Advanced Computing and Analysis Techniques in Physics Research

February 22-27, 2010, Jaipur, India

Methodology of Computations in Theoretical Physics

--- Summary ---

Peter Uwer

HUMBOLDT-UNIVERSITÄT ZU BERLIN

HEPTOOLS

Tools and Precision Calculations for Physics Discoveries at Colliders

PHYSICS AT THE TERA SCALE Helmholtz Alliance

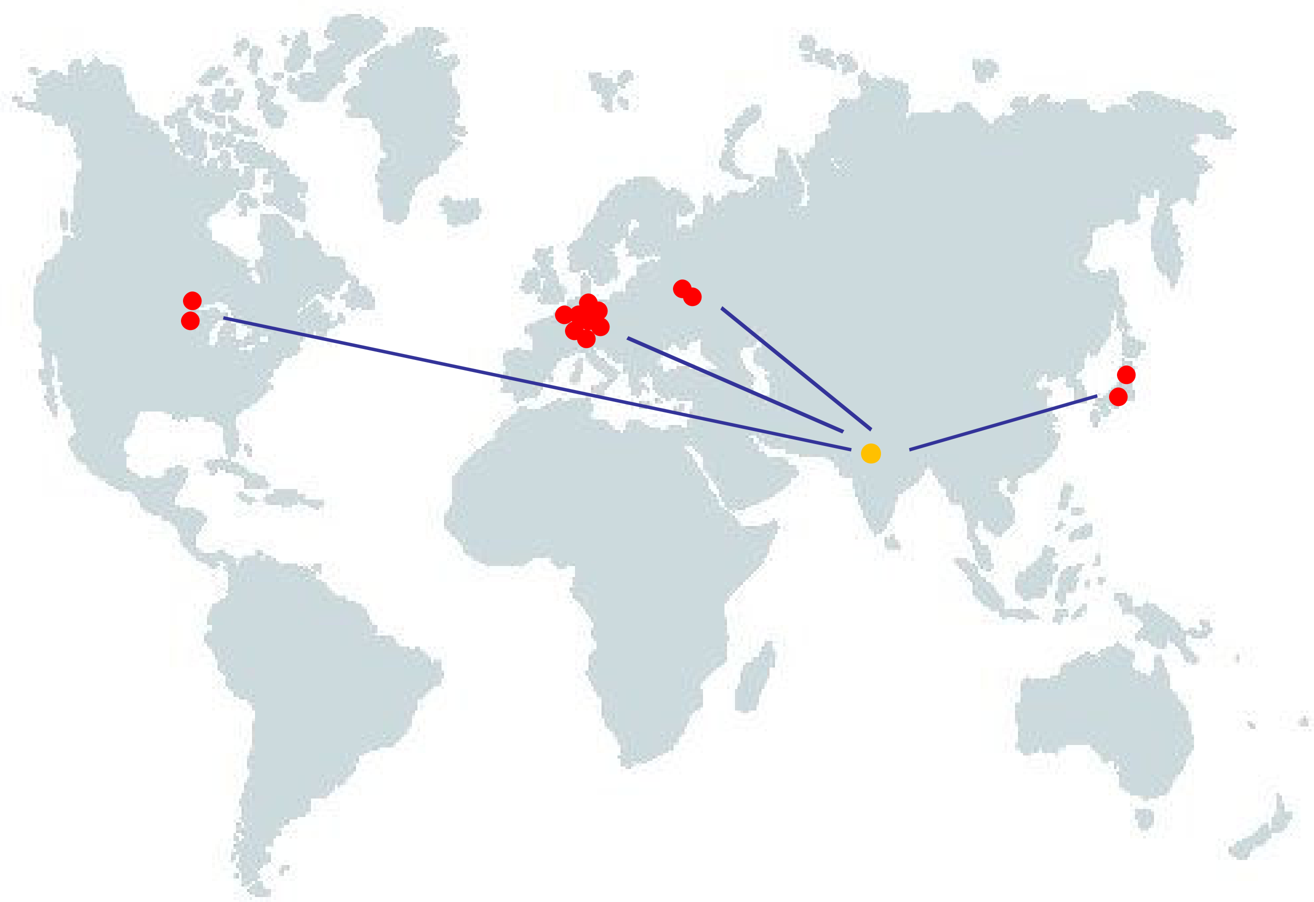
Computational Theoretical Particle Physics SFB TR9

Deutsche Forschungsgemeinschaft DFG

Statistics

- 5 + 4 + 6 = 15 presentations
- 450 min = 7.5 h
- In total 367 transparencies, 1.2 min / slide
- Average number of transparencies: 24.4 / talk
- Extreme values: min: 10, max: 56

