@TOOLS

17 June 2012

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

MadGraph5

Going Beyonder

Valentin Hirschi, EPFL

Core MG5 Team

F. Maltoni (CP3) , T. Sletzer (UIUC),
O. Mattelaer (FNRS/CP3), J. Alwall (Fermilab)

MG5@NLO Team

V. H.(EPFL) , M. Zaro (CP3), R. Frederix (UZH)

Special thanks to O. mattelaer and J. Alwall from whom many slides are inspired.

MG5, piece of a whole

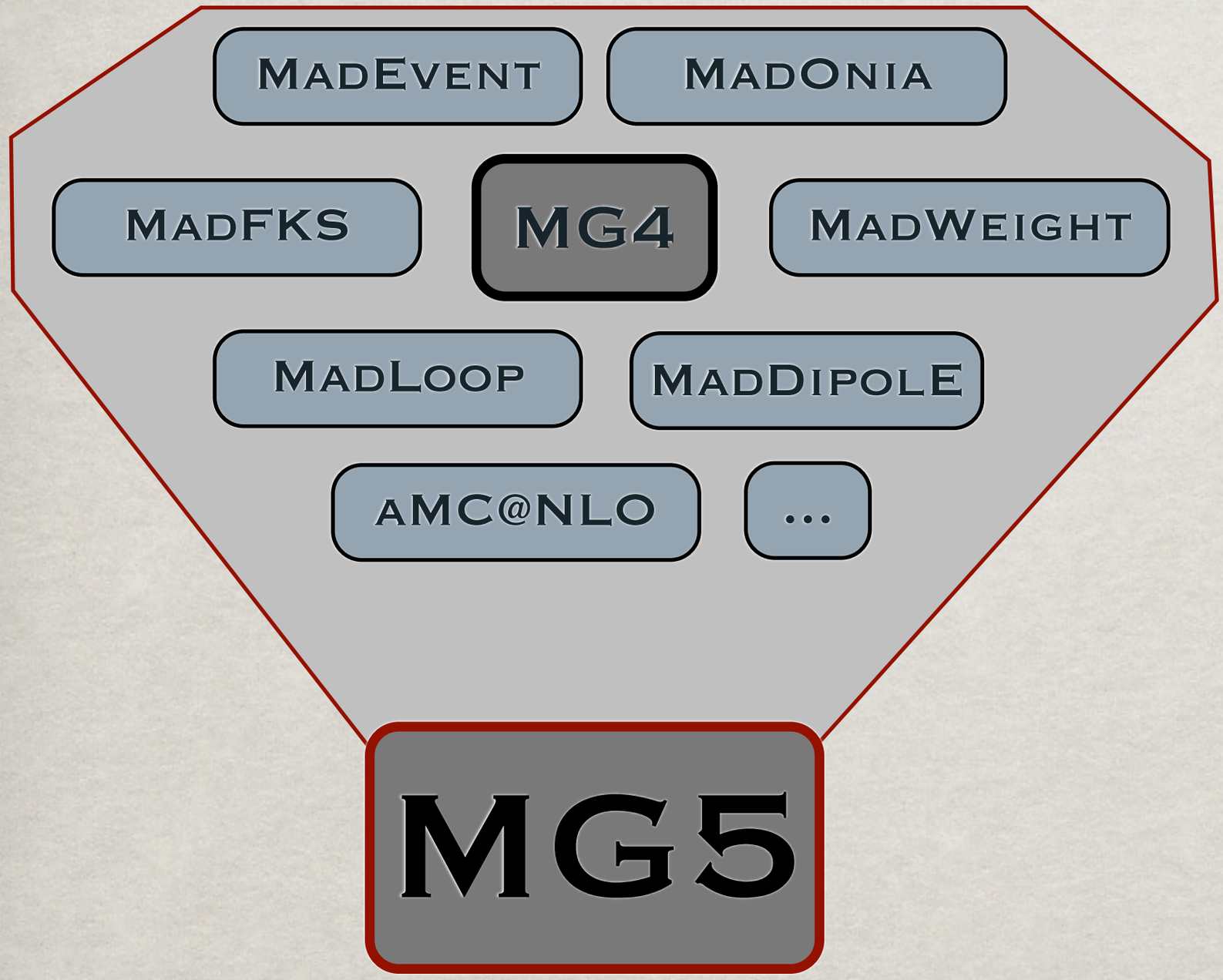
Theory

Exp. data

Model Building tools	Symmetries
FeynRules	Model
MadGraph 5	Matrix Element
MadEvent 5	Partonic Events
Pythia / Herwig	Hadron level
PGS/DELPHES	Detector level

MadAnalysis 5

A BIT OF HISTORY



1994 Core MG4
2002 MadEvent
2007 MadDipole
2008 MadOnia
2008 MadWeight
2009 MadFKS
2011 MadLoop

↓

2011 MG5

One code, to rule them all!

