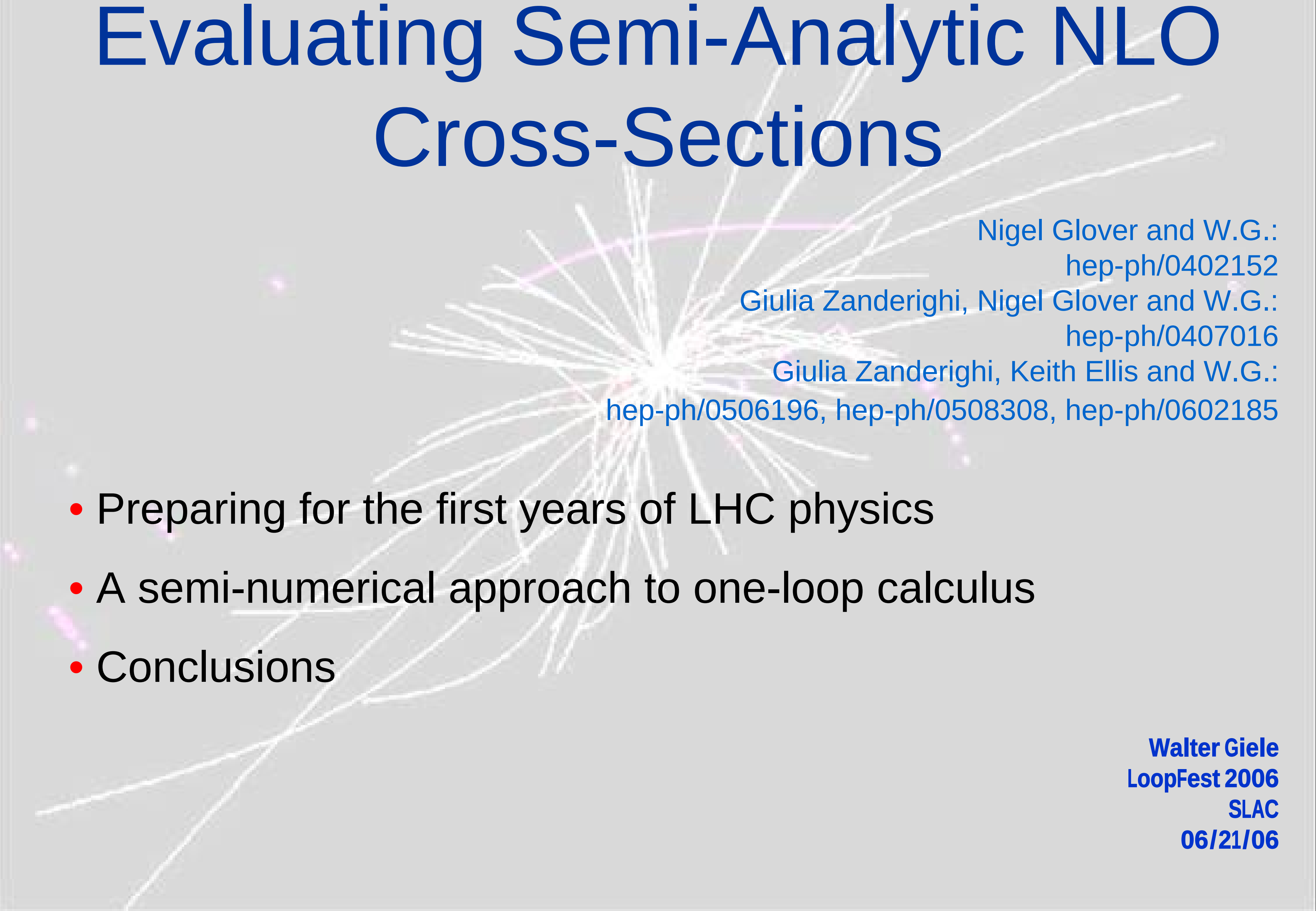
Evaluating Semi-Analytic NLO Cross-Sections



