Automation of One-Loop Calculations with Golem/Samurai

Giovanni Ossola

New York City College of Technology
City University of New York (CUNY)

In collaboration with:
G. Cullen, **N. Greiner**, **G. Heinrich**, **G. Luisoni**,
P. Mastrolia, **T. Reiter**, **F. Tramontano**

LoopFest X
Radiative Corrections for the LHC and future colliders
Northwestern University – May 12-14, 2011

Automation $\rightarrow$ the use of methods for controlling processes automatically, often reducing manpower

Virtues of Automation (for Calculations of Scattering Amplitudes):

- Optimization/Self-organization
- Avoid human mistakes
- Process-independent techniques

This is not a new idea $\rightarrow$ fully exploited at the tree-level

Automation at One-Loop

At the One-Loop level, amazing work has been done towards Automation:

- "Proofs of concept": **HELAC-NLO**, **MadLoop**

  Helac-nlo : Bevilacqua, Czakon, van Hameren, Papadopoulos, Pittau, Worek

  MadLoop: Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau

  - Upgrade of tree-level tools combined with integrand level reduction
  - Fully integrated with real radiation and subtraction terms to produce finite results

- There is also an **"algebraic way"** to Automation

  FeynArts/FormCalc/LoopTools: T. Hahn

  - Generate unintegrated amplitudes with Feynman diagrams
  - Manipulate and simplify them
  - Perform the reduction

This is the target of Golem/Samurai!

Automation at One-Loop

There are several other approaches to multi-leg NLO

- New results for very challenging processes
- They all involve automation
- They are more focused on a class of processes

Bredenstein, Denner, Dittmaier, Kallweit, Pozzorini

Blackhat: Berger, Bern, Dixon, Febres Cordero, Forde, Gleisberg, Ita, Kosower, Maitre

Rocket: Ellis, Melia, Melnikov, Rontsch, Zanderighi

VBFNLO: Zeppenfeld, Bozzi, Campanario, Englert, Hankele, Jaeger, Worek

Automation at One-Loop: "Algebraic Way"

Main features of the "Algebraic Way":

- Amplitudes generated with Feynman diagrams
- Algebraic manipulations are allowed before starting the numerical integration
- The generation of numerators is executed separately from the numerical reduction
- Optimization: grouping of diagrams, smart caching
- Control over sub-parts of the computation (move in/out subsets of diagrams)
- Algebra in dimension d, different schemes

Great flexibility in the reduction
Choice between different algorithms at runtime

Outline: Golem/Samurai

General **O**ne-**L**oop **E**valuator for **M**atrix elements
and
Scattering **AM**plitudes from **U**nitarity-based
Reduction **A**lgorithm at the **I**ntegrand-level

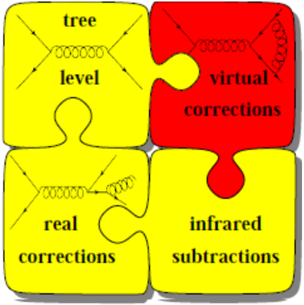
Golem/Samurai

Algebraic generation of **d-dimensional integrands** via Feynman diagrams

Reduction at the Integrand Level: **d-dimensional extension of OPP**

- Why Golem/Samurai?
- Brief Description of Samurai and Golem-2.0
- How can we Test our Numerical Results?
- Automated Calculations with Golem/Samurai
- Some examples

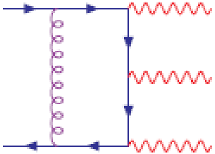
NLO CALCULATIONS



- LO tree-level $2 \to N$
- NLO virtual $2 \to N$
- NLO real $2 \to N+1$
- Subtraction terms

I will focus on the **virtual corrections**

real corrections and subtraction terms
see talk of **Nicolas Greiner**



Any m-point one-loop amplitude can be written, before integration, as

$$A(\bar{q}) = \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{m-1}}$$

where

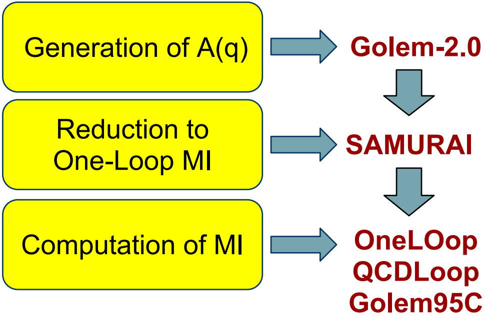
$$\bar{D}_i = (\bar{q} + p_i)^2 - m_i^2 \quad , \quad \bar{q}^2 = q^2 - \mu^2 \quad , \quad \bar{D}_i = D_i - \mu^2$$