MadGraph 5 Specs

- High-level language: Python
  - Complex data-structures allow for very general objects while keeping speed where needed.
    - Involved algorithms => Performance increase
- Built-in testing suite => Reliability
- User-interface and automatic doc. => User friendly
- Flexible and Modular => Developer friendly
  All-in-one distribution

Developing platform

Name	Status	Last Modified	Last Commit
lp:madgraph5 **Series:** trunk	Mature	2012-06-14	213. Change EWdim6 model accordingly to th...
lp:madgraph5/2.0 **Series:** 2.0	Experimental	2012-03-12	181. merge with 1.4.3
lp:~maddevelopers/madgraph5/GPU	Development	2012-06-15	220. first attempt to minimize the number ...
lp:~maddevelopers/madgraph5/ML5_faster_merged_1.4.6	Development	2012-06-15	230. remove a not working security
lp:~maddevelopers/madgraph5/FKS5_new_born	Development	2012-06-15	314. Fix in add_write_info.f related to th...
lp:~maddevelopers/madgraph5/new_color_ordering	Experimental	2012-06-15	240. Re-merge with 1.4.7 (fixes to cluster...
lp:~maddevelopers/madgraph5/1.4.7	Development	2012-06-15	233. Regenerate html pages after combine_runs
lp:~maddevelopers/madgraph5/1.5.0	Development	2012-06-14	218. fix external use of aloha routine. im...
lp:~maddevelopers/madgraph5/4fermion	Development	2012-06-13	221. fix another id problem
lp:~maddevelopers/madgraph5/FKS5	Development	2012-06-08	201. 1. Merged with latest push of this br...
lp:~maddevelopers/madgraph5/1.4.7_web	Development	2012-06-07	221. fix
lp:~maddevelopers/madgraph5/maddm	Development	2012-06-07	200. Fixed some more bugs. Improved test ...
lp:~maddevelopers/madgraph5/WWW	Development	2012-06-07	8. merged
lp:~maddevelopers/madgraph5/FKS5_merge_1.4.6	Development	2012-06-06	216. 1. Intermediate commit for the 1.5 me...
lp:~maddevelopers/madgraph5/negative_weights	Development	2012-06-06	216. Fixed forgotten DSIGN, fixed unit tes...
lp:~maddevelopers/madgraph5/faster_aloha	Development	2012-06-05	275. adding/modifying routine in order to ...
lp:~maddevelopers/madgraph5/madweight	Development	2012-06-03	209. correct dimension of integration for ...
lp:~maddevelopers/madgraph5/fr_decay	Development	2012-06-01	213. fix various problem (particles/anti-p...
lp:~maddevelopers/madgraph5/spin32	Development	2012-05-23	224. merge with 1.4.6
lp:~maddevelopers/madgraph5/Feynman_gauge	Development	2012-05-23	206. merge with version 1.4.6
lp:~maddevelopers/madgraph5/ML5_faster	Development	2012-05-22	212. 1. The whole skeletton for the open l...
lp:~maddevelopers/madgraph5/1.4.6	Development	2012-05-16	234. merge with 2.0
lp:~maddevelopers/madgraph5/FKS5_new_born_ko	Development	2012-05-07	253. working on an imporving of the madfks...
lp:~maddevelopers/madgraph5/mg5-systematics	Development	2012-03-28	216. Updated README.systematics
lp:~maddevelopers/madgraph5/usermodv5	Development	2012-03-24	225. merge with 1.4.3
lp:~maddevelopers/madgraph5/NLO_EW	Development	2012-03-21	5. Upload
lp:~maddevelopers/madgraph5/fermion_order_c_fix	Development	2012-03-09	221. Merged with old fix for inverted PID,...
lp:~maddevelopers/madgraph5/python_standalone	Development	2012-01-20	192. upgraded version
lp:~maddevelopers/madgraph5/decay_calculator	Experimental	2011-12-13	192. Delete intermediate interactions thor...
lp:~maddevelopers/madgraph5/reweight_alpha_s_for_g_to_bbbar	Development	2011-05-12	141. Changed alpha_s reweighting in reweig...

1 → 30 of 30 results

First • Previous • Next • Last

Supported Models

Color Code

New and in the public release!

Planned / Ongoing progress

Done and will be made public for **MG5 v2.0**

Effective Theories	N-Legs vertices, vN
color structures	Sextets, ε^{IJK}, virtually ALL
Lorentz Structures	ALL, thanks to Aloha
Spins supported	1, 1/2, (3/2), 2
Gauges	Unitary, Feynman
Complex Mass Scheme	Automatic Model Conversion Available for NLO too!
Model with loop info	Import UFO Loop-models
Decay widths computation	On-the-fly widths computation

Lorentz Structures	All, thanks to Aloha

Automatic Language-independent Output of Helicity Amplitude

O. Mattelaer *et al.* , arXiv:1108.2041 [hep-ph]

From UFO to MG5

ALOHA translate a UFO Lorentz structure

```
VVVV6 = Lorentz(name = 'VVVV6',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3) -Metric(1,3)*Metric(2,4)')
```

into pseudo-HELAS subroutine in a chosen language

```
      VERTEX = COUP*( (V4(1)*( (V2(1)*( (0, -1)*(V3(2)*V1(2))
     $ +(0, -1)*(V3(3)*V1(3))+(0, -1)*(V3(4)*V1(4))))+(V1(1)*( (0, 1)
     $ *(V3(2)*V2(2))+(0, 1)*(V3(3)*V2(3))+(0, 1)*(V3(4)*V2(4))))))
     $ +( (V4(2)*( (V2(2)*( (0, -1)*(V3(1)*V1(1))+(0, 1)*(V3(3)*V1(3))
     $ +(0, 1)*(V3(4)*V1(4))))+(V1(2)*( (0, 1)*(V3(1)*V2(1))+(0,
     $ -1)*(V3(3)*V2(3))+(0, -1)*(V3(4)*V2(4))))))+( (V4(3)*( (V2(3)
     $ *( (0, -1)*(V3(1)*V1(1))+(0, 1)*(V3(2)*V1(2))+(0, 1)*(V3(4)
     $ *V1(4))))+(V1(3)*( (0, 1)*(V3(1)*V2(1))+(0, -1)*(V3(2)*V2(2))
     $ +(0, -1)*(V3(4)*V2(4))))))+(V4(4)*( (V2(4)*( (0, -1)*(V3(1)
     $ *V1(1))+(0, 1)*(V3(2)*V1(2))+(0, 1)*(V3(3)*V1(3))))+(V1(4)
     $ *( (0, 1)*(V3(1)*V2(1))+(0, -1)*(V3(2)*V2(2))+(0, -1)*(V3(3)
     $ *V2(3)))))))))
      END
```

