7	$\{7$	8	3	6	5	$10\}$	$(4\,6)$
f_8	$\{7$	8	3	10	5	$6\}$	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



'Bridge' Decomposition

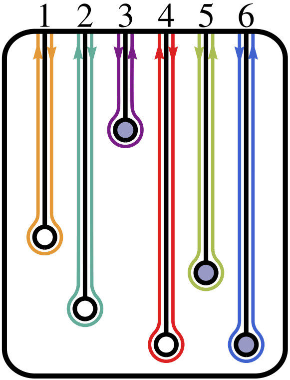
$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \end{array}$$

f_8 $\{7\ \ 8\ \ 3\ \ 10\ \ 5\ \ 6\ \}$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$



‘Bridge’ Decomposition

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \end{array}$$

$f_8\ \{7\ \ 8\ \ 3\ \ 10\ \ 5\ \ 6\ \}$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_8 = \prod_{a=\sigma(a)+n}\Big(\delta^4(\widetilde{\eta}_a)\,\delta^2(\widetilde{\lambda}_a)\Big)\prod_{b=\sigma(b)}\Big(\delta^2(\lambda_b)\Big)$$

‘Bridge’ Decomposition

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array} \qquad \tau$$

$$f_8\ \{7\ \ 8\ \ 3\ \ 10\ \ 5\ \ 6\ \}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_8 = \prod_{a=\sigma(a)+n}\left(\delta^4(\widetilde{\eta}_a)\,\delta^2(\widetilde{\lambda}_a)\right)\prod_{b=\sigma(b)}\left(\delta^2(\lambda_b)\right)$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

‘Bridge’ Decomposition

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \end{array}$$

$f_8\ \{7\ \ 8\ \ 3\ \ 10\ \ 5\ \ 6\ \}$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$

$$f_8 = \delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

'Bridge' Decomposition

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \end{array}$$

$$f_8\,\{7\;\;8\;\;3\;\;10\;\;5\;\;6\;\}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$

$$f_8 = \delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C \equiv \begin{array}{c} \\ \\ \end{array}\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \end{array}$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

‘Bridge’ Decomposition

$$\begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\ \\ f_8 & \{7 & 8 & 3 & 10 & 5 & 6\,\} \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$

$$f_7 = \frac{d\alpha_8}{\alpha_8}\delta^{3\times4}\big(C\cdot\widetilde{\eta}\big)\delta^{3\times2}\big(C\cdot\widetilde{\lambda}\big)\delta^{2\times3}\big(\lambda\cdot C^{\perp}\big)$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

$$(4\,6):\quad c_6 \mapsto c_6 + \alpha_8\, c_4$$

'Bridge' Decomposition

$$\begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\ f_7 & \{7 & 8 & 3 & 6 & 5 & 10\} & (4\,6) \\ f_8 & \{7 & 8 & 3 & 10 & 5 & 6\,\} & \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$

$$f_6 = \frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	
	↓	↓	↓	↓	↓	↓	τ
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_5 = \frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

$$(4\,5):\quad c_5 \mapsto c_5 + \alpha_6\,c_4$$

'Bridge' Decomposition

$$\begin{array}{ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & & \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \tau \\ f_5 & \{7 & 6 & 3 & 5 & 8 & 10\} & & (4\,5) \\ f_6 & \{7 & 6 & 3 & 8 & 5 & 10\} & & (2\,4) \\ f_7 & \{7 & 8 & 3 & 6 & 5 & 10\} & & (4\,6) \\ f_8 & \{7 & 8 & 3 & 10 & 5 & 6\} & & \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_4=\frac{d\alpha_5}{\alpha_5}\cdots\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&\alpha_5&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

$$(1\,2):\quad c_2\mapsto c_2+\alpha_5\,c_1$$

‘Bridge’ Decomposition

$$\begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\ f_4 & \{6 & 7 & 3 & 5 & 8 & 10\} & \\ & & & & & & & (1\,2) \\ f_5 & \{7 & 6 & 3 & 5 & 8 & 10\} & \\ & & & & & & & (4\,5) \\ f_6 & \{7 & 6 & 3 & 8 & 5 & 10\} & \\ & & & & & & & (2\,4) \\ f_7 & \{7 & 8 & 3 & 6 & 5 & 10\} & \\ & & & & & & & (4\,6) \\ f_8 & \{7 & 8 & 3 & 10 & 5 & 6\,\} & \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_3=\frac{d\alpha_4}{\alpha_4}\cdots\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

$$(2\,4):\quad c_4\mapsto c_4+\alpha_4\,c_2$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_2=\frac{d\alpha_3}{\alpha_3}\cdots\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\delta^{3\times2}(C\cdot\widetilde{\lambda})\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

$$(1\,2):\quad c_2\mapsto c_2+\alpha_3\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_1=\frac{d\alpha_2}{\alpha_2}\;\cdots\;\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}\big(C\cdot\widetilde{\eta}\big)\delta^{3\times2}\big(C\cdot\widetilde{\lambda}\big)\delta^{2\times3}\big(\lambda\cdot C^{\perp}\big)$$

$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

$$(2\,3):\quad c_3\mapsto c_3+\alpha_2\,c_2$$

‘Bridge’ Decomposition

	$1\atop\downarrow$	$2\atop\downarrow$	$3\atop\downarrow$	$4\atop\downarrow$	$5\atop\downarrow$	$6\atop\downarrow$	τ
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_0 = \frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}(C\cdot\widetilde{\eta})\,\delta^{3\times2}(C\cdot\widetilde{\lambda})\,\delta^{2\times3}(\lambda\cdot C^{\perp})$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_1\,c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:

$$f_0=\frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8}f_8$$

$$f_0=\frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_8}{\alpha_8}\delta^{3\times4}\big(C\cdot\widetilde{\eta}\big)\delta^{3\times2}\big(C\cdot\widetilde{\lambda}\big)\delta^{2\times3}\big(\lambda\cdot C^{\perp}\big)$$

