The GOLEM Project: Progress, Status and Prospects

Gavin Cullen

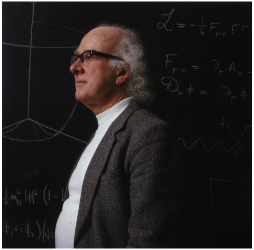
in collaboration with
A. Guffanti, J.P. Guillet, G. Heinrich, S. Karg, N. Kauer, T. Kleinschmidt, E. Pilon, T. Reiter, M. Rodgers, I. Wigmore

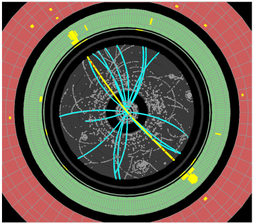
HP^2.3rd, Florence, Italy

14th September 2010

High Precision...

The LHC era is upon us

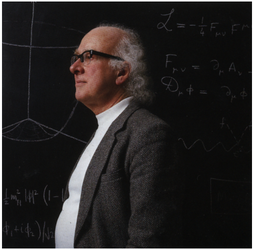




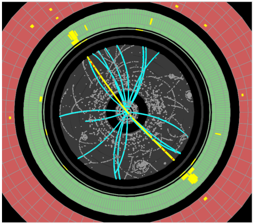
- ~ 1 million Higgs bosons per year
  ($\sigma \sim 10$ pb at yearly luminosity of ~ 100 fb^{-1})
- LHC runs for ~ 10 years at a cost of ~ 4 bn euro

High Precision...

The LHC era is upon us



- ~ 1 million Higgs bosons per year
  ($\sigma \sim 10$ pb at yearly luminosity of $\sim 100\ \text{fb}^{-1}$)
- LHC runs for ~ 10 years at a cost of ~ 4 bn euro

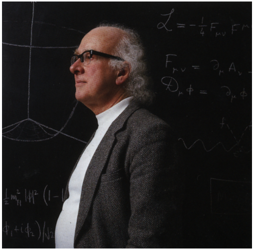


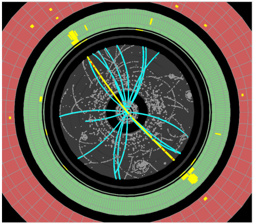
High Precision...

The LHC era is upon us

- $\sim$ 1 million Higgs bosons per year
  ($\sigma \sim$ 10 pb at yearly luminosity of $\sim$ 100 fb^{-1})
- LHC runs for $\sim$ 10 years at a cost of $\sim$ 4 bn euro
- $\Rightarrow$ 400 euro for each Higgs boson!

Precise knowledge of signal and background gives us value for money



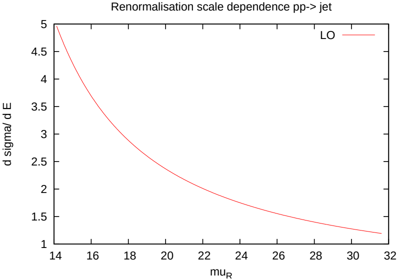


...for Hard Processes

$$\sigma_{AB} = \int dx_a dx_b f_{a/A}(x_a,\mu_F^2) f_{b/B}(x_b,\mu_F^2)\left[\hat{\sigma}_0 + \alpha_s(\mu_R)\hat{\sigma}_1 \cdots\right]_{ab\to X}$$

Truncating series introduces scale dependence: calculation at N^{th} order

$$\frac{d\hat{O}bs}{d\log(\mu_R^2)} = \mathcal{O}(\alpha_s^{N+1})$$



- At leading order huge scale variation
- At NLO scale dependence reduced

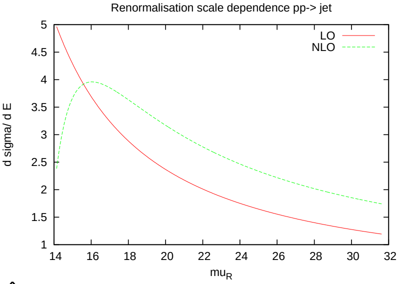
$$\hat{O}_{bs} = \sigma_0\alpha_s^2(\mu_R) + \alpha_s^3(\mu_R)(\sigma_1 + 2b_0\log(\mu_R)\sigma_0)$$