Our task is to calculate, for each phase space point:

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

Standard 3-Step Approach

Standard Golem/Samurai reduction:



This process is **fully automated**

Diagram Generation: Golem-2.0

Cullen, **Greiner**, **Heinrich**, **Luisoni**, **Mastrolia**, **G.O.**, **Reiter**, **Tramontano**

An automated amplitude generation based on Feynman diagrams
(distributed as a python package)

- FORM
  J.A.M. Vermaseren, (1991)
- QGRAF
  P. Nogueira, (1993)
- Haggies
  T. Reiter, (2009)
- Spinney
  Cullen, Koch-Janusz, Reiter, (2010)

Not released yet!

A walk through Golem-2.0

We use as example: $u\,\bar{d} \rightarrow \bar{s}\,c\,e^-\,\bar{\nu}_e\,\mu^+\,\nu_\mu$

- Preparation of the "card"
- Building the code
- Execution , Runtime Options

```
### in (comma separated list) ##########################################
#    A comma-separated list of initial state particles.                #
#    Which particle names are valid depends on the                     #
#    model file in use.                                                #
#                                                                      #
#    Examples (Standard Model):                                        #
#    1) in=u,u~                                                        #
#    2) in=e+,e-                                                       #
#    3) in=g,g                                                         #
#                                                                      #
########################################################################
in= u,d~

### out (comma separated list) #########################################
#    A comma-separated list of final state particles.                  #
#    Which particle names are valid depends on the                     #
#    model file in use.                                                #
#                                                                      #
#    Examples (Standard Model):                                        #
#    1) out=H,u,u~                                                     #
#    2) out=e+,e-,gamma                                                #
#    3) out=b,b~,t,t~                                                  #
#                                                                      #
########################################################################
out= nmu,mu+,e-,ne~,s~,c
```

A walk through Golem-2.0

- Preparation of the "card"

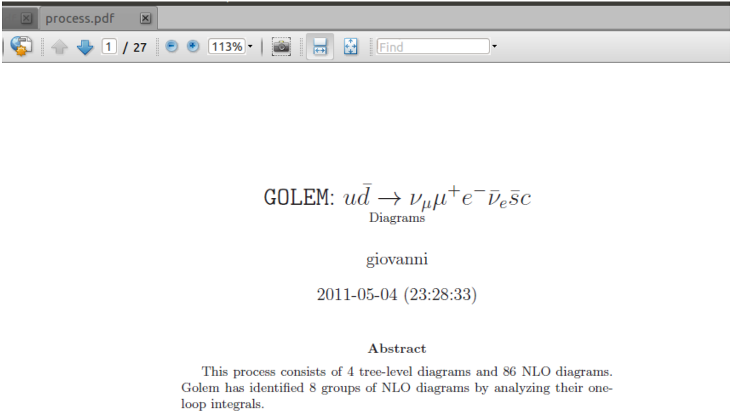
```
in= u,d~
out= nmu, mu+, e-, ne~, s~, c
model=smdiag
```

models can be added via **FeynRules** (Duhr) or **LanHEP** (Semenov)

```
order=gw,4,4; order=gs,2,4
zero=mB,mC,mS,mU,mD,me,mmu
one=gs,e
helicities=-+-+-++-
extensions=samurai, dred
```

A walk through Golem-2.0

- Building the code : check the details before the run



GOLEM: $u\bar{d} \to \nu_\mu \mu^+ e^- \bar{\nu}_e \bar{s} c$

Diagrams

giovanni

2011-05-04 (23:28:33)

Abstract

This process consists of 4 tree-level diagrams and 86 NLO diagrams. Golem has identified 8 groups of NLO diagrams by analyzing their one-loop integrals.