Where the speakers came from

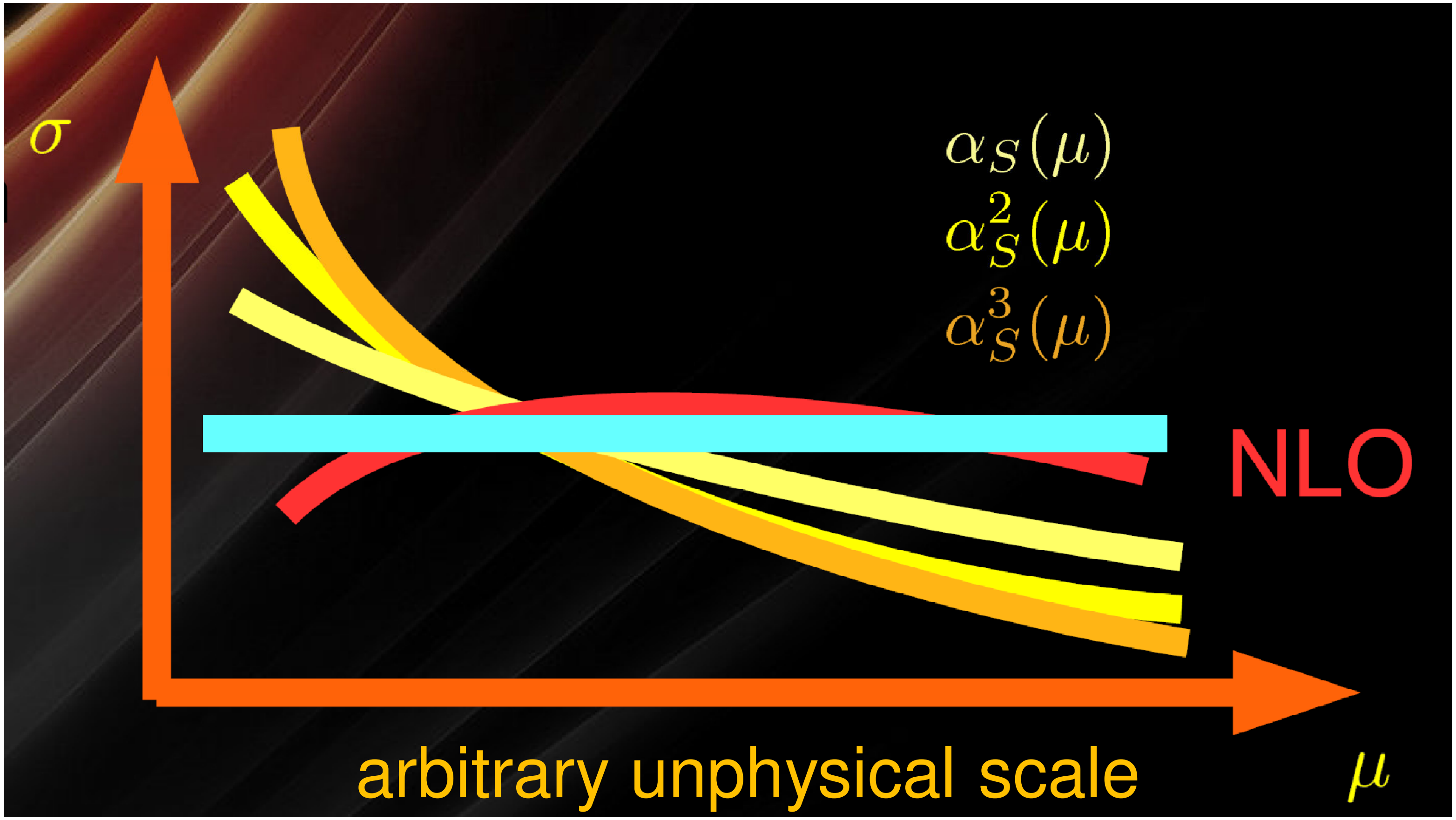


Europe: 8, Japan: 2, Russia: 2, US: 2

Main topics

- Automation of higher order corrections
  - Techniques for loop integrals 6
  - Computational aspects 2
  - Real corrections and subtractions 2
- Computer Algebra 2
- Various topics 3

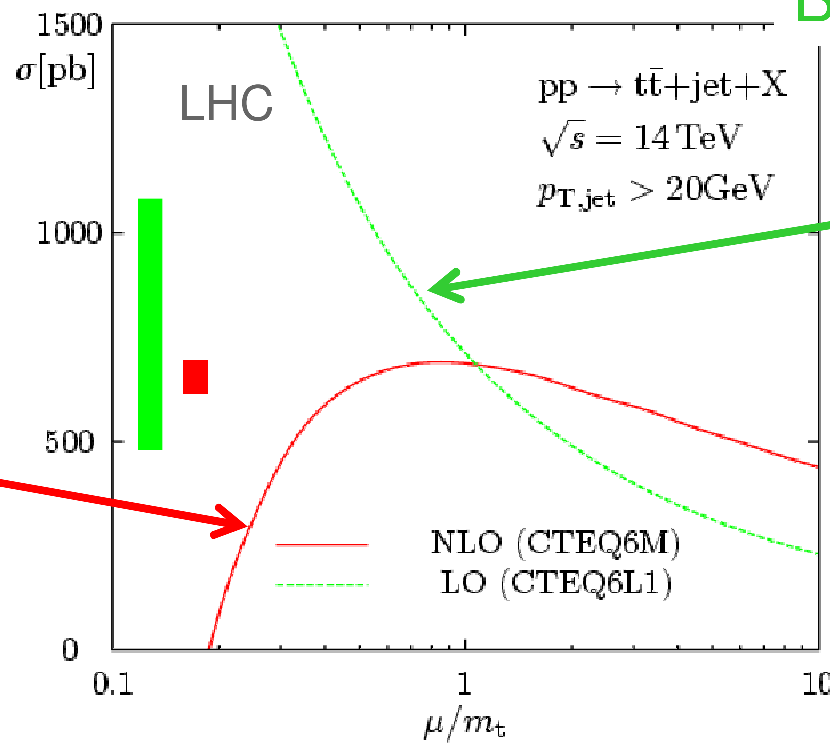
- What is the basic problem ?

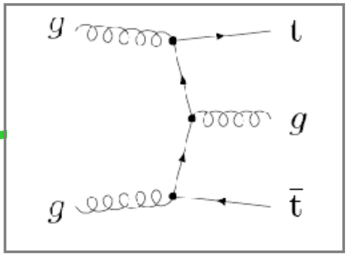


[Daniel Le Maitre]

Example: Top-quark production + 1 Jet

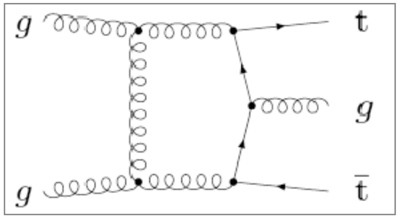
Born approximation





O(10) diagrams

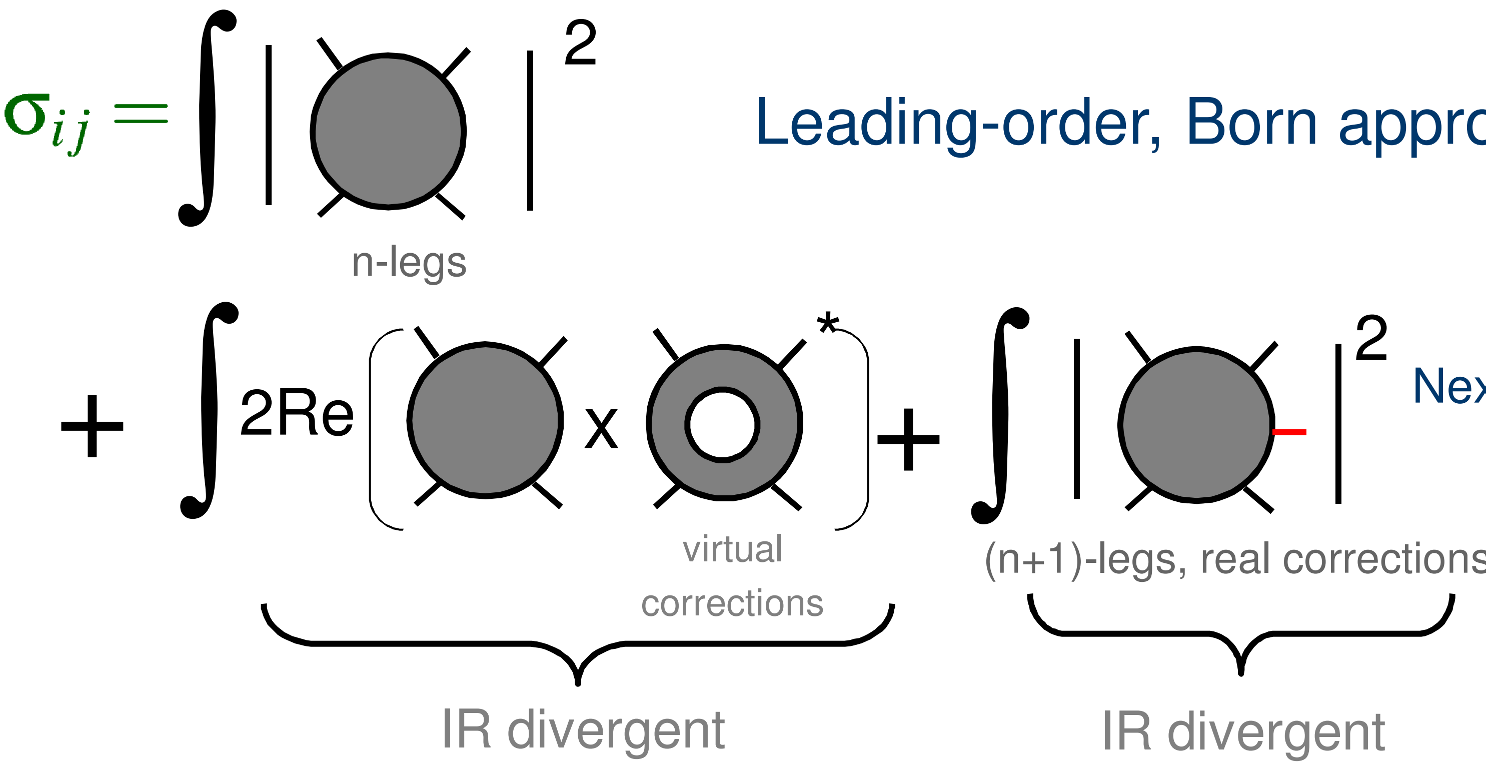
Next-to-leading order (NLO)