...for Hard Processes

$$\sigma_{AB} = \int dx_a dx_b f_{a/A}(x_a, \mu_F^2) f_{b/B}(x_b, \mu_F^2) \left[\hat{\sigma}_0 + \alpha_s(\mu_R)\hat{\sigma}_1 \cdots\right]_{ab \to X}$$

Truncating series introduces scale dependence: calculation at N^{th} order

$$\frac{d\hat{O}bs}{d\log(\mu_R^2)} = \mathcal{O}(\alpha_s^{N+1})$$



- At leading order huge scale variation
- At NLO scale dependence reduced

$$\hat{O}_{bs} = \sigma_0 \alpha_s^2(\mu_R) \quad + \alpha_s^3(\mu_R)(\sigma_1 + 2b_0 \log(\mu_R)\sigma_0)$$

Outline

- NLO calculations
- The Golem Method
- Golem-2.0
  - Spinney
  - Golem95
- Results
  - $q\bar{q} \to b\bar{b}b\bar{b}$

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

Automated
Multi Leg LO Tools

- AlpGen [Mangano et al]
- CalcHEP [Pukhov, Belyaev, Christensen]
- MadGraph [Maltoni, Stelzer]
- Grace [Fujimoto et al]
- Whizard [Kilian et al]
- Sherpa [Krauss et al]

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

Automated Infrared Subtraction Tools

- AutoDipole [Hasegawa, Moch, Uwer]
- MadDipole [Frederix,Gehrmann,Greiner]
- HELAC dipole [Czakon, Papadopoulus, Worek]
- Sherpa [Krauss et al]
- TevJet [Seymour,Tevlin]

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

Automated Public NLO Tools

- FeynArts/FormCalc [Hahn et al]
- MCFM [Campbell et al]
- MC@NLO [Frixione, Webber]

What goes into an NLO calculation?

What Tools are on the Market?

$$\sigma_{NLO} = \int_n d\sigma^{LO} + \int_n \left(d\sigma^V + \int_1 d\sigma^A \right) + \int_{n+1} \left(d\sigma^R - d\sigma^A \right)$$

- Tree level
- Virtual corrections
- Real emissions
- Subtraction terms for soft and collinear singularities

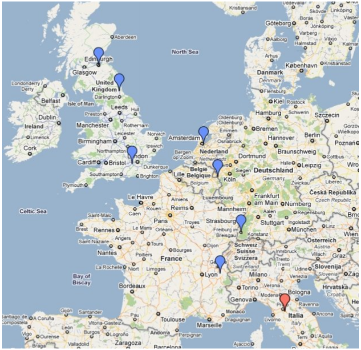
Towards automated merging of NLO with parton showers

- powheg-box, powheg-sherpa, herwig-menlops, ...

What is Golem?

GOLEM : General One Loop Evalulator of Matrix elements

- The Golem Collaboration
- The Golem Method:
  a method for evaluating one-loop Feynman diagrams
- Golem95:
  a library for one-loop integrals
- Golem-2.0:
  a one-loop matrix element generator



What is Golem?

GOLEM : General One Loop Evalulator of Matrix elements

- The Golem Collaboration
- The Golem Method:
  a method for evaluating one-loop Feynman diagrams
- Golem95:
  a library for one-loop integrals
- Golem-2.0:
  a one-loop matrix element generator

What is Golem?

GOLEM : General One Loop Evalulator of Matrix elements

- The Golem Collaboration
- The Golem Method:
  a method for evaluating one-loop Feynman diagrams
- Golem95:
  a library for one-loop integrals
- Golem-2.0:
  a one-loop matrix element generator

What is Golem?

