MadGraph 5 – The All-New Matrix Element generator for Everything

Johan Alwall
National Center for Theoretical Sciences
National Taiwan University

For the MadGraph 5 team:
J.A. Michel Herquet (NIKHEF)
Fabio Maltoni (Louvain) Olivier Mattelaer (Louvain)
Tim Stelzer (UIUC)

LHC Focus Group Workshop, Academia Sinica, Taipei
April 24, 2010
What will we need for the LHC?
What will we need for the LHC?

NLO
Multi-jet samples
Exotic models
Decay chains
Real corrections
Merging ME/PS
Testing/robustness

Exp-TH communication
Effective theories
Very exotic models
Advanced analysis techniques
Decay Packages
User friendliness

Matrix Elements
Cluster/Grid computing
MadGraph/MadEvent 4

- One of the most widely used automatized matrix element generators
  - Specify any process using simple syntax
  - > 1500 registered users (+ CDF/D0/CMS/ATLAS/...)
- Originally written by Tim Stelzer in 1994
- Phase space integrator/event generator
  MadEvent by F. Maltoni and T. Stelzer in 2002
- MadGraph/MadEvent v. 4 in 2006
### MadGraph 4

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading order matrix element generation</td>
<td>$\leq 8$ FS, $&lt;10000$ diag Max. W+4 jet/tt+3 jet</td>
</tr>
<tr>
<td>BSM, any renormalizable model</td>
<td>Yes</td>
</tr>
<tr>
<td>Decay Chains</td>
<td>Max 8 FS, slow</td>
</tr>
<tr>
<td>Color structures</td>
<td>Singlet/triplet/octet</td>
</tr>
<tr>
<td>Extended color structures $(6, 27, \varepsilon_{ijk})$</td>
<td>No</td>
</tr>
<tr>
<td>Effective theories $(&gt;4$-particle vx)</td>
<td>No</td>
</tr>
<tr>
<td>Recursion relations for multijet generation</td>
<td>No</td>
</tr>
<tr>
<td>NLO real corrections</td>
<td>Yes</td>
</tr>
<tr>
<td>NLO loop calculations</td>
<td>In progress</td>
</tr>
<tr>
<td>Output in any language/format</td>
<td>Only Fortran</td>
</tr>
</tbody>
</table>
Why new MadGraph?

• First version of core code from 1994
• Written in Fortran 77
  – Fixed array sizes
  – Limited (no) libraries
  – No recursion
  – Complicated file output
  – Difficult to modularize (no OO, dynamic libraries,...)
  – Difficult to extend
+ Intrinsically very fast
MadGraph 5

• Development started November 2009

• Modular program structure
  – Diagram generation / Color algebra / Helas objects / Diagram drawing / I/O libraries / ...

• Modern programming techniques
  – “Extreme programming”
  – Complete test suite including extensive module/function testing and integration/parallel tests
  – Functionality first, easy to modify/refactor/optimize/extend
Programming language: Python

- (Very) high level (Object Oriented, functional programming, ...)
- Easy to learn/write/maintain, concise (x4 compared to F77)
- Easily available on all platforms and no compilation required
- Slow, but fast standard library (99% of calculations) and easily extendable
- Automatic documentation
Innovations

- Completely new diagram generation algorithm
  - Makes optimal use of model information
  - Improves Helas call optimization by up to 90%
- Efficient multiprocesses (keep full track of discarded process crossings)
- Generic and “smart” new color calculation library
- New, faster and generic diagram drawing library
- Improved fermion flow treatment with Majorana particles
- Very efficient generation of decay chains
- Output formats: Fortran, C++, ...
- User friendly command line interface
- ... and (much) more to come !!!
### MadGraph 5