350 diagrams

→ Born approximation is not reliable, need to go beyond leading-order

Generic one-loop calculation



Leading-order, Born approximation

Next-to-leading order (NLO)

Automation of higher order corrections

Bottleneck in one-loop calculation:

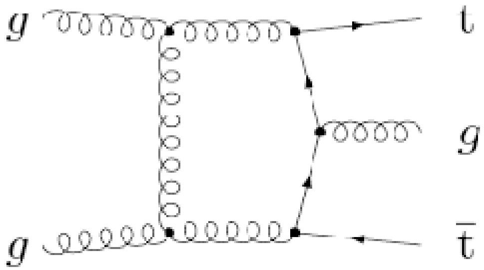
- Calculation of the virtual corrections
  - Many diagrams
  - Each with complicated analytical structure
  - Numerical stability and speed

- Combination of virtual corrections with real ones
  - Cancellation of IR singularities, conceptually solved, but cumbersome if done by hand

Need for new methods and automation

Algorithms crucial for automation

A typical one-loop diagram



$$E_{\mu\nu\alpha\ldots} = \int d^d\ell \frac{\ell_\mu \ell_\nu \ell_\alpha \ell_\beta}{(\ell^2 + i\varepsilon)((\ell + p_1)^2 + i\varepsilon)((\ell + p_1 - p_t)^2 - m_t^2 + i\varepsilon)} \times \frac{1}{((\ell - p_1 + p_{\bar{t}})^2 - m_t^2 + i\varepsilon)} \frac{1}{((\ell - p_2)^2 + i\varepsilon)}$$

→ complicated function of many variables

How do we calculate this efficiently?

Techniques for loop integrals

Different approaches:

- Refinement of mixed approaches
- Improved integration methods
- New algorithms

[Tord Riemann]

Basic idea:

Make use of the fact that all the scalar one-loop integrals are known analytically

→ Derive reduction avoiding leading Gram determinants in the denominator

Explicit reduction formulae are implemented in Computer code

Exceptional configuration with vanishing Gram determinants are handled by special reduction (extrapolation)

[Tord Riemann]

Using LoopTools call and our math numerics (preliminary):

```
x       D111

 7 :  0.007106204244698895      +0.004653980785027325 I   D0i[dd111]
     -3.15345811639208   -10    -3.318373348243635'-10 I   Z4d30,Z4d20,T4id20

-6 : -3.231307907858403 4-06    -2.896316001494784 6-06 I D0i[dd111]
      3.1479286753545824 10      3.318332145498356   10 I Z4d30,Z4d20,T4id20

 5 :  5.5231182028025025 09     +3.4832284324178667 09 I D0i[dd111]
      3.0926394107374516 10      3.3179201270079527 10 I Z4d30,Z4d20,T4id20

 x<  4:          LoopTools dies out

-4 : -3.1544928789869657-10     -3.33218368329059   -10 I D0i[dd111]
     -3.0798250216856066-10     -3.3447698103297804-10 I   flci

 x < -3:         loss of accuracy

-3 : -3.15374217566590 8 -10    -3.31639655233478   -10 I D0i[dd111]
     -3.1537481925176414-10     -3.3164147721227693-10 I   flci

-2 : -3.1500799889469005-10     -3.29915924109457   -10 I D0i[dd111]
     -3.150080001830792 -10     -3.2991592067243136-10 I   flci

-1 : -3.112267506942415 -10     -3.135823319774082 -10 I D0i[dd111]
     -3.1122675069507063-10     -3.1358233197649007-10 I   flci
```

x allows to test numerical stability

Techniques for loop integrals

[Giovanni Ossola]

General structure of one-loop amplitude:

$$\mathcal{M} = \int d^n\bar{q}\; A(\bar{q}) = \int d^n\bar{q} \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \dots \bar{D}_{m-1}}$$

$$\begin{aligned}\mathcal{M} \quad = \quad & \sum_i d_i \,\mathrm{Box}_i + \sum_i c_i \,\mathrm{Triangle}_i \\ + \quad & \sum_i b_i \,\mathrm{Bubble}_i + \sum_i a_i \,\mathrm{Tadpole}_i + \mathrm{R}\,,\end{aligned}$$

Reduction at the integrand level

[Giovanni Ossola]

$$
\begin{aligned}
N(q) &= \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d(i_0 i_1 i_2 i_3) + \tilde{d}(q; i_0 i_1 i_2 i_3) \right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \\
&+ \sum_{i_0<i_1<i_2}^{m-1} \left[c(i_0 i_1 i_2) + \tilde{c}(q; i_0 i_1 i_2) \right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\
&+ \sum_{i_0<i_1}^{m-1} \left[b(i_0 i_1) + \tilde{b}(q; i_0 i_1) \right] \prod_{i \neq i_0, i_1}^{m-1} D_i \\
&+ \sum_{i_0}^{m-1} \left[a(i_0) + \tilde{a}(q; i_0) \right] \prod_{i \neq i_0}^{m-1} D_i
\end{aligned}
$$

- Structures in red vanish after integration and their form is known → finite number of terms

→ Determine coefficients by solving linear system of equations

OPP method (Ossola, Pittau, Papadopoulos)

[Giovanni Ossola]

- OPP method very powerfull
- Available as Fortran program CutTools
- Can be combined with automated amplitude generation (combination with HELAC already done)
- Many new results recently (pp->Wjjj,pp->ttbb, pp->ttjj)

One-loop amplitudes solved ?