$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$f_0 = \frac{d\alpha_1}{\alpha_1}\frac{d\alpha_2}{\alpha_2}\frac{d\alpha_3}{\alpha_3}\frac{d\alpha_4}{\alpha_4}\frac{d\alpha_5}{\alpha_5}\frac{d\alpha_6}{\alpha_6}\frac{d\alpha_7}{\alpha_7}\frac{d\alpha_8}{\alpha_8} f_8$$

$$f_0 = \frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_8}{\alpha_8}\,\delta^{3\times4}\big(C\cdot\widetilde{\eta}\big)\,\delta^{3\times2}\big(C\cdot\widetilde{\lambda}\big)\,\delta^{2\times3}\big(\lambda\cdot C^{\perp}\big)$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

$$\begin{array}{lcccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 & \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau \\ f_0 & \{ & 3 & 5 & 6 & 7 & 8 & 10 & \} \\ & & & & & & & & (1\,2) \\ f_1 & \{ & 5 & 3 & 6 & 7 & 8 & 10 & \} \\ & & & & & & & & (2\,3) \\ f_2 & \{ & 5 & 6 & 3 & 7 & 8 & 10 & \} \\ & & & & & & & & (1\,2) \\ f_3 & \{ & 6 & 5 & 3 & 7 & 8 & 10 & \} \\ & & & & & & & & (2\,4) \\ f_4 & \{ & 6 & 7 & 3 & 5 & 8 & 10 & \} \\ & & & & & & & & (1\,2) \\ f_5 & \{ & 7 & 6 & 3 & 5 & 8 & 10 & \} \\ & & & & & & & & (4\,5) \\ f_6 & \{ & 7 & 6 & 3 & 8 & 5 & 10 & \} \\ & & & & & & & & (2\,4) \\ f_7 & \{ & 7 & 8 & 3 & 6 & 5 & 10 & \} \\ & & & & & & & & (4\,6) \\ f_8 & \{ & 7 & 8 & 3 & 10 & 5 & 6 & \} \end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

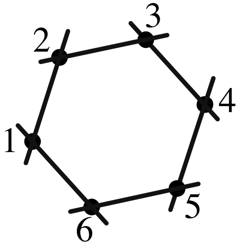
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

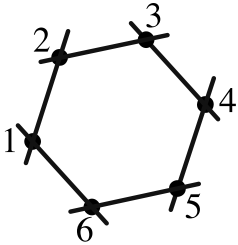
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

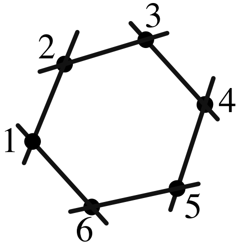
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

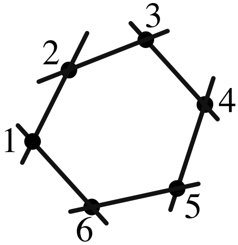
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\begin{array}{c}\\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

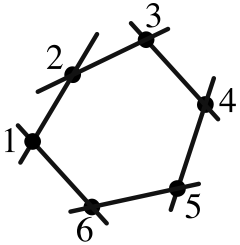
$$(6\,1):\quad c_1\mapsto c_1+\alpha_0\,c_6$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

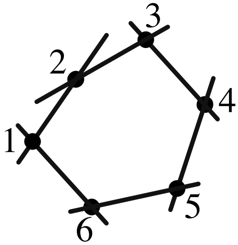
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

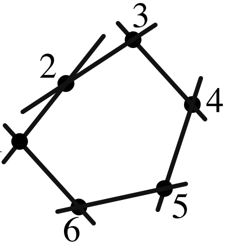
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

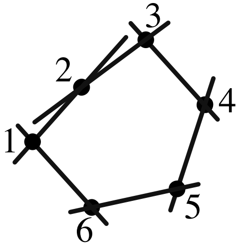
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

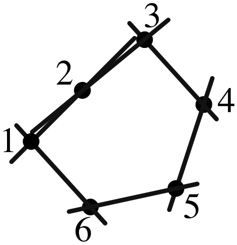
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

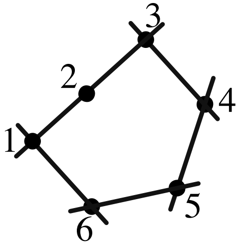
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



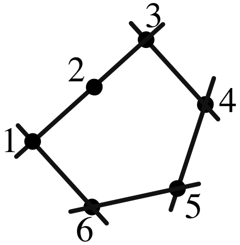
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



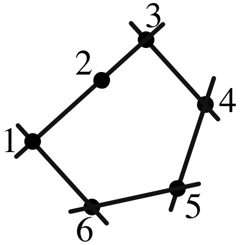
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



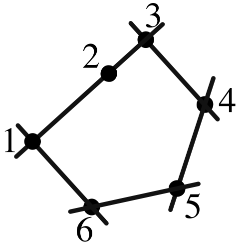
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



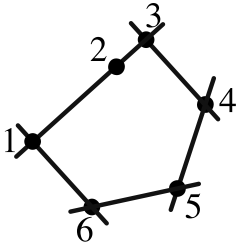
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



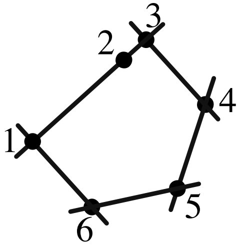
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:



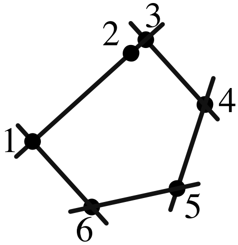
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



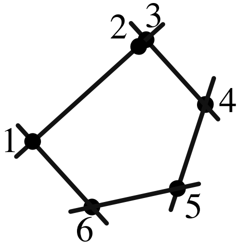
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



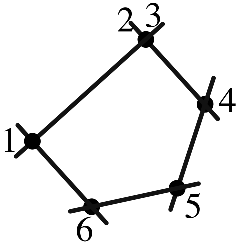
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



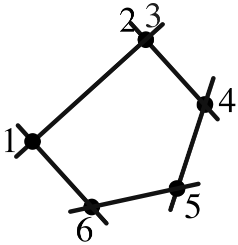
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



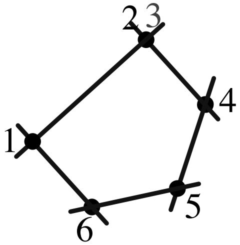
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



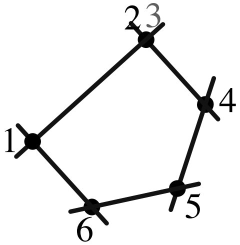
$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



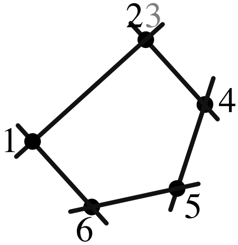
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



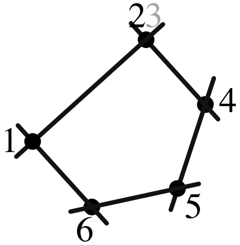
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:



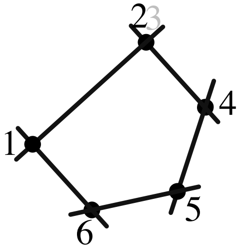
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



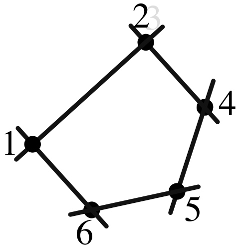
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



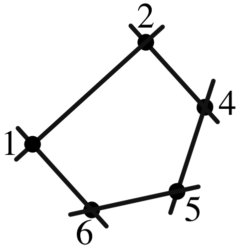
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



