The GOLEM Project

T. Reiter (Nikhef)

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

MC4LHC readiness, 29 March – 01 April 2010

Overview

The GOLEM Method

golem95 One-Loop Integral Library

Golem-2.0 Virtual Matrix Element Generator

Results

GOLEM Readiness

The GOLEM Method: Overview

GOLEM: General One-Loop Evaluator for Matrix Elements

- GOLEM = a method for evaluating one-loop Feynman diagrams
- GOLEM = a library for one-loop integrals (golem95)
- GOLEM = a matrix element generator at the one-loop level

Why Feynman Diagrams?

- No distinction between cut-constructible and rational part
  $\Rightarrow$ conceptually simple
- Gram determinant problem avoidable by dedicated tensor reduction ($\Rightarrow$ golem95)
- Combinatorial complexity of Feynman diagrams
  $\Rightarrow$ problematic only beyond $2 \to 4$

The GOLEM Method: Overview

GOLEM: General One-Loop Evaluator for Matrix Elements

- GOLEM = a method for evaluating one-loop Feynman diagrams
- GOLEM = a library for one-loop integrals (golem95)
- GOLEM = a matrix element generator at the one-loop level

Why Feynman Diagrams?

- No distinction between cut-constructible and rational part
  $\Rightarrow$ conceptually simple
- Gram determinant problem avoidable by dedicated tensor reduction ($\Rightarrow$ golem95)
- Combinatorial complexity of Feynman diagrams
  $\Rightarrow$ problematic only beyond $2 \to 4$

The GOLEM Method: Overview

GOLEM: General One-Loop Evaluator for Matrix Elements

- GOLEM = a method for evaluating one-loop Feynman diagrams
- GOLEM = a library for one-loop integrals (golem95)
- GOLEM = a matrix element generator at the one-loop level

Why Feynman Diagrams?

- No distinction between cut-constructible and rational part
  $\Rightarrow$ conceptually simple
- Gram determinant problem avoidable by dedicated tensor reduction ($\Rightarrow$ golem95)
- Combinatorial complexity of Feynman diagrams
  $\Rightarrow$ problematic only beyond $2 \to 4$

The GOLEM Method: Overview

The GOLEM method uses

- Feynman diagrams
- Helicity projections
- Improved tensor reduction

The GOLEM method is designed for

- any number of ext. particles ($\lesssim 6$ feasable)
- massless and massive particles
- QCD and EW corrections
- physics within and beyond the Standard Model

The GOLEM method is aiming at

- NLO "Plug In" for MC generators
  $\rightarrow$ see also Rikkert's talk

The GOLEM Method: Overview

The GOLEM method uses

- Feynman diagrams
- Helicity projections
- Improved tensor reduction

The GOLEM method is designed for

- any number of ext. particles ($\lesssim 6$ feasable)
- massless and massive particles
- QCD and EW corrections
- physics within and beyond the Standard Model

The GOLEM method is aiming at

- NLO "Plug In" for MC generators
  $\longrightarrow$ see also Rikkert's talk

The GOLEM Method: Overview

The GOLEM method uses

- Feynman diagrams
- Helicity projections
- Improved tensor reduction

The GOLEM method is designed for

- any number of ext. particles ($\lesssim 6$ feasable)
- massless and massive particles
- QCD and EW corrections
- physics within and beyond the Standard Model

The GOLEM method is aiming at

- NLO "Plug In" for MC generators
  $\rightarrow$ see also Rikkert's talk

The GOLEM Method: Implementation

$$\mathcal{A}^{\{\lambda\}}(\{p_j\};\{m_j\}) = \sum_{\{c_i\},\alpha} f^{\{c_i\}}\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\})$$

$$\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\}) = \int \frac{\mathrm{d}^n k}{i\pi^{n/2}} \frac{\mathcal{N}^{\{\lambda\}}(k)}{D_1\cdots D_N}$$
$$= \sum_r \mathcal{N}^{\{\lambda\}}_{\mu_1\ldots\mu_r}(\{p_j\};\{m_j\}) \cdot I_N^{n\,\mu_1\ldots\mu_r}(\{p_j\};\{m_j\})$$

