News from the FeynOnium project

Vladyslav Shtabovenko
Zhejiang University

in collaboration with
N. Brambilla, H. S. Chung and A. Vairo

The 13th International Workshop
on
Heavy Quarkonium
17th of May 2019, Torino, Italy

ZHEJIANG UNIVERSITY

Technische Universität München

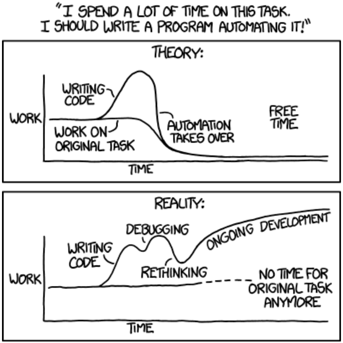
Outline

1. Motivation
2. Tree-level calculations with NREFTs
3. Loop-level calculations with NREFTs
4. Summary and Outlook

- FEYNCALC [Mertig et al., 1991; VS et al., 2016] is a package for semi-automatic symbolic QFT calculations.
- Available since 1990, my contributions started in 2014.
- Popular among EFT practitioners, especially in the heavy quarkonium community.
  - Familiar environment of MATHEMATICA.
  - Flat learning curve, usable documentation, many examples.
  - Friendly mailing list for questions & bug reports.
  - Easy to extend with own codes or interface with other tools.
- Yet, in my view, applying FEYNCALC to EFT calculations is still difficult.
  - Only Lorentz tensors: What about Cartesian?
  - Only standard $(p^2 - m^2 + i\eta)^{-1}$ propagators: Why not Euclidean or eikonal?
  - Manipulation of unusual integrals (abundant in NREFTs)?
  - Pauli algebra?

- Common solution: elaborated private codes.
- Often heavily tailored to a particular project.
- High probability to forget how things work in < 1 year.
- Been there, done that
  - EM Van der Waals forces between hydrogen atoms in pNRQED: A lot of Cartesian integrals to deal with [Brambilla, VS, Tarrús Castellà, Vairo, 2017].
  - Relativistic $\mathcal{O}(v^2)$ corrections to $e^+e^- \to \chi_{cJ}\,\gamma$: QCD-NRQCD matching via the threshold expansion method [Braaten & Chen, 1996] to extract Wilson coefficients of choromelectric operators $\Rightarrow$ Nonrelativistic expansions, tensor projections, Schouten's identities [Brambilla, Chen, Jia, VS, Vairo, 2017].
- Publication as the main goal, the codes are just by-products.
- Idea at the end of my PhD: Recycle and generalize the codes.
- Interesting side-project: (NR)EFTs made easy with FEYNCALC.

- Main focus on the strong sector: χPT [Weinberg, 1979; Gasser & Leutwyler, 1984], SCET [Bauer et al., 2000-2002; Beneke et al, 2002], NRQCD [Caswell & Lepage, 1986; Bodwin et al., 1995], pNRQCD [Pineda & Soto, 1998; Brambilla et al., 2000].
- Subdivided into 2 projects
  - FeynCalc 9.3: improved package with broad support for various EFTs.
  - FeynOnium: speicific extras for NREFTs (mainly NRQCD/pNRQCD).
- Reproduce key results from the literature to test the approach.
- As usual, initial time estimates too optimistic …



Source: xkcd.com/1319

- Many conceptual and technical issues to think over.
- Account for relations between Lorentz and Cartesian tensors.
- Every tensor that carries a Lorentz index can be decomposed into a temporal and a Cartesian piece

$$p^\mu = p_\nu g^{\mu\nu} = p_0 g^{\mu 0} + p_i g^{\mu i}.$$

- Distinction between upper (contravariant) and lower (covariant) indices

$$\boldsymbol{p}^i = -\boldsymbol{p}_i,$$

  metric signature dependent!
- Backward-compatibility: Extend the existing syntax but do not break it.
- Prevent over-engineering: Add only minimal amount of new symbols.
- Keep it simple: No new commands for standard operations (contractions, expansions etc.)

- Tree-level calculations with NR quantities: Mostly completed during 2016-2017, presented at QWG 2017

- Loop-level calculations with nonstandard integrals: somewhat harder than expected, ongoing work since 2017.