A walk through Golem-2.0

- Building the code : check the details before the run

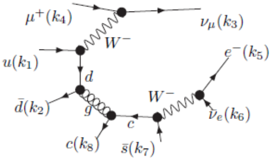


Diagram 1

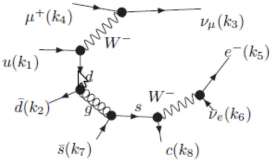


Diagram 2

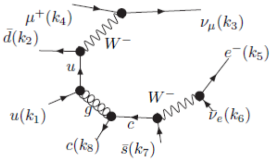


Diagram 3

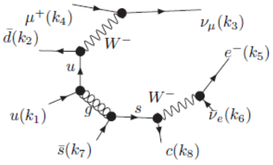


Diagram 4

A walk through Golem-2.0

- Building the code : check the details before the run

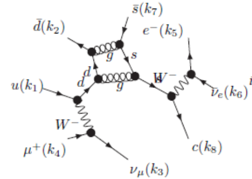


Diagram 36

$S' = S^{\{2\}}_{Q\to q+(-k3+k2+k1-k4)}$, rk $= 2$

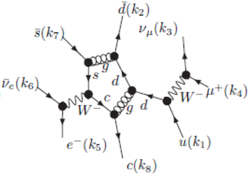


Diagram 3

$S' = S_{Q\to -q-(-k3+k2+k1-k4)}$,
rk $= 3$

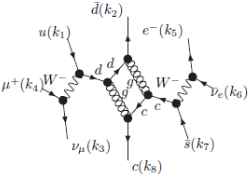
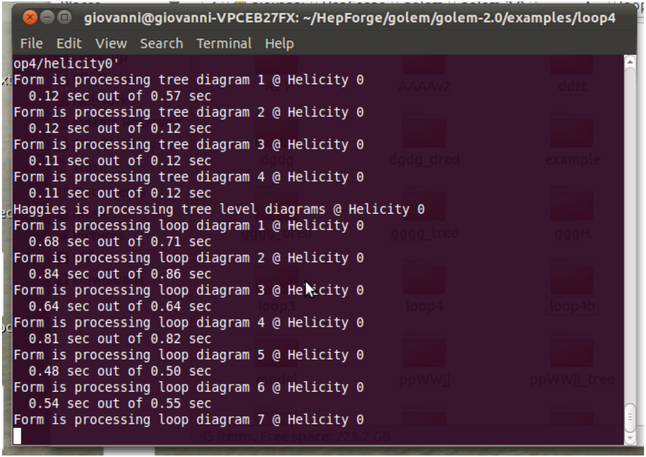


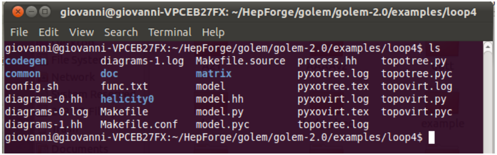
Diagram 31

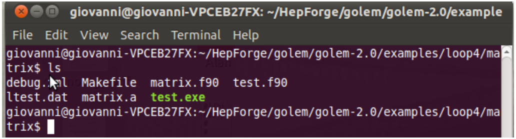
$S' = S^{\{3\}}_{Q\to -q-(-k3+k1-k4)}$, rk $= 2$

- Building the code : **Spinney+ Haggies**



- Execution : all the code is ready






```
NLO/LO, finite part -15.91575134226371
NLO/LO, single pole 7.587050691447690
NLO/LO, double pole -5.333333333333456
```