Idea: Split implementation into two steps

1. Numerically stable reduction of tensor integrals
2. Matrix element generator for one-loop amplitudes

The GOLEM Method: Implementation

$$\mathcal{A}^{\{\lambda\}}(\{p_j\};\{m_j\}) = \sum_{\{c_i\},\alpha} f^{\{c_i\}}\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\})$$

$$\begin{aligned}\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\}) &= \int\frac{\mathrm{d}^n k}{i\pi^{n/2}}\frac{\mathcal{N}^{\{\lambda\}}(k)}{D_1\cdots D_N}\\ &= \sum_r \mathcal{N}^{\{\lambda\}}_{\mu_1\ldots\mu_r}(\{p_j\};\{m_j\})\cdot I_N^{n\,\mu_1\ldots\mu_r}(\{p_j\};\{m_j\})\end{aligned}$$

Idea: Split implementation into two steps

1. Numerically stable reduction of tensor integrals
2. Matrix element generator for one-loop amplitudes

The GOLEM Method: Implementation

$$\mathcal{A}^{\{\lambda\}}(\{p_j\};\{m_j\}) = \sum_{\{c_i\},\alpha} f^{\{c_i\}} \mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\})$$

$$\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\}) = \int \frac{\mathrm{d}^n k}{i\pi^{n/2}} \frac{\mathcal{N}^{\{\lambda\}}(k)}{D_1 \cdots D_N}$$

$$= \sum_r \mathcal{N}^{\{\lambda\}}_{\mu_1\ldots\mu_r}(\{p_j\};\{m_j\}) \cdot I_N^{n\,\mu_1\ldots\mu_r}(\{p_j\};\{m_j\})$$

Idea: Split implementation into two steps

1. Numerically stable reduction of tensor integrals
2. Matrix element generator for one-loop amplitudes

The GOLEM Method: Implementation

$$\mathcal{A}^{\{\lambda\}}(\{p_j\};\{m_j\}) = \sum_{\{c_i\},\alpha} f^{\{c_i\}} \mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\})$$

$$\begin{aligned}\mathcal{G}_\alpha^{\{\lambda\}}(\{p_j\};\{m_j\}) &= \int \frac{\mathrm{d}^n k}{i\pi^{n/2}} \frac{\mathcal{N}^{\{\lambda\}}(k)}{D_1\cdots D_N} \\ &= \sum_r \mathcal{N}^{\{\lambda\}}_{\mu_1\ldots\mu_r}(\{p_j\};\{m_j\}) \cdot I_N^{n\,\mu_1\ldots\mu_r}(\{p_j\};\{m_j\})\end{aligned}$$

Idea: Split implementation into two steps

1. Numerically stable reduction of tensor integrals
2. Matrix element generator for one-loop amplitudes

$$I_N^{n\,\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S)=\int\frac{\mathrm{d}^nk}{i\pi^{n/2}}\frac{q_{a_1}^{\mu_1}\cdots q_{a_r}^{\mu_r}}{\prod_{j\in S}(q_j^2-m_j^2+i\delta)}$$

$$S_{ij}=(q_i-q_j)^2-m_i^2-m_j^2,\quad q_j=k+r_j$$

Decomposition into Lorentz invariant integrals:

$$I_N^{n\,\mu_1\ldots\mu_r}=(-1)^N\Gamma(N-n/2)\sum_{p,j_1,\ldots,j_p}T^{\mu_1\cdots\mu_r}(\{r_j\},g^{\cdot\cdot})\times$$
$$\int\mathrm{d}z_1\cdots\mathrm{d}z_N\delta(1-z_1-\ldots z_N)\frac{z_{l_1}\ldots z_{l_p}}{(-1/2z^TSz-i\delta)^{N-n/2}}$$

- Can be reduced further $\to I_{N-1}^n+I_N^{n+2}$
- Can be evaluated numerically (degenerated kinematics)

