WP1

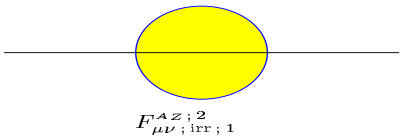
Giampiero PASSARINO

Dipartimento di Fisica Teorica, Università di Torino, Italy

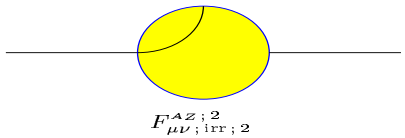
INFN, Sezione di Torino, Italy

February 15 2012, Torino

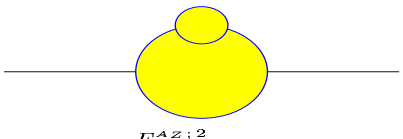
$G^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}} =$



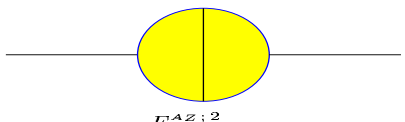
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,1}$



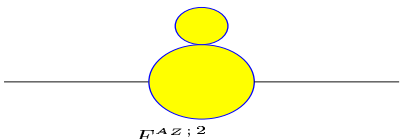
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,2}$



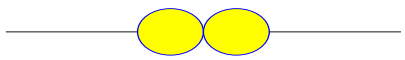
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,3}$



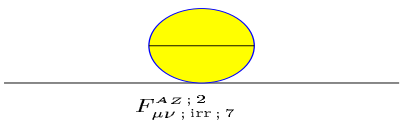
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,4}$



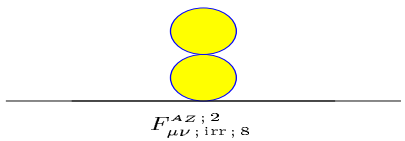
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,5}$



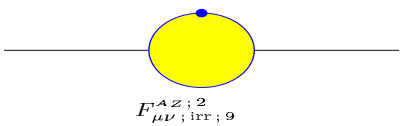
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,6}$



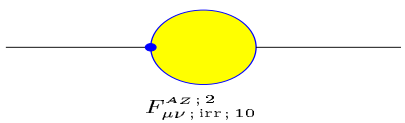
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,7}$



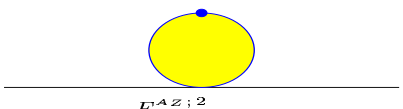
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,8}$



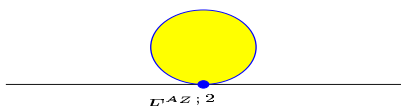
$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,9}$



$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,10}$



$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,11}$



$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,12}$

$F^{A\,Z\,;\,2}_{\mu\nu\,;\,\mathrm{irr}\,;\,13}$

Algebraic explosion

- For example, in $gg \to N$ gluons we need to compute:

N	diagrams
2	4
4	220
6	34,300
8	10,525,900

- Feynman rules in gauge theory

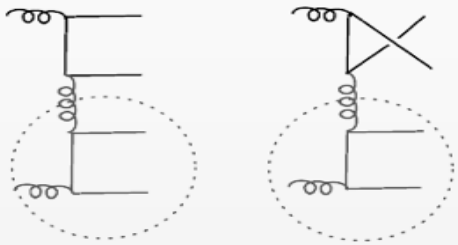
$$\mathcal{V}_{ggg} = f^{abc}\left[g_{\mu_1\mu_2}(p_1-p_2)^{\mu_3} + g_{\mu_2\mu_3}(p_2-p_3)^{\mu_1} + g_{\mu_3\mu_1}(p_3-p_1)^{\mu_2}\right]$$

- Algebra of γ matrices, colour algrebra, etc.

$$\begin{aligned} Tr(\gamma^{\mu_1}\gamma^{\mu_2}) &= \text{1 term} \\ Tr(\gamma^{\mu_1}\cdots\gamma^{\mu_8}) &= \text{105 terms} \\ Tr(\gamma^{\mu_1}\cdots\gamma^{\mu_{14}}) &= \text{26,931 terms} \end{aligned}$$

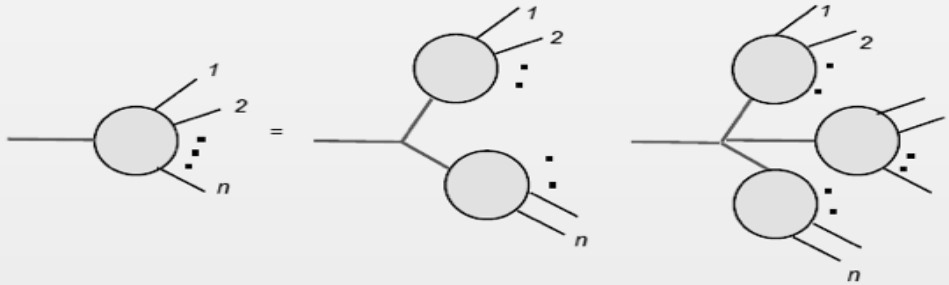
Recursion at tree-level

- Feynman diagrams contain sub-parts which we compute over and over.



