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

Exp-TH communication

Very exotic models

Multi-jet samples

Exotic models

Effective theories

Decay chains

Matrix Elements

Advanced analysis techniques

Real corrections

Decay Packages

Merging ME/PS

Cluster/Grid computing

User friendliness

Testing/robustness

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

Leading order matrix element generation	$\leq$ 8 FS, <10000 diag Max. W+4 jet/$t\bar{t}$+3 jet
BSM, any renormalizable model	Yes
Decay Chains	Max 8 FS, slow
Color structures	Singlet/triplet/octet
Extended color structures (6, 27, ε^{ijk})	No
Effective theories (>4-particle vx)	No
Recursion relations for multijet generation	No
NLO real corrections	Yes
NLO loop calculations	In progress
Output in any language/format	Only Fortran

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

MadGraph 5

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

Leading order matrix element generation	No limitations except time W+5 jets/$t\bar{t}$+4 jets realistic
BSM, any renormalizable model	Yes
Decay Chains	No limitations, fast
Color structures	No limitations
Extended color structures (6, 27, ϵ^{ijk})	Available (not yet tested)
Effective theories (>4-particle vx)	Yes, no limitations
Recursion relations for multijets	To be implemented
NLO real corrections	To be implemented
NLO loop calculations	To be implemented
Output in any language/format	No limitations, Fortran (MG/ME 4) available

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

Process	MG4	MG5	Definitions	Subprocs (after combine)	Diagrams	Comments
pp > jjj	29.02 s	54.38 s	p, j=u/u~/c/c~/d/d~/s/s~/g	34	307	
pp > jj l+l-	341 s (5:41 min)	258 s (4:18 min)	p, j=u/u~/c/c~/d/d~/s/s~/g l+-=e+-/mu+-/ta+-	108	1216	
pp > jjj e+e-	2444 s (40:44 min)	993 s (16:33 min)	p, j=u/u~/c/c~/d/d~/s/s~b/b~/g	141	9012	
uu~>e+e-e+e-e+e-	772 s (12:52 min)	175 s (2:55 min)		1	3474	MG4: 3194 wavefunctions MG5: 301 wavefunctions
gg > ggggg	2788 s (46:28 min)	1049 s (17:29 min)		1	7245	MadGraph standalone output MG4: 3745 wavefunctions MG5: 898 wavefunctions
pp > jj (W+ > l+vl)	146 s (2:26 min)	70 s (1:10 min)	p, j=u/u~/c/c~/d/d~/s/s~/g l+=e+/mu+/ta+, vl=ve/vm/vt	82	304	
pp > t t~ with full decays	5640 s (1:34 h)	22.0 s	p=u/u~/c/c~/d/d~/s/s~/g W+/W->du/sc/eve/muvm/tavt	27	45	MG4: 12 proc defs MG5: single proc def
pp>sq sq	222 s (3:42 min)	286 s (4:46 min)	p=u/u~/c/c~/d/d~/s/s~/g sq=go/ul/ur/cl/cr/dl/dr/sl/sr+conj	313	475	
gg>(go>u(ul~>u~(n2>Zn1)))(go>ud~x1-)	383 s (7:23 min)	5.2 s		1	6	7 FS decay chain, single diagram
gg>(go>uu~n1)(go>uu~n1)	70 s	5.5 s		1	48	6 FS decay chain, mult.diag.
pp>(go>jjn1)(go>jjn1)	3 h - >>1 year	551 s (9:11 min)	p, j=u/u~/c/c~/d/d~/s/s~/g	144	11008	