Table 8 of arXiv:1104.2327: **Melia, Melnikov, Rontsch, Zanderighi**

A walk through Golem-2.0

- Runtime Options : available without regenerating the code

giovanni@giovanni-VPCEB27FX: ~/HepForge/golem/golem-2.0/example

File Edit View Search Terminal Help

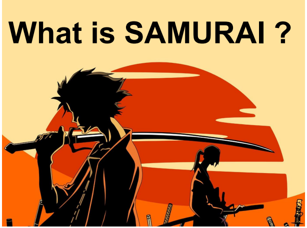
GNU nano 2.2.4 File: config.f90

```
! they can also be set using the subroutine parse in model.f90
integer :: samurai_scalar = 2
integer :: samurai_verbosity = 0
integer :: samurai_test = 0
! The following parameter sets the 'istop' argument in all samurai
! calls. Unless you really know what you do, you should stick to the
! default value.
integer :: samurai_istop = 0
logical :: samurai_group_numerators = .true.

! Options to control the interoperation between different
! reduction methods
integer :: reduction_interoperation = 0
! 0: use samurai only
! 1: golem95 only
! 2: try samurai first, use golem95 if samurai fails
! 3: tens. reconstruction with golem95, reduction with samurai
! 4: tens. reconstruction with golem95, reduction with samurai,
!    use golem95 if samurai fails
! Parameter to switch UV-Counterterms on or off
integer :: renormalisation = 0

! Parameter: Use stable accumulation of diagrams or builtin sum
!            Stable accumulation is implemented in accu.f90
logical :: use_sorted_sum = .false.

! Flag to decide if results should be converted to CDR
! if they are not already in that scheme
logical :: convert_to_cdr = .true.
```

```
real(ki), parameter :: e = 1.0_ki
real(ki), parameter :: gs = 1.0_ki
real(ki), parameter :: mB = 0.0_ki
real(ki) :: mBMS = 4.20_ki
real(ki), parameter :: mC = 0.0_ki
real(ki), parameter :: mD = 0.0_ki
real(ki), parameter :: me = 0.0_ki
real(ki) :: mH = 114.4_ki
real(ki), parameter :: mmu = 0.0_ki
real(ki), parameter :: mS = 0.0_ki
real(ki) :: mT = 171.2_ki
real(ki) :: mtau = 1.77684_ki
real(ki), parameter :: mU = 0.0_ki
real(ki) :: mW = 80.3760000_ki
real(ki) :: mZ = 91.1876000_ki
real(ki) :: NC = 3.0_ki
real(ki) :: Nf = 5.0_ki
real(ki) :: Nfgen = 5.0_ki
real(ki) :: sw =  0.4723042_ki
```

Reduction: "the way of the samurai"

Scattering **AM**plitudes from **U**nitarity-based
Reduction **A**lgorithm at the **I**ntegrand-level



- OPP Reduction Algorithm
  G.O., Papadopoulos, Pittau (2007)
- d-dimensional extension
  Ellis, Giele, Kunszt, Melnikov (2008)
- Coefficients of Polynomials via DFT
  Mastrolia, G.O., Papadopoulos, Pittau (2008)
- Model-independent Computation of the full Rational Term

Mastrolia, G.O., Reiter, Tramontano
JHEP 1008:080,2010

ORIGINAL "MASTER" FORMULA FOR THE INTEGRAND

4-dim identity at the integrand level for $N(q)$ in terms of **4-dim** D_i

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

$\tilde{d}(q)$, $\tilde{c}(q)$, $\tilde{b}(q)$, $\tilde{a}(q)$ are "spurious" terms that vanish upon integration

ORIGINAL "MASTER" FORMULA FOR THE INTEGRAND

4-dim identity at the integrand level for $N(q)$ in terms of **4-dim** D_i

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

Rational Terms

In this approach, Rational Terms require a separate computation

G.O., Papadopoulos, Pittau (2007)

Draggiotis, Garzelli, Malamos, Pittau (2009-2011)

Important points

1. The **functional form** of the **OPP master formula** is **universal** (**process independent**)

2. The **only information required in order to extract the coefficients** of the master integrals is the knowledge of **the numerical value of the numerator function for a finite set of values $\{q_i\}$ of the integration momentum**

3. The process becomes particularly simple if we **choose** $\{q_i\}$ such that sets of **denominators D_i vanish** ("cuts")

G.O., Papadopoulos, Pittau (2007)

IDENTITY IN D-DIMENSIONS: $q \to \bar{q}$

Ellis, Giele, Kunszt, Melnikov (2008)
Melnikov, Schulze (2010)

$$
\begin{aligned}
N(\bar{q}) \;=\; & \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\
+ \; & \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h
\end{aligned}
$$

1. Add a spurious pentagon term in μ^2

$$\Delta_{ijk\ell m}(\bar{q}) = c_{5,0}^{(ijk\ell m)}\,\mu^2$$

2. The coefficients have a more complicated structure

$$\Delta_{ijk\ell}(\bar{q}) = c_{4,0} + c_{4,2}\,\mu^2 + c_{4,4}\,\mu^4 + \left(c_{4,1} + c_{4,3}\,\mu^2\right)\tilde{F}(q)$$