- Manipulations of Cartesian tensors (common in NREFTs) are easy.
- Calculating $\left(\delta^{ij} + \frac{\boldsymbol{n}^i \boldsymbol{k}^j + \boldsymbol{k}^i \boldsymbol{n}^j}{k \cdot n}\right) \boldsymbol{q}^i \boldsymbol{r}^j (\boldsymbol{q} \cdot \boldsymbol{r})$ amounts to

```
In[2]:= (KD[i, j] + (CV[k, j] CV[n, i] + CV[k, i] CV[n, j])/SP[k, n]) CV[q, i] CV[r, j] CSP[q, r] //
          Contract
```

$$\text{Out[2]=}\ (\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})^2 + \frac{(\overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{r}})(\overline{\boldsymbol{n}} \cdot \overline{\boldsymbol{q}})(\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})}{\overline{k} \cdot \overline{n}} + \frac{(\overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{q}})(\overline{\boldsymbol{n}} \cdot \overline{\boldsymbol{r}})(\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})}{\overline{k} \cdot \overline{n}}$$

```
In[3]:= % // LorentzToCartesian
```

$$\text{Out[3]=}\ (\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})^2 + \frac{(\overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{r}})(\overline{\boldsymbol{n}} \cdot \overline{\boldsymbol{q}})(\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})}{k^0 n^0 - \overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{n}}} + \frac{(\overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{q}})(\overline{\boldsymbol{n}} \cdot \overline{\boldsymbol{r}})(\overline{\boldsymbol{q}} \cdot \overline{\boldsymbol{r}})}{k^0 n^0 - \overline{\boldsymbol{k}} \cdot \overline{\boldsymbol{n}}}$$

- Lorentz (Cartesian) vectors in 4-dim (3-dim) have a bar.
- `KD[i,j]` $\sim \delta^{ij}$: Cartesian metric (Kronecker Delta).
- `CV[p,i]` $\sim \boldsymbol{p}^i$: Cartesian 3-vector.
- `SP[k,n]` $\sim k \cdot n$: Scalar product of two 4-vectors.
- `CSP[q,r]` $\sim \boldsymbol{q} \cdot \boldsymbol{r}$: Scalar product of two 3-vectors.
- `TC[k]` $\sim k^0$: Temporal component of a 4-vector.
- `Contract` handles the contractions of Lorentz and Cartesian indices.
- `LorentzToCartesian` for breaking the manifest Lorentz covariance.

- Differentiation w.r.t 3-vectors (projections, series expansions, differential equations) is straightforward.
- Let us calculate $\frac{\partial}{\partial \boldsymbol{q}^i} \frac{1}{(\boldsymbol{q}^2+m^2)(\boldsymbol{p}+\boldsymbol{q})^2}$

```
In[4]:= 1 / (CSPD[q] + m ^ 2)   1 / (CSPD[q + p])
```

Out[4]= $\frac{1}{(m^2 + \boldsymbol{q}^2)(\boldsymbol{p} + \boldsymbol{q})^2}$

```
In[5]:= ThreeDivergence[%, CVD[q, i]]
```

Out[5]= $-\frac{2\,\boldsymbol{p}^i}{(m^2 + \boldsymbol{q}^2)(2(\boldsymbol{p}\cdot\boldsymbol{q}) + \boldsymbol{p}^2 + \boldsymbol{q}^2)^2} - \frac{2\,\boldsymbol{q}^i}{(m^2 + \boldsymbol{q}^2)^2(2(\boldsymbol{p}\cdot\boldsymbol{q}) + \boldsymbol{p}^2 + \boldsymbol{q}^2)} - \frac{2\,\boldsymbol{q}^i}{(m^2 + \boldsymbol{q}^2)(2(\boldsymbol{p}\cdot\boldsymbol{q}) + \boldsymbol{p}^2 + \boldsymbol{q}^2)^2}$

- `CSP[q,r]` $\sim \boldsymbol{q}\cdot\boldsymbol{r}$: Scalar product of two 3-vectors in $D-1$ dimensions.
- `ThreeDivergence[f, CVD[q,i]]` $\sim \frac{\partial}{\partial \boldsymbol{q}^i} f$.
- In general, Cartesian tensors can live in 3, $D-1$ or $D-4$ dimensions.