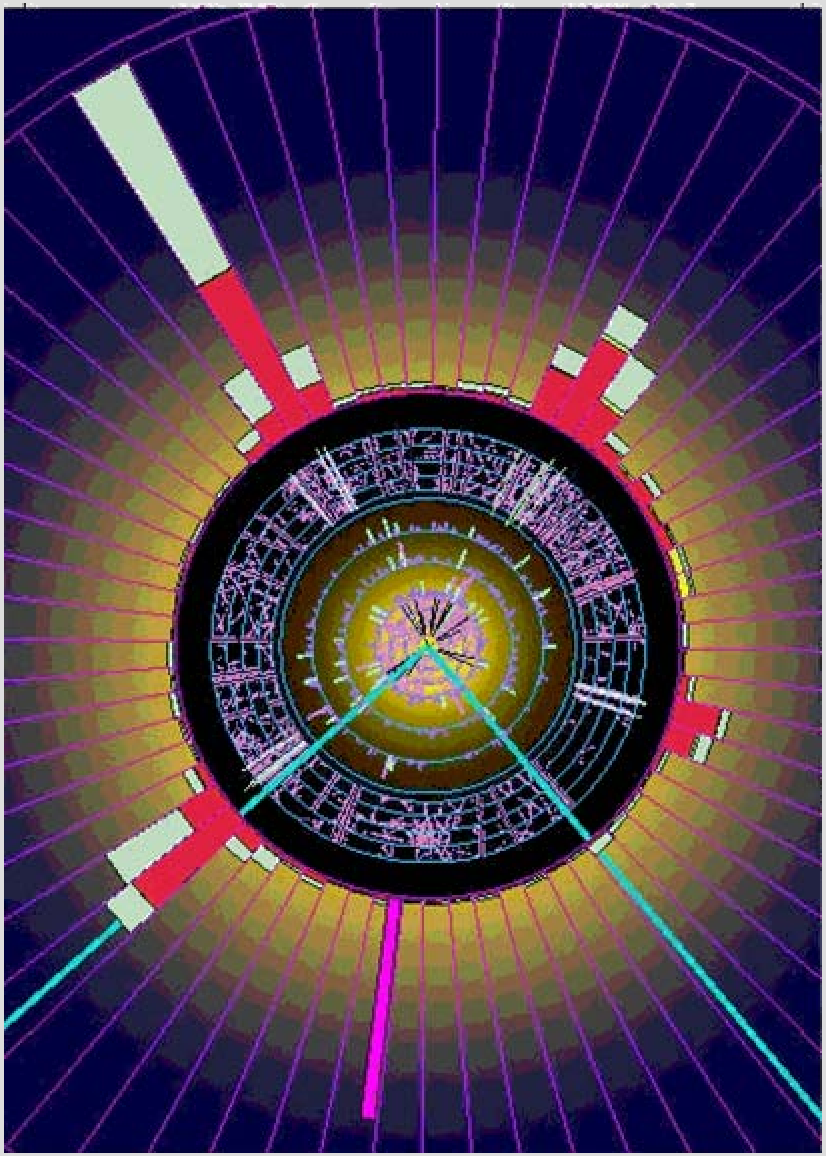
Nigel Glover and W.G.:
hep-ph/0402152
Giulia Zanderighi, Nigel Glover and W.G.:
hep-ph/0407016
Giulia Zanderighi, Keith Ellis and W.G.:
hep-ph/0506196, hep-ph/0508308, hep-ph/0602185

- Preparing for the first years of LHC physics
- A semi-numerical approach to one-loop calculus
- Conclusions