[Elise de Doncker]

Basic idea:

Recursive (deterministic) integration over Feynman parameter, combined with extrapolation

$$\int_0^1 dx \int_0^1 dy \frac{2\alpha y}{(x+y-1)^2+\alpha^2}, \quad \alpha = 10^{-p}$$

	DQAGE × DQAGE		DCUHRE	
p	ABS. ERR.	# EVAL.	ABS. ERR.	# EVAL.
1	0.00e+00	21255	2.06e-12	144165
2	2.40e-13	93135	5.96e-12	1998675
3	3.49e-13	208035	1.37e-12	21040551
4	1.58e-13	388125	8.04e-12	99999963
5	4.49e-13	561585	4.40e-07	99999963
6	1.69e-09	527205	3.38e-02	99999963
7	1.42e-10	686745	1.99e+00	99999963
8	3.94e-10	902145	3.04e+00	99999963
9	3.20e-08	106965	3.13e+00	99999963
10	4.32e-09	1964385	3.14e+00	99999963
11	1.87e-01	58651365	3.14e+00	99999963

- use DQAGE from QUADPACK recursively

Techniques for loop integrals

[Elise de Doncker]

Six-point scalar integrals are reduced algebraically to 3- and 4-point scalar integrals

3- and 4-point integrals are then evaluated numerically

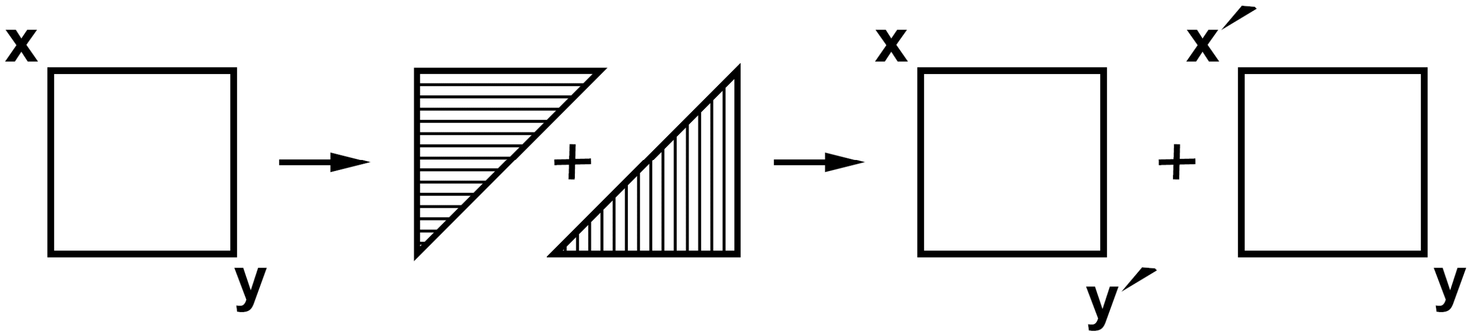
Technique also applicable to tensor integrals

Recursive integration might be interesting also for other fields

Sector decomposition to isolate singularities [Mikhail Tentyukov]

$$\int_0^1 \frac{dx dy f(x,y)}{(x+y)^{2-\varepsilon}}$$

with $\varepsilon \to 0$ problems in both $x = 0$ and $y = 0$ *simultaneously*.



$$\int_0^1 dx \int_0^x dy + \int_0^1 dy \int_0^y dx \quad \begin{pmatrix} y \to xy' \\ x \to x'y \end{pmatrix} = \int_0^1 \frac{dx dy' f(x, xy')}{x^{1-\varepsilon}(1+y')^{2-\varepsilon}} + \int_0^1 \frac{dx' dy f(x'y, y)}{y^{1-\varepsilon}(1+x')^{2-\varepsilon}}$$

Singularities are factorized, can be integrated analitycally

to find the decomposition of a complicated integrand highly non-trivial

FIESTA [Mikhail Tentyukov]

Feynman Integral Evaluation by a Sector decompositTion Approach

Computer algebra part in Mathematica combined with numerical integration routine

Important:

- Publicly available
- Different Algorithms for sector decomposition
- Applicable to multi-loop integrals
- Important new 4-loop results
- Circumvent memory problem in Mathematica
- Own interpreter to process formulae of TeraByte length

Alternative algorithm for sector decomposition

[Toshiaki Kaneko]

Sector decomposition based on computational geometry

Diagram	A	B	C	S	X	H	This method	Exponential S.D.
Bubble	2	2	2	2*	2		2	2
Triangle	3	3	3	3*	3		3	3
Box	12	12	12	12	12		12	8
Tbubble	58	48	48	48*	48		48	36
Double box, $p_i^2 = 0$	775	586	586	362	293	282	266	106
Double box, $p_4^2 \neq 0$	543*	245*	245*	230*	192*	197	186	100
Double box, $p_i^2 = 0$ nonplanar	1138	698	698	441*	395		360	120
D420	8898	564	564	180	F		168	100
3 loop vertex (A8)	4617*	1196*	1196*	871*	750*	684	684	240
Triple box	M	114256	114256	22657	10155		6568	856

- "H" : G. Heinrich, Int. J. Mod. Phys. A23 (2008) 1457.
- "A" , "B", "C", "S", "X" (without "*") is cited from Borger & Weinzierl, and Smirnov &Tentyukov.

→ implementation underway

Techniques for loop-integrals

Methods rely on

- Increased computational power
- Increased main memory
- Parallelization is used frequently