- Dirac matrices may carry Cartesian or temporal indices.
- We can simplify chains of such matrices and compute traces as in

$$\mathrm{Tr}\left(\not{\boldsymbol{p}}\,\gamma^0 P_R\,\not{\boldsymbol{q}}\,P_L\not{\boldsymbol{\varepsilon}}(n)\right),$$

with $\not{\boldsymbol{p}} = \gamma^i \boldsymbol{p}^i$.

```
In[9]:= DiracTrace[CGS[p].TGA[].GA[6].CGS[q].GA[7].CGS[Polarization[n, I]]]
```

Out[9]= $\mathrm{tr}((\overline{\gamma}\cdot\overline{p}).\overline{\gamma}^0.\overline{\gamma}^6.(\overline{\gamma}\cdot\overline{q}).\overline{\gamma}^7.(\overline{\gamma}\cdot\overline{\varepsilon}(n)))$

```
In[10]:= % // DiracSimplify
```

Out[10]= $2\,i\,\bar{\epsilon}^{\overline{p}\,\overline{q}\,\overline{\varepsilon}(n)}$

- $\bar{\epsilon}^{\bar{\boldsymbol{p}}\bar{\boldsymbol{q}}\bar{\boldsymbol{\varepsilon}}(n)}$ stands for $\epsilon^{ijk}\boldsymbol{p}^i\boldsymbol{q}^j\boldsymbol{\varepsilon}^k(n)$.
- `CGS[p]` $\sim \gamma^i\boldsymbol{p}^i$: Dirac matrix contracted with a 3-vector.
- `GA[6/7]` $\sim \frac{1}{2}(1\pm\gamma_5)$: Chirality projectors.
- `TGA[]` $\sim \gamma^0$.
- `DiracSimplify` handles the Dirac algebra.

- Pauli matrices are indispensable in NR calculations, can handle that too.

```
In[10]:= PauliEta[-i].CSIS[q].CSIS[Polarization[k1, -I]].CSIS[k1].CSIS[q].
          PauliXi[i]
```

Out[10]=

$$\eta^{\dagger}.(\overline{\sigma}\cdot\overline{\boldsymbol{q}}).(\overline{\sigma}\cdot\overline{\varepsilon}^{*}(\mathrm{k1})).(\overline{\sigma}\cdot\overline{\mathbf{k1}}).(\overline{\sigma}\cdot\overline{\boldsymbol{q}}).\xi$$

```
In[11]:= PauliTrick[%, PauliReduce → False]
```

Out[11]=

$$2\,(\overline{\boldsymbol{q}}\cdot\overline{\varepsilon}^{*}(\mathrm{k1}))\,\eta^{\dagger}.(\overline{\sigma}\cdot\overline{\mathbf{k1}}).(\overline{\sigma}\cdot\overline{\boldsymbol{q}}).\xi-2\,(\overline{\mathbf{k1}}\cdot\overline{\boldsymbol{q}})\,\eta^{\dagger}.(\overline{\sigma}\cdot\overline{\varepsilon}^{*}(\mathrm{k1})).(\overline{\sigma}\cdot\overline{\boldsymbol{q}}).\xi+\overline{\boldsymbol{q}}^{2}\,\eta^{\dagger}.(\overline{\sigma}\cdot\overline{\varepsilon}^{*}(\mathrm{k1})).(\overline{\sigma}\cdot\overline{\mathbf{k1}}).\xi$$

- `CSIS[p]` $\sim \sigma^i \boldsymbol{p}^i$: Pauli matrix contracted to a 3-vector.
- `PauliEta`, `PauliXi` denote the Pauli 2-spinors.
- `PauliTrick` simplifies a chain of Pauli matrices.
- `PauliReduce` governs the application of

$$\sigma^i \sigma^j = \delta^{ij} + i\varepsilon^{ijk}\sigma^k.$$

Important for DR calculations, ε^{ijk} is tricky in $D-1$ dimensions.

