Monte Carlo Orientation

OSET/Simplified Model Tutorial
CERN, Nov 3 2010

What do we want?

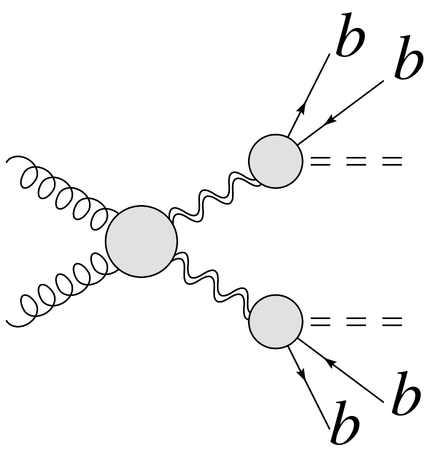
- **Flexibility** in specifying 2→2 process

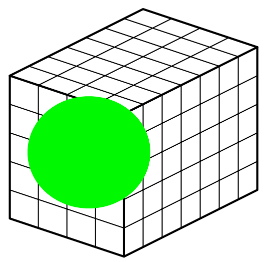
- Flexibility in decay modes, ability to tune branching ratios and masses separately

  ⇒ **don't want to work in model parameter space**
  ⇒ **Use decay tables (SLHA, Pythia native)**

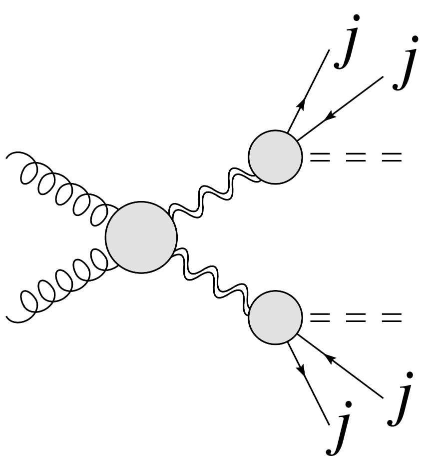
- Book-keeping to generate all allowed processes individually [MARMOSET]

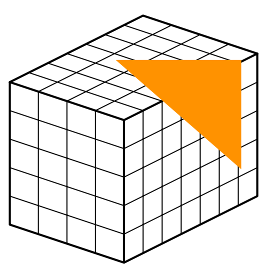
Organizing Process Sets



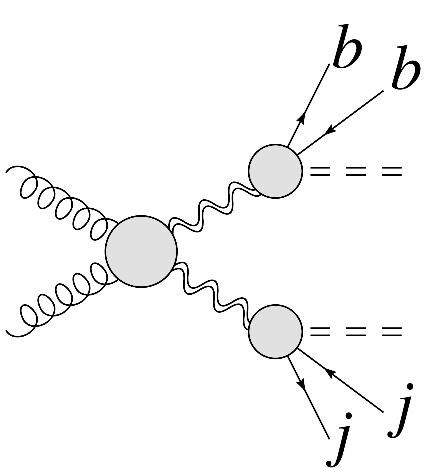
$=$  $\times\, \sigma_1 \times \mathrm{Br}_{1a} \times \mathrm{Br}_{1b}$

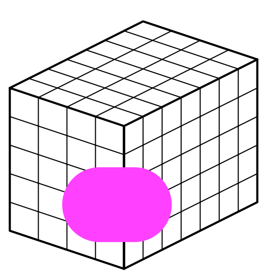
topo 1



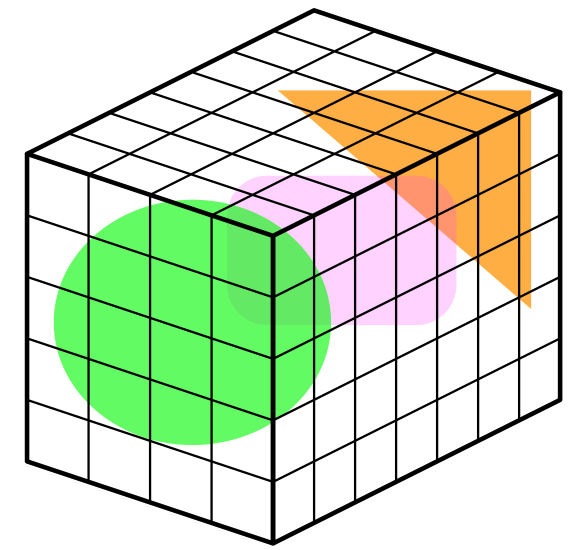
$=$  $\times\, \sigma_2 \times \mathrm{Br}_{2a} \times \mathrm{Br}_{2b}$

topo 2



$=$  $\times\, \sigma_3 \times \mathrm{Br}_{3a} \times \mathrm{Br}_{3b}$

topo 3



distributions

→ most invariant correlations
predictive power!