Speed benchmarks

Full MadEvent subprocess directory output, including diagram drawing

Computer: Sony Vaio TZ

Process	MG4	MG5	Definitions	Subprocs (after combine)	Diagrams	Comments
pp > jjj	29.02 s	54.38 s	p, j=u/u~/c/c~/d/d~/s/s~/g	34	307	
pp > jj l+l-	341 s (5:41 min)	258 s (4:18 min)	p, j=u/u~/c/c~/d/d~/s/s~/g l+-=e+-/mu+-/ta+-	108	1216	
pp > jjj e+e-	40 min	16 min	p, j=u/u~/c/c~/d/d~/s/s~b/b~/g	141	9012	
uu~>e+e-e+e-e+e-	772 s (12:52 min)	175 s (2:55 min)		1	3474	MG4: 3194 wfs MG5: 301 wfs
uu~ > e+e-e+e-e+e-	38 s (46:28 min)	1049 s (17:29 min)		1	7245	MG5: 898 wavefunctions
pp > jj (W+ > l+vl)	146 s (2:26 min)	70 s (1:10 min)	p, j=u/u~/c/c~/d/d~/s/s~/g l+=e+/mu+/ta+, vl=ve/vm/vt	82	304	
pp > tt~ + decays	1:34 h	22 s	p=u/u~/c/c~/d/d~/s/s~/g W+/W->du/sc/eve/muvm/tavt	27	45	12 proc defs single proc def
pp>sq sq	222 s (3:42 min)	286 s (4:46 min)	p=u/u~/c/c~/d/d~/s/s~/g sq=go/ul/ur/cl/cr/dl/dr/sl/sr+conj	313	475	
gg>(go>u(ul~>u~(n2>Zn1)))(go>ud~x1-)	383 s (7:23 min)	5.2 s		1	6	7 FS decay chain, single diagram
gg>(go>uu~n1)(go>uu~n1)	70 s	5.5 s		1	48	6 FS decay chain, mult.diag.
pp > $(go>jj\chi^0)(go>jj\chi^0)$	>> 1 year	9 min	p, j=u/u~/c/c~/d/d~/s/s~/g	144	11008	