- Validation is crucial for such software frameworks!
- FEYNCALC has a decent test-coverage with a large test-suite.
- Still, unit-tests cannot replace real-life calculations.
- Reproducing analytic results from the literature is very time-consuming.
- Many publications are very concise regarding the technical details.
- A selection of results reproduced using FEYNCALC+FEYNONIUM
  - $Q\bar{Q}({}^3P_0^{[1]}) \to gg$ at $\mathcal{O}(\alpha_s v^0)$ [Petrelli et al., 1998]
  - $\eta_c/\chi_{cJ} \to 2\gamma$ at $\mathcal{O}(\alpha_s^0 v^4)$ [Brambilla et al., 2006]
  - $\eta_c \to 2\gamma$ at $\mathcal{O}(\alpha_s v^2)$ [Jia et al., 2011]
  - $H \to J/\psi\,\gamma$ at $\mathcal{O}(\alpha_s^0 v^2)$ [Bodwin et al., 2014]
  - See also today's talk of Hee Sok Chung on $\mathcal{O}(\alpha_s^0 v^4)$ corrections to $H \to J/\psi\,\gamma$!
- No manipulations of nonstandard loop integrals were required here.

- What is so special about loop integrals in (NR)EFTs?
- For standard integrals at 1-loop we have the time-proven Passarino-Veltman (PaVe) [Passarino & Veltman, 1979] technique.
- Old method, but still convenient and easy to apply.
- Implemented in FeynCalc since version 1.0, subsequently improved in the newer versions.
- The FeynCalc 1-loop toolbox includes
  - `FDS`: Simplification using symmetries and loop momentum shifts.
  - `ApartFF`: Partial fractioning, based on [Feng, 2012].
  - `ToPaVe`: Conversion of scalar integrals to PaVe functions.
  - `PaVeUVPart` (new in 9.3): extract the UV part of arbitrary PaVe functions, based on [Sulyok, 2017].
  - `$KeepLogDivergentScalelessIntegrals`: switch to prevent $\left(\frac{1}{\varepsilon_{\rm UV}} - \frac{1}{\varepsilon_{\rm IR}}\right)$-type integrals from being set to zero.
  - `TID`: 1-loop tensor reduction (all-in-one).

- Nonstandard integrals (not $(p^2 - m^2)^{-1}$-type) are not that easy.
- Not all tensor integrals reducable to scalars with unit numerators (even at 1-loop)
- No universal basis like Passarino-Veltman.
- Master integrals usually evaluated on case-by-case basis.
- Need special care when simplifying the denominators, e. g. for eikonal

$$\int \frac{d^D q}{[q^2 + i\eta]\,[-2p \cdot q - m^2 + i\eta]} = -\int \frac{d^D q}{[q^2 + i\eta]\,[2p \cdot q + m^2 - i\eta]}$$

- Most useful manipulations still applicable to almost all integrals.
- Partial fractioning (linear independence of propagators)

$$\frac{4(\boldsymbol{p}\cdot\boldsymbol{q})}{\boldsymbol{q}^2(\boldsymbol{q}+p)^2(\boldsymbol{q}-p)^2} = \frac{1}{\boldsymbol{q}^2(\boldsymbol{q}-\boldsymbol{p})^2} + \frac{1}{\boldsymbol{q}^2(\boldsymbol{q}+\boldsymbol{p})^2}$$

- Tensor reduction to remove loop momenta with uncontracted indices

$$\int d^D q \frac{q^\mu q^\nu}{q^2\,(q\cdot p - m^2)} = \frac{m^4}{(D-1)p^4}(D p^\mu p^\nu - p^2 g^{\mu\nu}) \int \frac{d^D q}{q^2\,(q\cdot p - m^2)}$$

- Those are useful at 1-loop but also in multiloop calculations.
- Need a simple and consistent way to represent various (NR)EFT propagators in a symbolic code.

- Surprisingly, in FeynCalc only 3 new shortcuts are needed.
- In addition to the standard `FAD` (`FeynAmpDenominator`) introduce
  - `SFAD` (`StandardFeynAmpDenominator`): Covariant integrals.
  - `CFAD` (`CartesianFeynAmpDenominator`): Cartesian integrals.
  - `GFAD` (`GenericFeynAmpDenominator`): anything else.