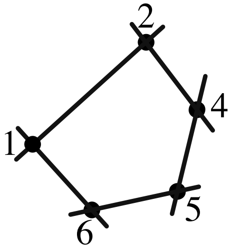
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



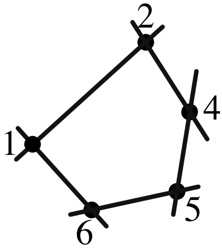
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



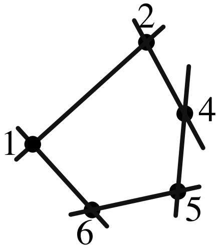
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



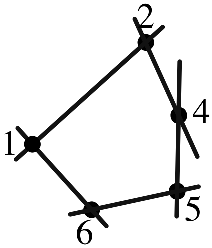
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



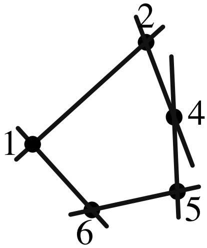
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



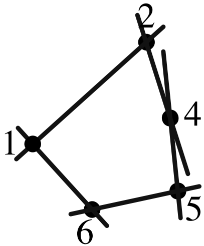
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



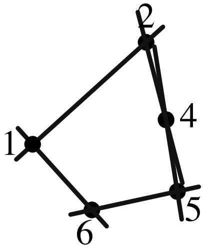
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



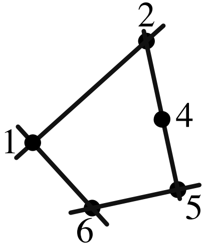
$$C \equiv \begin{array}{c} \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



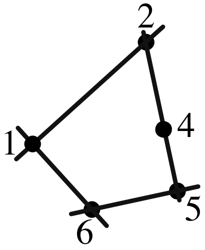
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



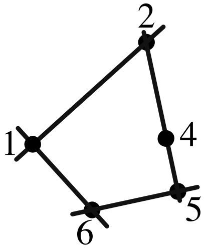
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



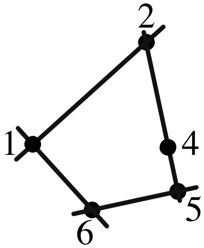
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&\alpha_5&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



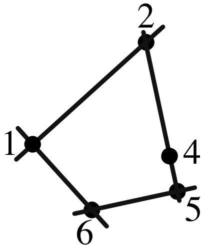
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



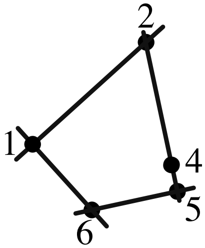
$$C \equiv \begin{array}{c} \\ \\ \end{array}\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



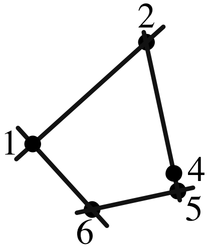
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&\alpha_5&0&\alpha_4\alpha_5&0&0\\0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



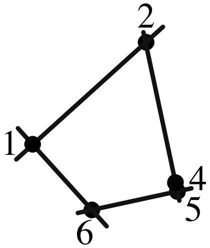
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



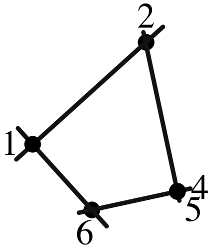
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



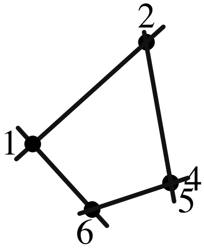
$$C \equiv \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}$$

‘Bridge’ Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



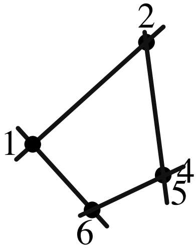
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



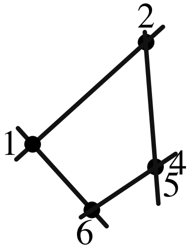
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&\alpha_5&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



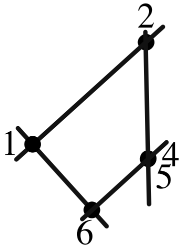
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



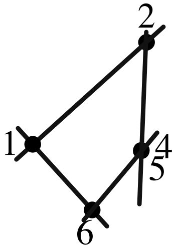
$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0\\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



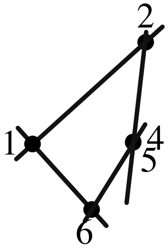
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&\alpha_5&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



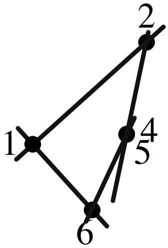
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



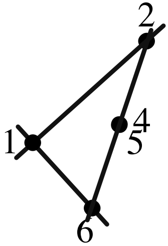
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&\alpha_5&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



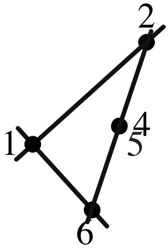
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



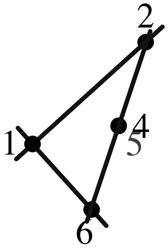
$$C \equiv \begin{array}{c} \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



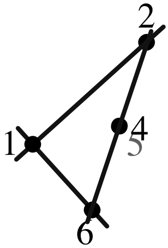
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



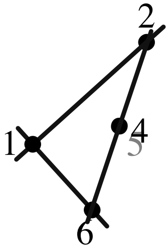
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



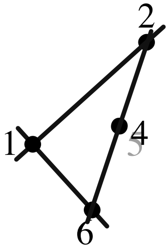
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



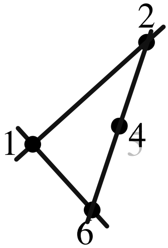
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



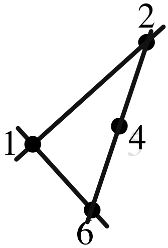
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



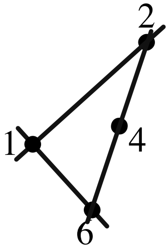
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



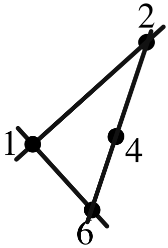
$$C \equiv \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}$$

‘Bridge’ Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



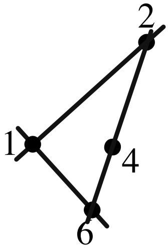
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



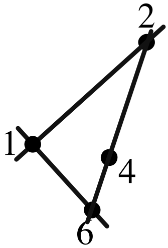
$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \alpha_7 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



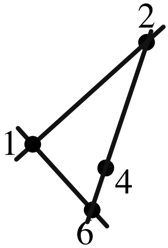
$$C\equiv\left(\begin{array}{cccccc}1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \alpha_7 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & \alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