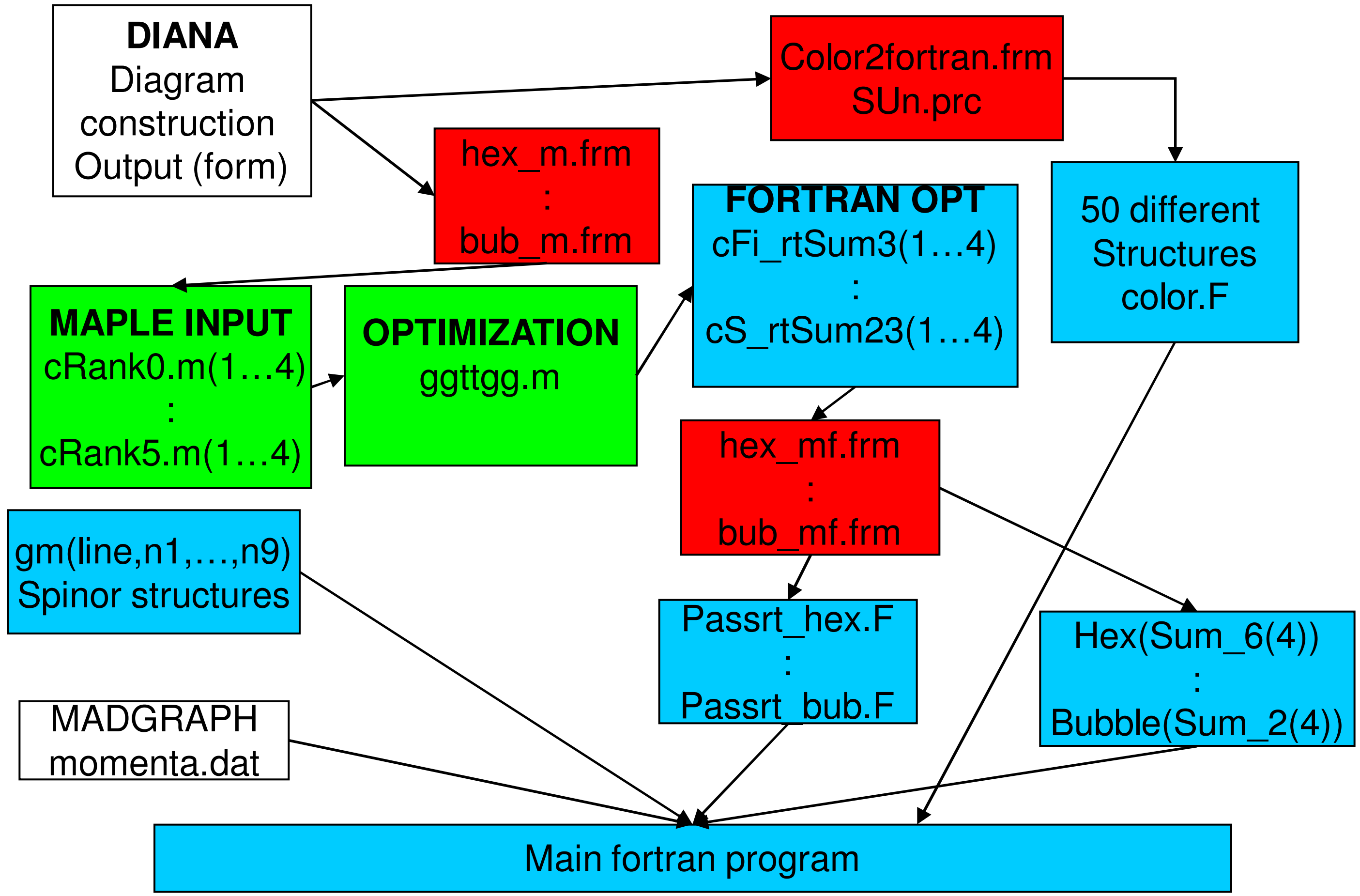
[Theodoros Diakonidis]

Apply reduction scheme
presented by Tord Riemann
to gg→ttgg @ 1-loop

O(1000) Feynman diagrams with complex structure

→ Automation needed

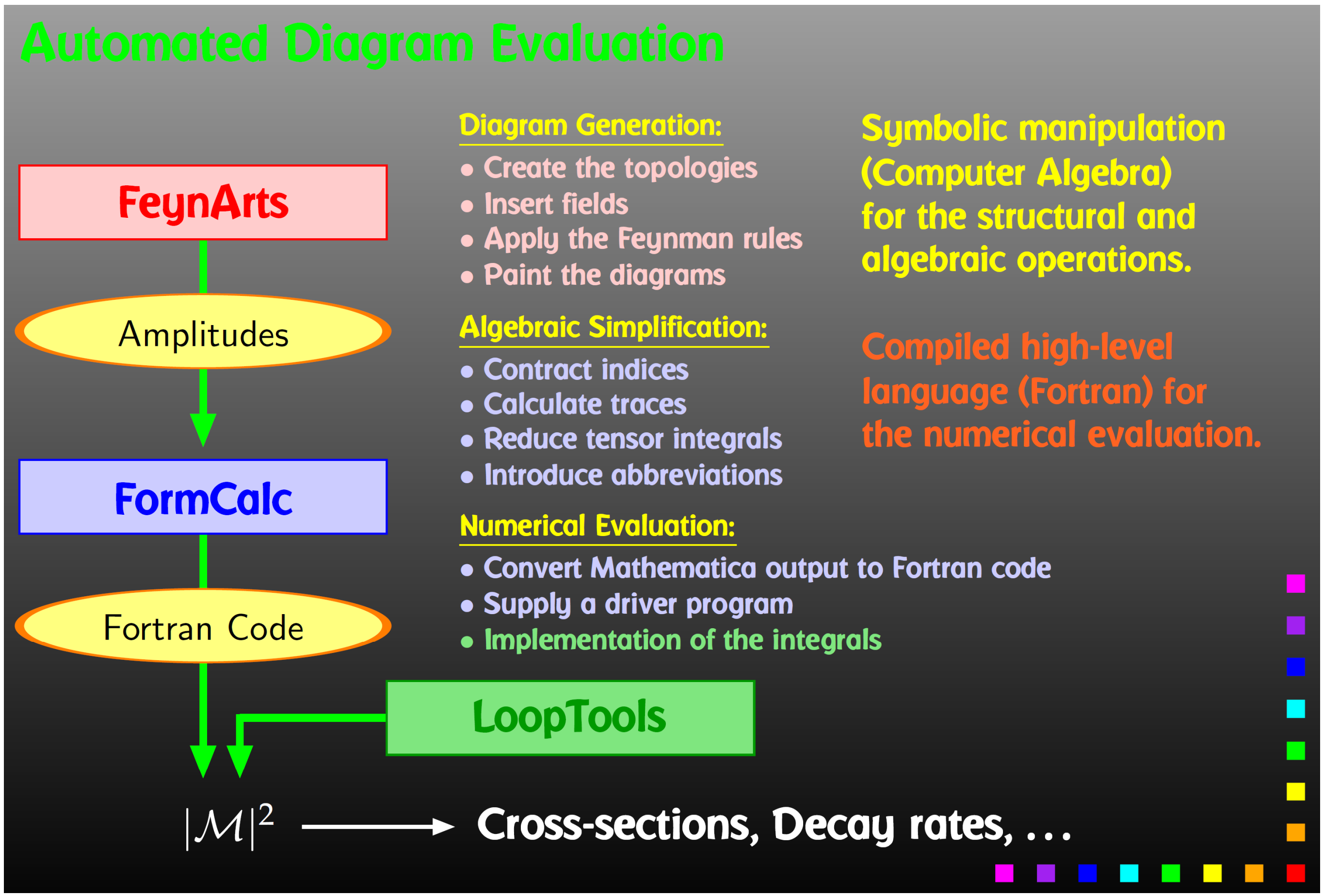
[Theodoros Diakonidis] (Qgraf→Form→Maple→Fortran)