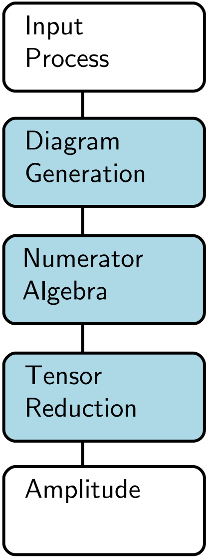
GOLEM : General One Loop Evalulator of Matrix elements

- The Golem Collaboration
- The Golem Method:
  a method for evaluating one-loop Feynman diagrams
- Golem95:
  a library for one-loop integrals
- Golem-2.0:
  a one-loop matrix element generator

The Golem Method

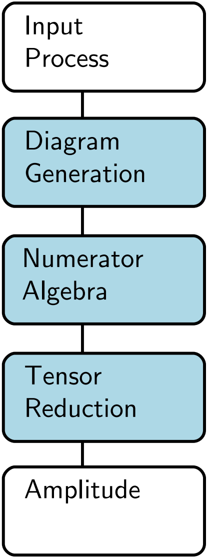
Aim: Automate the evaluation of one loop amplitudes for multi-leg and multi-scale processes within and beyond the Standard Model

Golem Method Overview



- The Golem method:
  - Feynman Diagrammatic
  - Uses Helicity projections
  - Improved tensor reduction
- and is designed for
  - $\leq$ 6 external particles
  - Massless and massive particles
  - QCD and EW corrections...
  - ...and beyond the standard model

Golem-2.0 [T.Reiter]

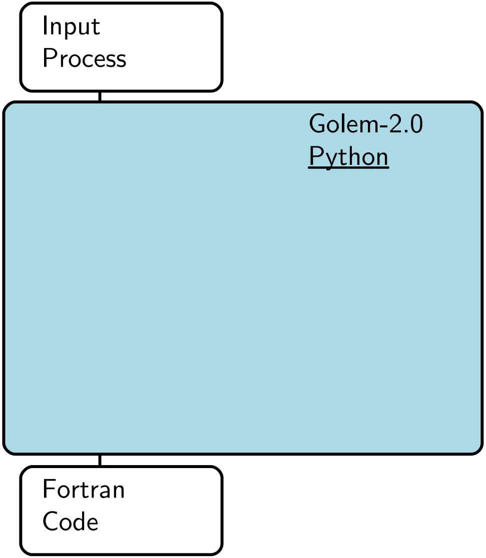


Golem-2.0: One loop matrix element generator based on Python scripts, using QGRAF, FORM and translation to golem95 form factor representation.

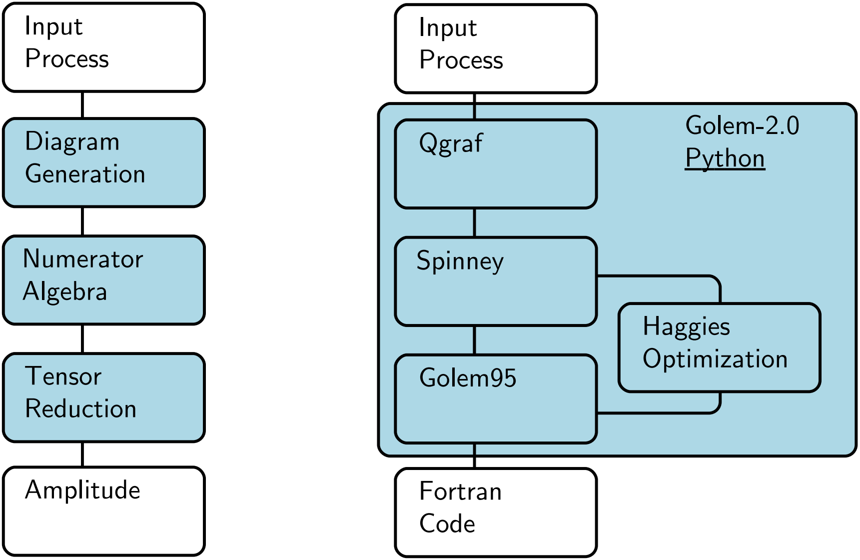
Golem-2.0 [T.Reiter]