Master Integrals

The blue coefficients, such as $c_{4,0}$ or $c_{4,4}$ multiply non-vanishing integrals:

$$\int d^d\bar{q}\ A(\bar{q}) = c_{4,0}\int d^d\bar{q}\frac{1}{\bar{D}_i\bar{D}_j\bar{D}_k\bar{D}_\ell} + c_{4,4}\int d^d\bar{q}\frac{\mu^4}{\bar{D}_i\bar{D}_j\bar{D}_k\bar{D}_\ell} + \dots$$

In addition to the standard **scalar integrals**

$$\int d^d\bar{q}\frac{1}{\bar{D}_i\bar{D}_j} \quad , \quad \int d^d\bar{q}\frac{1}{\bar{D}_i\bar{D}_j\bar{D}_k} \quad , \quad \int d^d\bar{q}\frac{1}{\bar{D}_i\bar{D}_j\bar{D}_k\bar{D}_l}$$

we need the following additional integrals

$$\int d^d\bar{q}\frac{\mu^2}{\bar{D}_i\bar{D}_j} \quad , \quad \int d^d\bar{q}\frac{\mu^2}{\bar{D}_i\bar{D}_j\bar{D}_k} \quad , \quad \int d^d\bar{q}\frac{\mu^4}{\bar{D}_i\bar{D}_j\bar{D}_k\bar{D}_l}$$

According to the chosen scheme the numerator can be written as:

$$\mathcal{N}(\bar{q}, \epsilon) = N_0(\bar{q}) + \epsilon N_1(\bar{q}) + \epsilon^2 N_2(\bar{q})$$

N_0, N_1 and N_2, are functions of q^ν and μ^2

Rational Terms

- the term proportional to μ^2 is automatically included in the reduction of N_0.
- the term proportional to ϵ, if present, can be reduced using the same algorithm on N_1

Last Step: multiply all coefficients with the corresponding Master Integral

QCDloop
(**Ellis, Zanderighi**)

OneLOop
(**van Hameren**)

Golem95C
(**Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, Reiter, Rodgers**)

- Upgrade of previous *Golem95* library
- Real and complex masses are supported
- Interface for tensorial reconstruction

`http://projects.hepforge.org/~golem/95/`

ALTERNATIVE PATH: THE "TENSORIAL WAY"

Tensorial Reconstruction at the Integrand Level
Heinrich, G.O., Reiter, Tramontano JHEP 1010:105,2010

In this work:

- We **tested** the methods for the **detection of instabilities**
- We proposed a **"rescue-system"** alternative to higher precision routines
- We proposed an **optimized reconstruction method**

Idea: tensorial reconstruction performed at the integrand level by means of a sampling in the integration momentum.

$$\mathcal{N}(q) = \sum_{r=0}^{R} C_{\mu_1 \ldots \mu_r} q_{\mu_1} \ldots q_{\mu_r} \implies \hat{N}(q)$$

$\hat{N}(q)$ is the "reconstructed numerator" written as a tensor
– numerically identical to the initial $\mathcal{N}(q)$ –

Precision Tests in samurai

We use again the decomposition of $N(\bar{q})$ after determining all coefficients

$$
\begin{aligned}
N(\bar{q}) \; &= \; \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\
&+ \; \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h
\end{aligned}
$$

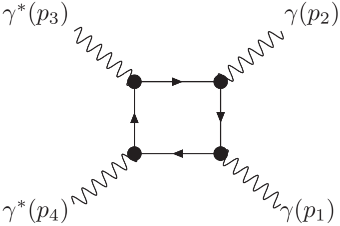
1. **Global** $(N = N)$**-test**
2. **Local** $(N = N)$**-test**
3. **Power-test**

Are those methods **reliable** in detecting **unstable phase space points**?

