MADGRAPH 5

The All New Matrix Element Generator for Everything

O. Mattelaer
INFN-Roma Tre

F. Maltoni (CP3)
T. Sletzer (UIUC)

J. Alwall (Taiwan)
M. Herquet (NIKHEF)

CP3 Lunch: 19/05/10
WHAT WE WILL NEED FOR THE LHC?
WHAT WE WILL NEED FOR THE LHC?

- NLO
- Multi-jet samples
- Exotic models
- Decay chains
- Real corrections
- Merging ME/PS
- Testing/robustness
- Exp-TH communication
- Effective theories
- Matrix Elements
- Advanced analysis techniques
- Decay Packages
- Very exotic models
- User friendliness
One of the most widely used matrix element event generator

- Specify any process using simple syntax
- More than 1500 users (CMS/ATLAS/DO/CDF/...)

mercredi 19 mai 2010
A long time ago in a galaxy far, far away....
A long time ago in a galaxy far, far away....
A long time ago in a galaxy far, far away....
One of the most widely used matrix element event generator

- Specify any process using simple syntax
- More than 1500 users (CMS/ATLAS/DO/CDF/...)

Originally written by T. Sletzer in 1994

- Phase Space Integrator/Event Generator MadEvent in 2002. (F. Maltoni & T. Sletzer)
- MadGraph v4 in 2006
Code can be generated either by:

**I. Fill the form:**

**Model:**

- **SM**
- Model descriptions

**Input Process:**

- Examples

**Max QCD Order:**

- 99

**Max QED Order:**

- 99

**p and j definitions:**

- $p = j = d, u, s, c, d, u, s, c, g$

**Sum over leptons:**

- $l^+ = e^+, u^+, l^- = e^-, u^-, v_l = v_e, v_l = v_e$, $v_l = v_e$

[Submit]
## MADGRAPH 4
### CHECKLIST

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</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>

*mercredi 19 mai 2010*
WHY A MADGRAPH 5

- First version of the core code from 1994
- Written in fortran 77
  - Fixed array size
  - Limited (no) Libraries
  - No pointer/ No recursion
  - Complicated file output
  - No Object Oriented (not modular)
  - Difficult to extend
- But intrinsically very fast
MADGRAPH 5

- Development starts in September 2009 at MadGraph 2009

- Modular Program Structure
  - Diagram Generation/ Color Algebra / Helas Object / diagram drawing / IO libraries

- Modern Programming Technique
  - Complete test suite including extensive module/function testing
  - “Extreme Programming”
MADGRAPH 5

MadGraph Version 4
UCL UIUC Fermi
by the MG/ME Development team

Python

MG5
WHY PYTHON?

- (Very) **High Level** (object oriented, functional programming, multi-heritage, ...)
- **Easy to learn, to write, to read and to maintain.**
  - Example: the memory is automatically cleaned
- **Easily Available** on all computers, no compilation required
- **Slow but fast library** (90% of the computation)
- 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 Multiprocess (keep track of discarded process crossing)
- Generic and smart new color calculation library
- Faster and generic diagram drawing
- Very efficient decay chain package
- Command line interface
- And (much) more to come
## MADGRAPH 5

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading order matrix element generation</td>
<td>No limitations except time W+5 jets/(t\bar{t}+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 (v_x))</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 0.4.0 available since last week
Beta 0.4.1 available today (MG5 on the web)

- Full Matrix Element generation for any v4 model
- Complete Majorana treatment
- Full decay chain generation
- Complete MadGraph StandAlone / Madevent Output
- Secure mode in order to run on the web
- Extensively tested against MG4 (SM + MSSM)
### Speed Benchmark