User sees “black box“

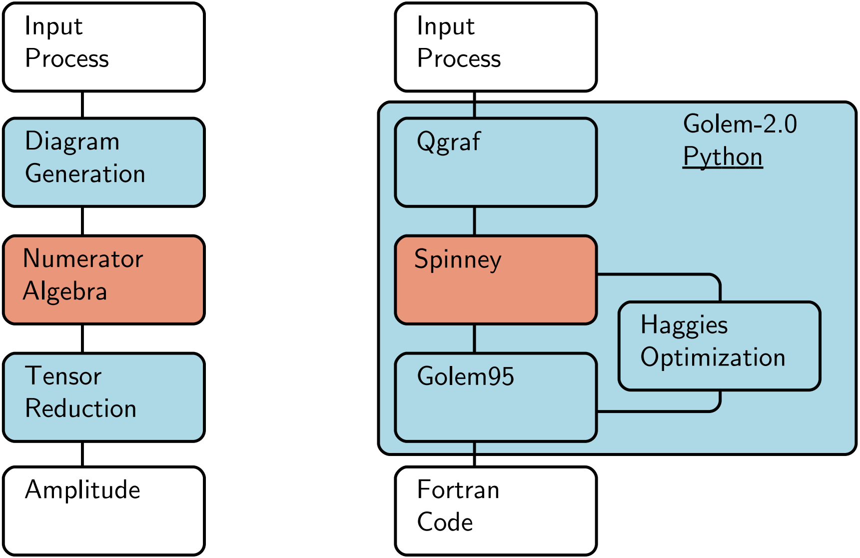
- In
  - Specify process (Process.in)
- Out
  - Optimised fortran95 code
  - process.ps



Golem-2.0 [T.Reiter]



Golem-2.0


Input Process
Diagram Generation
Numerator Algebra
Tensor Reduction
Amplitude
Input Process
Golem-2.0
Python
Qgraf
Spinney
Haggies Optimization
Golem95
Fortran Code

Spinney- A Form Library for Helicity Spinors

[GC, M. Koch-Janusz, T. Reiter]

Numerator Algebra

Form [Vermaseren] is a symbolic manipulation program

- Form can handle large intermediate expressions
- Form's language = tensors, Lorentz indices, Dirac algebra, traces

Problem:

- Many approaches (including Golem) use helicity projections
- Not implemented in Form

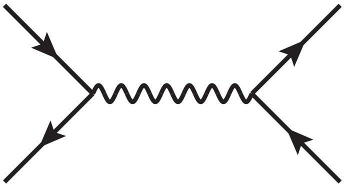
Spinney- A Form Library for Helicity Spinors

Spinney is a Form library
Uses the language of Form to

- implement helicity spinors

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$



Spinor helicity conventions

$$\Pi_+ u(p_i) = \Pi_+ v(p_i) = |i\rangle$$
$$\Pi_- u(p_i) = \Pi_- v(p_i) = |i]$$
$$\bar{u}(p_i)\Pi_- = \bar{v}(p_i)\Pi_- = [i|$$
$$\bar{u}(p_i)\Pi_+ = \bar{v}(p_i)\Pi_+ = \langle i|$$

$$\begin{aligned} A &= \frac{g_{\mu\nu}}{s}[\bar{v}(p_2)(\Pi_+ + \Pi_-)\gamma^\mu(\Pi_+ + \Pi_-)u(p_1)] \\ &\qquad [\bar{u}(p_4)(\Pi_+ + \Pi_-)\gamma^\nu(\Pi_+ + \Pi_-)v(p_3)] \\ &= A^{++++} + A^{----} + A^{++--} + A^{--++} \end{aligned}$$

$$\Rightarrow A^{++++} = \frac{g_{\mu\nu}}{s}[2|\gamma^\mu|1\rangle[4|\gamma^\nu|3\rangle$$

```
Local Amp = d4(mu,nu)*UbarSpb(k2) * Sm4(mu)
* USpa(k1) * UbarSpb(k4) * Sm4(nu) * USpa(k3);
```