Parameters are masses, cross-sections, and branching ratios

can be varied by
reweighting events

Outline

1. Approaches to each of these three questions
   - Generating general hard process
   - Arbitrary decay chains (controlled indep. of mass)
   - Organize individual-process generation

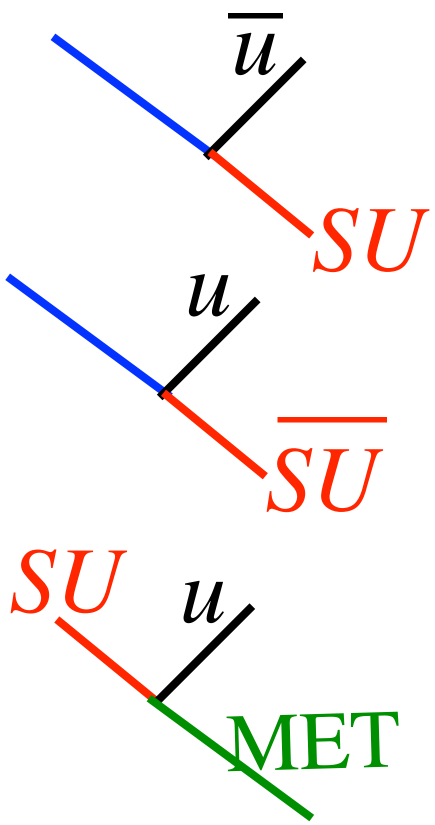
2. Implemented (and in-progress) ways of chaining them together
   - inputs & workflow
   - advantages/disadvantages of each
   - resources & (partial) examples

Hard Process Generation

- How do I generate arbitrary hard ($2 \rightarrow 2$, resonance) process?
  - built-in pythia process
    - limited choices; particular physics matrix element
    - see Pythia manual "subprocess summary table"
  - "generic" pythia process
    - anything allowed by quantum numbers; flat $2 \rightarrow 2$ matrix element or resonance [new in 6.4.24; previous implementation in MARMOSET]
  - MadGraph
    - flexible, physical matrix element (matched ISR)
    - Need to match onto another generator for multi-body decays

Decay Chain Generation

- How do I generate arbitrary hard (2→2, resonance) process?
  - built-in pythia process
  - “generic” pythia process
  - MadGraph
- How do I generate arbitrary decay chains?
  - Pythia-native decay tables (PYUPDA)




```
6000004  GL         0  2  0   600.00000     0.00010     0.00010  0.00000E+00  2  1
           1    0    0.500000         -2   6000003        0          0          0
           1    0    0.500000          2  -6000003        0          0          0
6000003  SU   SU~   2  1  1   540.00000     0.00010     0.00010  0.00000E+00  2  1
           1    0    1.000000    1000022         2        0          0          0
```

  - SLHA decay tables

```
DECAY 6000004   1.00
         0.50000            2       -2   6000003
         0.50000            2        2  -6000003
DECAY 6000003   1.00
          1.00000            2  1000022          2
```

[Pretty much functionally equivalent; SLHA now standard, more easily integrated in experiments’ software frameworks]

Book-Keeping

- Often useful to separately generate processes with different decay modes…
- Nice not to worry about formatting, have “natural language” syntax
  - particularly for asymmetric decay chains.

This is where MARMOSET comes in as a wrapper…
Easiest to explain via syntax examples.