Available in
Python, C++ and F77

ALOHA available as
a standalone release

New on ALOHA

- ALOHA is optimizing the way it does analytical computation

Model name	Loading time, new ALOHA	Loading time, old ALOHA
SM	1.2 s	3 s
MSSM	1.4 s	5 s
Randall-Sundrum	90 s	15 min

- Abbreviation usage improves compilation and running time (up to 40%)
- Possibility to create ALOHA subroutine from the MG5 shell

```
mg5> output aloha FFV1_3
```

- New Outputs/Options in progress (not yet into the public release)

Quadruple precision, Feynman Gauge, Spin 3/2,
Complex Mass Scheme, Open Loops techniques, anomalous couplings

Matrix Element Generation

Decay Chains	Fast, no limitation
Output Languages	Python, Fortran, C++
MadEvent5	Less channels, Compact
NLO, Virtual	≡MadLoop5 using OPP
NLO, Real	≡MadFKS5 FKS formalism
Recursion relations for multi-jets	BG Color-Ordered amps
GPU output	Long-standing work

Diagram Generation
speed benchmark

Process	MadGraph 4	MadGraph 5	Subprocesses	Diagrams
$pp \to jjj$	29.0 s	25.8 s	34	307
$pp \to jjl^+l^-$	341 s	103 s	108	1216
$pp \to jjje^+e^-$	1150 s	134 s	141	9012
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	772 s	242 s	1	3474
$gg \to ggggg$	2788 s	1050 s	1	7245
$pp \to jj(W^+ \to l^+\nu_l)$	146 s	25.7 s	82	304
$pp \to t\bar{t}$+full decays	5640 s	15.7 s	27	45
$pp \to \tilde{q}/\tilde{g}\,\tilde{q}/\tilde{g}$	222 s	107 s	313	475
7 particle decay chain	383 s	13.9 s	1	6
$gg \to (\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)(\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)$	70 s	13.9 s	1	48
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10^7 years	251 s	144	11008
$gg \to (\tilde{g} \to u(\bar{\tilde{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to ud\tilde{\chi}_1^-)$				

Very fast decay chains opening the way for new types of processes!

MadEvent5 now able to handle such large decay chains.

Decay Chains	Fast, no limitation

pp>tt~w+, (t>w+b,w+>l+vl),(t~>w-b~,w->jj),w+>l+vl

- Separately generate core process and decays
  Combining them iteratively at the time of the output

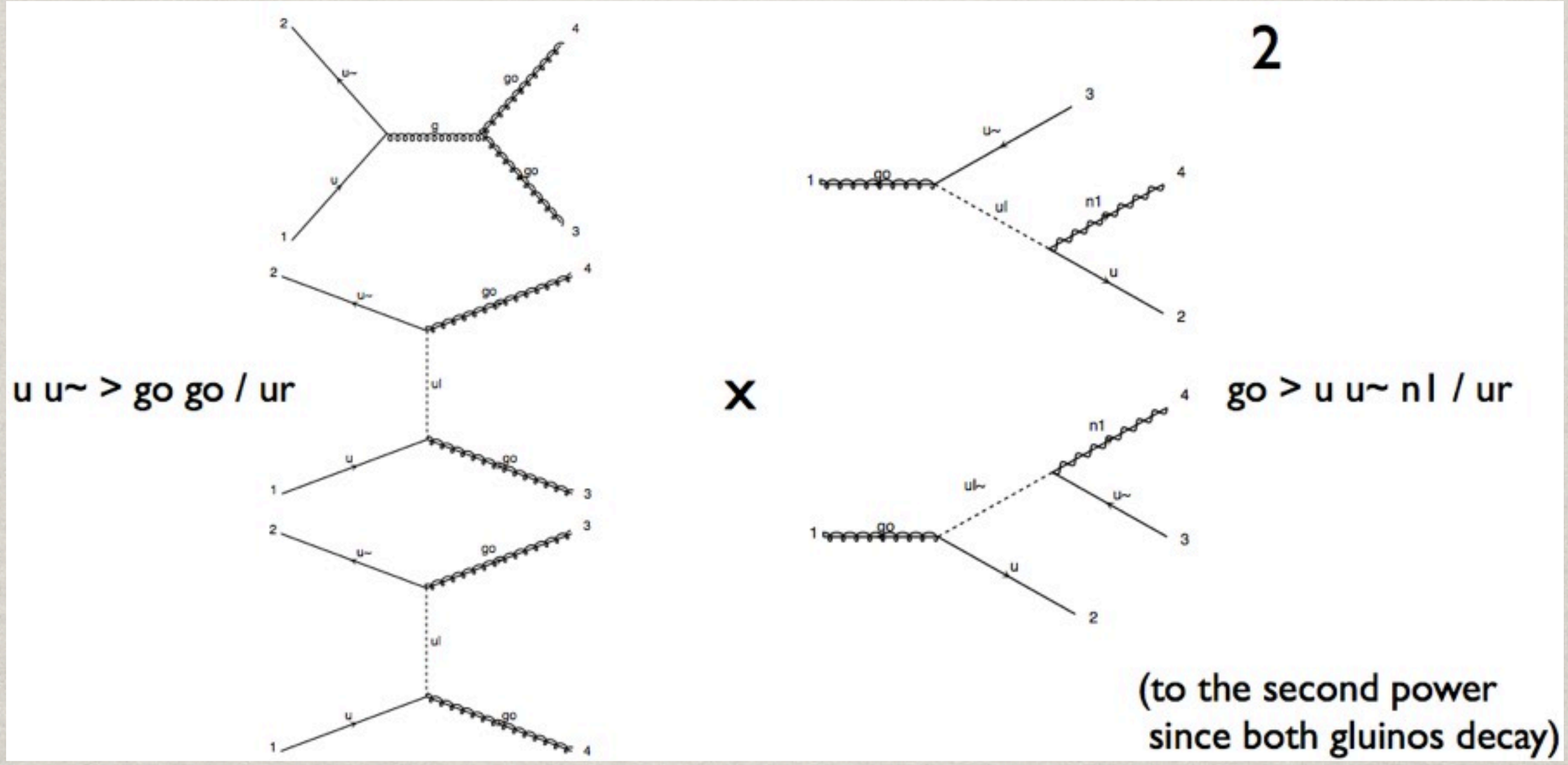
- Retain full matrix element compatible with the decay
  So full width effect and full spin correlations