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Vectors k1,k2,k3,k4;
Indices mu, nu;
#include spinney.hh
Local Amp = UbarSpb(k2) * Sm4(mu) * USpa(k1) *
UbarSpb(k4) * Sm4(mu) * USpa(k3)*d4(mu,nu);
#call SpCollect
#call SpContractMetrics
#call SpContract
#call SpOpen
print;
.end
```

Amp =UbarSpb(k2) * Sm4(mu) * USpa(k1) * UbarSpb(k4) * Sm4(nu) * USpa(k3)*d4(mu,nu)

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Vectors k1,k2,k3,k4;
Indices mu, nu;
#include spinney.hh
Local Amp = UbarSpb(k2) * Sm4(mu) * USpa(k1) *
UbarSpb(k4) * Sm4(nu) * USpa(k3)*d4(mu,nu);
#call SpCollect
#call SpContractMetrics
#call SpContract
#call SpOpen
print;
.end
```

Amp =d4(mu,nu)*Spba(k2,mu,k1)*Spba(k4,nu,k3)

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Vectors k1,k2,k3,k4;
Indices mu, nu;
#include spinney.hh
Local Amp = UbarSpb(k2) * Sm4(mu) * USpa(k1) *
UbarSpb(k4) * Sm4(nu) * USpa(k3)*d4(mu,nu);
#call SpCollect
#call SpContractMetrics
#call SpContract
#call SpOpen
print;
.end
```

Amp =Spba(k2,nu,k1)*Spba(k4,nu,k3)

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Vectors k1,k2,k3,k4;
Indices mu, nu;
#include spinney.hh
Local Amp = UbarSpb(k2) * Sm4(mu) * USpa(k1) *
UbarSpb(k4) * Sm4(nu) * USpa(k3)*d4(mu,nu);
#call SpCollect
#call SpContractMetrics
#call SpContract
#call SpOpen
print;
.end
```

Amp =2*Spaa(k3,k1)*Spbb(k2,k4)

Spinney: An Example

Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Vectors k1,k2,k3,k4;
Indices mu, nu;
#include spinney.hh
Local Amp = UbarSpb(k2) * Sm4(mu) * USpa(k1) *
UbarSpb(k4) * Sm4(nu) * USpa(k3)*d4(mu,nu);
#call SpCollect
#call SpContractMetrics
#call SpContract
#call SpOpen
print;
.end
```

Amp =- 2*Spa2(k1,k3)*Spb2(k2,k4)

Spinney: An Example

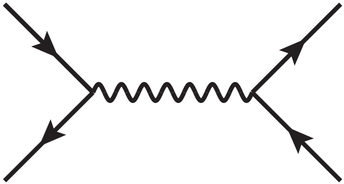
Helicity amplitude for $u\bar{u} \to d\bar{d}$

```
Amp =
      - 2*Spa2(k1,k3)*Spb2(k2,k4);
```

i.e. $A^{++++} = -2\langle 13\rangle\langle 24\rangle$

Spinney: An Example

Helicity amplitude for $u\bar{u} \to t\bar{t}$



Massive final state

$$p_I^\mu = k_i^\mu + \frac{(p_I)^2}{2p_I \cdot q} q^\mu$$

$$q^2 = 0$$

$$k^2 = 0$$

$$p_I^2 = m_I^2$$

$$|I\rangle = |i\rangle + \frac{m_I}{[iq]} |q]$$