$\Rightarrow$ Gram determinants can be avoided

golem95 One-Loop Integral Library

$$I_N^{n\,\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S)=\int\frac{\mathrm{d}^n k}{i\pi^{n/2}}\frac{q_{a_1}^{\mu_1}\cdots q_{a_r}^{\mu_r}}{\prod_{j\in S}(q_j^2-m_j^2+i\delta)}$$

$$S_{ij}=(q_i-q_j)^2-m_i^2-m_j^2,\quad q_j=k+r_j$$

Decomposition into Lorentz invariant integrals:

$$I_N^{n\,\mu_1\ldots\mu_r}=(-1)^N\Gamma(N-n/2)\sum_{p,j_1,\ldots,j_p}T^{\mu_1\cdots\mu_r}(\{r_j\},g^{\cdot\cdot})\times$$

$$\int \mathrm{d}z_1\cdots\mathrm{d}z_N\delta(1-z_1-\ldots z_N)\frac{z_{l_1}\ldots z_{l_p}}{(-1/2z^TSz-i\delta)^{N-n/2}}$$

- Can be reduced further $\rightarrow I_{N-1}^n+I_N^{n+2}$
- Can be evaluated numerically (degenerated kinematics)

$\Rightarrow$ Gram determinants can be avoided

Reminder: PV reduction and Gram determinants

$$\int \frac{\mathrm{d}^n k}{(2\pi)^n} \frac{p \cdot k}{(k+r_1)^2(k+r_2)^2(k+r_3)^2 k^2}$$

- If $\{r_1, r_2, r_3\}$ linearly independent:

$$p^\mu = \alpha_1 r_1^\mu + \alpha_2 r_2^\mu + \alpha_3 r_3^\mu + \alpha_\perp \epsilon^{\mu\nu\rho\sigma} r_{1\nu} r_{2\rho} r_{3\sigma}$$

- Since $2r_i \cdot k = (k+r_i)^2 - k^2 - r_i^2$
  $\Rightarrow$ decomposition into scalar integrals
- Need to solve: (which introduces Gram determinant)

$$\left(\begin{array}{ccc|c} r_1 \cdot r_1 & r_1 \cdot r_2 & r_1 \cdot r_3 & 0 \\ r_2 \cdot r_1 & r_2 \cdot r_2 & r_2 \cdot r_3 & 0 \\ r_3 \cdot r_1 & r_3 \cdot r_2 & r_3 \cdot r_3 & 0 \\ \hline 0 & 0 & 0 & \det G \end{array}\right) \cdot \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \hline \alpha_\perp \end{array}\right) = \left(\begin{array}{c} p \cdot r_1 \\ p \cdot r_2 \\ p \cdot r_3 \\ \hline \epsilon^{p r_1 r_2 r_3} \end{array}\right)$$

Reminder: PV reduction and Gram determinants

$$\int \frac{\mathrm{d}^n k}{(2\pi)^n} \frac{p \cdot k}{(k+r_1)^2(k+r_2)^2(k+r_3)^2 k^2}$$

- If $\{r_1, r_2, r_3\}$ linearly independent:

$$p^\mu = \alpha_1 r_1^\mu + \alpha_2 r_2^\mu + \alpha_3 r_3^\mu + \alpha_\perp \epsilon^{\mu\nu\rho\sigma} r_{1\nu} r_{2\rho} r_{3\sigma}$$

- Since $2r_i \cdot k = (k+r_i)^2 - k^2 - r_i^2$
  $\Rightarrow$ decomposition into scalar integrals
- Need to solve: (which introduces Gram determinant)

$$\left(\begin{array}{ccc|c} r_1\cdot r_1 & r_1\cdot r_2 & r_1\cdot r_3 & 0 \\ r_2\cdot r_1 & r_2\cdot r_2 & r_2\cdot r_3 & 0 \\ r_3\cdot r_1 & r_3\cdot r_2 & r_3\cdot r_3 & 0 \\ \hline 0 & 0 & 0 & \det G \end{array}\right) \cdot \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \hline \alpha_\perp \end{array}\right) = \left(\begin{array}{c} p\cdot r_1 \\ p\cdot r_2 \\ p\cdot r_3 \\ \hline \epsilon^{p r_1 r_2 r_3} \end{array}\right)$$