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading order matrix element generation</td>
<td>No limitations except time W+5 jets/tt+4 jets realistic</td>
</tr>
<tr>
<td>BSM, any renormalizable model</td>
<td>Yes</td>
</tr>
<tr>
<td>Decay Chains</td>
<td>No limitations, fast</td>
</tr>
<tr>
<td>Color structures</td>
<td>No limitations</td>
</tr>
<tr>
<td>Extended color structures (6, 27, $\varepsilon^{ijk}$)</td>
<td>Available (not yet tested)</td>
</tr>
<tr>
<td>Effective theories (&gt;4-particle $\nu\chi$)</td>
<td>Yes, no limitations</td>
</tr>
<tr>
<td>Recursion relations for multijets</td>
<td>To be implemented</td>
</tr>
<tr>
<td>NLO real corrections</td>
<td>To be implemented</td>
</tr>
<tr>
<td>NLO loop calculations</td>
<td>To be implemented</td>
</tr>
<tr>
<td>Output in any language/format</td>
<td>No limitations, Fortran (MG/ME 4) available</td>
</tr>
</tbody>
</table>
Present status

Beta v. 0.4.0 available next week!

- Full matrix element generation for any model that is available for MadGraph 4
- Complete Majorana particle treatment
- Full decay chain generation
- Complete MadGraph Standalone and MadEvent output
- Extensively tested against MG 4 (SM+MSSM)
# Speed benchmarks

Full MadEvent subprocess directory output, including diagram drawing

Computer: Sony Vaio TZ

<table>
<thead>
<tr>
<th>Process</th>
<th>MG4</th>
<th>MG5</th>
<th>Definitions</th>
<th>Subprocs (after combine)</th>
<th>Diagrams</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>pp &gt; jji</td>
<td>29.02 s</td>
<td>54.38 s</td>
<td>p, j=(\mu/\nu/\tau)</td>
<td>34</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jj</td>
<td>341 s (5:41 min)</td>
<td>268 s (4:18 min)</td>
<td>p, j=(\mu/\nu/\tau)</td>
<td>108</td>
<td>1216</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jji e+e-</td>
<td>2444 s (40:44 min)</td>
<td>993 s (16:33 min)</td>
<td>p, j=(\mu/\nu/\tau)</td>
<td>141</td>
<td>9012</td>
<td></td>
</tr>
<tr>
<td>uu&gt;e+e+e+e+e-</td>
<td>772 s (12:52 min)</td>
<td>175 s (2:56 min)</td>
<td>1</td>
<td>3474</td>
<td>MG4: 3194 wavefunctions, MG5: 301 wavefunctions</td>
<td></td>
</tr>
<tr>
<td>gg &gt; gggg</td>
<td>2768 s (46:28 min)</td>
<td>1649 s (17:29 min)</td>
<td>1</td>
<td>7245</td>
<td>MadGraph standalone output, MG4: 3745 wavefunctions, MG5: 888 wavefunctions</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jj (W+ &gt; H+H)</td>
<td>146 s (2:26 min)</td>
<td>70 s (1:10 min)</td>
<td>p, j=(\mu/\nu/\tau)</td>
<td>82</td>
<td>304</td>
<td></td>
</tr>
<tr>
<td>pp &gt; tt&gt; &gt; with full decays</td>
<td>5640 s (1:34 h)</td>
<td>22.0 s</td>
<td>27</td>
<td>45</td>
<td>MG4: 12 proc def, MG5: single proc def</td>
<td></td>
</tr>
<tr>
<td>pp&gt;sq sq</td>
<td>222 s (3:42 min)</td>
<td>286 s (4:46 min)</td>
<td>p, j=(\mu/\nu/\tau)</td>
<td>313</td>
<td>476</td>
<td></td>
</tr>
<tr>
<td>gg&gt;(q&gt;u&gt;(u&gt;((n2&gt;(\nu1))))&gt;(q&gt;d&gt;&gt;(x1))&gt;l)</td>
<td>383 s (7:23 min)</td>
<td>5.2 s</td>
<td>1</td>
<td>6.7 FS decay chain, single diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gg&gt;(q&gt;u&gt;((\nu1))&gt;q&gt;u&gt;((\nu1))&gt;l)</td>
<td>70 s</td>
<td>5.5 s</td>
<td>1</td>
<td>48.6 FS decay chain, mult.diag.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pp&gt;(q&gt;j&gt;((\nu1))&gt;q&gt;j&gt;((\nu1))&gt;l)</td>
<td>3 h &gt; &gt; &gt; year</td>
<td>561 s (9:11 min)</td>
<td>144</td>
<td>11008</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Speed benchmarks