```
#call LightConeDecomposition(p3,k4,k2,m);
```

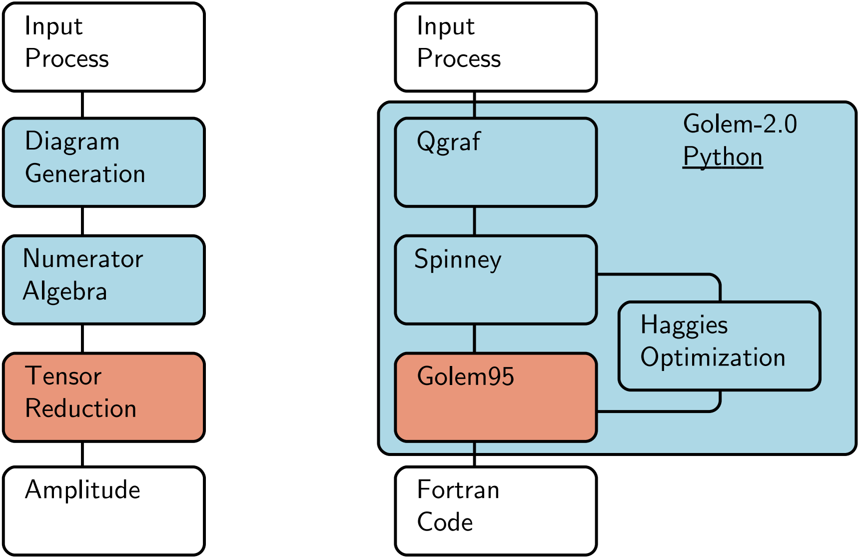
```
Amp = - 2*Spa2(k1,k4)*Spb2(k2,k3);
```

Spinney- A Form Library for Helicity Spinors

Spinney is a Form library
Uses the language of Form to

- implement helicity spinors
- massive and massless
- includes flipping rules for Majorana fermions
- includes t'Hooft-Veltman scheme for dimensional splitting
- functions and procedures named to allow easy migration to S@M [D. Maitre, P. Mastrolia, 0710.5559] $\Rightarrow$ numerical evaluation of spinor products

Golem-2.0


Input
Process
Diagram
Generation
Numerator
Algebra
Tensor
Reduction
Amplitude
Input
Process
Golem-2.0
Python
Qgraf
Spinney
Haggies
Optimization
Golem95
Fortran
Code

Golem95

[T. Binoth, GC, J.Ph. Guillet, G. Heinrich, T. Kleinschmidt, E. Pilon, T. Reiter, M. Rodgers]

One-loop amplitudes $\Rightarrow$

- Dimensionally regulated one-loop integrals

$$I_N^{d,\mu_1\cdots\mu_r}(S) = \int \frac{d^d k}{i\pi^{d/2}} \frac{k^{\mu_1}\cdots k^{\mu_r}}{\prod_{j=1}^N \left[(k+r_j)^2 - m_j^2 + i\delta\right]}$$

with $S_{ij} = (r_i - r_j)^2 - m_i^2 - m_j^2$.

- Strip away Lorentz structure $\rightarrow$ Form Factor rep.

$$\begin{aligned} I_N^{d,\mu_1\ldots\mu_r}(S) = & \sum_{j_1,\ldots,j_r} [r_{j_1}^{\cdot}\ldots r_{j_r}^{\cdot}]^{\mu_1\ldots\mu_r} A_N^r(j_1,\cdots,j_r;S) \\ & + \sum_{j_1,\ldots,j_{r-2}} [r_{j_1}^{\cdot}\ldots r_{j_{r-2}}^{\cdot} g^{\cdot\cdot}]^{\mu_1\ldots\mu_r} B_N^r(j_1,\ldots,j_{r-2};S) \\ & + \sum_{j_1,\ldots,j_{r-4}} [r_{j_1}^{\cdot}\ldots r_{j_{r-4}}^{\cdot} g^{\cdot\cdot} g^{\cdot\cdot}]^{\mu_1\ldots\mu_r} C_N^r(j_1,\ldots,j_{r-4};S) \end{aligned}$$

Golem95

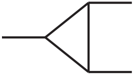
Form Factors are linear combinations of

$$I_N^d(l_1, \ldots, l_p, S) = (-1)^N \Gamma\left(N - \frac{d}{2}\right) \int d^N z \frac{\delta(1 - \sum z_j) z_{l_1} \ldots z_{l_p}}{\left[-\frac{1}{2} z^T S z - i\delta\right]^{N-d/2}}$$

- Reduce to scalar integrals
- can introduce dangerous inverse gram determinants for N=3,4
- if det G small Golem95 $\Rightarrow$ avoids this step, instead completes numerical one-dimensional integration

Golem95: Simple Example

3-point, rank 2



$$I_3^{\mu\nu}(S) = \int d\bar{k} \frac{k^\mu k^\nu}{[(k+r_1)^2 - m_1^2][(k+r_2)^2 - m_2^2][k^2 - m_3^2]}$$