Steering with shell scripts, process specific

Automation of higher order corrections

[Thomas Hahn] Process independent automation based on Feynman diagrams and tensor reduction



Automation of higher order corrections

[Thomas Hahn]

Many new features in the Feynarts-System

- Tweaking model files
- Diagram selection
- Linear combination of fields (mass vs gauge states)

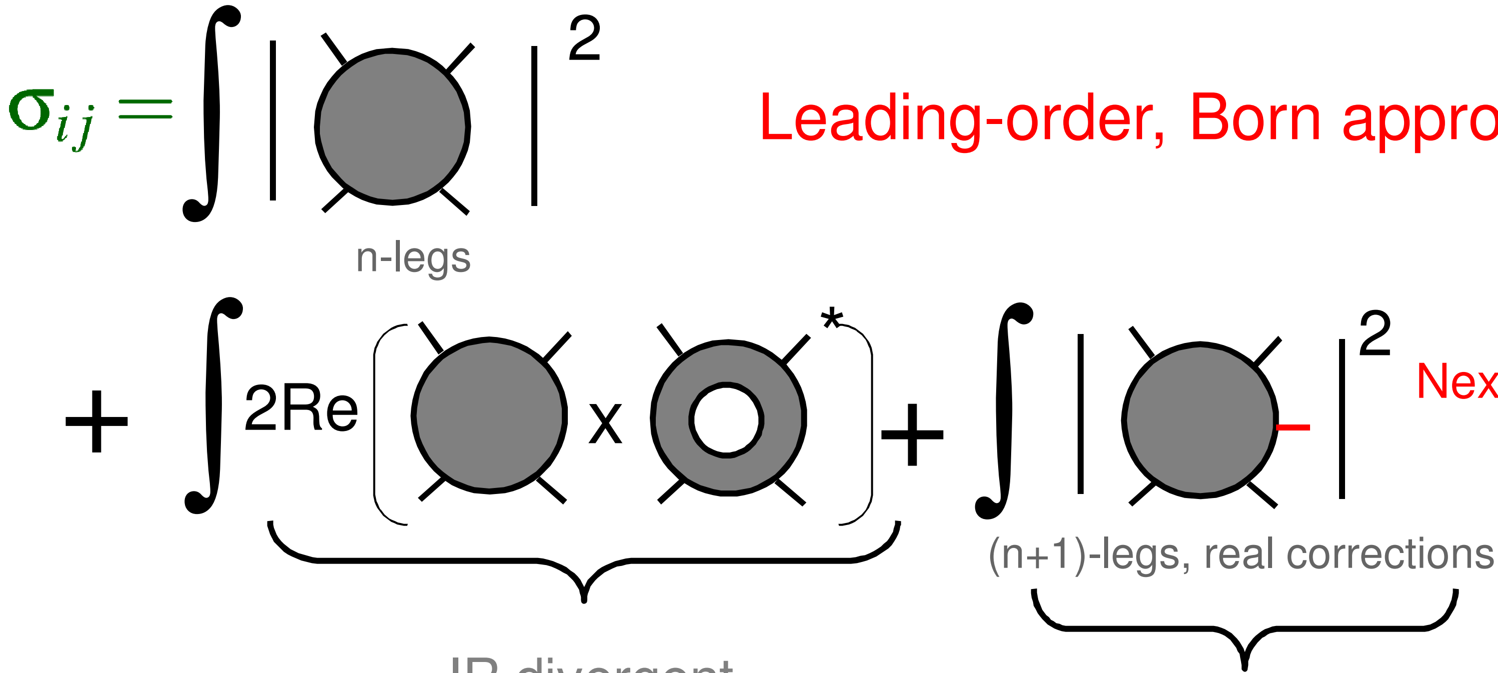
Computational aspects

- Efficient Fortran code generation, abbreviations to remove common subexpressions
- Parallelization of parameter scans

→ very powerful tool but:

- **On the other hand, you can only go where there are roads. You can't climb a mountain with your car.**

- Generic one-loop calculation



Leading-order, Born approximation

Next-to-leading order (NLO)

IR divergent

$1/\varepsilon^2, 1/\varepsilon$ poles
appear in scalar loop integrals

IR divergent

poles appear after phase integration in d dimensions

Real corrections and subtractions

Problem: Phase space integration cannot be done in d dim.

→ Add and subtract a counterterm which is easy enough to be integrated analytically:

$$\int_0^\alpha dx \frac{1}{x} f(x) x^\epsilon$$

$$= \int_0^\alpha dx \frac{1}{x} (f(x) - f(0)) x^\epsilon + \frac{1}{x} f(0) x^\epsilon$$

$$= +\frac{1}{\epsilon}\alpha^\epsilon + \int_0^\alpha \frac{1}{x}(f(x) - f(0)) + O(\epsilon)$$

Can be done numerically

Construction of subtraction for real corrections more involved,

Fortunately a general solution exists:

→Dipole subtraction formalism

Real corrections and subtractions

$$\sigma_{\text{sub}} = \sum_{\text{dipoles}} \mathcal{D}_{ij,k}(p_i, p_j, p_k)$$

Generic form of individual dipol:

Leading-order amplitudes
Vector in color space

$$\mathcal{D}_{ij;k} = -\frac{1}{(p_i+p_j)-m_{ij}^2}\langle \ldots, \widetilde{ij}, \ldots, \tilde{k}, \ldots \left| \frac{\mathbf{T}_a \cdot \mathbf{T}_{ij}}{\mathbf{T}_{ij}} V_{ij,k} \right| \ldots, \widetilde{ij}, \ldots, \tilde{k}, \ldots \rangle$$

universal

Color charge operators,
induce color correlation !

Spin dependent part,
induces spin correlation !

Example gg→ttgg: 36 (singular) dipoles $\mathcal{D}_{g_1 g_3, t}, \ldots$

→ Automation required

[Rikkert Frederix]

Automation of NLO subtraction terms

Two different methods:

- Catani-Seymour subtraction
- Frixione-Kunszt-Signer subtraction

→ useful to interface with MC@NLO

→ Fully automated based on Madgraph: MadFKS

[Paolo Bolzoni]

Extension of subtraction method to NNLO

Much more involved due to double unresolved configuration

Analytic integration of subtraction terms highly non trivial

→ Solution using:

- Mellin-Barnes representation
- Special summation algorithms for nested sums (XSummer in Form)

Computer Algebra -- Form

Standard Tool in Theoretical Particle Physics if large expressions are encountered:

Form by Jos Vermaseren et al.