Reminder: PV reduction and Gram determinants

$$\int \frac{\mathrm{d}^n k}{(2\pi)^n} \frac{p \cdot k}{(k+r_1)^2(k+r_2)^2(k+r_3)^2 k^2}$$

- If $\{r_1, r_2, r_3\}$ linearly independent:

$$p^\mu = \alpha_1 r_1^\mu + \alpha_2 r_2^\mu + \alpha_3 r_3^\mu + \alpha_\perp \epsilon^{\mu\nu\rho\sigma} r_{1\nu} r_{2\rho} r_{3\sigma}$$

- Since $2r_i \cdot k = (k+r_i)^2 - k^2 - r_i^2$
  $\Rightarrow$ decomposition into scalar integrals
- Need to solve: (which introduces Gram determinant)

$$\left(\begin{array}{ccc|c} r_1\cdot r_1 & r_1\cdot r_2 & r_1\cdot r_3 & 0 \\ r_2\cdot r_1 & r_2\cdot r_2 & r_2\cdot r_3 & 0 \\ r_3\cdot r_1 & r_3\cdot r_2 & r_3\cdot r_3 & 0 \\ \hline 0 & 0 & 0 & \det G \end{array}\right) \cdot \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \hline \alpha_\perp \end{array}\right) = \left(\begin{array}{c} p\cdot r_1 \\ p\cdot r_2 \\ p\cdot r_3 \\ \hline \epsilon^{p r_1 r_2 r_3} \end{array}\right)$$

Reminder: PV reduction and Gram determinants

$$\int \frac{\mathrm{d}^n k}{(2\pi)^n} \frac{p \cdot k}{(k+r_1)^2 (k+r_2)^2 (k+r_3)^2 k^2}$$

- If $\{r_1, r_2, r_3\}$ linearly independent:

$$p^\mu = \alpha_1 r_1^\mu + \alpha_2 r_2^\mu + \alpha_3 r_3^\mu + \alpha_\perp \epsilon^{\mu\nu\rho\sigma} r_{1\nu} r_{2\rho} r_{3\sigma}$$

- Since $2r_i \cdot k = (k+r_i)^2 - k^2 - r_i^2$
  $\Rightarrow$ decomposition into scalar integrals
- Need to solve: (which introduces Gram determinant)

$$\left(\begin{array}{ccc|c} r_1 \cdot r_1 & r_1 \cdot r_2 & r_1 \cdot r_3 & 0 \\ r_2 \cdot r_1 & r_2 \cdot r_2 & r_2 \cdot r_3 & 0 \\ r_3 \cdot r_1 & r_3 \cdot r_2 & r_3 \cdot r_3 & 0 \\ \hline 0 & 0 & 0 & \det G \end{array}\right) \cdot \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \hline \alpha_\perp \end{array}\right) = \left(\begin{array}{c} p \cdot r_1 \\ p \cdot r_2 \\ p \cdot r_3 \\ \hline \epsilon^{p r_1 r_2 r_3} \end{array}\right)$$

Reminder: PV reduction and Gram determinants

$$\int \frac{\mathrm{d}^n k}{(2\pi)^n} \frac{p \cdot k}{(k+r_1)^2(k+r_2)^2(k+r_3)^2 k^2}$$

- If $\{r_1, r_2, r_3\}$ linearly independent:

$$p^\mu = \alpha_1 r_1^\mu + \alpha_2 r_2^\mu + \alpha_3 r_3^\mu + \alpha_\perp \epsilon^{\mu\nu\rho\sigma} r_{1\nu} r_{2\rho} r_{3\sigma}$$