Walter Giele
LoopFest 2006
SLAC
06/21/06

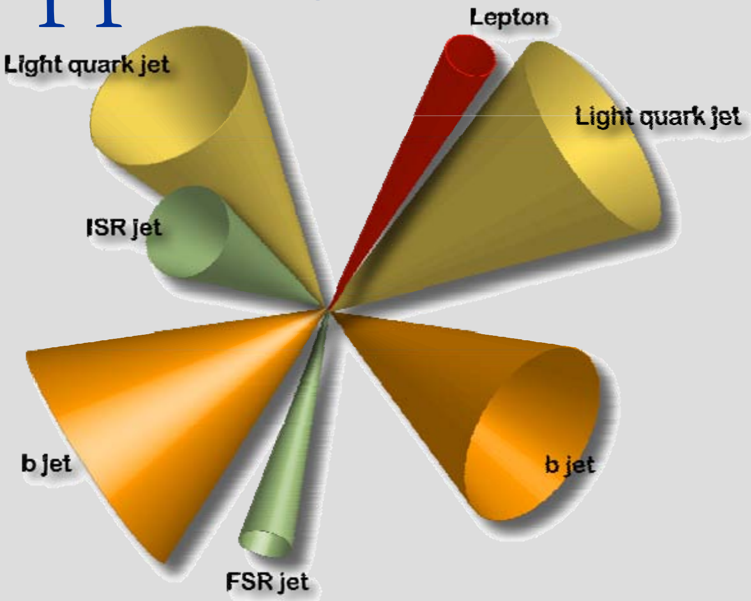
Preparing for the LHC era

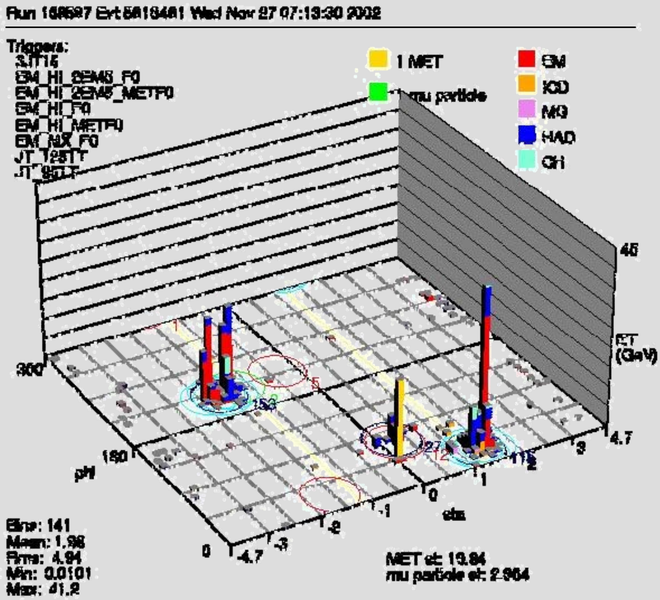
- At the start of run I at the Tevatron the outstanding issue was the top search.
  - Initiated many developments in LO multi-parton generation for *PP → W+jets* (numerical recursion and algebraic generation of tree level amplitudes)
  - An unexpected challenge in the top discovery was the importance of matching issues between matrix elements and shower monte carlo's.
- Detailed QCD studies at CDF/DØ initiated development of "numerical" partonic NLO jet MC's (e.g. EKS, JETRAD).
- In the coming years all new challenges for NLO are encapsulated by Higgs searches at CDF/DØ and ATLAS/CMS.



A semi-numerical approach

- The proof of any method, especially numerical methods, is actually applying the method to perform new calculations.
- We will need a minimum numerical accuracy (better than $O\left(10^{-8}\right)$).
- For one-loop amplitudes in NLO calculations time is not that important:
  - → We can generate 1,000,000 one-loop events, calculate the amplitude and store them in a file for use in a NLO parton MC.
  - → As a benchmark one could take 1 minute/event. This gives in one week on a 100 processor farm/grid 100x60x24x7=1,008,000 events/week