Shortcut in FeynCalc	Meaning
$\texttt{FAD}[l - p_1 - \ldots,\ m]$	$\frac{1}{(l-p_1-\ldots)^2-m^2}$
$\texttt{SFAD}[l - p_1 - \ldots,\ \pm l.(q_1 + \ldots),\ \pm m^2,\ \pm 1,\ n]$	$\left[\frac{1}{(l-p_1-\ldots)^2 \pm l.(q_1+\ldots) \mp m^2 \pm i\eta}\right]^n$
$\texttt{CFAD}[l - p_1 - \ldots,\ \pm l.(q_1 + \ldots),\ \pm m^2,\ \pm 1,\ n]$	$\left[\frac{1}{(\boldsymbol{l}-\boldsymbol{p}_1-\ldots)^2 \pm \boldsymbol{l}.(\boldsymbol{q}_1+\ldots) \pm m^2 \pm i\eta}\right]^n$
$\texttt{GFAD}[x,\ \pm 1,\ n]$	$\left[\frac{1}{x \pm i\eta}\right]^n$

- Notice that the mass term and $i\eta$ can be positive or negative.

- Simplify the 1-loop heavy nucleon propagator in baryonic χPT [Ecker & Mojzis, 1996] (example from S. Scherer's lectures [Scherer, 2003])

$$\int \frac{d^D k}{(2\pi)^D} (-S_v \cdot k)\sigma^i \frac{1}{v \cdot (r-k) + i\eta} \frac{1}{k^2 - M^2 + i\eta} (S_v \cdot k)\sigma^i,$$

with $S_v^\mu = -\frac{1}{2}\gamma_5(\gamma^\mu \not{v} - v^\mu)$.

- Painful in FEYNCALC 9.2, trivial with version 9.3

```
In[40]:= exp = (-(-1/2 GA[5].(GAD[μ].GSD[v] - FVD[v, μ]).CSI[i] FVD[k, μ])).
    SFAD[{{0, v.(r - k)}}, {k, M^2}].(-1/2 GA[5].
      (GAD[ν].GSD[v] - FVD[v, ν]).CSI[i]) FVD[k, ν]
```

Out[40]=

$$k^\nu \left(\frac{1}{2} k^\mu \bar{\gamma}^5.(\gamma^\mu.(\gamma \cdot v) - v^\mu).\bar{\sigma}^i\right).\frac{1}{((r-k)\cdot v + i\eta).(k^2 - M^2 + i\eta)}.\left(-\frac{1}{2}\bar{\gamma}^5.(\gamma^\nu.(\gamma \cdot v) - v^\nu).\bar{\sigma}^i\right)$$

```
In[41]:= TID[PauliTrick[exp], k, FCE → True]
```

Out[41]=

$$\frac{3(r \cdot v)}{4(k^2 - M^2 + i\eta)} - \frac{3((r \cdot v)^2 - M^2 v^2)}{4(k^2 - M^2 + i\eta).((r-k)\cdot v + i\eta)}$$

- Tensor reduction of Cartesian integrals e. g.

$$\int \frac{d^{D-1}k}{(2\pi)^{D-1}} \frac{\boldsymbol{k}^i \boldsymbol{k}^j}{[\boldsymbol{k}^2 + m^2][(\boldsymbol{k}-\boldsymbol{q})^2 + m^2]}$$

- Very common in pNRQCD calculations if you prefer to handle spatial and temporal integrations separately.
- *Sort of* possible in FeynCalc 9.2, but not straightforward.
- Now it *just works*

```
In[44]:= exp = CVD[k, i] CVD[k, j] CFAD[{k, m^2}, {k - q, m^2}]
```

Out[44]=

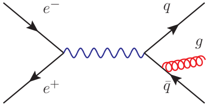
$$\frac{\boldsymbol{k}^i \boldsymbol{k}^j}{(\boldsymbol{k}^2+m^2-i\eta).((\boldsymbol{k}-\boldsymbol{q})^2+m^2-i\eta)}$$

```
In[47]:= TID[exp, k]
```

Out[47]=

$$\frac{-(D-1)\,\boldsymbol{q}^i\,\boldsymbol{q}^j - \boldsymbol{q}^2\,\delta^{ij} + 2\,\boldsymbol{q}^i\,\boldsymbol{q}^j}{2\,(2-D)\,\boldsymbol{q}^2\,(\boldsymbol{k}^2+m^2-i\eta)} - \frac{(D-1)\,\boldsymbol{q}^2\,\boldsymbol{q}^i\,\boldsymbol{q}^j - 4\,m^2\,\boldsymbol{q}^2\,\delta^{ij} + 4\,m^2\,\boldsymbol{q}^i\,\boldsymbol{q}^j - \boldsymbol{q}^4\,\delta^{ij}}{4\,(2-D)\,\boldsymbol{q}^2\,(\boldsymbol{k}^2+m^2-i\eta).((\boldsymbol{k}-\boldsymbol{q})^2+m^2-i\eta)}$$