~ 2.5 times faster evaluation for produced matrix elements

Diagram examples

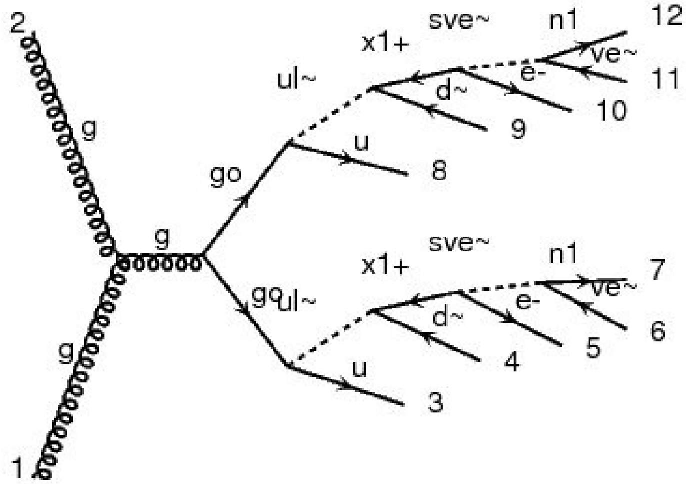


diagram 1

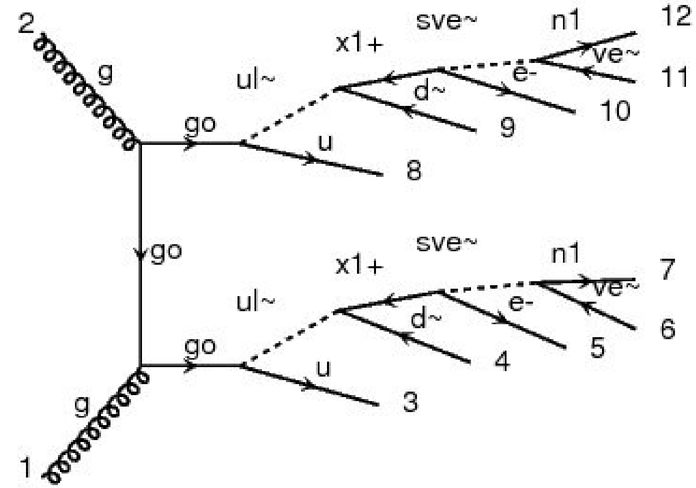


diagram 2

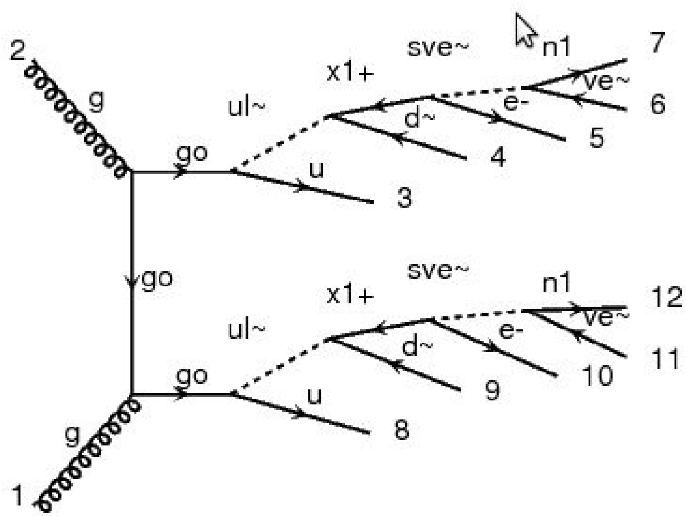
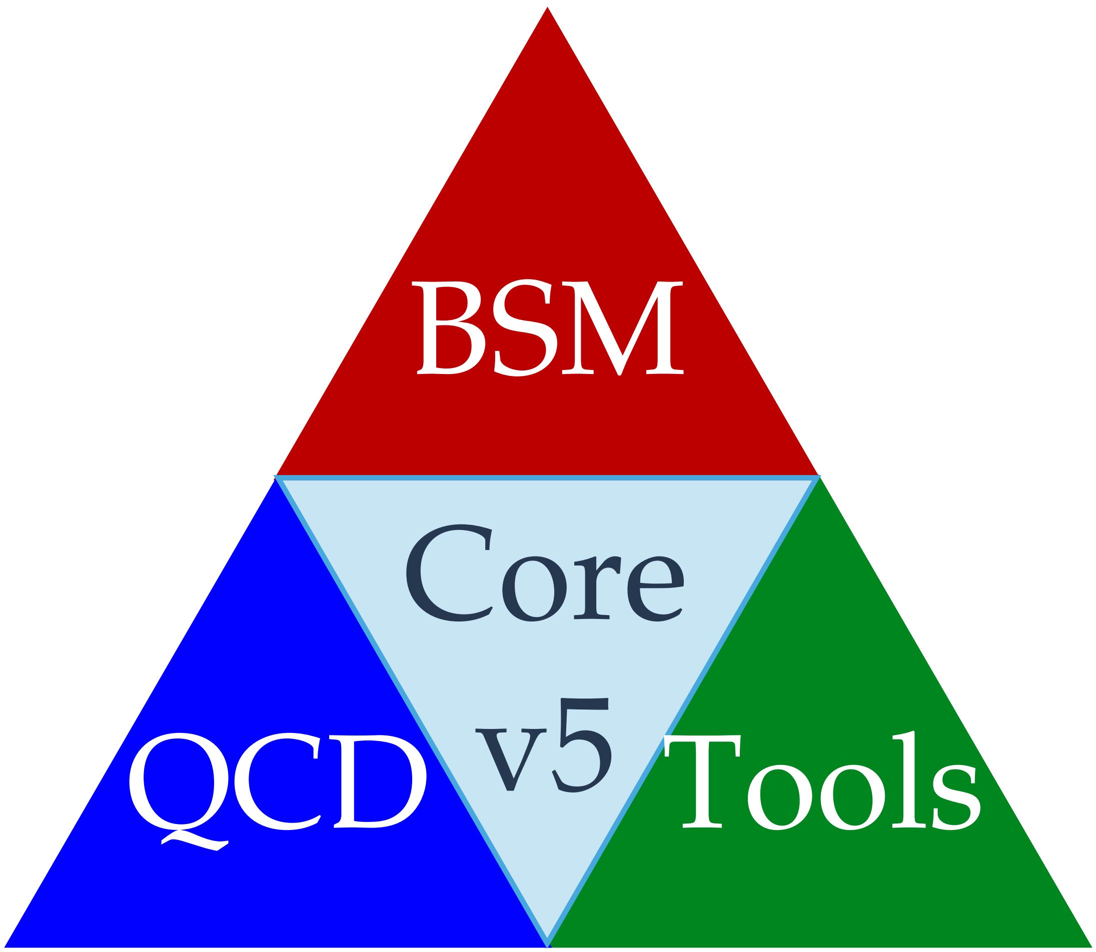