A semi-numerical approach

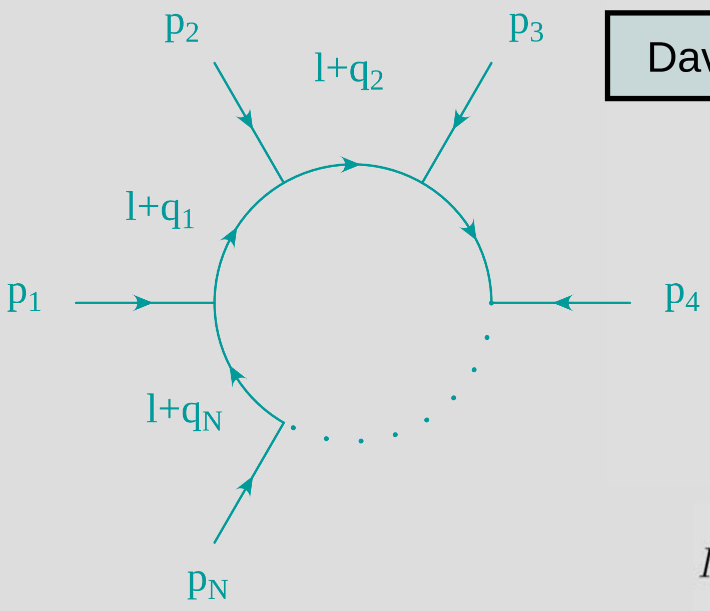
- These methods decompose a NLO scattering amplitude by numerically evaluating the (D-dimensional algebraic) coefficients of the master integrals (the D=4-2e self energy, triangle and box integrals)
- The master integrals are evaluated as analytic formula.
- Instead of extracting the singular terms analytical (by subtraction) we can simply treat all “numbers” as Laurent series instead of complex numbers in the computer code (and code up multiplications, divisions… between Laurent series.
  (van Hameren, Vollinga, Weinzierl)

$$L = \frac{c_{-2}}{\varepsilon^2} + \frac{c_{-1}}{\varepsilon} + c_0 + c_1\varepsilon + c_2\varepsilon^2 + O\left(\varepsilon^3\right)$$

This is an important step towards automatization

A semi-numerical approach

- We numerically implemented a definite algorithmic solution (based on the integration-by-parts method) to calculate one-loop tensor integrals semi-numerical:
  (Chetyrkin, F. V. Tkachov; Tarasov; T. Binoth, J. P. Guillet, G. Heinrich; G. Duplancic, B. Nizic)



Davydychev

$$
\begin{aligned}
I^{\mu_1\mu_2}(q_1,q_2,\ldots,q_N) &= -\frac{1}{2}I(D+2;\{1\}_{k=1}^{N})\; g^{\mu_1\mu_2} \\
&\quad + \sum_{P(\mu_1,\mu_2)}^{2!}\sum_{i_1\le i_2}^{N} I(D+4;\{1+\delta_{i_1k}+\delta_{i_2k}\}_{k=1}^{N})\; q_{i_1}^{\mu_{i_1}} q_{i_2}^{\mu_{i_2}} \\
&= -\frac{1}{2}I(D+2;1,1,1,\ldots,1)\; g^{\mu_1\mu_2} \\
&\quad +2I(D+4;3,1,1,\ldots,1)\; q_1^{\mu_1}q_1^{\mu_2} \\
&\quad +\; I(D+4;2,2,1,\ldots,1)\; (q_1^{\mu_1}q_2^{\mu_2}+q_1^{\mu_2}q_2^{\mu_1}) \\
&\quad +\cdots
\end{aligned}
$$

$$
I(D;\nu_1,\nu_2,\ldots,\nu_N) = \int d^D l\, \frac{1}{d_1^{\nu_1}d_2^{\nu_2}\cdots d_N^{\nu_N}}
$$

Unfortunately no analytic expression for this integral

The generalized scalar integral coefficients need to be evaluated semi-numerically.