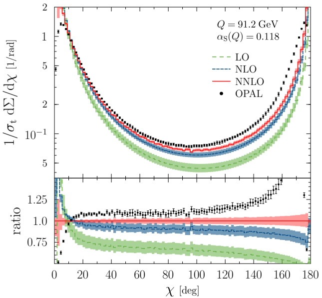
- My favorite new propagator type: `GFAD`, the generic propagator
- `GFAD` can be anything, so it works with *highly* nonstandard propagators
- The idea for this came from my work on the Energy-Energy-Correlation (EEC) event shape variable in QCD.



- EEC [Basham et al., 1978] is a classical (IR safe) hadronic observable in e^+e^- annihilation: $e^+e^- \to \gamma^*/Z^0 \to a + b + X$.
- Formal definition ($\cos\theta_{ab} = \hat{\boldsymbol{p}}_a \cdot \hat{\boldsymbol{p}}_b$)

$$\frac{1}{\sigma_{\text{tot}}} \frac{d\Sigma(\chi)}{d\cos\chi} = \sum_{a,b} \int \underbrace{\frac{E_a E_b}{Q^2} \delta(\cos\theta_{ab} - \cos\chi)}_{\text{measurement function}} \times d\sigma_{a+b+X}.$$

- 2 calorimeters at relative angle χ measure the energies of the hadrons a and b.



Source: [Tulipánt et al., 2017].

- Diff. angular distribution from measuring the energies (E_a, E_b) of and the angle θ_{ab} between all hadrons pairs produced in each event.

- Numerically, EEC is currently known at NNLO [Del Duca et al., 2016; Tulipánt et al., 2017].
- Analytically (fixed-order), the LO result was state of the art for 40 years [Basham et al., 1978]

$$\frac{1}{\sigma_{\text{tot}}}\frac{d\Sigma(\chi)}{d\cos\chi} = \frac{\alpha_s(\mu)}{2\pi} C_F \frac{3-2z}{4(1-z)z^5}\left[3z(2-3z)+2(2z^2-6z+3)\log(1-z)\right] + \mathcal{O}(\alpha_s^2), \quad \text{with } z=(1-\cos\chi)/2.$$

- With my collaborators we obtained fixed-order analytic NLO results for
  - $e^+e^- \to \gamma^* \to \text{partons}$ in pQCD [Dixon, Luo, VS, Yang, Zhu, 2018]
  - $H \to \text{gluons} \to \text{partons}$ in HEFT [Luo, VS, Yang, Zhu, 2019]
- "NLO" in this context: by calculating 3-loop phase space integrals with special propagators from the measurement function.
- Done using familiar multiloop tools (FORM [Vermaseren, 2000], Color [van Ritbergen et al., 1999], FIRE 5 [Smirnov, 2015], LiteRed [Lee, 2012], Fuchsia [Gituliar & Magerya, 2017])
- In the HEFT calculation FeynCalc was employed at some intermediate stages.

- EEC: $\frac{1}{\sigma_{\text{tot}}} \frac{d\Sigma(\chi)}{d \cos \chi} = \sum_{a,b} \int \frac{E_a E_b}{Q^2} \delta(\cos \theta_{ab} - \cos \chi) d\sigma_{a+b+X}$.
- Measurement function in the rest frame, setting $Q^2 = 1$

$$E_a E_b \, \delta(\cos \theta_{ab} - \cos \chi) = (p_a \cdot Q)^2 (p_b \cdot Q)^2 \delta \left(2z \, p_a \cdot Q \, p_b \cdot Q - p_a \cdot p_b\right).$$

- Analytic calculation of phase-space integrals using the reverse unitarity method [Anastasiou & Melnikov, 2002; Anastasiou et al., 2003]

$$\delta \left(2z \, p_a \cdot Q \, p_b \cdot Q - p_a \cdot p_b\right) \sim \frac{1}{2z \, p_a \cdot Q \, p_b \cdot Q - p_a \cdot p_b}\bigg|_{\text{cut}}$$

- $[2z \, p_a \cdot Q \, p_b \cdot Q - p_a \cdot p_b]^{-1}$ is nonlinear in the scalar products.
- Nonstandard propagator that is not supported by public codes.
- Initially we used “hacked” LiteRed to derive IBPs [Chetyrkin & Tkachov, 1981] and differential equations [Kotikov, 1991a, 1991b, 1991c; Bern et al., 1994; Remiddi, 1997; Gehrmann & Remiddi, 2000].
- Later we discovered interesting tricks to optimize the IBP equations and simplify the integrands.
- For “experiments” I needed a more convenient software framework.

