



# Some random thoughts on the use of fast simulators

Fast Simulators for the LHC, June 8<sup>th</sup> 2012

Martin White (with help/thoughts from Andy Buckley, Nicholas Rodd and Kenji Hamano)

- I've used a variety of “public” fast simulators over the years
  - AcerDet
  - ATLFAST 1 (really this is just modified AcerDet, and it used to be public)
  - DELPHES
- I mostly use public fast simulators for model interpretation
  - i.e. *up to a point* I don't care if they reproduce every detail of ATLAS or CMS
  - the only important thing is to get a good description of the efficiency

# Thoughts on AcerDet

- I'm not sure anyone uses this anymore
- Last time I used it, I ran into problems on 64 bit systems due to the use of PAW ntuples
  - there are hardcoded memory limits in odd bits of CERNLIB
  - things go haywire on 64 bit architectures when they are exceeded
  - I hacked out the bit that made ntuples, and did everything from the original common blocks
- AcerDet has nice features (if you are not FORTRAN-phobic):
  - easy to integrate with the main event loop in HERWIG
  - very quick to run
  - it should in theory be very close to ATLFAST I

# One recent example of fast simulator use

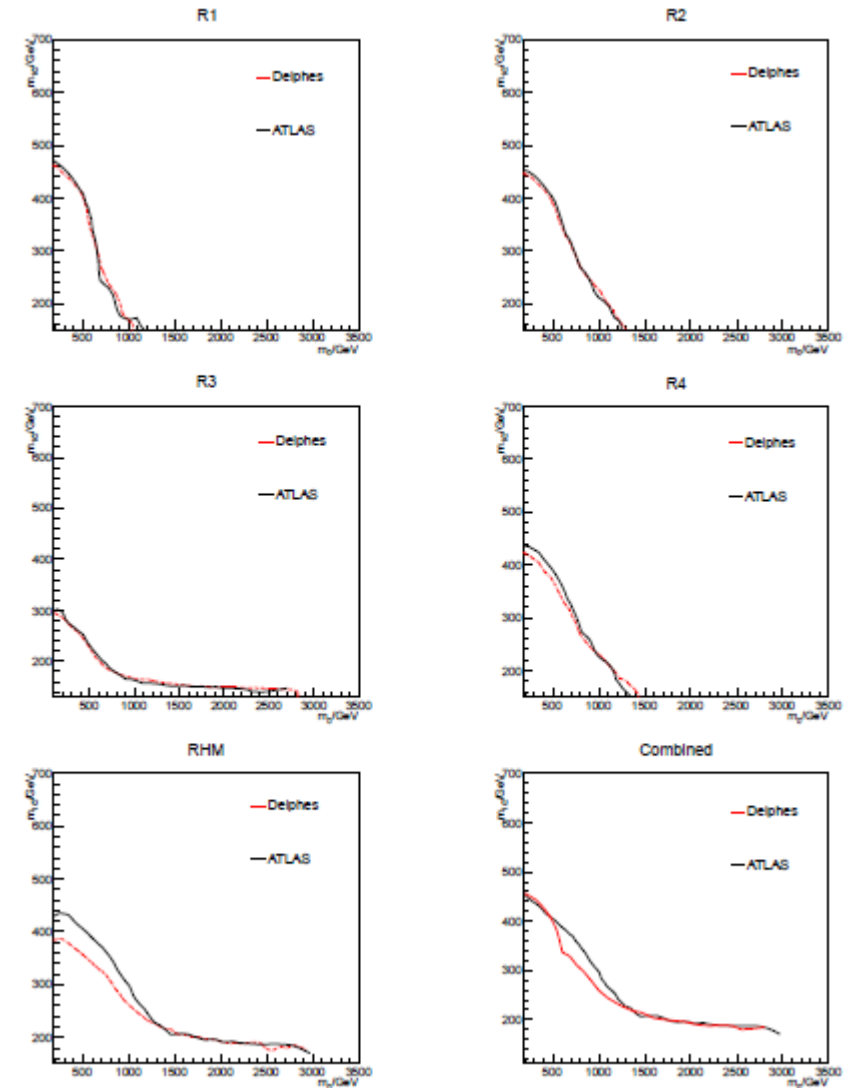
• Constraints on “stochastic supersymmetry” parameters from the ATLAS 0 lepton SUSY search (Phys.Rev. D85 (2012) 075023)

- this is a two parameter model developed by Melbourne theorists (Kobakhidze, Pesor and Volkas)
- used Herwig++ and DELPHES to simulate a grid of points in the model space
- applied the ATLAS 0 lepton SUSY search results