Full MadEvent subprocess directory output, including diagram drawing

Computer: Sony Vaio TZ

<table>
<thead>
<tr>
<th>Process</th>
<th>MG4</th>
<th>MG5</th>
<th>Definitions</th>
<th>Subprocs (after combine)</th>
<th>Diagrams</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>pp &gt; jji</td>
<td>29.02 s</td>
<td>54.38 s</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>34</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jji</td>
<td>341 s (5:41 min)</td>
<td>268 s (4:18 min)</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>108</td>
<td>1216</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jji e+e-</td>
<td>40 min</td>
<td>16 min</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>141</td>
<td>9012</td>
<td></td>
</tr>
<tr>
<td>uu~ &gt; e+e+e+e+e-</td>
<td>772 s (12:52 min)</td>
<td>176 s (2:56 min)</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>1</td>
<td>3474</td>
<td></td>
</tr>
<tr>
<td>uu~ &gt; e+e+e+e+e+e-</td>
<td>38 s (46:28 min)</td>
<td>1049 s (17:29 min)</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>1</td>
<td>7245</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jji (W+ &gt; tt)</td>
<td>46 s (2:26 min)</td>
<td>70 s (1:10 min)</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>82</td>
<td>304</td>
<td></td>
</tr>
<tr>
<td>pp &gt; tt~ + decays</td>
<td>1:34 h</td>
<td>22 s</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>27</td>
<td>45</td>
<td>12 proc def, single proc def</td>
</tr>
<tr>
<td>pp &gt; sq sq</td>
<td>222 s (3:42 min)</td>
<td>286 s (4:46 min)</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>313</td>
<td>475</td>
<td></td>
</tr>
<tr>
<td>gg&gt;(go&gt;u/u~(u~(n2&gt;2n1)))(go&gt;u/d~e1)</td>
<td>383 s (7:23 min)</td>
<td>5.2 s</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>1</td>
<td>67 FS decay chain, single diagram</td>
<td></td>
</tr>
<tr>
<td>gg&gt;(go&gt;u/<del>u</del>(n1&gt;2n1))(go&gt;u/u~1)</td>
<td>70 s</td>
<td>5.5 s</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>1</td>
<td>486 FS decay chain, multi diag.</td>
<td></td>
</tr>
<tr>
<td>pp &gt; (go&gt;jjx^0)(go&gt;jjx^0)</td>
<td>&gt;&gt; 1 year</td>
<td>9 min</td>
<td>p, j=u/u/c/c<del>d/d</del>/s/s~/g</td>
<td>144</td>
<td>11008</td>
<td></td>
</tr>
</tbody>
</table>

~ 2.5 times faster evaluation for produced matrix elements
Diagram examples

Process: \( gg \rightarrow go \; go \)

- Decay: \( go \rightarrow u \; ul^\sim \)
  - Decay: \( ul^\sim \rightarrow d^\sim \; x1^- \)
  - Decay: \( x1^- \rightarrow e^- \; sve^\sim \)
  - Decay: \( sve^\sim \rightarrow ve^- \; n1 \)

- Decay: \( go \rightarrow u \; ul^\sim \)
  - Decay: \( ul^\sim \rightarrow d^\sim \; x1^- \)
  - Decay: \( x1^- \rightarrow e^- \; sve^\sim \)
  - Decay: \( sve^\sim \rightarrow ve^- \; n1 \)

(10 FS particles. Generation time: 5 s)
Development directions
Sidenote: FeynRules