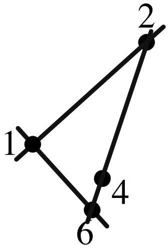
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



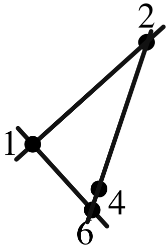
$$C \equiv \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ & 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



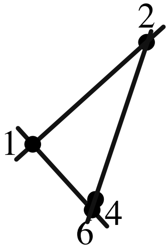
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



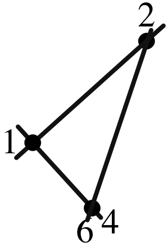
$$C \equiv \left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&0&0\\ 0&0&0&1&0&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



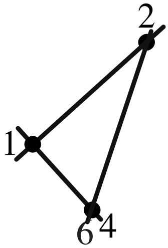
$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&1&0&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



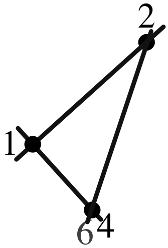
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



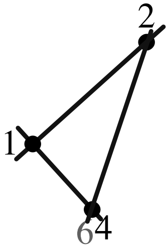
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



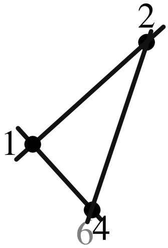
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&\alpha_8\end{array}\right)$$

'Bridge' Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



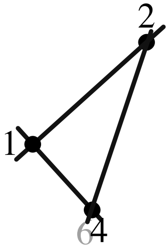
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:



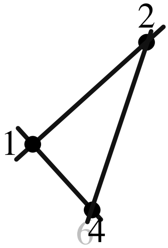
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



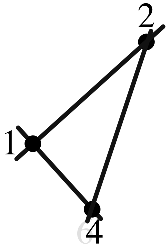
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&\alpha_8\end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



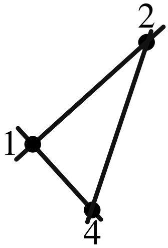
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



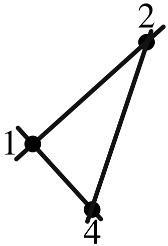
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



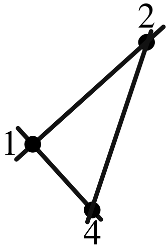
$$C \equiv \begin{array}{c} \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



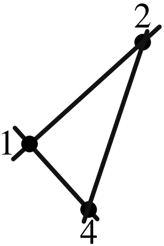
$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



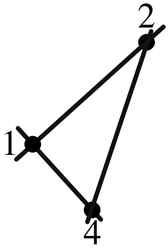
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



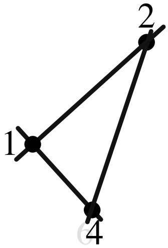
$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\end{array}\right)$$

'Bridge' Decomposition

$$\begin{array}{c|cccccc|c}
 & 1 & 2 & 3 & 4 & 5 & 6 & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \tau\\
f_0 & \{3 & 5 & 6 & 7 & 8 & 10\} & (1\,2)\\
f_1 & \{5 & 3 & 6 & 7 & 8 & 10\} & (2\,3)\\
f_2 & \{5 & 6 & 3 & 7 & 8 & 10\} & (1\,2)\\
f_3 & \{6 & 5 & 3 & 7 & 8 & 10\} & (2\,4)\\
f_4 & \{6 & 7 & 3 & 5 & 8 & 10\} & (1\,2)\\
f_5 & \{7 & 6 & 3 & 5 & 8 & 10\} & (4\,5)\\
f_6 & \{7 & 6 & 3 & 8 & 5 & 10\} & (2\,4)\\
f_7 & \{7 & 8 & 3 & 6 & 5 & 10\} & (4\,6)\\
f_8 & \{7 & 8 & 3 & 10 & 5 & 6\} &
\end{array}$$

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&1&0&\alpha_8\end{array}\right)$$

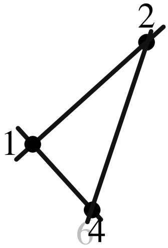
$$(4\,6):\quad c_6\mapsto c_6+\alpha_8\,c_4$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & \alpha_8\end{array}\right)$$

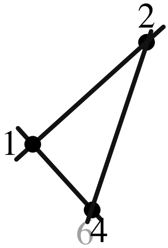
$$(4\,6):\quad c_6\mapsto c_6+\alpha_8\,c_4$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

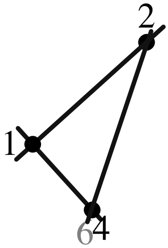
$$(4\,6): \quad c_6 \mapsto c_6 + \alpha_8\, c_4$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

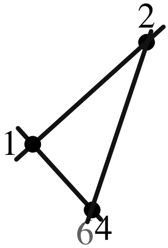
$$(4\,6): \quad c_6 \mapsto c_6 + \alpha_8\, c_4$$

'Bridge' Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	$\{3$	5	6	7	8	$10\}$	$(1\,2)$
f_1	$\{5$	3	6	7	8	$10\}$	$(2\,3)$
f_2	$\{5$	6	3	7	8	$10\}$	$(1\,2)$
f_3	$\{6$	5	3	7	8	$10\}$	$(2\,4)$
f_4	$\{6$	7	3	5	8	$10\}$	$(1\,2)$
f_5	$\{7$	6	3	5	8	$10\}$	$(4\,5)$
f_6	$\{7$	6	3	8	5	$10\}$	$(2\,4)$
f_7	$\{7$	8	3	6	5	$10\}$	$(4\,6)$
f_8	$\{7$	8	3	10	5	$6\}$	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&\alpha_8\end{array}\right)$$

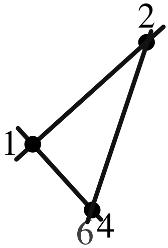
$(4\,6):\quad c_6\mapsto c_6+\alpha_8\,c_4$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

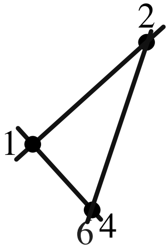
$$(4\,6): \quad c_6 \mapsto c_6 + \alpha_8\, c_4$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}$$

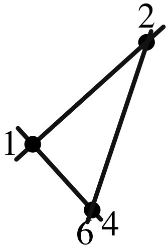
$$(4\,6): \quad c_6 \mapsto c_6 + \alpha_8\, c_4$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

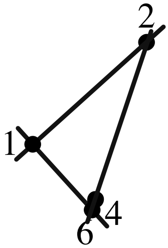
$(4\,6):\quad c_6\mapsto c_6+\alpha_8\,c_4$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

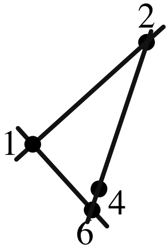
$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