- Using GFADs it is easy to put the nonlinear propagators into FEYNCALC
- Example: calculate

$$\int d^D p_1 d^D p_2 \frac{\partial}{\partial p_1^\mu} \frac{p_1^\mu (p_2 \cdot Q)}{p_1^2 p_2^2 (Q - p_1 - p_2)^2 (Q - p_1)^2 (Q - p_2)^2} \frac{1}{2z\, p_a \cdot Q\, p_b \cdot Q - p_a \cdot p_b}$$

```
In[72]:= SFAD[p1, p2, Q - p1 - p2, Q - p1, Q - p2] GFAD[2 z SPD[p1, Q] SPD[p2, q] - SPD[p1, p2]]
```

Out[72]=

$$\frac{1}{(\mathrm{p1}^2+i\eta).(\mathrm{p2}^2+i\eta).((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta).((Q-\mathrm{p1})^2+i\eta).((Q-\mathrm{p2})^2+i\eta)\,(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta)}$$

```
In[73]:= FourDivergence[FVD[p1, μ] SPD[p2, Q] %, FVD[p1, μ]]
```

Out[73]=

$$\frac{D\,(\mathrm{p2}\cdot Q)}{((Q-\mathrm{p2})^2+i\eta).((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta).(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta).(\mathrm{p1}^2+i\eta).(\mathrm{p2}^2+i\eta).((Q-\mathrm{p1})^2+i\eta)} -$$

$$\frac{2\,\mathrm{p1}^2\,(\mathrm{p2}\cdot Q)}{(\mathrm{p1}^2+i\eta)^2.((Q-\mathrm{p1})^2+i\eta).((Q-\mathrm{p2})^2+i\eta).((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta).(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta).(\mathrm{p2}^2+i\eta)} +$$

$$\frac{2(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot Q)-2\,\mathrm{p1}^2(\mathrm{p2}\cdot Q)}{((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta).(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta).(\mathrm{p1}^2+i\eta).(\mathrm{p2}^2+i\eta).((Q-\mathrm{p1})^2+i\eta)^2.((Q-\mathrm{p2})^2+i\eta)} +$$

$$\frac{-2\,\mathrm{p1}^2(\mathrm{p2}\cdot Q)-2(\mathrm{p1}\cdot\mathrm{p2})(\mathrm{p2}\cdot Q)+2(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot Q)}{((Q-\mathrm{p2})^2+i\eta).(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta).(\mathrm{p1}^2+i\eta).(\mathrm{p2}^2+i\eta).((Q-\mathrm{p1})^2+i\eta).((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta)^2} +$$

$$\frac{(\mathrm{p1}\cdot\mathrm{p2})(\mathrm{p2}\cdot Q)-2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)(\mathrm{p2}\cdot Q)}{((-\mathrm{p1}-\mathrm{p2}+Q)^2+i\eta).(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot q)-\mathrm{p1}\cdot\mathrm{p2}+i\eta)^2.(\mathrm{p1}^2+i\eta).(\mathrm{p2}^2+i\eta).((Q-\mathrm{p1})^2+i\eta).((Q-\mathrm{p2})^2+i\eta)}$$

- In FeynCalc 9.3 propagators can *in principle* have symbolic powers.
- With little effort one can write a custom function for deriving IBPs for the given topology à la LiteRed

```
In[29]:= SPD[Q] = 1;

In[30]:= topo = FCTopology[1,
  {SFAD[p1], SFAD[p2], SFAD[-p1 - p2 + Q], SFAD[-p2 + Q], SFAD[-p1 + Q],
    GFAD[-SPD[p1, p2] + 2 z SPD[p1, Q] SPD[p2, Q]]}]
```