diagram 3

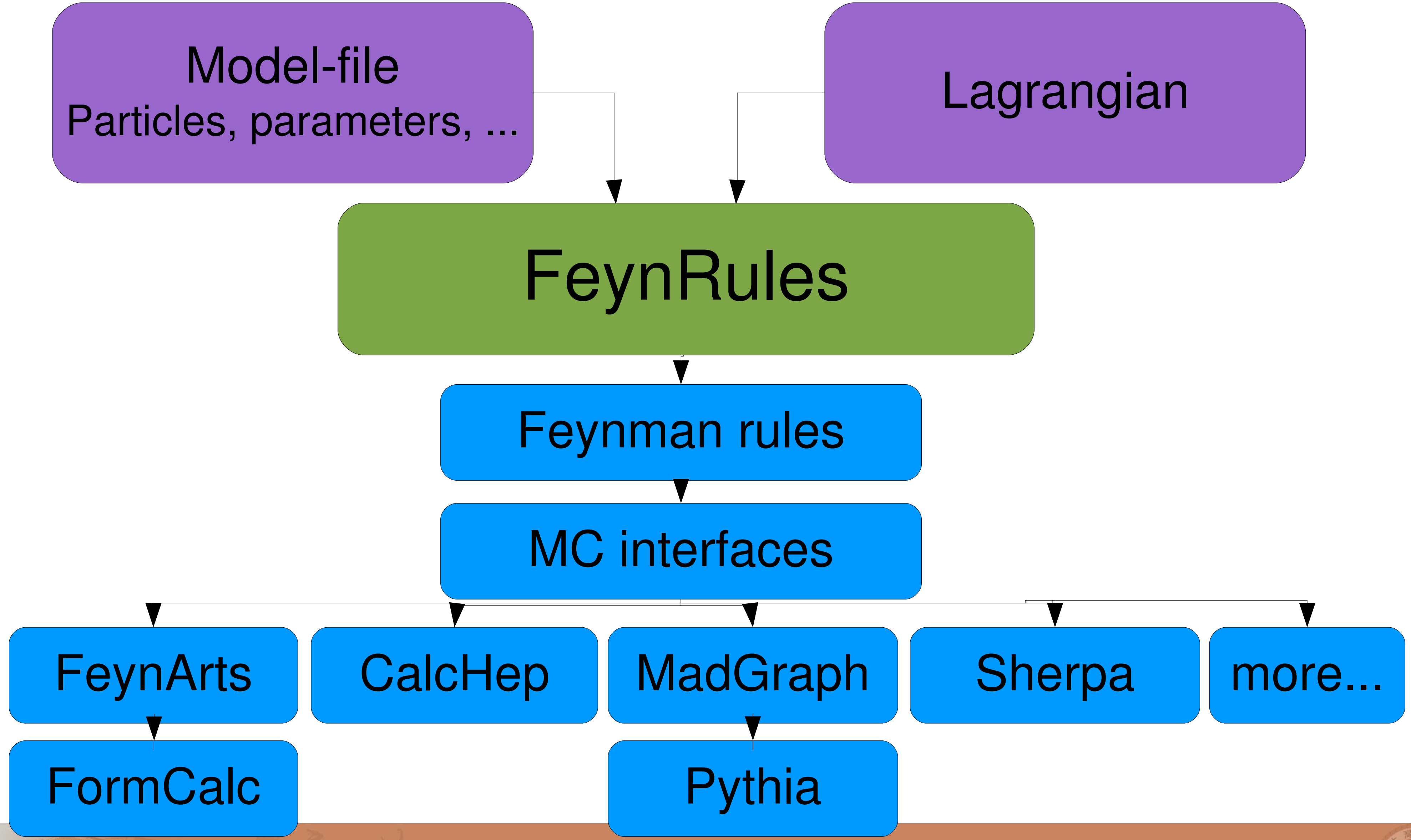
Process: g g > go go
  Decay: go > u ul~
    Decay: ul~ > d~ x1-
      Decay: x1- > e- sve~
        Decay: sve~ > ve~ n1
  Decay: go > u ul~
    Decay: ul~ > d~ x1-
      Decay: x1- > e- sve~
        Decay: sve~ > ve~ n1
(10 FS particles. Generation time: 5 s)