- However no interference with non-resonant diagrams.
  Description only valid close to pole mass
  Therefore cutoff at $|m \pm n\Gamma|$

- Madevent5 capable of handling decays as large as 2>14 !

Decay Chains	Fast, no limitation

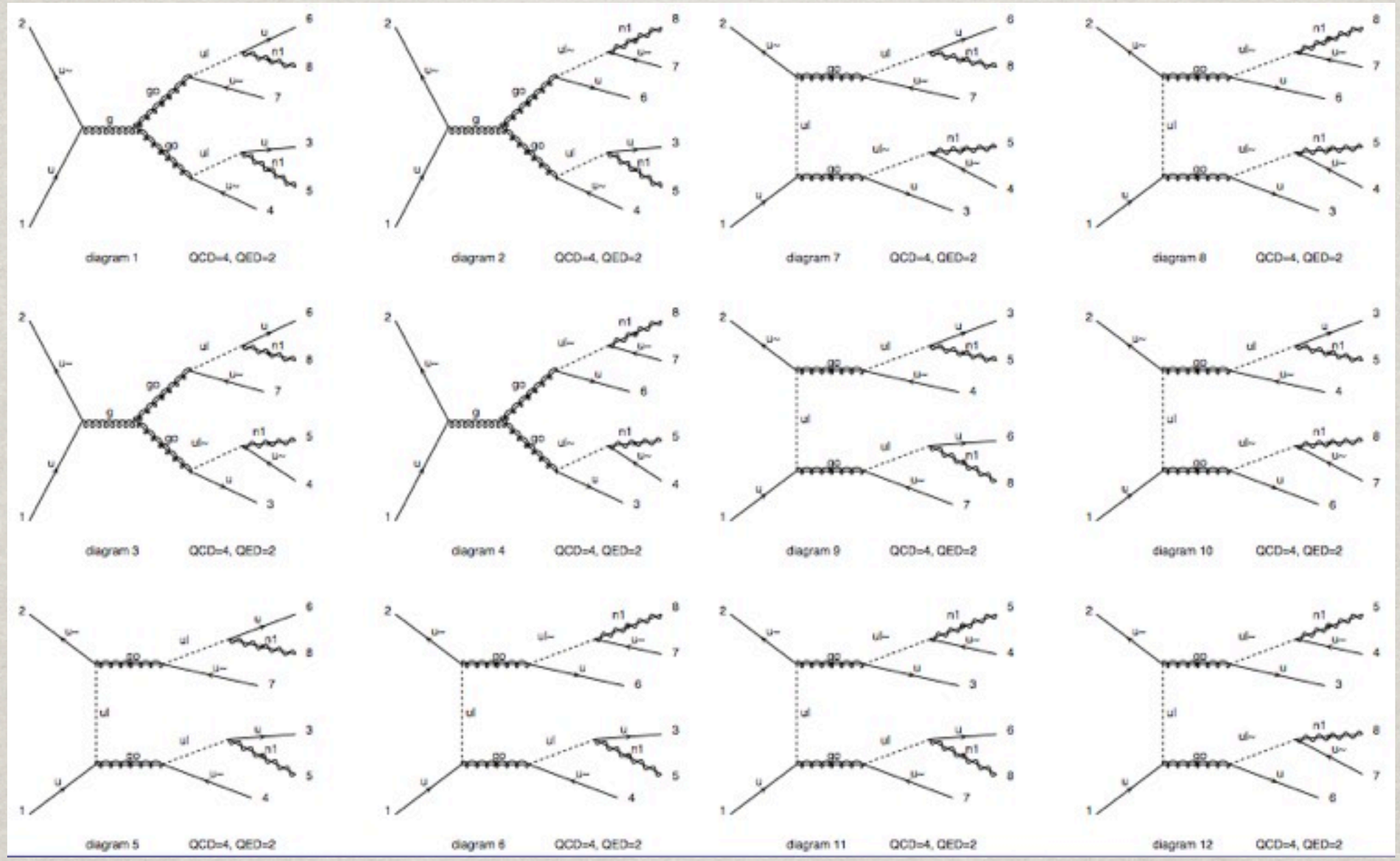
Example `u u~ > go go / ur, go > u u~ nl /ur`



yields...