<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; j j</td>
<td>29.02 s</td>
<td>54.38 s</td>
<td>p, j = u/c/c/d/s/g</td>
<td>34</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>pp &gt; j j+</td>
<td>34.1 s (5:41 min)</td>
<td>258 s (4:18 min)</td>
<td>s, j = u/c/c/d/s/g</td>
<td>108</td>
<td>1216</td>
<td></td>
</tr>
<tr>
<td>uu&gt;ee+ee+ee-</td>
<td>2444 s (40:44 min)</td>
<td>993 s (16:33 min)</td>
<td>s, j = u/c/c/d/s/g</td>
<td>141</td>
<td>9012</td>
<td></td>
</tr>
<tr>
<td>gg &gt; ggggg</td>
<td>2788 s (46:28 min)</td>
<td>1049 s (17:29 min)</td>
<td></td>
<td>1</td>
<td>7245</td>
<td></td>
</tr>
<tr>
<td>pp &gt; j j (W+ &gt; H+j+)</td>
<td>146 s (2:26 min)</td>
<td>70 s (1:10 min)</td>
<td>p, j = u/c/c/d/s/g</td>
<td>82</td>
<td>304</td>
<td></td>
</tr>
<tr>
<td>pp &gt; tt- with full decays</td>
<td>5640 s (1:34 h)</td>
<td>22.0 s</td>
<td>p, j = u/c/c/d/s/g</td>
<td>27</td>
<td>45</td>
<td>MG4: 12 proc defs, MG5: single proc def</td>
</tr>
<tr>
<td>pp&gt;ss sq</td>
<td>222 s (3:42 min)</td>
<td>286 s (4:46 min)</td>
<td>p, j = u/c/c/d/s/g</td>
<td>313</td>
<td>475</td>
<td></td>
</tr>
<tr>
<td>qq&gt;(qq&gt;u(u-&gt;(n2-&gt;n1))((qq&gt;ud-&gt;x1))</td>
<td>383 s (7:23 min)</td>
<td>6.2 s</td>
<td></td>
<td>1</td>
<td>67 FS decay chain, single diagram</td>
<td></td>
</tr>
<tr>
<td>qq&gt;(qq&gt;u(n1))((qq&gt;uu-&gt;n1))</td>
<td>70 s</td>
<td>5.5 s</td>
<td></td>
<td>1</td>
<td>48</td>
<td>FS decay chain, multidiag.</td>
</tr>
<tr>
<td>pp&gt;(qq&gt;j1)(qq&gt;j1)</td>
<td>3 h -&gt; &gt;1 year</td>
<td>551 s (9:11 min)</td>
<td>p, j = u/c/c/d/s/g</td>
<td>144</td>
<td>11008</td>
<td></td>
</tr>
</tbody>
</table>
### Speed Benchmark

<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; jj</td>
<td>29.02 s</td>
<td>54.38 s</td>
<td>p, j=ulu/~cl/c/<del>d/d</del>s/s/g</td>
<td>34</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jj H+</td>
<td>341 s (5:41 min)</td>
<td>258 s (4:18 min)</td>
<td>p, j=ulu/~cl/c/<del>d/d</del>s/s/g</td>
<td>108</td>
<td>1216</td>
<td></td>
</tr>
<tr>
<td>pp &gt; j j e+e-</td>
<td>40 min</td>
<td>16 min</td>
<td>p=ulu/~cl/c/<del>d/d</del>s/s-b/b/g</td>
<td>141</td>
<td>9012</td>
<td>MG4: 3194 wfs MG5: 301 wfs</td>
</tr>
<tr>
<td>uu~ &gt; e+e-e+e-e+e</td>
<td>1.772 s (1:26:52 min)</td>
<td>1.75 s (2:55 min)</td>
<td></td>
<td>1</td>
<td>3474</td>
<td></td>
</tr>
<tr>
<td>uu~ &gt; e+e-e+e+e+e</td>
<td>13 s (46:28 min)</td>
<td>1049 s (17:29 min)</td>
<td></td>
<td>1</td>
<td>7245</td>
<td></td>
</tr>
<tr>
<td>pp &gt; jj (W+ -&gt; l+l)</td>
<td>146 s (2:26 min)</td>
<td>70 s (1:10 min)</td>
<td>p, j=ulu/~cl/c/<del>d/d</del>s/s/g l=ee+/-</td>
<td>82</td>
<td>304</td>
<td>MG5: 390 wavefunctions</td>
</tr>
<tr>
<td>pp &gt; tt~ + decays</td>
<td>1:34 h</td>
<td>22 s</td>
<td>p=ulu/~cl/c/<del>d/d</del>s/s/g W+ -&gt; du/du/ee/mumu/tau</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=ulu/~cl/c/<del>d/d</del>s/s/g ee/mumu</td>
<td>313</td>
<td>475</td>
<td></td>
</tr>
<tr>
<td>oo &gt; (u&gt;u-&gt;q2, Z1) (o-&gt;x1)</td>
<td>383 s (7:23 min)</td>
<td>5.2 s</td>
<td>sq=go/ul/s1/qi/du/sc/eve/mumu/text</td>
<td>1</td>
<td>67 FS decay chain single diagram</td>
<td></td>
</tr>
<tr>
<td>oo &gt; (u-&gt;nu+1) (o-&gt;nu+1)</td>
<td>76 s</td>
<td>5.6 s</td>
<td></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, i=ulu/~cl/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
DEVELOPMENT DIRECTIONS