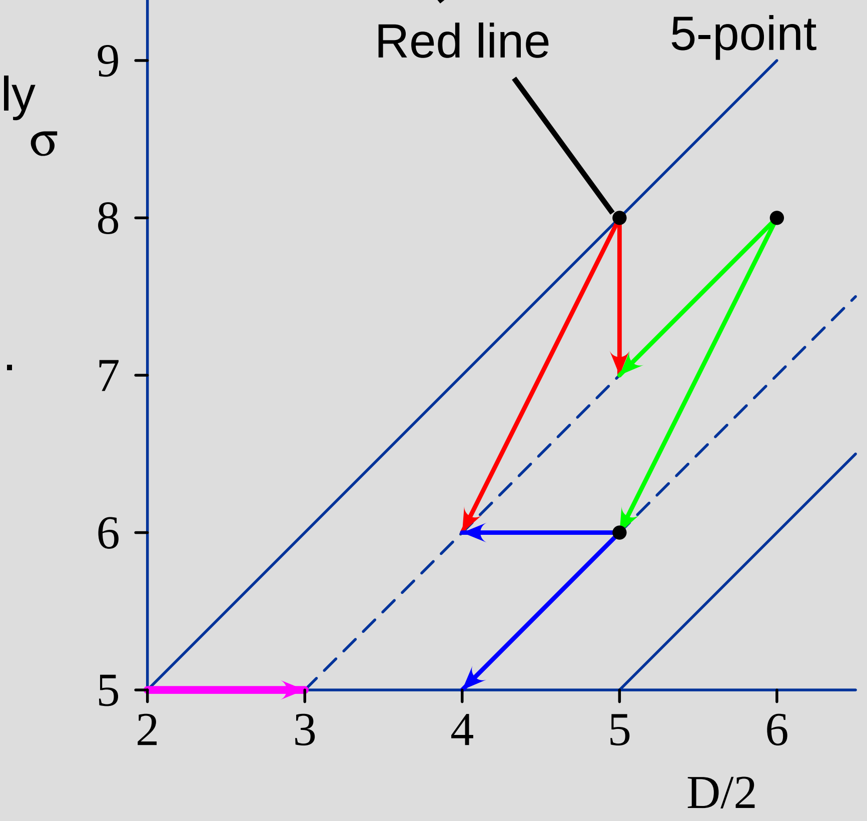
A semi-numerical approach

(See also talk Gudrun Heinrich)

$$(\nu_l - 1)I(D;\{\nu_k\}_{k=1}^{N})$$
$$= -\sum_{i=1}^{N} S_{li}^{-1} I(D-2;\{\nu_k - \delta_{ik} - \delta_{lk}\}_{k=1}^{N}) - b_l\,(D-\sigma)\,I(D;\{\nu_k - \delta_{lk}\}_{k=1}^{N})$$

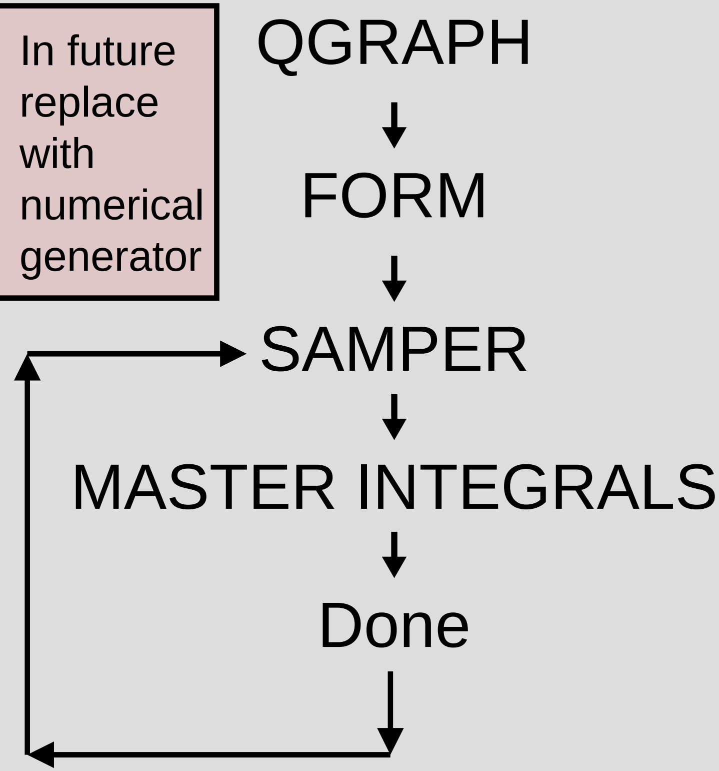
$$\sigma \equiv \sum_{i=1}^{N} \nu_i; \quad b_i \equiv \sum_{j=1}^{N} S_{ij}^{-1}; \quad B \equiv \sum_{i=1}^{N} b_i = \sum_{i,j=1}^{N} S_{ij}^{-1} \quad S_{ij} \equiv (q_i - q_j)^2$$

- The generalized scalar integrals are recursively reduced in the numerical program to scalar 2-, 3- and 4-point integrals (in 4 dimensions).
- Each step in the recursion involves a matrix inversion (or other manipulations of the matrix).
- There are potential numerical instabilities associated with the matrix inversion.
- This is a purely algebraic procedure. Masses (real or complex) can be trivially included.