- Since $2r_i \cdot k = (k+r_i)^2 - k^2 - r_i^2$
  $\Rightarrow$ decomposition into scalar integrals
- Need to solve: (which introduces Gram determinant)

$$\left(\begin{array}{ccc|c} r_1\cdot r_1 & r_1\cdot r_2 & r_1\cdot r_3 & 0 \\ r_2\cdot r_1 & r_2\cdot r_2 & r_2\cdot r_3 & 0 \\ r_3\cdot r_1 & r_3\cdot r_2 & r_3\cdot r_3 & 0 \\ \hline 0 & 0 & 0 & \det G \end{array}\right) \cdot \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \hline \alpha_\perp \end{array}\right) = \left(\begin{array}{c} p\cdot r_1 \\ p\cdot r_2 \\ p\cdot r_3 \\ \hline \epsilon^{p r_1 r_2 r_3} \end{array}\right)$$

golem95 One-Loop Integral Library

Current version of golem95

- http://lappweb.in2p3.fr/lapth/Golem/golem95.html
- algebraic separation of IR poles
- cache avoiding multiple evaluation
- all required integrals for $N \leq 6$, massless
- documentation, examples available

Under development

- version with propagator masses
- currently: finite box (D_0) by call to LoopTools [T. Hahn]

Early stage of development

- Complex propagator masses

golem95 One-Loop Integral Library

Current version of golem95

- http://lappweb.in2p3.fr/lapth/Golem/golem95.html
- algebraic separation of IR poles
- cache avoiding multiple evaluation
- all required integrals for $N \leq 6$, massless
- documentation, examples available

Under development

- version with propagator masses
- currently: finite box (D_0) by call to LoopTools [T. Hahn]

Early stage of development

- Complex propagator masses

golem95 One-Loop Integral Library

Current version of golem95

- http://lappweb.in2p3.fr/lapth/Golem/golem95.html
- algebraic separation of IR poles
- cache avoiding multiple evaluation
- all required integrals for $N \leq 6$, massless
- documentation, examples available

Under development

- version with propagator masses
- currently: finite box (D_0) by call to LoopTools [T. Hahn]

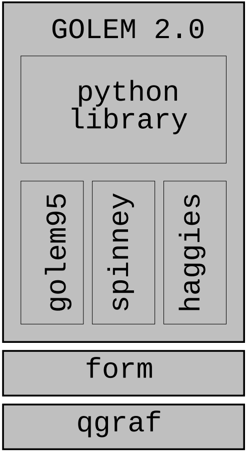
Early stage of development

- Complex propagator masses

GOLEM-2.0: One-Loop Matrix Element Generator

Overview

- implementation of the Golem method
- very modular
  - python library (command line tools)
  - spinney: helicity spinors in Form
  - haggies: optimizing code generator
  - golem95: integral library
- based on Form and QGraf [Vermaseren;Nogueira]
- Fortran 95 matrix element code



- create configuration file
- enter process, here: $gg \longrightarrow s\bar{s}b\bar{b}$ @ NLO in QCD
- set up process directory
- generate code and draw diagrams

shell

```
$ golem-main.py --template process.in
$
```

GOLEM-2.0: Matrix Elements Made Easy

- create configuration file
- enter process, here: $gg \longrightarrow s\bar{s}b\bar{b}$ @ NLO in QCD
- set up process directory
- generate code and draw diagrams

editor: process.in

```
process_path=<a directory>
in=g,g
out=s,s~,b,b~
order=gs,4,6
model=sm

# more settings optional
...
```

GOLEM-2.0: Matrix Elements Made Easy

- create configuration file
- enter process, here: $gg \longrightarrow s\bar{s}b\bar{b}$ @ NLO in QCD
- set up process directory
- generate code and draw diagrams

shell

```
$ golem-main.py --template process.in
$ edit process.in
$ golem-main.py process.in
$
```

- create configuration file
- enter process, here: $gg \to s\bar{s}b\bar{b}$ @ NLO in QCD
- set up process directory
- generate code and draw diagrams

shell

```
$ golem-main.py --template process.in
$ edit process.in
$ golem-main.py process.in
$ make dist # -> matrix.tar.gz
$
```