$$= r_1^\mu r_1^\nu A_{1,1}^{3,2}(S) + r_1^\mu r_2^\nu A_{1,2}^{3,2}(S) + r_2^\mu r_1^\nu A_{2,1}^{3,2}(S) + r_2^\mu r_2^\nu A_{2,2}^{3,2}(S)$$
$$+ g^{\mu\nu} B^{3,2}(S)$$

and

$$A_{i,j}^{3,2}(S) = I_3^n(i,j,S) \sim \frac{1}{(detG)^2} I_3^n(S) \quad B^{3,2}(S) = -\frac{1}{2} I_3^{n+2}(S)$$

For N=3,4:

- (N=3) Infra-red divergent $\rightarrow$ explicit expressions
- det G small $\rightarrow$ one-dimensional numerical integration (only for massless propagtors so far)
- otherwise: reduce to scalar integrals

Golem95

Dedicated Fortran 95 library for the reduction and evaluation of tensor integrals

Latest version 1.1.1 available online

http://lappweb.in2p3.fr/lapth/Golem/golem95.html

including:

- Inclusion of internal masses (Internal call to OneLOop [A. van Hameren] for finite massive scalar box/triangle)
- Scale μ has been added
- Contains all tensor coefficients up to rank six, six point integrals for massive and massless integrals (IR/ UV divergent and finite)
- Can also be used as a library for all types of scalar integrals

Future plans:

- completion of numerical option for all types of integrals
- complex masses

Golem-2.0: Summary and Outlook

New features:

- Can handle massive loops
- Implementation of Majorana Fermions and higher spins $\Rightarrow$ BSM
- import of CalcHep Feynman Rules
- interface to SAMURAI (unitary based) [Mastrolia, Ossola, Reiter, Tramontano, hep-ph 1006.0710]

In progress:

- Check of MSSM model file
- FeynRules model files [C. Duhr et al]
- Les Houches interface
- PowHeg-Box interface [Alioli,Nason,Oleari,Reiter]
- user-friendly "black box" with detailed documentation

Aim: Public and open source: after validation of $gg \to b\bar{b}b\bar{b}$

Golem Results

Golem method has been used for

- $\gamma\gamma \to 4\gamma$ [Binoth, Gehrmann, Heinrich, Mastrolia]
- $gg \to W^*W^* \to l\nu l'\nu$ [Binoth,Ciccolini,Kauer,Krämer]
- $gg \to HH, HHH$ [Binoth,Karg,Kauer,Rückl]
- $pp \to Hjj$ (VBF/GF) [Andersen,Binoth,Heinrich,Smillie]
- $q\bar{q} \to b\bar{b}b\bar{b}$ [Binoth,Greiner,Guffanti,Guillet,Reiter,Reuter]
- $pp \to VVj$ [Binoth,Gleisberg,Karg,Kauer,Sanguinetti]
- $pp \to$ Graviton $+j$ [Karg et al.]
- $gg \to b\bar{b}b\bar{b}$ (in progress)

Golem Results

Golem method has been used for

- $\gamma\gamma \to 4\gamma$ [Binoth, Gehrmann, Heinrich, Mastrolia]