A semi-numerical approach

- Once a program is constructed to evaluate tensor one-loop integrals it is straightforward to calculate the one-loop amplitudes:



In future replace with numerical generator

QGRAPH

FORM

SAMPER

MASTER INTEGRALS

Done

$$M = \sum_k m_k$$

$$M = \sum_k K_{\mu_1 \cdots \mu_k}(p,\psi) \times I^{\mu_1 \cdots \mu_k}(p)$$

$$M = \sum_{n=2}^{4} C_n(p,\psi) \times I_n(p)$$

The coefficient C_n is a numerically evaluated Laurent series of complex numbers.

The master integrals $I_n(p)$ are known Laurent series.

A semi-numerical approach

- With this method we calculated the one-loop *H+4 partons* (through gluon fusion)
  - → Accuracy equal or better than $O\left(10^{-12}\right)$ for
    - gauge invariance
    - singular term parts (proportional to Born).
    - comparison with analytically calculated *H+4 quarks*.
  - → No evaluation speed issues, however method can be speed up significantly
    - "unrolling" recursion (i.e. hardcode some of the simpler generalized scalar integrals $I_N\left(D;\nu_1,\ldots,\nu_N\right)$). E.g. "hardcode" all higher dimensional 4-points, etc…
- It is now straightforward to start constructing the still missing 2→3 parton level Monte Carlo's
- Numerical implementation of *PP→H+2 jets* almost done (Campbell, Ellis, Zanderighi).

A semi-numerical approach

- To see what is needed for a one-loop *2→4* we have done a feasibility study by looking at the calculation of *2g→4g* by “pure” semi-numerical methods:
  - → Can we use the brute force QGRAPH→FORM method to generate the amplitude Feynman diagram by Feynman diagram? (~8,000+300 feynman graphs; rank 6 6-point tensor integrals with 6 3-gluon vertices;…)
  - → Or do we need more sophisticated “factorized” generation using multi-particle sources attached to loops and more numerical implementation of Feynman rules using partial numerical double off-shell recurrence relations? (Mahlon)
  - → Are there numerical issues with the methods?
  - → Analytic results exist for almost all helicity amplitudes.
  - → Phenomenological application for NLO *PP→4 jets* is of limited interest.

A semi-numerical approach

- Findings:
  - For one-loop *2→4* we have as of yet not established sufficient accuracy using integration-by-parts on itself.
  - However, other semi-numerical methods are easily developed. For this case we used a generalization of the Van Neerven/Vermaseren method to reduce M-point tensor integrals to (M-1)-point tensor integrals (M>4).
  - This method is simply based on the 4-dimensionality of space time. As a consequence the loop momentum can be written as a linear combination of 4 of the external momenta (with the coefficients proportional to denominators).

Equivalent to method developed by Denner & Dittmaier

A semi-numerical approach

$$I_N^{\mu_1\cdots\mu_M}(p_1,\ldots,p_6) = \int \frac{d^Dl}{i\pi^{D/2}} \frac{l^{\mu_1}\ldots l^{\mu_M}}{d_1 d_2 \ldots d_N}, \quad d_i \equiv (l+q_i)^2, \quad q_i \equiv \sum_{j=1}^{i} p_j$$

$$l^\mu = \sum_{i=1}^{4} l\cdot p_{k_i} v_{k_i}^\mu = V^\mu + \frac{1}{2}\sum_{i=1}^{4}\left(d_{k_i} - d_{k_i-1}\right) v_{k_i}^\mu$$

$$v_{k_i}^\mu = \sum_{j=1}^{4} [G^{-1}]_{ij} p_{k_j}^\mu, \quad G_{ij} = p_{k_i}\cdot p_{k_j}$$

Tensor reduction method

$$V^\mu = -\frac{1}{2}\sum_{i=1}^{4}(r_{k_i} - r_{k_i-1})v_{k_i}^\mu, \quad r_k = q_k^2$$

$$I_N^{\mu_1\cdots\mu_M} = I_N^{\mu_1\cdots\mu_{M-1}} V^{\mu_M} + \frac{1}{2}\sum_{i=1}^{4}\left(I_{N,k_i}^{\mu_1\cdots\mu_{M-1}} - I_{N,k_i-1}^{\mu_1\cdots\mu_{M-1}}\right) v_{k_i}^{\mu_M}$$