BSM

Core v5

QCD

Tools
BSM : UFO

UFO = Universal Feynrules Output

- New FeynRules (python) output including color and lorentz structures. Output for golem-herwig-MG5 [Duhr et al]

- Automatic Helas Amplitude Generation for any new model (including effective theory) [P. de Aquino, W. Link, OM]

  - Output for fortran/C++
  - reproduces the SM and spin2

From Lagrangian to event generation in ANY model
For multijet generation (≥ 4 jets), Feynman diagram formalism expensive (factorial growth)

Helicity amplitude optimization (in MG4/5) reduces run time by factor ~ 10 for complex processes

Recursion relations (such as Berhreends-Giele) can reduce run time by additional orders of magnitude

MG5 perfect framework for implementation and development

Work started with exciting prospects in near future!
\[ \sigma^{\text{NLO}} = \int_m d^{(d)} \sigma^V + \int_{m+1} d^{(d)} \sigma^R + \int_m d^{(4)} \sigma^B \]
NLO: VIRTUALS

- Use MG to generate n+2 amplitudes to build NLO result (CutTools Technique) [V. Hirschi, R. Pittau, M. V. Garzielli, R. Frederix]

- Rely on external Tools (BlackHat, Golem, ...)
  - less generic
  - more possibilities for optimization
NLO: REAL

- **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]

mercredi 19 mai 2010
MADGRAPH 5 will significantly simplify the development for both real and virtual contributions.

- **Clear structure** -> easy to extract what is needed
- **Modular** -> easy to extend to add new features
- **Flexible** -> output not limited to fortran
<table>
<thead>
<tr>
<th></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><strong>MG</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MadGraph v4</td>
<td></td>
<td>Development v5</td>
<td></td>
<td>Release core MG v5</td>
<td></td>
</tr>
<tr>
<td><strong>ME</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MadEvent v4</td>
<td></td>
<td></td>
<td></td>
<td>Start dvlpt. ME v5</td>
<td></td>
</tr>
<tr>
<td><strong>BSM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FeynRules interface v4 + USRMOD2</td>
<td></td>
<td>Dvlpt. FR if v5 + autom. HELAS</td>
<td></td>
<td>Generic MG5</td>
<td></td>
</tr>
<tr>
<td><strong>NLO V</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dvlpt. CutTools</td>
<td></td>
<td>Physics results v4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NLO R</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dvl. MadFKS</td>
<td></td>
<td>Physics results MadFKS + stable MadDipole</td>
<td></td>
<td>Dvl.MadFKS v5</td>
<td></td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MadWeight, MadOnia, etc. released and stable for ME v4</td>
<td></td>
<td></td>
<td></td>
<td>Move to ME v5</td>
<td></td>
</tr>
</tbody>
</table>
CONCLUSION

- **MG/ME v4** is a mature, well established and stable code with many features for BSM and QCD physics and numerous peripheral tools.

- **MG5** is available with important and unprecedented improvements in all directions.

- Still many **new features** to come in the near future.

- A **tutorial of MG5** will be given in a few minutes.