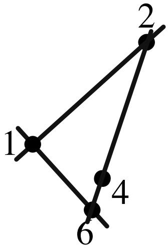
$(2\,4): \quad c_4 \mapsto c_4 + \alpha_7\, c_2$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&\alpha_7&0&0\\0&0&0&1&0&\alpha_8\end{array}\right)$$

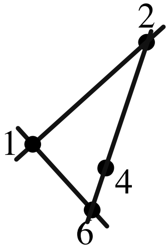
$(2\,4):\quad c_4\mapsto c_4+\alpha_7\,c_2$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

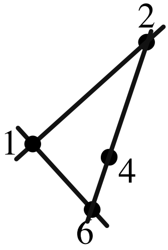
$(2\,4)$: $c_4 \mapsto c_4 + \alpha_7\, c_2$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

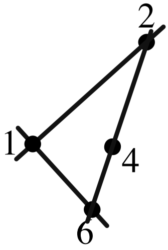
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	
	↓	↓	↓	↓	↓	↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

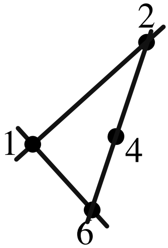
$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \end{array}$$

$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

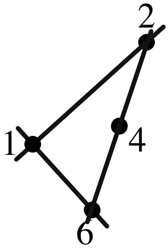
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \alpha_8 \end{array}\right)$$

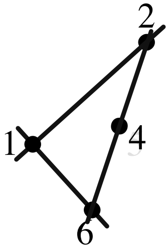
$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_7\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

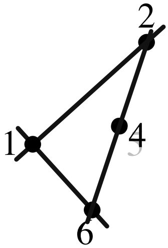
$$(4\,5):\quad c_5\mapsto c_5+\alpha_6\,c_4$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&0&0&0&0&0\\ 0&1&0&\alpha_7&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

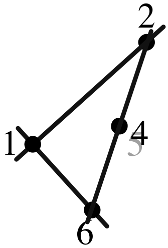
$(4\,5):\quad c_5\mapsto c_5+\alpha_6\,c_4$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0\\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

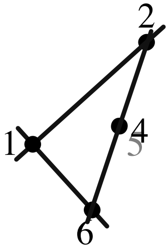
$$(4\,5):\quad c_5\mapsto c_5+\alpha_6\,c_4$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

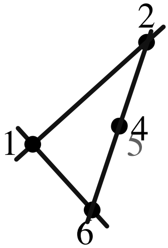
$(4\,5)$: $\quad c_5 \mapsto c_5 + \alpha_6\, c_4$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

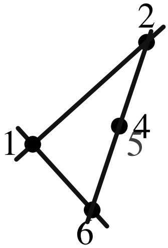
$$(4\,5):\quad c_5\mapsto c_5+\alpha_6\,c_4$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&0&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

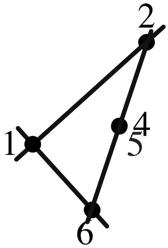
$(4\,5):\quad c_5\mapsto c_5+\alpha_6\,c_4$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

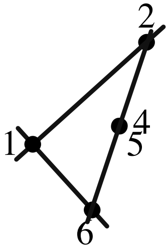
$$(4\,5): \quad c_5 \mapsto c_5 + \alpha_6\, c_4$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}$$

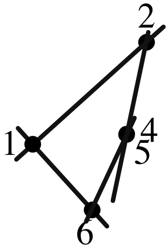
$$(4\,5):\quad c_5 \mapsto c_5 + \alpha_6\, c_4$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\begin{array}{c}\begin{array}{cccccc}1&2&3&4&5&6\end{array}\\\left(\begin{array}{cccccc}1&\alpha_5&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)\end{array}$$

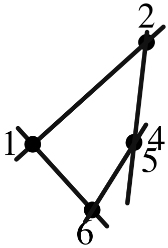
$$(1\,2):\quad c_2\mapsto c_2+\alpha_5\,c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

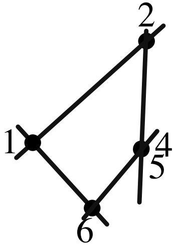
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_5\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

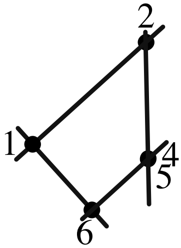
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_5\, c_1$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&\alpha_5&0&0&0&0\\0&1&0&\alpha_7&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

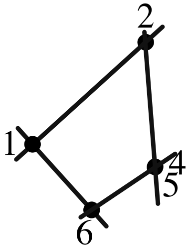
$(1\,2)$: $\quad c_2\mapsto c_2+\alpha_5\,c_1$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

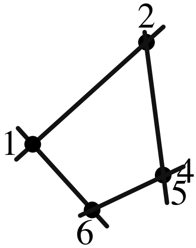
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_5\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

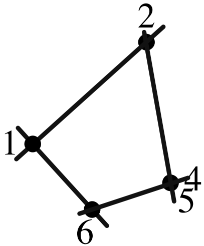
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_5\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0\\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

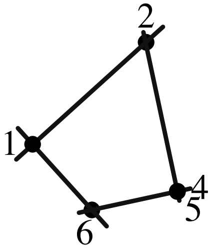
$$(1\,2):\quad c_2\mapsto c_2+\alpha_5\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha_7 & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

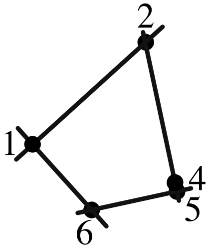
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_5\, c_1$$

'Bridge' Decomposition

	$\overset{1}{\downarrow}$	$\overset{2}{\downarrow}$	$\overset{3}{\downarrow}$	$\overset{4}{\downarrow}$	$\overset{5}{\downarrow}$	$\overset{6}{\downarrow}$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

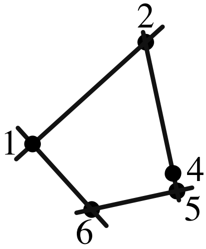
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&\alpha_5&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

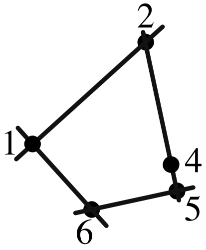
$$(2\,4):\quad c_4\mapsto c_4+\alpha_4\,c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&\alpha_5&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

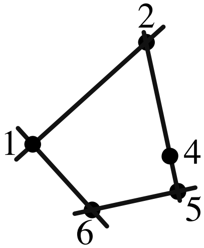
$$(2\,4):\quad c_4\mapsto c_4+\alpha_4\,c_2$$

'Bridge' Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ & 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}$$

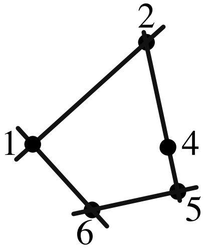
$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