- $gg \to W^* W^* \to l\nu l' \nu$ [Binoth,Ciccolini,Kauer,Krämer]
- $gg \to HH, HHH$ [Binoth,Karg,Kauer,Rückl]
- $pp \to Hjj$ (VBF/GF) [Andersen,Binoth,Heinrich,Smillie]
- $q\bar{q} \to b\bar{b}b\bar{b}$ [Binoth,Greiner,Guffanti,Guillet,Reiter,Reuter]
- $pp \to VVj$ [Binoth,Gleisberg,Karg,Kauer,Sanguinetti]
- $pp \to$ Graviton $+j$ [Karg et al.]
- $gg \to b\bar{b}b\bar{b}$ (in progress)

The four b amplitude

[T. Binoth, N. Greiner, A. Guffanti, T.Reiter, J. Reuter]

An important background for BSM Higgs searches:
For example: in MSSM: at large $\tan\beta$ the $Hb\bar{b}$ coupling is enhanced

- Approximations: $m_b = 0$, $m_t \to \infty$, $q \in \{u, d, s, c\}$
- LHC kinematics and cuts
  - $\sqrt{s} = 14\text{TeV}$
  - p_T cut: $p_T > 30$ GeV
  - rapidity cut: $|\eta| < 2.5$
  - separation cut: $\Delta R > 0.8$

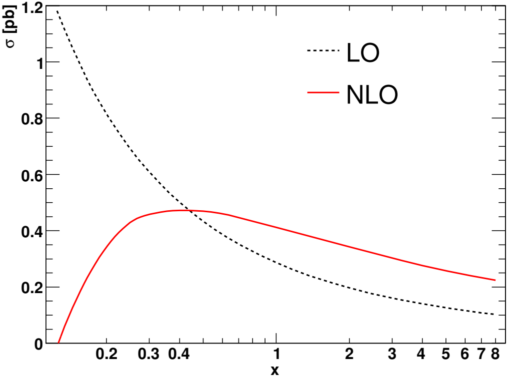
The process $q\bar{q} \to b\bar{b}b\bar{b}$

Method:

- virtual corrections: Golem-2.0
- born part: Madgraph [F. Maltoni, T. Steltzer]
- real corrections: MadGraph
- subtraction terms: MadDipole [R. Frederix, T. Gehrmann, N. Greiner]
- integration/analysis (MadEvent [Maltoni, Stelzer])
- "plug and play" single subroutine call from Madevent to Golem

$q\bar{q} \to b\bar{b}b\bar{b}$: Results

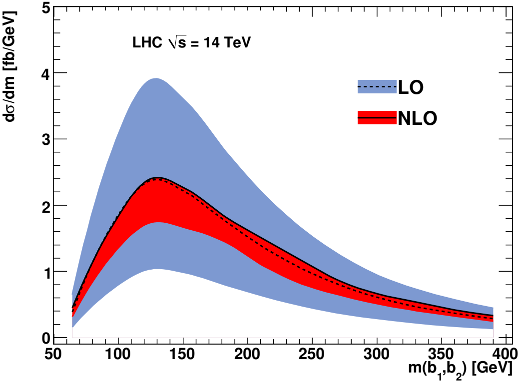
$$\mu_R = x\mu_0;\ \mu_0 = \sqrt{\sum_{j=1}^{4} |p_T(b_j)|^2}$$



- reduction of scale dependence
- stabilization of result
- study of dependence on μ_F after all channels computed
- the error bands $\mu_0 < \mu_R < 2\mu_0$

$q\bar{q} \to b\bar{b}b\bar{b}$: Results

m_{bb} of leading b-jets



- reduction of scale dependence
- stabilization of result
- study of dependence on μ_F after all channels computed
- the error bands $\mu_0 < \mu_R < 2\mu_0$

Conclusions and outlook

- high precision = beyond leading order
- Golem
  - Golem is designed for automated one-loop calculations
  - Numerically safe (avoids inverse Gram determinants)
  - massive and massless particles
  - Golem95- tensor integral library available at http://lappweb.in2p3.fr/lapth/Golem/golem95.html
  - Spinney- Form Library for Helicity Spinors available at http://www.nikhef.nl/~thomasr/
  - Golem techniques being used for processes beyond the Standard Model
  - Golem-2.0- matrix element generator public soon