Decay Chains

Fast, no limitation



Tough bookkeeping ...

Output Languages | Python, Fortran, C++

- Python for fast user local checks of chosen processes

```
mg5> check p p > j j
```

```
28 processes checked in 6.636 s
Gauge results:
Process          matrix           BRS              ratio            Result
g g > g g        1.2407989312e+02 1.2138126173e-27 9.7825085655e-30 Passed
g g > u u~       2.3629996644e+00 1.9002508785e-31 8.0416891595e-32 Passed
Summary: 2/2 passed, 0/2 failed
Lorentz invariance results:
Process          Min element      Max element      Relative diff.   Result
g g > g g        4.2713055572e+02 4.2713055572e+02 2.3954772746e-15 Passed
g g > u u~       1.0314340809e+01 1.0314340809e+01 4.8222171854e-15 Passed
u u > u u        5.8114902657e+01 5.8114902657e+01 1.9562424184e-14 Passed
u c > u c        1.5184393643e+01 1.5184393643e+01 1.2868426421e-15 Passed
u d > u d        6.7102369730e+00 6.7102369730e+00 5.2944682775e-16 Passed
u s > u s        1.0050745419e+00 1.0050745419e+00 4.2417315701e-14 Passed
u s > c d        1.8741700240e-01 1.8741700240e-01 8.8857174941e-16 Passed
d d > d d        1.4179370573e+01 1.4179370573e+01 8.5188735607e-15 Passed
d s > d s        4.6071223798e+00 4.6071223798e+00 1.1567026180e-15 Passed
Summary: 9/9 passed, 0/9 failed
Not checked processes: g g > c c~, g g > d d~, g g > s s~, c c > c c, c d >
Process permutation results:
Process          Min element      Max element      Relative diff.   Result
g g > g g        1.3704079118e+02 1.3704079118e+02 2.0739598178e-16 Passed
g g > u u~       7.3262576044e-01 7.3262576044e-01 1.5154026579e-15 Passed
u u > u u        2.0931560511e+01 2.0931560511e+01 1.8670299544e-15 Passed
u c > u c        1.7726210646e+00 1.7726210646e+00 0.0000000000e+00 Passed
u d > u d        4.1597645298e+00 4.1597645298e+00 1.7081321086e-15 Passed
u s > u s        1.0967268231e+00 1.0967268231e+00 0.0000000000e+00 Passed
u s > c d        1.1260362474e-01 1.1260362474e-01 3.9438269493e-15 Passed
d d > d d        5.6082819971e+01 5.6082819971e+01 1.6470383568e-15 Passed
d s > d s        9.7692549705e+00 9.7692549705e+00 1.8183135201e-16 Passed
Summary: 9/9 passed, 0/9 failed
```

Also be available for loops using slower compiled form

Output Languages | Python, Fortran, C++

- C++ for neat interface with Pythia 8

Run exactly as if it was an internal Pythia standard process

Allows using Pythia for ANY 2>1,2,3 process in ANY model!

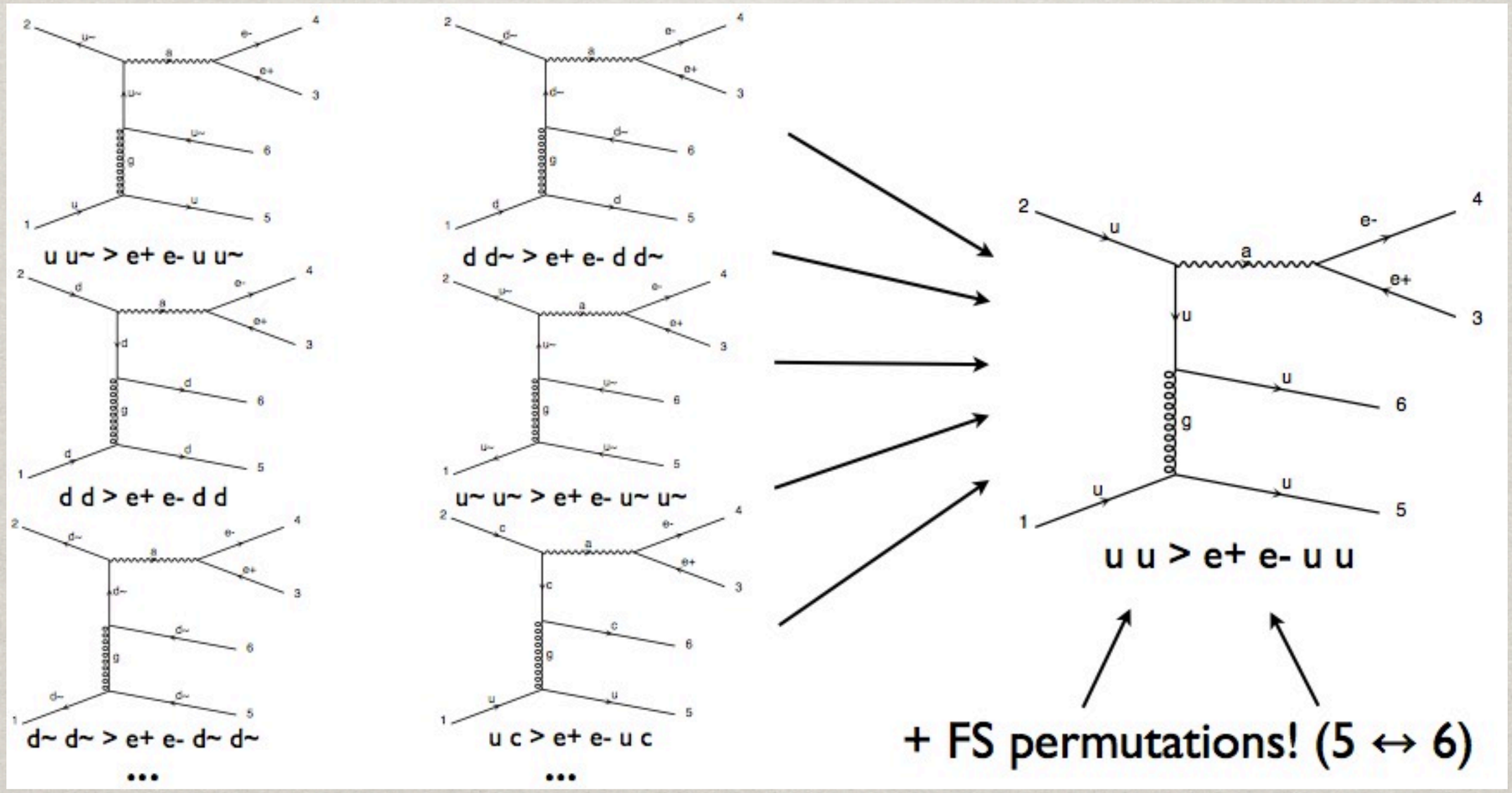
- F77 for the MadEvent output.