[used to be also for easy generic process generation – now supported by Pythia]

OSET Language 1

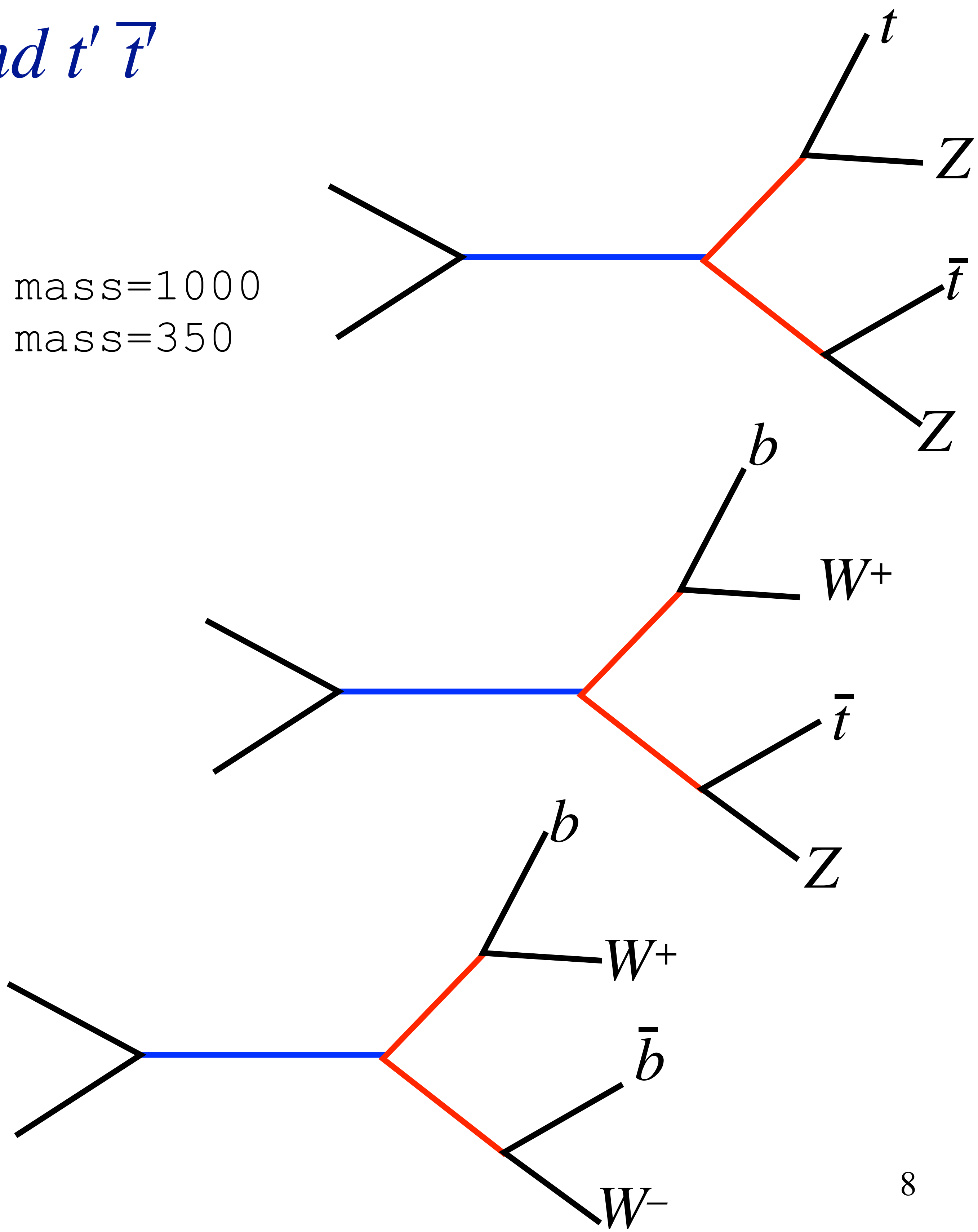
- *Z' that decays to e+e–, $u\bar{u}$, and $t'\,\bar{t}'$*
- *t' pair-production*

```
#new particles
Zprime : charge=0 color=0 mass=1000
T4     : charge=2 color=3 mass=350

#decays
Zprime > e+ e-
Zprime > u ubar
Zprime > T4 T4~

T4 > t Z0
T4 > b W+

#production
u ubar > Zprime
g g > T4 T4~
```