Out[30]=

$$\text{FCTopology}\left(1, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)}, \frac{1}{((Q-\text{p2})^2+i\eta)}, \frac{1}{((Q-\text{p1})^2+i\eta)}, \frac{1}{(2z(\text{p1}\cdot Q)(\text{p2}\cdot Q)-\text{p1}\cdot\text{p2}+i\eta)}\right\}\right)$$

```
In[33]:= FCCreateIBPEquationsEEC[topo, {p1, p2}][[1 ;; 3]]
```

Out[33]=

$$\{-a(3)\,Y(3)\,\text{Ym}(1)-a(5)\,Y(5)\,\text{Ym}(1)+a(3)\,Y(3)\,\text{Ym}(4)+a(5)\,Y(5)-2\,a(1)-a(3)-a(5)-a(6)+D,$$

$$-\frac{1}{2}a(6)\,Y(6)\,\text{Ym}(1)^2 z-a(6)\,Y(6)\,\text{Ym}(1)(z-1)+a(6)\,Y(6)\,\text{Ym}(5)\,\text{Ym}(1)\,z-\frac{1}{2}a(6)\,Y(6)\,\text{Ym}(5)^2 z+ a(6)\,Y(6)\,\text{Ym}(5)\,z-a(3)\,Y(3)\,\text{Ym}(1)+a(4)\,Y(4)\,\text{Ym}(1)-a(2)\,Y(2)\,\text{Ym}(3)-a(4)\,Y(4)\,\text{Ym}(3)+ a(2)\,Y(2)\,\text{Ym}(4)+a(3)\,Y(3)\,\text{Ym}(4)+a(2)\,Y(2)\,\text{Ym}(5)-\frac{1}{2}a(6)\,Y(6)\,z-a(2)\,Y(2)-a(3)+a(4),$$

$$-\frac{1}{2}a(6)\,Y(6)\,\text{Ym}(2)^2 z-a(6)\,Y(6)\,\text{Ym}(2)(z-1)+a(6)\,Y(6)\,\text{Ym}(4)\,\text{Ym}(2)\,z-\frac{1}{2}a(6)\,Y(6)\,\text{Ym}(4)^2 z+ a(6)\,Y(6)\,\text{Ym}(4)\,z-a(3)\,Y(3)\,\text{Ym}(2)+a(5)\,Y(5)\,\text{Ym}(2)-a(1)\,Y(1)\,\text{Ym}(3)-a(5)\,Y(5)\,\text{Ym}(3)+ a(1)\,Y(1)\,\text{Ym}(4)+a(1)\,Y(1)\,\text{Ym}(5)+a(3)\,Y(3)\,\text{Ym}(5)-\frac{1}{2}a(6)\,Y(6)\,z-a(1)\,Y(1)-a(3)+a(5)\}$$

- `Y[i]` (`Ym[i]`) increase (decrease) the power of the i^{th} propagator
- `a[i]` denotes the original power of the i^{th} propagator.
- The output can be directly put into FIRE for the IBP reduction.

- The packages are *publicly* (GNU GPLv3) available on GitHub (https://github.com/FeynCalc)
- Works on Mathematica starting with version 8.0.
- If you are brave, you can try out the development version https://github.com/FeynCalc/feyncalc/wiki/Installation
- ...or just wait until the preprints appear on the arXiv.
- There will be 2 publications related to this project very soon
  - FeynCalc 9.3 (for the broad audience): coding finished, documentation updated, writing everything up. Huge changelog: https://github.com/FeynCalc/feyncalc/blob/master/FeynCalc/Changelog.md
  - FeynOnium 1.0 (for the NREFT community): employed in several ongoing projects, already usable but needs a bit of polishing.

Summary

- Our long-term (2016-2019) project to develop a toolbox for (NR)EFT calculations is now in its final stage.
- Publicly available code to handle nonrelativistic expressions and a wide range of nonstandard integrals.
- Not a fully automatic all-in-one solution, but a handy tool for knowledgeable people.
- The focus is on tree and 1-loop level.
- Validation by reproducing existing results from the literature and obtaining new predictions.
- Good progress, given the side-project nature of this work.

Outlook

- Currently no user feedback, expect a sharp rise after the publications.
- Many things that could be implemented or improved, unfortunately only finite amount of time.