Sigma_sm_qq_ttx.h

```
#include "SigmaProcess.h"
#include "Parameters_sm.h"

using namespace std;

namespace Pythia8
{
//==========================================================
// A class for calculating the matrix elements for
// Process: u u~ > t t~
// Process: c c~ > t t~
// Process: d d~ > t t~
// Process: s s~ > t t~
//----------------------------------------------------------
// Process: s s~ > t t~
//----------------------------------------------------------
class Sigma_sm_qq_ttx : public Sigma2Process
{
  public:

    // Constructor.
    Sigma_sm_qq_ttx() {}

    // Initialize process.
    virtual void initProc();

    // Calculate flavour-independent parts of cross section.
    virtual void sigmaKin();

    // Evaluate sigmaHat(sHat).
    virtual double sigmaHat();

    // Select flavour, colour and anticolour.
    virtual void setIdColAcol();
...
```

MadEvent5 | Less channels, Compact

- Combine all processes with same initial/final state (color, spin, mass, width)
- Combine all channels with same pole structure (and permutations)



MadEvent5	User-friendly Too

Example: The process p p > l l j j has the following subprocess directories

content	gg>llqq	gq>llgq	qq>llgg	qq>llqq
# matrix elements	2	4	2	20
# integration channels	8	16	10	14

So only 48 integration channels in MG5 compared to the 486 in MG4!

- USER-FRIENDLY with neat '`install`', '`launch`' and '`help`' commands
- Browser-based monitoring of the runs and results

MadEvent5 | Interfacing to MC tools

MadEvent5 supervises the running of subsequent MC and Analysis tools

- Pythia8, Delphes, PGS and MadAnalysis incorporated.

Matching implemented (CKKW / MLM), may be extended in the future

MadAnalysis5 soon interfaced within ME shell => One framework

Event Generation
speed benchmark

Generation of 10,000 unweighted events
Computer: Sony Vaio TZ laptop / *128-core cluster

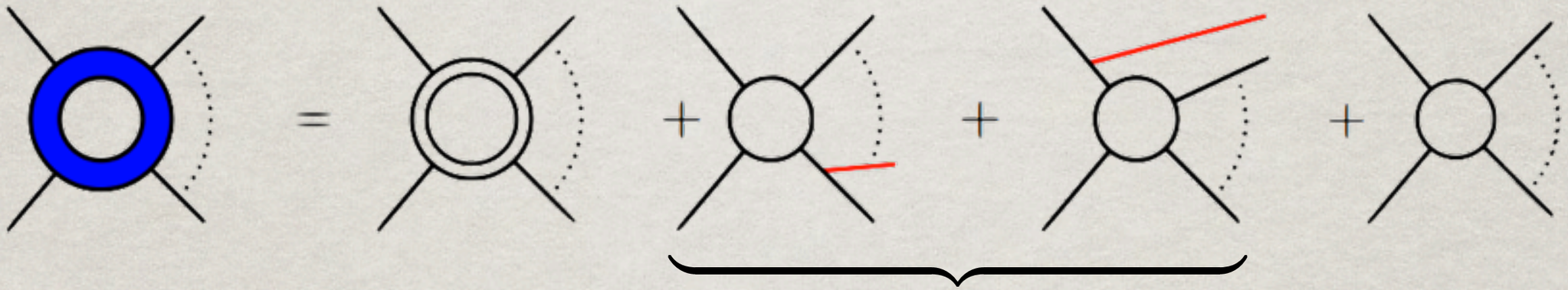
Process	Subproc. dirs.		Channels		Directory size		Event gen. time	
	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5
$pp \to W^+j$	6	2	12	4	79 MB	35 MB	3:15 min	1:55 min
$pp \to W^+jj$	41	4	138	24	438 MB	64 MB	9:15 min	4:19 min
$pp \to W^+jjj$	73	5	1164	120	842 MB	110 MB	21:41 min*	8:14 min*
$pp \to W^+jjjj$	296	7	15029	609	3.8 GB	352 MB	2:54 h*	46:50 min*
$pp \to W^+jjjjj$	-	8	-	2976	-	1.5 GB	-	11:39 h*
$pp \to l^+l^-j$	12	2	48	8	149 MB	44 MB	21:46 min	3:00 min
$pp \to l^+l^-jj$	54	4	586	48	612 MB	83 MB	2:40 h	11:52 min
$pp \to l^+l^-jjj$	86	5	5408	240	1.2 GB	151 MB	49:18 min*	16:38 min*
$pp \to l^+l^-jjjj$	235	7	65472	1218	5.3 GB	662 MB	7:16 h*	2:45 h*
$pp \to t\bar{t}$	3	2	5	3	49 MB	39 MB	2:39 min	1:55 min
$pp \to t\bar{t}j$	7	3	45	17	97 MB	56 MB	10:24 min	3:52 min
$pp \to t\bar{t}jj$	22	5	417	103	274 MB	98 MB	1:50 h	32:37 min
$pp \to t\bar{t}jjj$	34	6	3816	545	620 MB	209 MB	2:45 h*	23:15 min*