[Christiansen, Duhr, arXiv:0806.4194]
MadGraph 5 BSM

- New FeynRules interface including color and Lorentz structures [C. Duhr, M. Herquet, et al]
- Automatic Helicity Amplitude (HELAS) output for any new model (including effective theories) [P. de Aquino, W. Link, O. Mattelaer]
- Automatic HELAS routines in Fortran/C/C++/… From Lagrangean to matrix elements/
decays/event generation in ANY model!
MadGraph 5 Multijets

- For multijet generation ($\geq 4$ jets), Feynman diagram formalism expensive (factorial growth)
- Helicity amplitude optimization (in MG4/5) reduces run times by factor $\sim 10$ for complex processes
- Recursion relations (such as Berhreends-Giele) can reduce run times by additional orders of magnitude
- MG5 perfect framework for implementation and development
- Work started with exciting prospects in near future!
MadGraph NLO

\[ \sigma_{NLO} = \int_m d^{(d)} \sigma^V + \int_{m+1} d^{(d)} \sigma^R + \int_m d^{(4)} \sigma^B \]
MadGraph NLO

- **Virtuals**: two (complementary) approaches:
  - Use MG to generate diagrams and calculate n+2 amplitudes to build the NLO result (CutTools technique), e+e- → 2 and 3 jets already checked (MG4).
  - Advantages: valid for any BSM model

  \[ V. \text{Hirschi, R. Pittau, M. V. Garzielli; R. Frederix} \]

  - Rely on external tool(s) (BlackHat, Rocket, Golem, ...) using the Binoth-LHA accord.
  - Various e+e- and hadronic processes checked.
  - Advantage: strong optimization possibilities.
MadGraph NLO

- **Real contributions: two approaches:**
  
  - **MadDipole:** Catani-Seymour dipole subtraction scheme, standalone implementation (TH), cancellation of singularities checked, and dipoles checked against MCFM.
  
  - **MadFKS:** Frixione-Kunszt-Signer subtraction scheme, integration is available (TH+PH), cancellation of singularities checked.

- **Both:** usable both for SM and BSM processes, and for massless and massive external particles.

[R. Frederix, S. Frixione, et al]
MadGraph 5 NLO

- MadGraph 5 will significantly simplify the continued development efforts for both virtual and real contributions:
  - Clear structure – easy to extract exactly what is needed
  - Modular – Easy to extend with new features
  - Flexible – Output not limited to Fortran
### Timeline for MadGraph 5

<table>
<thead>
<tr>
<th>Tools</th>
<th>Sept 09</th>
<th>Dec 09</th>
<th>Mar 10</th>
<th>June 10</th>
<th>Sept 10</th>
<th>Dec 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MadWeight, MadOnia, etc. released and stable for ME v4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dvl. MadFKS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics results MadFKS + stable MadDipole</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dvl. CutTools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics results v4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dvlpt. CutTools v5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics res. v5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FeynRules interface v4 + USRMODE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dvlpt. FR iff v5 + autom. HELAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generic MG5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MadEvent v4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start dvlpt. ME v5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MadGraph v4 Development v5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release core MG v5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Details

- **MG (MadGraph)**
  - madgraph v4 Development v5
  - Release core MG v5

- **BSM (BSM)**
  - FeynRules interface v4 + USRMODE2
  - Dvlpt. FR iff v5 + autom. HELAS
  - Generic MG5

- **NLO V (NLO v5)**
  - Dvlpt. CutTools
  - Physics results v4

- **NLO R (NLO v5)**
  - Dvl. MadFKS
  - Physics results MadFKS + stable MadDipole

- **Tools**
  - MadWeight, MadOnia, etc. released and stable for ME v4
  - Move to ME v5
Conclusions

- MG/ME v4 is a mature, well established and stable code with many features for BSM and QCD physics, and numerous peripheral tools.
- MG/ME v5 is here, with important and unprecedented improvements in all directions.
- Beta release of core MadGraph 5 v. 0.4.0 already next week, many developments in the near future!