Approaching the Gram - I

- We approach a kinematic configuration which can lead to large cancellations

- Fermion loop with two massless and two massive vector particles

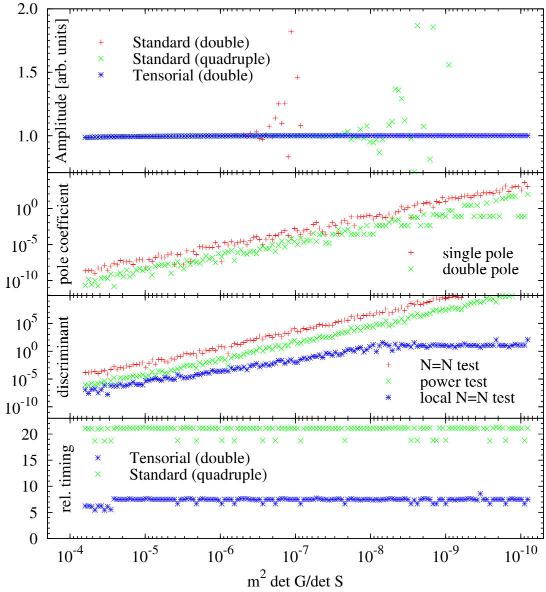


$$
\begin{aligned}
&p_{1,2} = (E, 0, 0, \pm E) \quad p_{1,2}^2 = 0 \\
&p_{3,4} = (E, 0, \pm Q \sin\theta, \pm Q \cos\theta) \\
&p_{3,4}^2 = m^2 \\
&E = \sqrt{m^2 + Q^2}
\end{aligned}
$$

- The Gram-det vanishes when $Q \to 0$ (m and θ are fixed)

$$\det G = 32\, E^4 Q^2 \sin^2\theta$$

Approaching the Gram - II



Ways to use the Tensorial Reduction

- **"Rescue-system"**
  - Unstable points will be automatically reprocessed using the tensorial decomposition + tensor integrals with `Golem 95`
  - Tensorial "master" integrals appears to be less costly than multi-precision routines
- **"Hybrid method"** for improved timing
  - The reduction of $\hat{N}(q)$ can be faster than that of $\mathcal{N}(q)$

# Lines	Time ratio "hybrid"/standard	
N	Rank = 4	Rank = 6
1	1.3	1.6
10	1.1	1.4
100	0.51	0.85
1000	0.30	0.59
10000	0.27	0.55

Alternative reduction paths

Samurai/Tensorial Reduction/Golem95

$u\bar{u} \rightarrow d\bar{d}$

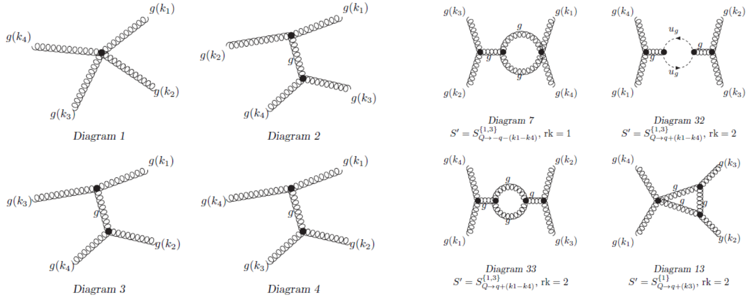
1. Evaluation with Samurai, sampling of diagram groups
2. Evaluation with Samurai, sampling of individual diagrams
3. Tensorial Reconstruction + Reduction of numetens with Samurai
4. Evaluation with Golem95

Method	finite part	single pole	double pole
1	-3.433053565229151	-14.62937842683104	-5.333333333333338
2	-3.433053565229129	-14.62937842683102	-5.333333333333342
3	-3.433053565229163	-14.62937842683104	-5.333333333333342
4	-3.433053565229146	-14.62937842683102	-5.333333333333332

Calculations tested with Golem/Samurai

1. $\gamma + \gamma \to \gamma + \gamma$
2. $\bar{u} + d \to e^- + \bar{\nu}_e$
3. $\bar{u} + u \to d + \bar{d}$
4. $d + g \to d + g$
5. $u + \bar{d} \to e^+ + \nu_e + g$
6. $u + \bar{d} \to e^+ + \nu_e + s + \bar{s}$
7. $u + \bar{d} \to e^+ + \nu_e + g + g$
8. $d + \bar{d} \to e^+ + e^- + g$
9. $d + \bar{d} \to W^+ + W^-$
   (with/without leptonic decays)
10. $d + \bar{d} \to t + \bar{t}$
11. $b + g \to H + b$
12. $u + \bar{u} \to g + \gamma$
13. $u + g \to u + \gamma$
14. $g + g \to g + \gamma$
15. $g + g \to g + g$
16. $g + g \to Z + g$
17. $g + g \to Z + Z$
    (with/without leptonic decays)
18. $g + g \to W^+ + W^-$
    (with/without leptonic decays)
19. $e^+e^-(\to Z) \to \bar{d}dgg$
20. $u\bar{d} \to \bar{c}se^+\nu_e\mu^+\nu_\mu$
21. $u\bar{d} \to \bar{s}ce^-\bar{\nu}_e\mu^+\nu_\mu$