A semi-numerical approach

$$
\begin{aligned}
I_N^{\mu_1\cdots\mu_M}(p_1,p_2,p_3,p_4,p_5,\ldots,p_N) &= I_N^{\mu_1\cdots\mu_{M-1}}(p_1,p_2,p_3,p_4,p_5,\ldots,p_N)V^{\mu_M}(p_1,p_2,p_3,p_4)\\
&+\frac{1}{2}\left(I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_1+p_2,p_3,p_4,p_5,\ldots,p_N)-I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_2,p_3,p_4,p_5,\ldots,p_N)\right)\\
&\quad\times v_1^{\mu_M}(p_1,p_2,p_3,p_4)\\
&+\frac{1}{2}\left(I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_1,p_2+p_3,p_4,p_5,\ldots,p_N)-I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_1+p_2,p_3,p_4,p_5,\ldots,p_N)\right)\\
&\quad\times v_2^{\mu_M}(p_1,p_2,p_3,p_4)\\
&+\frac{1}{2}\left(I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_1,p_2,p_3+p_4,p_5,\ldots,p_N)-I_{N-1}^{\mu_1\cdots\mu_{M-1}}(p_1,p_2+p_3,p_4,p_5,\ldots,p_N)\right)\\
&\quad\times v_3^{\mu_M}(p_1,p_2,p_3,p_4)
\end{aligned}
$$

Repeated application leaves us with 4-point tensor integrals (up to rank 4) and 4-dimensional scalar integrals. These can subsequently be calculated using integration-by-part (or other) techniques.

In many ways these type of tensor reduction techniques are numerically simpler than the integration-by-part techniques.

A semi-numerical approach

- With this method we calculated the one-loop *6 gluon* amplitude:
  - Accuracy equal or better than $O\left(10^{-8}\right)$ for
    - gauge invariance
    - singular term parts (proportional to Born).
    - comparison with analytically known 6 gluon amplitudes. (see e.g. Bern, Kosower, Dixon, Berger, Forde)
  - Evaluation speed is significantly slower than for the previous calculation, code can be easily improved:
    - “unrolling” recursion (i.e. hard-coding some of the tensor integrals; e.g. all 2-, 3- and 4-point tensor integrals).
    - Optimizing code.
  - However, while speed is already sufficient for use in NLO monte carlo in a “real” application (e.g. *PP → tt+2 jets*) one would do as much as possible analytic (FORM writing out explicitly part of the recursion). This is a balance between computer speed and size of generated code the compiler has to deal with.

A semi-numerical approach

- There is one caveat common to both methods:
  - ➜ There will be numerical instabilities in certain phase space regions when external momenta become linearly dependent on each other:

$$G_4(p_{i_1}, p_{i_2}, p_{i_3}, p_{i_4}) \to 0$$

    i.e. the Gram determinant becomes very small (for example planar events, thresholds and more complicated geometrical configurations).
  - ➜ These regions are easily detected by a numerical program and can be treated differently.

A semi-numerical approach

- There are several methods to deal with the instabilities:
  - → An interpolation method (while keeping track of things like gauge invariance and singular term factors). (Oleari,Zeppenfeld)
  - → Using integration-by-parts techniques to derive expansion formulae (in a small parameter $\Delta \sim G_4$).
    (Glover,Zanderighi,Ellis,WG; Dittmaier,Denner,Roth,Wieders,Weber,Bredenstein)
  - → Construct the entire tensor reduction method for 5-point and higher in such a manner that the instabilities are avoided. This method is a good alternative to integration-by-parts method.
    (Denner,Dittmaier)