No problem running pp>tt~jj on a laptop!

MG5@LOOP Teaser

NLO Basics

NLO contributions have two parts



$$\sigma^{\mathrm{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

Virtual part

- Used to be bottleneck of NLO computations
- This work brings automation using MadGraph5 exploiting the OPP implemented in CutTools.

Real emission part

- Challenge is the systematic extraction of singularities
- FKS subtraction method implemented on MadGraph5

L-CUT DIAGRAMS

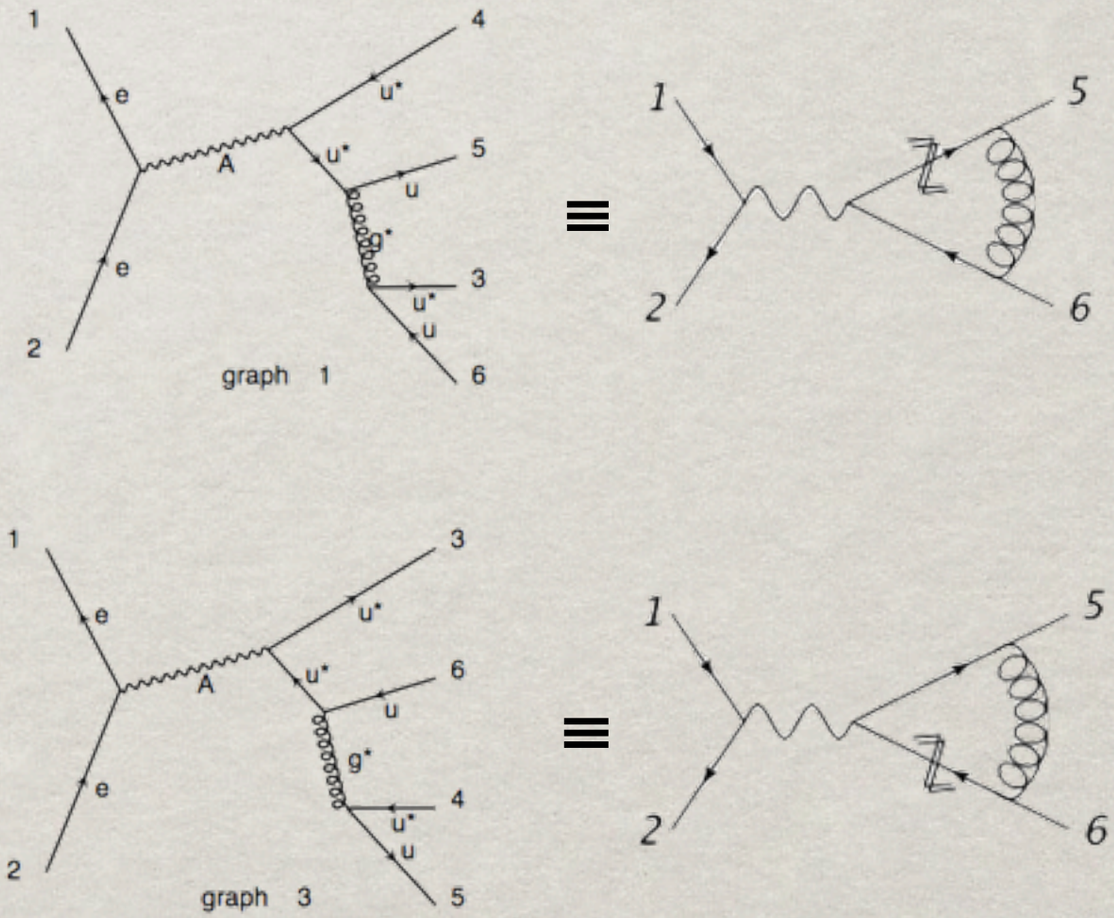
Tree diagram generation algorithms at work for loops

- Loop diagrams are nothing but tree diagrams with two FS merged.

Take advantage of MG5 efficient tree-diagram generation

Filter out tadpoles and wf renorm. loops on the fly.

Disregard loop-particles already considered as L-Cut particles.