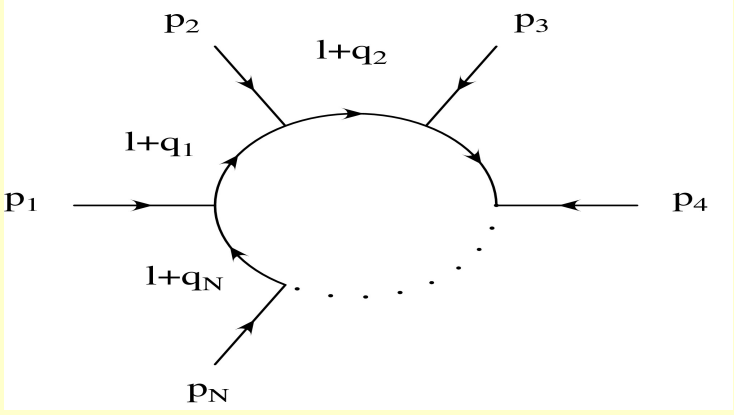
- It is possible to organize the evaluation of tree amplitudes recursively

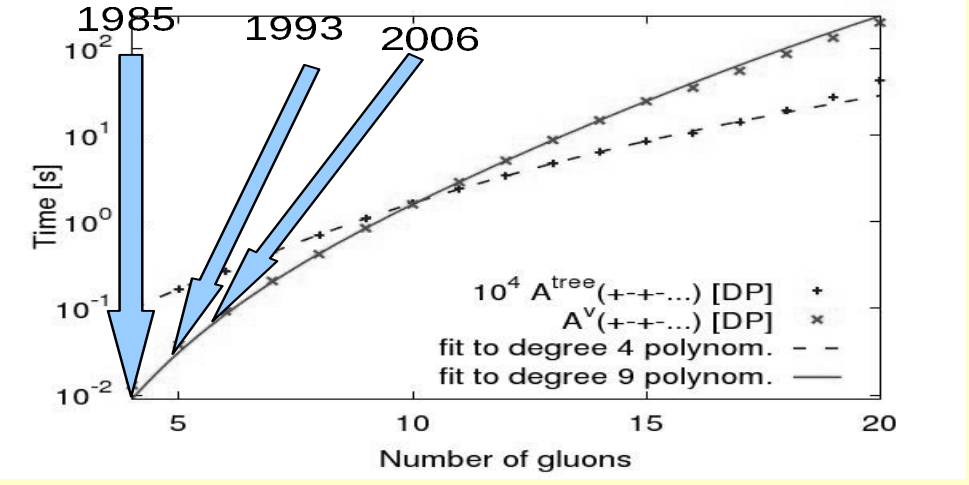
e.g. Berends, Giele



The power of unitarity: N-gluon amplitudes

$$20! \approx 2.4 \times 10^{18}$$ ⟺ 100 years of calculating





Giele, Zanderighi

N-gluon amplitudes can be calculated for arbitrary N. Explicit numerical results available for N through 20. Factorial growth in the number of Feynman diagrams makes this computation impossible with traditional methods.

Some words in the beginning

- Fast tree-level event generators are needed for multi-particle final states.
  - $\Rightarrow$ evaluation time for event generation is crucial as one needs to average over many events to obtain good statistics for cross sections and observables
  - $\Rightarrow$ not only @ LO, also @ NLO to calculate real-emission corrections ...
  - $\Rightarrow$ ... and tree-level matrix elements when using generalized unitarity-cut methods to determine the virtual corrections
- Throw large computer farms/grids at the problem.
  - expensive; require certain infrastructure and maintenance
  - What if problem could be handled on a single, affordable PC ?
- Graphical Processor Units (GPUs) in addition to CPUs give an option. Explore capabilities.
  - $\Rightarrow$ first applications within the framework of HELAS ME generator [HAGIWARA, KANZAKI, OKAMURA, RAINWATER, STELZER, ARXIV:0909.5257, ARXIV:0908.4403]
  - $\Rightarrow$ can tame but not overcome factorial scaling of Feynman diagrammatic approach
- Define the project: LO LC n-gluon scattering cross sections. $\Rightarrow$ Tools needed ...
  - unit-weight phase-space generator ... implementation of RAMBO [KLEISS, STIRLING, ELLIS]
  - strong-coupling evaluation, PDFs using LHAPDF and observables
  - $gg \longrightarrow 2, \ldots, 10$ g MEs ... Berends–Giele ordered recursions, use threading to tame n^4-scaling