A semi-numerical approach

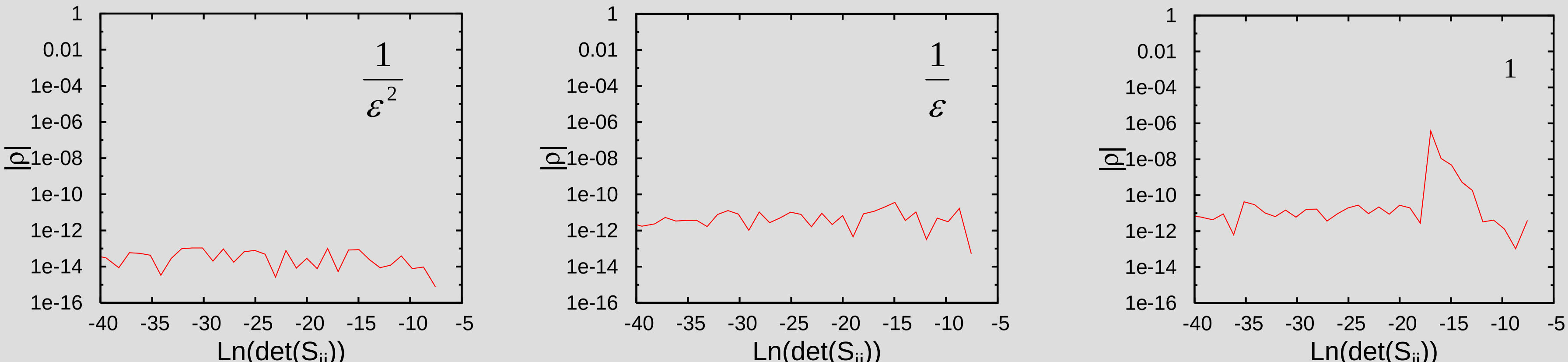
- We developed the integration-by-parts method as a way to deal with the exceptional momenta configurations. An whole alternative system of recursion relations can be set up which reduces tensor integrals to master integrals for exceptional momenta configurations (up to terms of $O\left(\Delta^{K}\right)$ for any chosen K)

The region $\det(S) \leq S_0$ and $\det(G) \leq G_0$

$$I(D;\{\nu_k\}_{k=1}^{N}) = \sum_{i=1}^{N} r_i I(D;\{\nu_k - \delta_{ik}\}_{k=1}^{N}) + \sum_{i=1}^{N} \Delta_i \nu_i I(D+2;\{\nu_k + \delta_{ik}\}_{k=1}^{N})$$

$$S_{ij} r_j = (1,1,\ldots,1) + \Delta_i$$

Comparison (relative accuracy) of numerical result to analytic result for *H→4 quarks*



A semi-numerical approach

- Inspired by the Denner-Dittmaier treatment of the gram-determinant issue in the tensor reduction method we can again use the Vermaseren-Van Neerven method:

$$
\begin{aligned}
l_\mu l^\nu \overline{W}^\mu_\nu &= l_\mu l^\nu \delta^{\mu p_1 p_2 p_3 p_4}_{\nu p_1 p_2 p_3 p_4} \\
&= \Delta_4\, l^2 - \sum_{i,j=1}^{4} (d_i - d_{i-1} - r_i + r_{i-1}) Y^{ij} (d_j - d_{j-1} - r_j + r_{j-1}) \\
&= Y + \Delta_4 d_0 - \sum_{i=1}^{4} (d_i - d_{i-1}) Y^i - \sum_{i,j=1}^{4} (d_i - d_{i-1}) Y^{ij} q_j^\alpha\, l_\alpha \qquad (28)
\end{aligned}
$$

where

$$
Y^{11} = \delta^{p_2 p_3 p_4}_{p_2 p_3 p_4};\ Y^{12} = \delta^{p_2 p_3 p_4}_{p_1 p_3 p_4};\ Y^{23} = \delta^{p_1 p_3 p_4}_{p_1 p_2 p_4}; \cdots \qquad (29)
$$

$$
Y^i = \sum_{j=1}^{4} Y^{ij} (r_j - r_{j-1});\ Y = \sum_{j=1}^{4} Y^j (r_j - r_{j-1}) \qquad (30)
$$

Conclusions

- We demonstrated we are able to calculate *2→3* processes with sufficient accuracy and speed to start construction of the parton level MC generators
- For *2→4* processes we can use brute force matrix element generation. Also speed and accuracy are fine.
- With all the key methods in place, now the hard works start:
  The construction of efficient event generators which integrate out the bremsstrahlung contributions over the unresolved phase space.
- Finally we should not forget the importance to provide an interface with shower MC programs.