OSET Language II

```
##### New Particles #####
MPT : pdg=1000022 charge=0 color=0 mass=100

Q1          : charge=2 color=3 mass=800

X20         : charge=0 color=0 mass=440
X2+ X2-     : charge=3 color=0 mass=440

Smu- Smu+   : charge=-3 color=0 mass=300
Se- Se+     : charge=-3 color=0 mass=300
Snu  Snu~   : charge=0 color=0 mass=300

##### Production ####
g g    > Q1~ Q1

##### Q1 decay ######
Q1 > X20 u
Q1 > X2+ d

##### X2 decay ######
X20 > Se- e+    $ bX0l
X20 > Smu- mu+  $ bX0l
X20 > Snu~ nu_e $ bX0nu
```

Generation Paths 1
Madgraph→Pythia

- MadGraph hard process → SLHA-based decays in Pythia

  http://madgraph.hep.uiuc.edu/

- Wrapper scripts to help with generation for particular simplified models

  http://www.lhcnewphysics.org/wiki/index.php?title=SimplifiedModels:GluinoOneStage

  questions on script generation/use: Mariangela Lisanti

- Pros: ISR, general process w/ specific physical ME
- Cons: less standard to interface w/ ATLAS/CMS frameworks, separation of processes by hand (if desired)
  [interface to MARMOSET was partially developed, could be resumed if interest]

Generation Paths 1.5
Madgraph→BRIDGE→Pythia

- MadGraph hard process → BRIDGE decays

  http://www.lepp.cornell.edu/Research/TPP/BridgeSoftware.html

- Example [Lisanti?]

- Pros: ISR, general process w/ specific physical ME, keep spin correlations of particular model in **decays** (this is where they matter most!)
- Cons: even less standard interface w/ ATLAS/CMS frameworks, but can use LHE

Generation Paths 2
Pythia-Native 'Generic' Process

- Pythia 6.4.24 "generic process" plus SLHA decay table

  questions: ask Steve Mrenna!

  input 2→2 initial and final states as "decay modes" of special particle

  http://home.fnal.gov/~mrenna/topology/topology.html

- Pros: Universal technique; Standard interface w/ ATLAS/CMS frameworks can take SLHA input
- Cons: separation of processes by hand (if desired)
  [interface to MARMOSET will be developed very soon for this!]
- Cons (debatable): Flat matrix element not literally equivalent to any physical model
  - NOTE: same workflow with a standard Pythia process should also work.

Generation Paths 3
MARMOSET

- Pythia 6.4.24 “generic process” plus SLHA decay table

  questions: ask Steve Mrenna!

  input 2→2 initial and final states as “decay modes” of special particle

  http://home.fnal.gov/~mrenna/topology/topology.html

- Pros: Separation of processes is automatic
- Cons:
  - separation of processes is **mandatory**
  - requires *intermediate* special input file → CMS interface cumbersome – through stored LHE events

    [both to be fixed in near-future revision to use “Path 2” internally]
  - (debatable) Flat matrix element not literally equivalent to any physical model

What to choose?

- Priority: physical ME — MadGraph or (for 'canonical' examples) Pythia standard process
- Priority: generality & simple framework-interface — Pythia 6.4.24 w/ either generic or standard builtin processes
- Priority: simple user-input, or separation of processes — MARMOSET (stay tuned for update!)

What to choose?

- Priority: physical ME — MadGraph or (for 'canonical' examples) Pythia standard process
- Priority: generality & simple framework-interface — Pythia 6.4.24 w/ either generic or standard builtin processes
- Priority: simple user-input, or separation of processes — MARMOSET (stay tuned for update!)

These two will play ***very*** nicely together

[Steve: Pythia generic production slides]

DECISION POINT

Hands-on MC or Discussion?

Downloads
MARMOSET

- Can follow tutorial at

  http://www.marmoset-mc.net/wiki/doku.php?id=lhe_install_workflow

  (which also contains some assistance for generating an OSET of t-tbar)

Downloads
Pythia-Native 'Generic' Process

- Pythia 6.4.24

  http://www.hepforge.org/archive/pythia6/pythia-6.4.24.f.gz

- Example main file, card file, SLHA file

  http://ntoro.stanford.edu/~toro/slha/

Downloads:
Madgraph→Pythia

- MadGraph hard process → SLHA-based decays in Pythia

  http://madgraph.hep.uiuc.edu/

- Wrapper scripts to help with generation for particular simplified models

  http://www.lhcnewphysics.org/wiki/index.php?title=SimplifiedModels:GluinoOneStage