Important features:

- Expression size only limited by disk space (TB)
- Only local operations i.e. no factorization

→ Many ongoing developments

Talks by Irina Pushkina and Mikhail Tentyukov

Computer Algebra -- Form

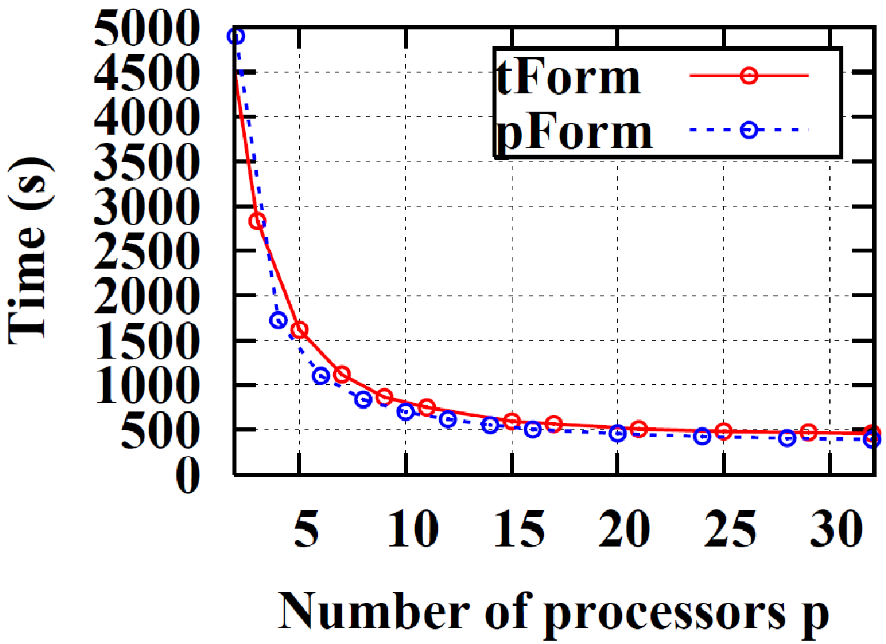
[Irina Pushkina, Mikhail Tentyukov]

New features:

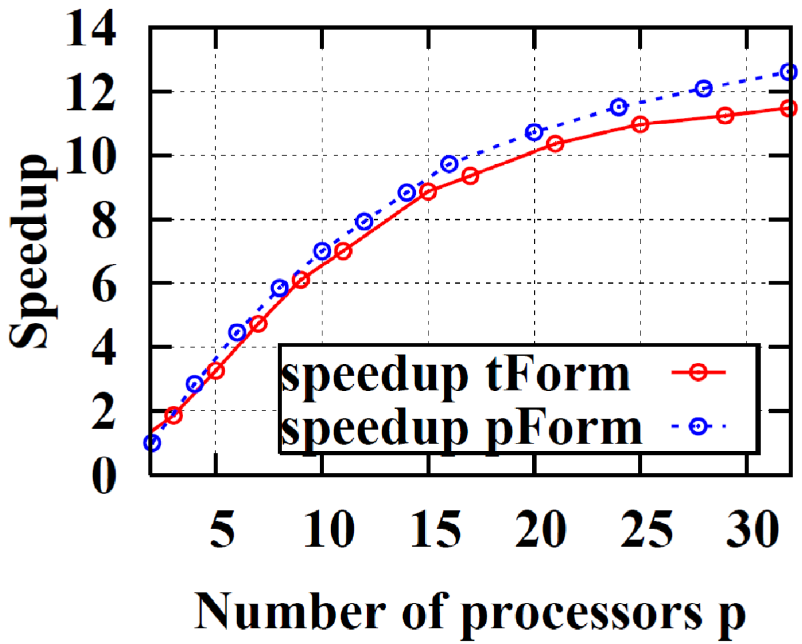
- Architecture independent file storage (32bit vs 64bit)
- Checkpoints to save intermediate states
- Steps towards open source (summer 2010?)
- Two approaches to parallelisation:
  - Parform based on MPI for cluster
  - Tform based on threads for multi-core machines
- Improved load balancing
- Link to Grace system

Speed up in Form:

[Mikhail Tentyukov]



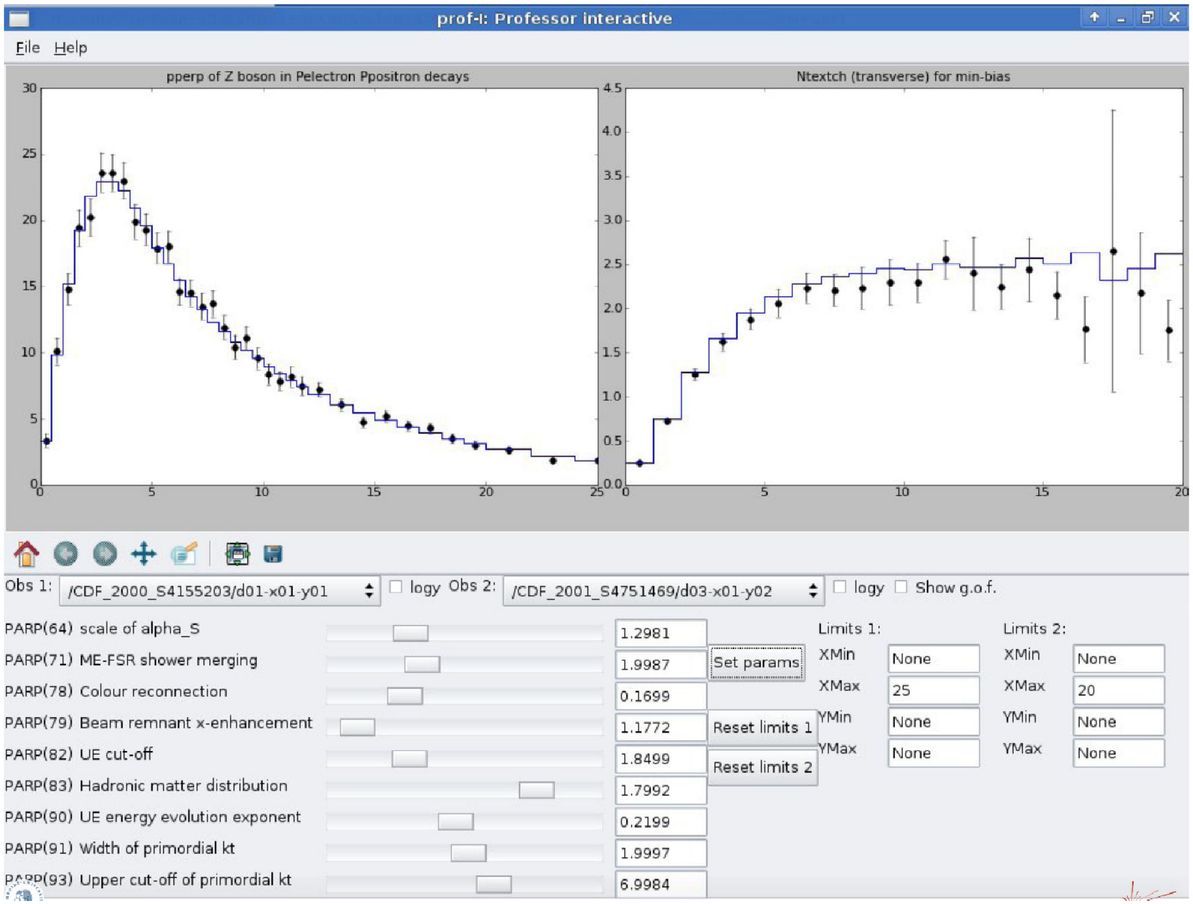
BAICER N=12



$$S(p) = \frac{T(1)}{T(p)}$$

Tuning framework *Professor*

[James Monk]