GOLEM-2.0: Matrix Elements Made Easy

- create configuration file
- enter process, here: $gg \longrightarrow s\bar{s}b\bar{b}$ @ NLO in QCD
- set up process directory
- generate code and draw diagrams

shell

```
$ golem-main.py --template process.in
$ edit process.in
$ golem-main.py process.in
$ make dist # -> matrix.tar.gz
$ make doc # -> process.ps
$
```

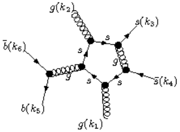


Diagram 1109

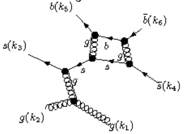


Diagram 1111

golem-2.0: Work in Progress

Features not fully implemented (but planned/in progress):

- Les Houches interface
- FeynRules import [C. Duhr]
- Renormalisation of massive theories

Implemented but not fully tested:

- Majorana fermions and higher spins
- massive processes

+ improvements considering size, speed and interface

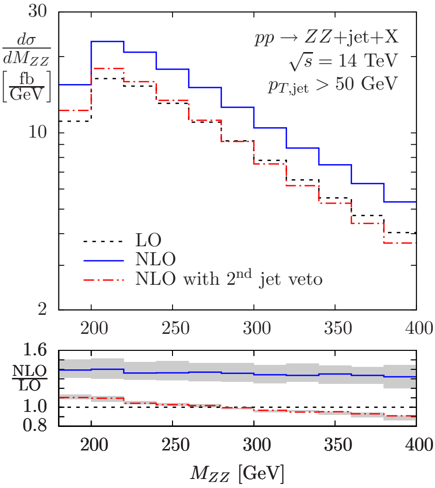
Some recent results

GOLEM method has been used for

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

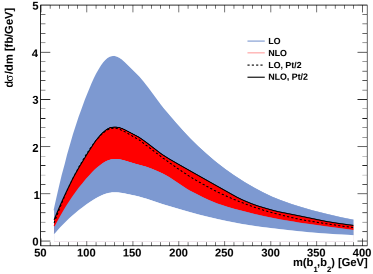
- high priority "wishlist" process
- algebraic reduction of tensor integrals
- at most pentagon diagrams
- successful comparison with

[Dittmaier,Kallweit,Uwer]



$q\bar{q} \rightarrow b\bar{b}b\bar{b}$ [Binoth,Greiner,Guffanti,Guillet,TR,Reuter]

- added to "wishlist" 2007
- background to BSM Higgs search
- calculation using `golem-2.0` and `golem95`
- $gg \rightarrow b\bar{b}b\bar{b}$ missing
  $\Rightarrow$ to be completed $\approx$ June



GOLEM Readiness

golem95

massless version

real propagator masses (due: May)

complex propagator masses

golem-2.0

massless processes

massive processes

LanHEP interface

Majorana fermions

FeynRules interface

Les Houches interface

Automatic renormalisation

- To appear in first release this summer

GOLEM Readiness

golem95

- massless version
- real propagator masses (due: May)
- complex propagator masses

golem-2.0

- massless processes
- massive processes
- LanHEP interface
- Majorana fermions
- FeynRules interface
- Les Houches interface
- Automatic renormalisation

▸ To appear in first release this summer

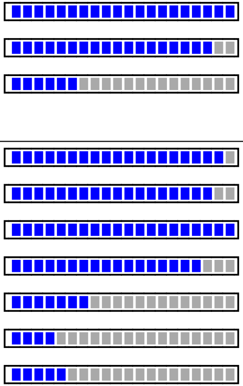
GOLEM Readiness

golem95

- massless version
- real propagator masses (due: May)
- complex propagator masses

golem-2.0

- massless processes
- massive processes
- LanHEP interface
- Majorana fermions
- FeynRules interface
- Les Houches interface
- Automatic renormalisation



▶ To appear in first release this summer