EXAMPLE: $gg \to gg$

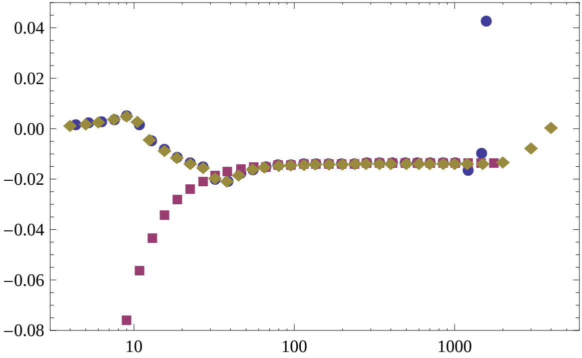


	Golem/Samurai	hep-ph/0609054
LO	14.120983050796795	14.120983050796804
NLO/LO finite	-124.02475579423496	-124.02475579423495
NLO/LO $1/\epsilon$	44.003597347101028	44.003597347101035
NLO/LO $1/\epsilon^2$	-12.000000000000002	-12.000000000000000

Comparison with: hep-ph/0609054 Binoth, Guillet, Heinrich

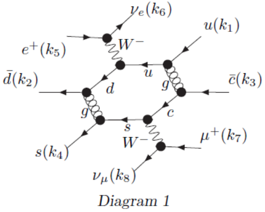
EXAMPLE: $e^+e\,(\to Z) \to \bar{d}dgg$

Pure-axial contributions to $e^+e\,(\to Z) \to \bar{d}dgg$ as a function of M_{top}



MadLoop (Blue) – MENLO PARC (Red) – Golem/Samurai (Yellow)

$u\bar{d} \to \bar{c}se^+\nu_e\mu^+\nu_\mu$



Helicities

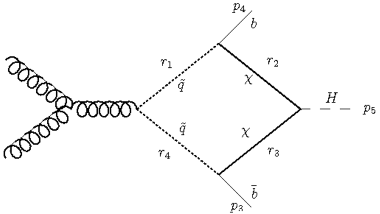
Index	1	2	3	4	5	6	7	8
0	−	+	+	−	+	−	+	−

Golem/Samurai (NLO/LO):

```
finite part 23.3596455167118
single pole 13.6255429251954
double pole -5.33333333333343
```

Comparison with Melia, Melnikov, Rontsch, Zanderighi

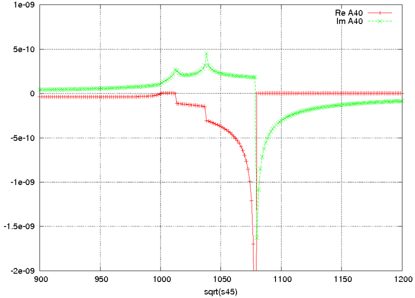
Example: MSSM Higgs and Complex Masses



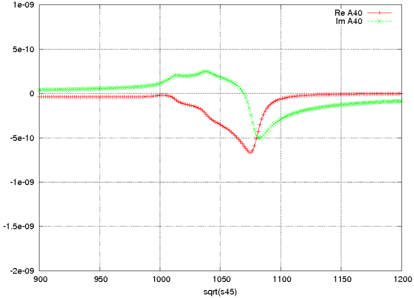
Production of a heavy neutral MSSM Higgs boson and a $\bar{b}b$ pair in gluon fusion.

The loop contains two squarks and two neutralinos

Real Masses



Complex Masses



Golem95C – arXiv:1101.5595

Conclusions: Golem/Samurai

There are many valuable approaches/codes to One-Loop Calculations
(too much to fit in a 30' talk)

Golem/Samurai is a flexible and broadly applicable tool

- it is based on Feynman diagrams
- it uses a d-dimensional reduction (no additional techniques required for rational terms)
- it will be publicly available, as soon as we complete the testing
- it uses some of the best techniques on the market

We look forward to **interacting/interfacing** with other tools!