‘Bridge’ Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	$\{3$	5	6	7	8	$10\}$	$(1\,2)$
f_1	$\{5$	3	6	7	8	$10\}$	$(2\,3)$
f_2	$\{5$	6	3	7	8	$10\}$	$(1\,2)$
f_3	$\{6$	5	3	7	8	$10\}$	$(2\,4)$
f_4	$\{6$	7	3	5	8	$10\}$	$(1\,2)$
f_5	$\{7$	6	3	5	8	$10\}$	$(4\,5)$
f_6	$\{7$	6	3	8	5	$10\}$	$(2\,4)$
f_7	$\{7$	8	3	6	5	$10\}$	$(4\,6)$
f_8	$\{7$	8	3	10	5	$6\}$	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ & 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ & 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}$$

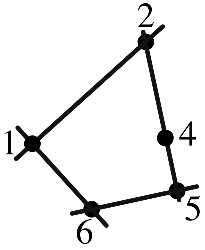
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

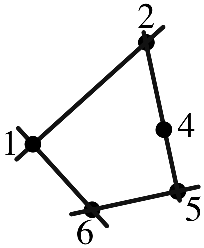
$$(2\,4):\quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

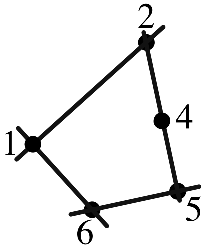
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \alpha_5 & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

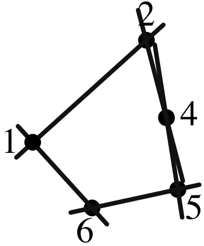
$$(2\,4): \quad c_4 \mapsto c_4 + \alpha_4\, c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

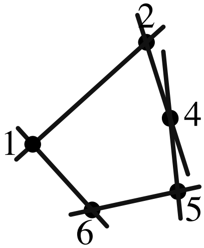
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

'Bridge' Decomposition

	$1 \downarrow$	$2 \downarrow$	$3 \downarrow$	$4 \downarrow$	$5 \downarrow$	$6 \downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

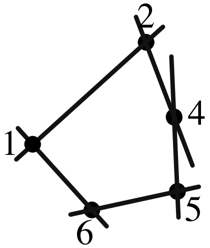
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

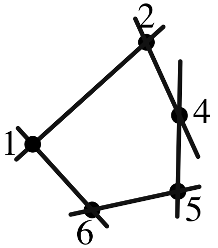
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

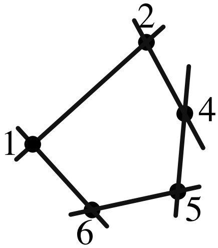
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

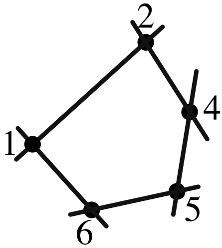
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_3\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&0&\alpha_4\alpha_5&0&0\\ 0&1&0&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

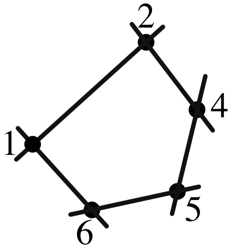
$$(1\,2):\quad c_2\mapsto c_2+\alpha_3\,c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

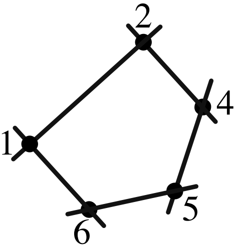
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & 0 & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & 0 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

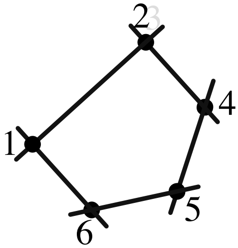
$$(1\,2): \quad c_2 \mapsto c_2 + \alpha_3\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

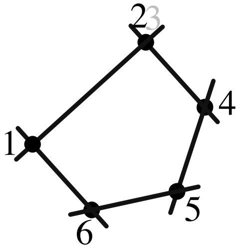
$$(2\,3):\quad c_3\mapsto c_3+\alpha_2\,c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

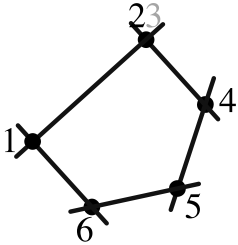
$$(2\,3): \quad c_3 \mapsto c_3 + \alpha_2\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

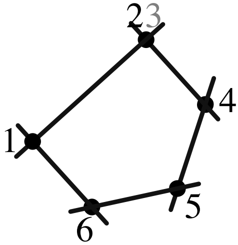
$$(2\,3):\quad c_3\mapsto c_3+\alpha_2\,c_2$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\, b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

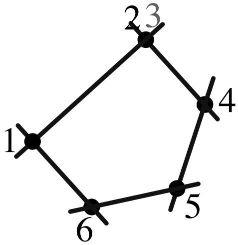
$$(2\,3):\quad c_3 \mapsto c_3 + \alpha_2\, c_2$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

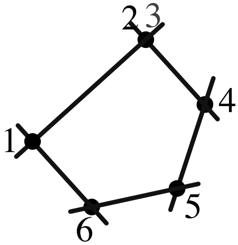
$$(2\,3): \quad c_3 \mapsto c_3 + \alpha_2\, c_2$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

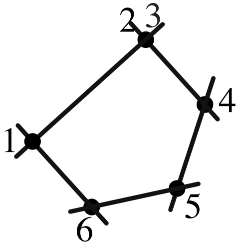
$$(2\,3):\quad c_3\mapsto c_3+\alpha_2\,c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

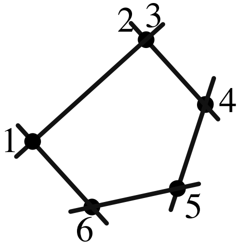
$$(2\,3): \quad c_3 \mapsto c_3 + \alpha_2\, c_2$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

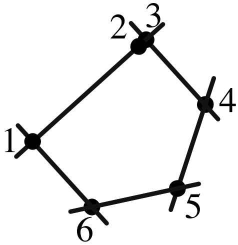
$$(2\,3): \quad c_3 \mapsto c_3 + \alpha_2\, c_2$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\\hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

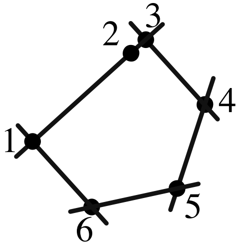
$$(1\,2):\quad c_2\mapsto c_2+\alpha_1\,c_1$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

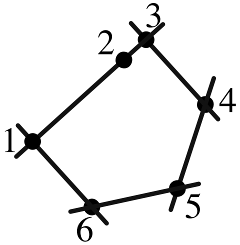
$$(1\,2):\quad c_2\mapsto c_2+\alpha_1\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\ 0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\ 0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