GPU hardware and programming principles

[Giele, Stavenga, Winter, arXiv:1002.3446]

- The C1060 Nvidia Tesla GPU is a plug-in card for your desktop. GPU has its own memory.
- The Tesla chip is designed for numerical applications and programmable in C/limited C++.
- The chip has 30 multi-processors (MPs), each comes with 1024 processors (threads).
  Each thread has an unique number (for I/O etc.).
  Threads essentially execute same processor instructions over different data
  (... can skip ahead and wait for others to catch up).
- Desirable: trivial parallelization (Monte Carlo algorithms: 1 event per thread).
  So, in principle we can run 30720 MC generators in parallel, each running N events
  ... a speed-up of 30000 !!
- Approach limited by amount of available fast-access memory.
  - off-chip slow-access memory: 4 Gb; use for I/O only ... transfer to and bin results on CPU
  - on-chip fast-access memory: only kbs; registers and shared memory

    $\Rightarrow$ 16384 32-bit registers per MP; once assigned only seen by specific thread; temporary storage for function evaluations

    $\Rightarrow$ 16384 bytes shared memory per MP; accessible to all threads

Memory layout

[Giele, Stavenga, Winter, arXiv:1002.3446]

- The n-gluon recursion relation needs n momenta and $n(n-1)/2$ currents for a total of $n(n+1)/2$ single precision 4-vectors.
  - Recursion relations are very suitable for GPU (memory efficient & algorithmically simple).
- We need $(4 \cdot 4)\ n(n+1)/2$ bytes of fast accessible memory per event.
- This means $16384/(8\ n(n+1))$ events per MP.
- One constraint though: implementation needs 35 registers per thread, i.e. 16384/35=468 threads are useable per MP.

n	4	5	6	7	8	9	10	11	12
events per MP	102	68	48	36	28	22	18	15	13
avail. threads / evt	10	15	21	28	36	45	55	66	78
useable threads / evt	4	6	9	13	16	21	26	31	36

$\Rightarrow$ used threads per event $= n - 1$.

Timing

[Giele, Stavenga, Winter, arXiv:1002.3446]

- Compare evaluation time per event on GPU with that of running the same algorithm on CPU [AMD Phenom(tm) II X4 940 (3 GHz)]. $P_n(m) = [(n-1)/n]\ \sqrt[m]{T_n/T_{n-1}}$
  - Speed-ups occur because events are evaluated in parallel.

n	T_n^{GPU} (seconds)	$P_n(3)$	T_n^{CPU} (seconds)	$P_n(4)$	G_n
4	2.975×10^{-8}		8.753×10^{-6}		294
5	4.438×10^{-8}	0.91	1.247×10^{-5}	0.87	281
6	8.551×10^{-8}	1.03	1.966×10^{-5}	0.93	230
7	2.304×10^{-7}	1.19	3.047×10^{-5}	0.96	132
8	3.546×10^{-7}	1.01	4.736×10^{-5}	0.98	133
9	4.274×10^{-7}	0.94	7.263×10^{-5}	0.99	170
10	6.817×10^{-7}	1.05	1.044×10^{-4}	0.99	153
11	9.750×10^{-7}	1.02	1.529×10^{-4}	1.00	157
12	1.356×10^{-6}	1.02	2.129×10^{-4}	1.00	158

$\Rightarrow$ Gain.

Summary

- We obtained encouraging results in our first exploration of the potential of using multi-threaded GPU based workstations for Monte Carlo programming.
- Testbed chosen: leading-order leading colour n-gluon scattering.
  $\Rightarrow$ **Tess** Monte Carlo program.
- Wrt the CPU based implementation, we found speed-ups ranging from $\mathcal{O}(300)$ and $\mathcal{O}(150)$ for 4-gluon and 12-gluon scattering, respectively.
- Outlook: @ LO – include quarks, vector bosons, subleading colour contributions, replace Rambo by Sarge. [Van Hameren, Papadopoulos]
  $\Rightarrow$ application to NLO
- GPU chips are still evolving rapidly ... next generation, Fermi chip (Fall 2010).