LO MG5 POWER BROUGHT TO NLO

- ANY SM process *can* be generated, including those with 4-gluon vertices
- *Expecting* ANY renormalizable loop-model to be handled by MG5.
- Mixed order perturbation expansion
- Quadruple-precision output available for handling unstable PS points.
- Complex mass scheme and Feynman gauge available (under checks)
- Automatic checks for internal consistency and vs independent codes.
- Easy implementation of optimizations at the output level:
  - ➥ Sum over color/hels before OPP calls. Done!
  - ➥ Open loops methods and diagram grouping Work in progress

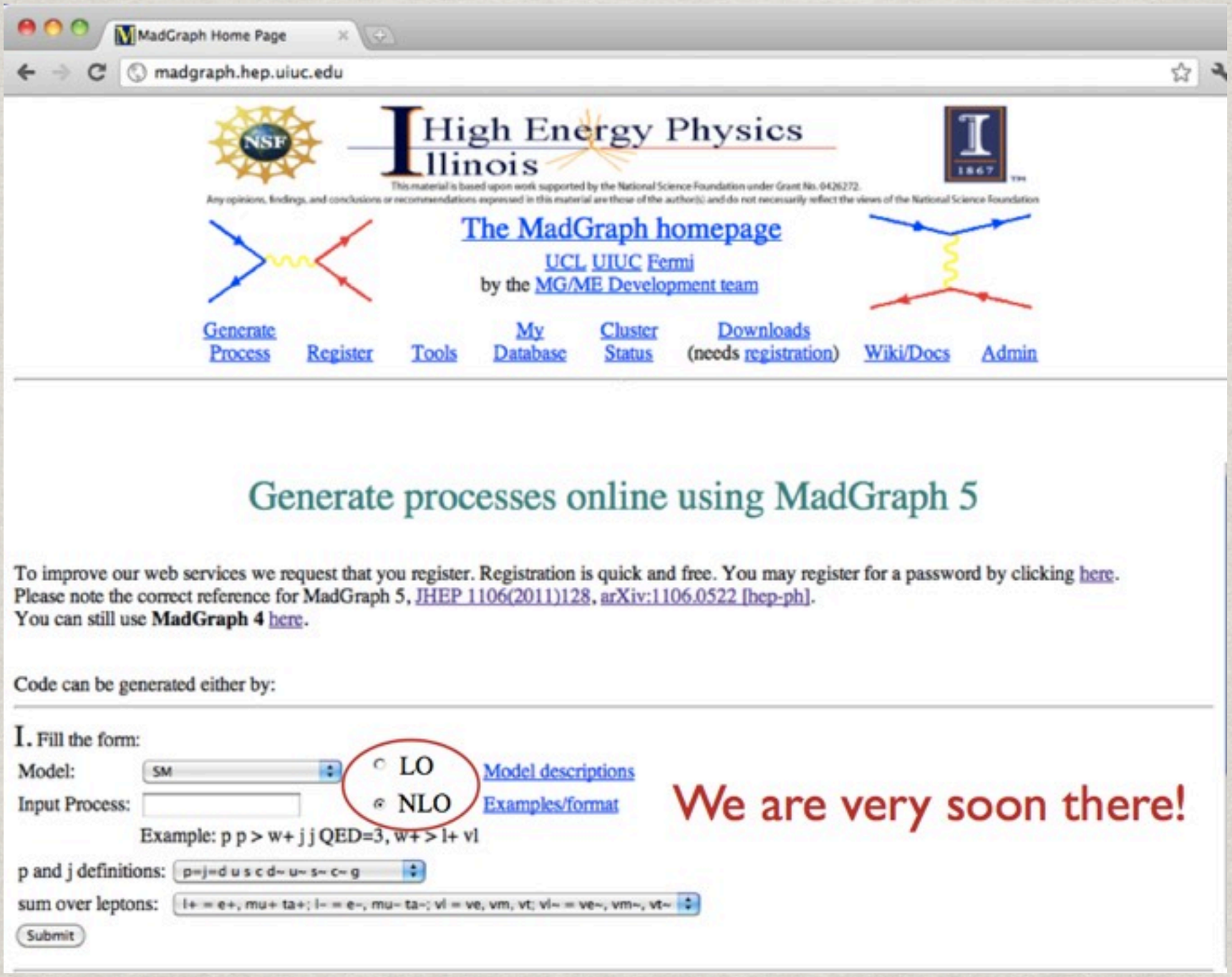
MG5@Loops: Results

1-loop process	**Generation time**		Output size[1]		Compilation time		**Running time[2]**	
`d d~ > u u~`	**3.5 s**	**5.378 s**	68 Kb	268 Kb	0.8 s	2.996 s	**2.2 ms**	**9.4 ms**
`d d~ > d d~ g`	**17.8 s**	**104.8 s**	228 Kb	1.7 Mb	2.4 s	19.181 s	**125 ms**	**0.74 s**
`d d~ > d d~ u u~`	**36.7 s**	**2094 s**	372 Kb	3.3 Mb	4.1 s	45.02 s	**291 ms**	**2.34 s**
`g g > g g`	**13.5 s**	×	372 Kb	×	1.9 s	×	**212 ms**	×
`g g > g g g`	**3 min 23s**	×	180 Kb	×	15.7 s	×	**10.2 s**	×
`g g > h h`	**4 s**	×	28 Kb	×	0.5 s	×	**44 ms**	×
`g g > g h h`	**11.4 s**	×	64 Kb	×	1.0 s	×	**1.16 s**	×

[2] : Of the equivalent `matrix.f` file.

MG5@NLO = ♦, MadLoop (v4) = ♦

[4] : Per PS points, Color / Helicity summed amplitude.



In the mean time, go check http://amcatnlo.cern.ch/

Conclusions, MG5 ...

... is a reliable and generic mature ME generator

... is flexible and modular for easy integration of new modules

... has a powerful user-friendly event generator linked to many analysis and MC tools (Pythia, PGS, MadAnalysis,...)

... aims at becoming an automated, competitive and self-contained NLO generator => (i.e. public aMC@NLO)

... **v2.0** soon released with lots of new features!