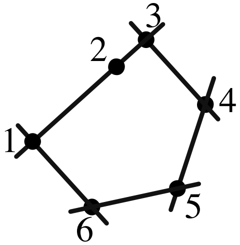
$$(1\,2):\quad c_2\mapsto c_2+\alpha_1\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

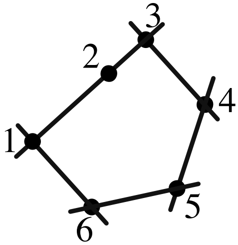
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_1\, c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc}1&2&3&4&5&6\\ \hline 1&(\alpha_1+\alpha_3+\alpha_5)&\alpha_2(\alpha_3+\alpha_5)&\alpha_4\alpha_5&0&0\\0&1&\alpha_2&(\alpha_4+\alpha_7)&\alpha_6\alpha_7&0\\0&0&0&1&\alpha_6&\alpha_8\end{array}\right)$$

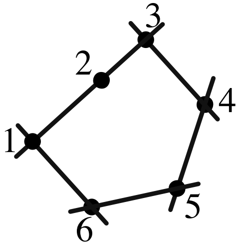
$$(1\,2):\quad c_2\mapsto c_2+\alpha_1\,c_1$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

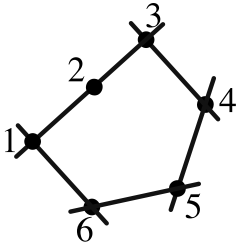
$$(1\,2):\quad c_2\mapsto c_2+\alpha_1\,c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

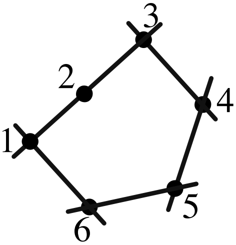
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_1\, c_1$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ 0 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

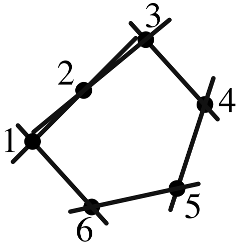
$$(1\,2):\quad c_2 \mapsto c_2 + \alpha_1\, c_1$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

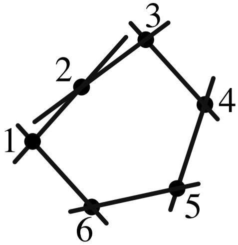
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

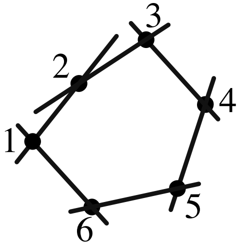
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

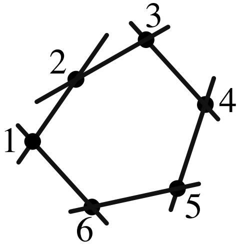
$$(6\,1):\quad c_1\mapsto c_1+\alpha_0\,c_6$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	
	↓	↓	↓	↓	↓	↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau\equiv(a\,b)$ such that $\sigma(a)<\sigma(b)$:



$$C\equiv\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6\\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

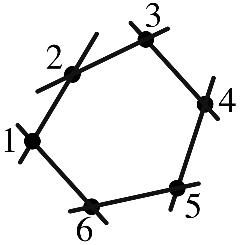
$$(6\,1):\quad c_1\mapsto c_1+\alpha_0\,c_6$$

'Bridge' Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

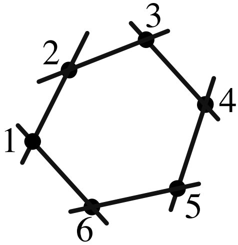
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	$1\downarrow$	$2\downarrow$	$3\downarrow$	$4\downarrow$	$5\downarrow$	$6\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

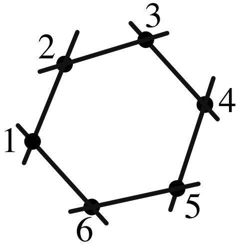
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

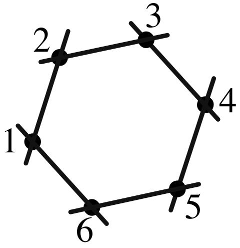
$$(6\,1): \quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

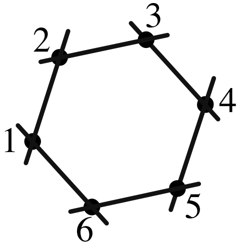
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

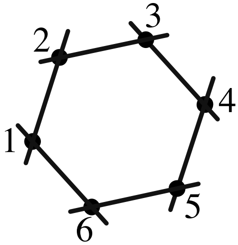
$$(6\,1):\quad c_1 \mapsto c_1 + \alpha_0\, c_6$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:



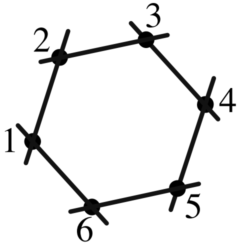
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 $\downarrow$	2 $\downarrow$	3 $\downarrow$	4 $\downarrow$	5 $\downarrow$	6 $\downarrow$	τ
f_0	{3	5	6	7	8	10}	$(1\,2)$
f_1	{5	3	6	7	8	10}	$(2\,3)$
f_2	{5	6	3	7	8	10}	$(1\,2)$
f_3	{6	5	3	7	8	10}	$(2\,4)$
f_4	{6	7	3	5	8	10}	$(1\,2)$
f_5	{7	6	3	5	8	10}	$(4\,5)$
f_6	{7	6	3	8	5	10}	$(2\,4)$
f_7	{7	8	3	6	5	10}	$(4\,6)$
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

There are many ways to decompose a permutation into transpositions—*e.g.*, always choose the first transposition $\tau \equiv (a\,b)$ such that $\sigma(a) < \sigma(b)$:

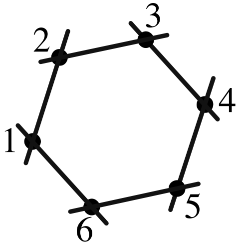


$$C \equiv \begin{array}{c} \\ \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)\end{array}$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions



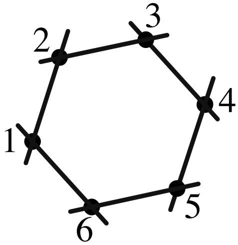
$$C\equiv\left(\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6\\ \hline & 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0\\ & 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0\\ & \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

'Bridge' Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{6,3} \equiv \frac{d\alpha_0}{\alpha_0}\cdots\frac{d\alpha_8}{\alpha_8}$$



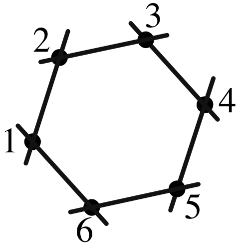
$$C \equiv \begin{array}{c} \\ \\ \end{array}\left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{6,3} \equiv \frac{d\alpha_0}{\alpha_0} \cdots \frac{d\alpha_8}{\alpha_8}$$