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++/…

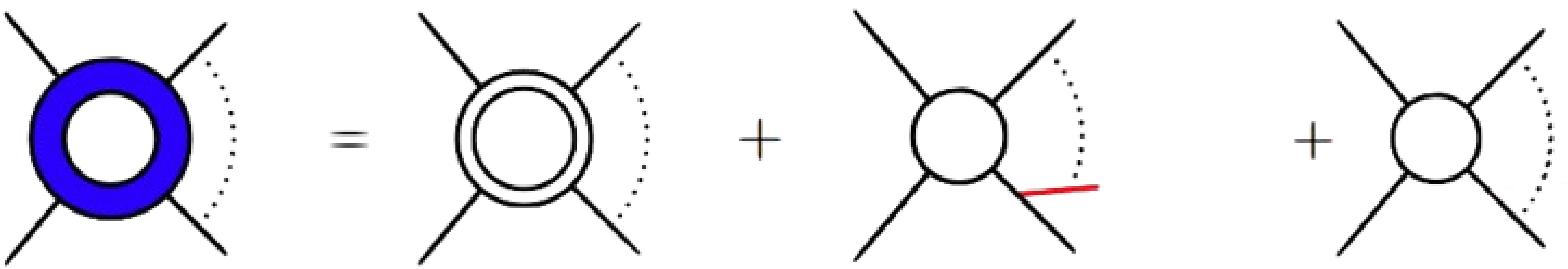
From Lagrangean to matrix elements/ decays/event generation in ANY model!

MadGraph 5 Multijets

- For multijet generation ($\gtrsim$ 4 jets), Feynman diagram formalism expensive (factorial growth)
- Helicity amplitude optimization (in MG4/5) reduces run times by factor ~ 10 for complex processes
- Recursion relations (such as Berhrends-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

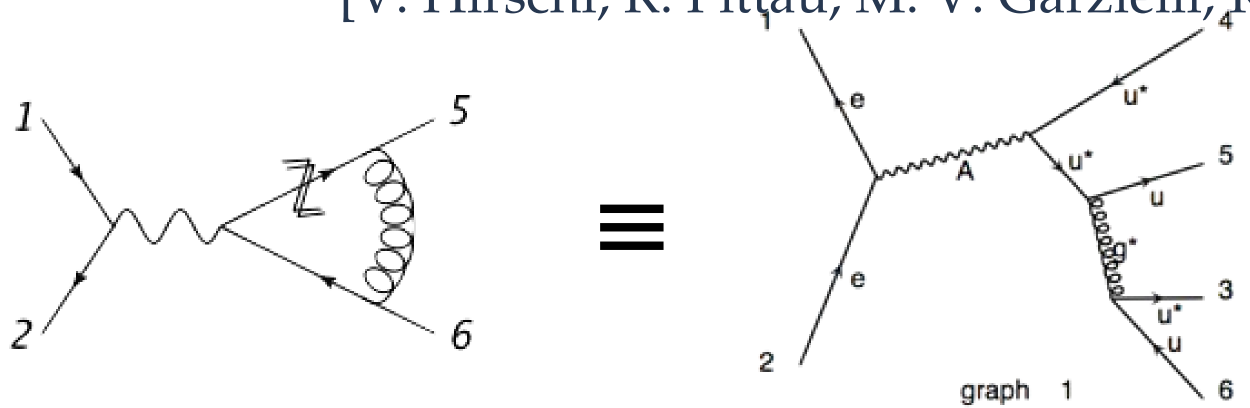
NLO Virtual Real Born



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

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- $\to$ 2 and 3 jets already checked (MG4). Advantages: valid for any BSM model
    [V. 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:

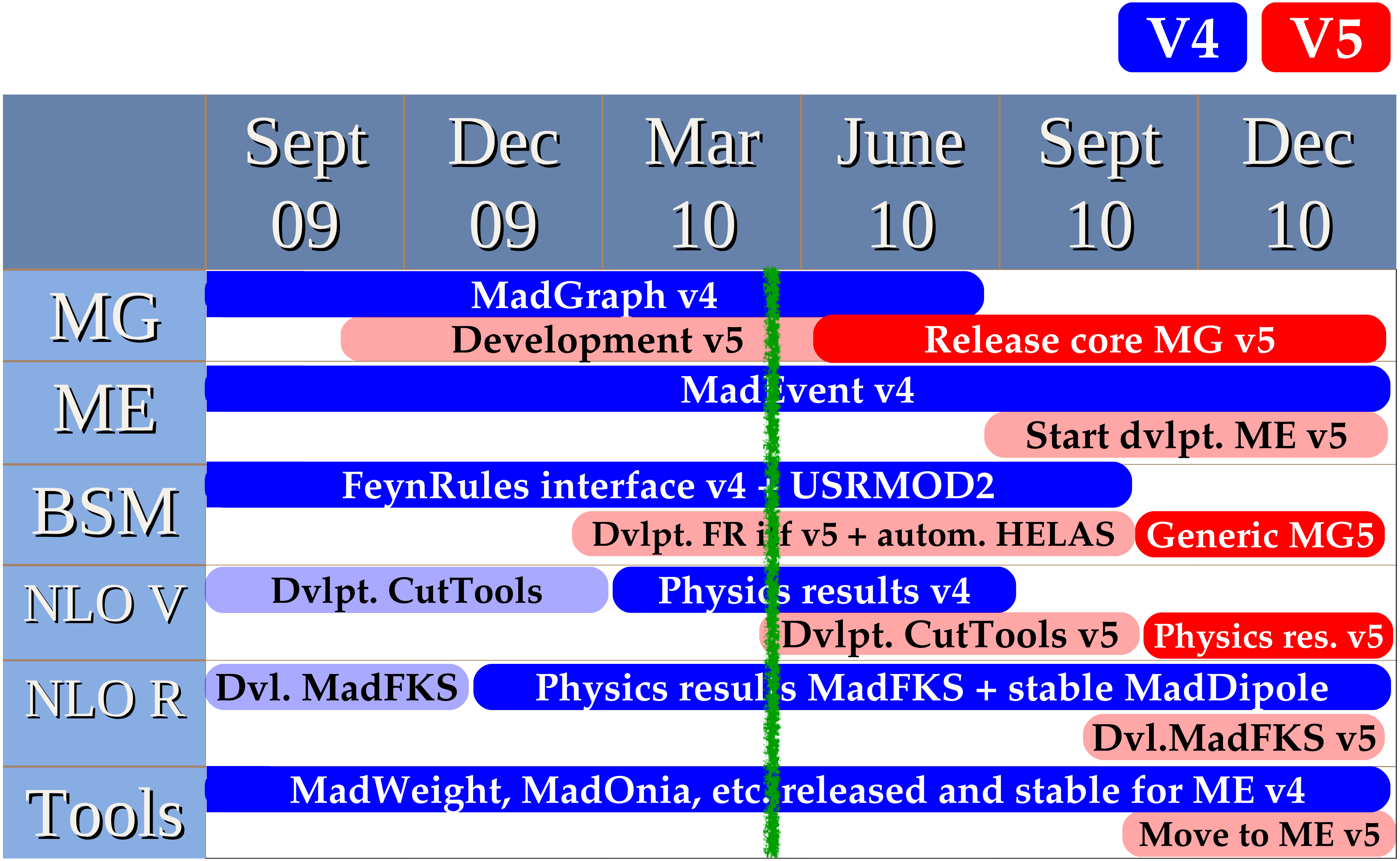
  [R. Frederix, S. Frixione, et al]

  - MadDipole: Catani-Seymour dipole substraction scheme, standalone implementation (TH), cancellation of singularities checked, and dipoles checked against MCFM
  - MadFKS: Frixione-Kunszt-Signer substraction 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

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



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!