Multi-dimensional interpolation

Change Generator Parameters on the fly through interpolation in parameter space

Remarks

- Main programming language in Theoretical Particle Physics

Fortran

- Powerful mixed approach:

Analytic part is combined with numerical part,
a chain of different tools is connected using scripts

- Numerical instabilities

Switch to quadrupel and higher accuracy

→ Very active field, many new and important developments recently

Remarks

Important progress concerning the automation of one-loop amplitudes

→ OPP Method

Final Remarks

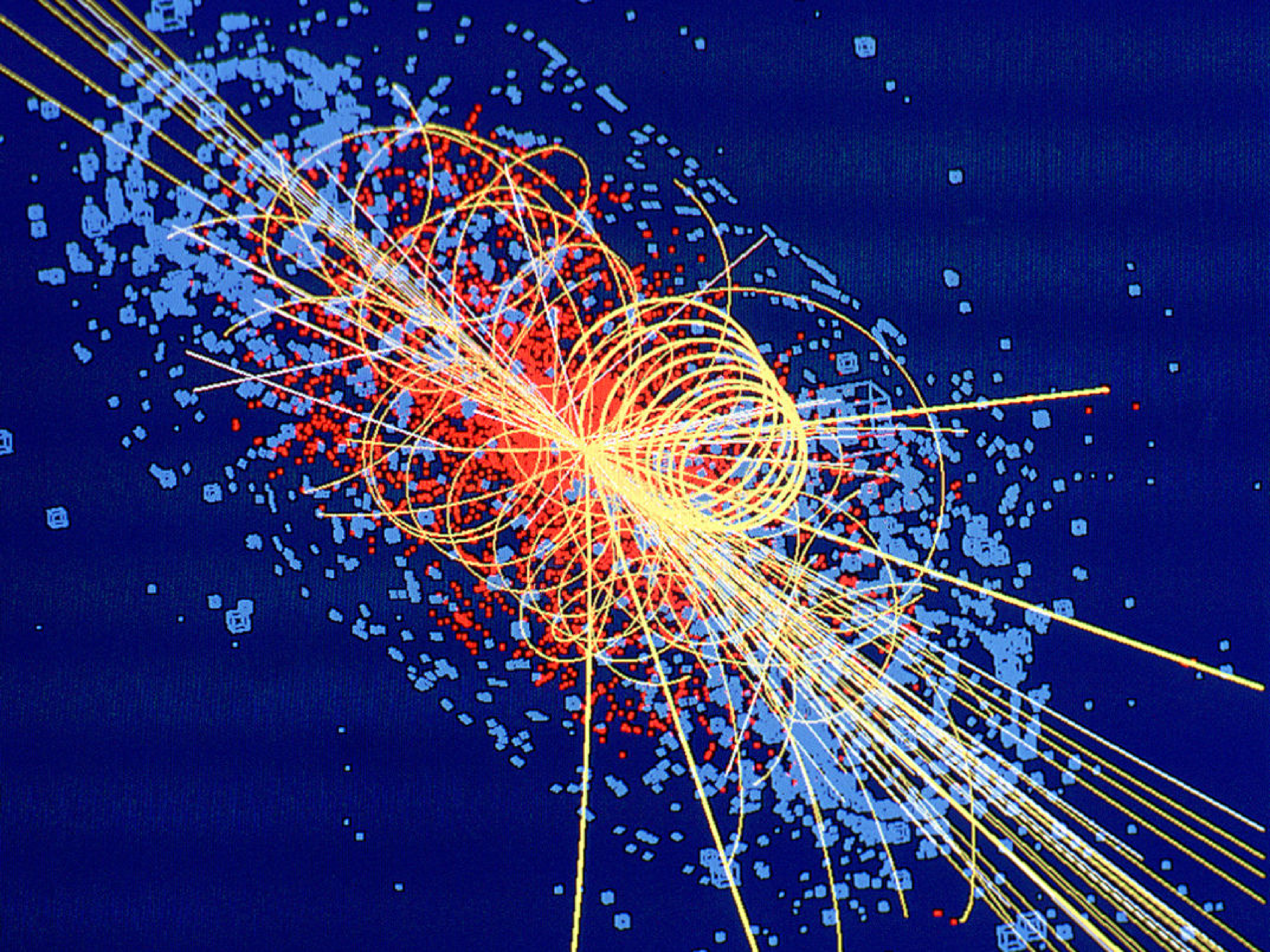
- Apologies that not everybody could be mentioned
- Many thanks to all the speakers
- Many thanks to the audience of track 3 for their contribution in many lively discussions

All talks are uploaded, if you want to see the details check Indico

Thanks!

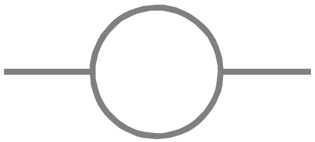
- Thanks to IAC for insight
- Sudhir for everlasting help
- All the chairs for creating wonderful workshop atmosphere
- Students etc for flawless technical support
- Speakers for a mosaic of present and future
  - and their help with this summary
- Let's see what the future brings to Brunel!

2010-02-27 Axel Naumann - ACAT 2010 Track 1 26



Automation of higher order corrections

Traditional approach to tensor reduction



$$\int d^d\ell \frac{\ell_\mu}{(\ell^2 - m_0^2 + i\varepsilon)((\ell+p)^2 - m_1^2 + i\varepsilon)} = p_\mu B_1$$

$$B_1 = \frac{1}{2p^2}\left(A(m_0) - A(m_1) + (m_1^2 - m_0^2 - p^2) B_0 \right)$$

$$A(m) = \int d^d\ell \frac{1}{\ell^2 - m^2 + i\varepsilon} \qquad B_0 = \int d^d\ell \frac{1}{(\ell^2 - m_0^2 + i\varepsilon)((\ell+p)^2 - m_1^2 + i\varepsilon)}$$

[Passarino, Veltman 78]

Major problem:

(spurious) numerical instabilities for
exceptional momentum configurations
vanishing Gram determinants

in the example above: $p^2 \to 0$ (integral bases degenerates “0/0”)