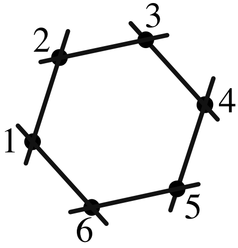
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{6,3} \equiv \frac{d\alpha_0}{\alpha_0} \cdots \frac{d\alpha_8}{\alpha_8} \quad = \frac{d^{3\times 6}C}{\mathrm{vol}(GL(3))} \frac{1}{(123)(234)(345)(456)(561)(612)}$$



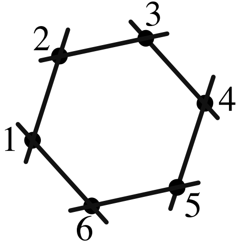
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{n,k} \equiv \frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_{k(n-k)}}{\alpha_{k(n-k)}} = \frac{d^{k\times n}C}{\mathrm{vol}(GL(k))}\,\frac{1}{(1\cdots k)\,(2\cdots k{+}1)\cdots(n\cdots k{-}1)}$$



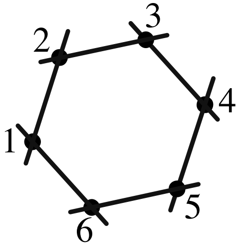
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{n,k} \equiv \frac{d\alpha_1}{\alpha_1} \cdots \frac{d\alpha_{k(n-k)}}{\alpha_{k(n-k)}} = \frac{d^{k\times n}C}{\mathrm{vol}(GL(k))}\,\frac{1}{(1\cdots k)\,(2\cdots k{+}1)\cdots(n\cdots k{-}1)}$$



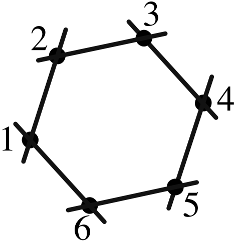
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{n,k} \equiv \frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_{k(n-k)}}{\alpha_{k(n-k)}} = \frac{d^{k\times n}C}{\text{vol}(GL(k))}\,\frac{1}{(1\cdots k)\,(2\cdots k{+}1)\cdots(n\cdots k{-}1)}$$



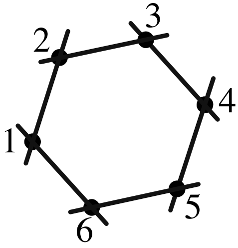
$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1	2	3	4	5	6	τ
	↓	↓	↓	↓	↓	↓	
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6 }	

Canonical Coordinates for Computing On-Shell Functions

$$\mathcal{L}_{n,k} \equiv \frac{d\alpha_1}{\alpha_1}\cdots\frac{d\alpha_{k(n-k)}}{\alpha_{k(n-k)}} = \frac{d^{k\times n}C}{\mathrm{vol}(GL(k))}\,\frac{1}{(1\cdots k)\,(2\cdots k{+}1)\cdots(n\cdots k{-}1)}$$



$$C \equiv \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & (\alpha_1+\alpha_3+\alpha_5) & \alpha_2(\alpha_3+\alpha_5) & \alpha_4\alpha_5 & 0 & 0 \\ 0 & 1 & \alpha_2 & (\alpha_4+\alpha_7) & \alpha_6\alpha_7 & 0 \\ \alpha_0\alpha_8 & 0 & 0 & 1 & \alpha_6 & \alpha_8 \end{array}\right)$$

‘Bridge’ Decomposition

	1 ↓	2 ↓	3 ↓	4 ↓	5 ↓	6 ↓	τ
f_0	{3	5	6	7	8	10}	(1 2)
f_1	{5	3	6	7	8	10}	(2 3)
f_2	{5	6	3	7	8	10}	(1 2)
f_3	{6	5	3	7	8	10}	(2 4)
f_4	{6	7	3	5	8	10}	(1 2)
f_5	{7	6	3	5	8	10}	(4 5)
f_6	{7	6	3	8	5	10}	(2 4)
f_7	{7	8	3	6	5	10}	(4 6)
f_8	{7	8	3	10	5	6}	

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

$$f = \int \frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_d}{\alpha_d} \; \delta^{k\times 4}\big(C(\vec{\alpha})\cdot\widetilde{\eta}\big)\delta^{k\times 2}\big(C(\vec{\alpha})\cdot\widetilde{\lambda}\big)\delta^{2\times(n-k)}\big(\lambda\cdot C(\vec{\alpha})^{\perp}\big)$$

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

$$f = \int \frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_d}{\alpha_d} \; \delta^{k\times 4}\big(C(\vec{\alpha})\cdot\widetilde{\eta}\big)\delta^{k\times 2}\big(C(\vec{\alpha})\cdot\widetilde{\lambda}\big)\delta^{2\times(n-k)}\big(\lambda\cdot C(\vec{\alpha})^{\perp}\big)$$

Measure-preserving diffeomorphisms leave the function invariant

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

$$f = \int \frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_d}{\alpha_d}\ \delta^{k\times 4}\big(C(\vec{\alpha})\cdot\widetilde{\eta}\big)\delta^{k\times 2}\big(C(\vec{\alpha})\cdot\widetilde{\lambda}\big)\delta^{2\times(n-k)}\big(\lambda\cdot C(\vec{\alpha})^{\perp}\big)$$

Measure-preserving diffeomorphisms leave the function invariant, but—via the δ-functions—can be recast variations of the kinematical data.

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

$$f = \int \frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_d}{\alpha_d} \; \delta^{k\times 4}\big(C(\vec{\alpha})\cdot\widetilde{\eta}\big)\delta^{k\times 2}\big(C(\vec{\alpha})\cdot\widetilde{\lambda}\big)\delta^{2\times(n-k)}\big(\lambda\cdot C(\vec{\alpha})^{\perp}\big)$$

Measure-preserving diffeomorphisms leave the function invariant, but—via the δ-functions—can be recast variations of the kinematical data.

The *Yangian* corresponds to those diffeomorphisms that simultaneously preserve the measures of *all* on-shell diagrams.

Canonical Coordinates and the Manifestation of the *Yangian*

All on-shell diagrams, in terms of canonical coordinates, take the form:

$$f = \int \frac{d\alpha_1}{\alpha_1} \wedge \cdots \wedge \frac{d\alpha_d}{\alpha_d} \; \delta^{k\times 4}\big(C(\vec{\alpha})\cdot\widetilde{\eta}\big)\delta^{k\times 2}\big(C(\vec{\alpha})\cdot\widetilde{\lambda}\big)\delta^{2\times(n-k)}\big(\lambda\cdot C(\vec{\alpha})^{\perp}\big)$$

Measure-preserving diffeomorphisms leave the function invariant, but—via the δ-functions—can be recast variations of the kinematical data.

The *Yangian* corresponds to those diffeomorphisms that simultaneously preserve the measures of *all* on-shell diagrams.