Signal Region	$\geq 2$ -jet	$\geq 3$ -jet	$\geq 4$ -jet	High mass
$E_T^{\text{miss}}$	$> 130$	$> 130$	$> 130$	$> 130$
Leading jet $p_T$	$> 130$	$> 130$	$> 130$	$> 130$
Second jet $p_T$	$> 40$	$> 40$	$> 40$	$> 80$
Third jet $p_T$	–	$> 40$	$> 40$	$> 80$
Fourth jet $p_T$	–	–	$> 40$	$> 80$
$\Delta\phi(\text{jet}, \vec{P}_T^{\text{miss}})_{\text{min}}$	$> 0.4$	$> 0.4$	$> 0.4$	$> 0.4$
$E_T^{\text{miss}}/m_{\text{eff}}$	$> 0.3$	$> 0.25$	$> 0.25$	$> 0.2$
$m_{\text{eff}}$	$> 1000$	$> 1000$	$> 500/1000$	$> 1100$

Process	Signal Region				
	$\geq 2$ -jet	$\geq 3$ -jet	$\geq 4$ -jet, $m_{\text{eff}} > 500$ GeV	$\geq 4$ -jet, $m_{\text{eff}} > 1000$ GeV	High mass
$Z/\gamma$ +jets	$32.3 \pm 2.6 \pm 6.9$	$25.5 \pm 2.6 \pm 4.9$	$209 \pm 9 \pm 38$	$16.2 \pm 2.2 \pm 3.7$	$3.3 \pm 1.0 \pm 1.3$
$W$ +jets	$26.4 \pm 4.0 \pm 6.7$	$22.6 \pm 3.5 \pm 5.6$	$349 \pm 30 \pm 122$	$13.0 \pm 2.2 \pm 4.7$	$2.1 \pm 0.8 \pm 1.1$
$t\bar{t}$ + single top	$3.4 \pm 1.6 \pm 1.6$	$5.9 \pm 2.0 \pm 2.2$	$425 \pm 39 \pm 84$	$4.0 \pm 1.3 \pm 2.0$	$5.7 \pm 1.8 \pm 1.9$
QCD multi-jet	$0.22 \pm 0.06 \pm 0.24$	$0.92 \pm 0.12 \pm 0.46$	$34 \pm 2 \pm 29$	$0.73 \pm 0.14 \pm 0.50$	$2.10 \pm 0.37 \pm 0.82$
Total	$62.4 \pm 4.4 \pm 9.3$	$54.9 \pm 3.9 \pm 7.1$	$1015 \pm 41 \pm 144$	$33.9 \pm 2.9 \pm 6.2$	$13.1 \pm 1.9 \pm 2.5$
Data	58	59	1118	40	18

- I first reproduced the ATLAS CMSSM limits in each channel
  - this owes a heavy debt to Allanach's AMSB paper (JHEP 1111 (2011) 132)
    - i.e. use limits on  $\sigma \times A \times \epsilon$
    - DELPHES gives very good results once a “fudge factor” is applied in each channel (essentially this is playing the role of the signal systematic uncertainty)
    - the high multiplicity channel shows the greatest deviation (probably not a detector simulation issue)



# A more tricky recent example

- A Bayesian analysis of the CMSSM using the same ATLAS results (arxiv:1205.1568)
- This required evaluating the simulation results for millions of points in the parameter space
  - this is not feasible directly!
- Instead, we generated a grid of training data and used a Bayesian neural net to interpolate between the points
  - required simulating only 10,000 points
  - get a “superfast” likelihood calculator for LHC studies
  - see also the proof of principle paper: *Comput.Phys.Commun.* 183 (2012) 960-970

# Thoughts on DELPHES

- *Many thanks to the DELPHES authors for providing a nice tool!*
  - coding up the ATLAS analyses was very easy
  - the examples provided are very useful (i.e. they do most of the work for you)
  - the manual is excellent
  - it is way more up to date and easy to compile than AcerDet
- For small projects with little total simulation it works well
- A simple change would greatly improve DELPHES for large simulation projects:
  - at present we had to run Herwig++ to produce a very large HepMC output file
  - this then has to be read *in full* by DELPHES at the start of the run
  - this is unnecessary (each event is of course independent)
- A better solution:
  - allow DELPHES to read an incoming MC stream (from a pipe?)
  - can then put the analysis code in the main event loop
  - c.f. HERWIG and AcerDet
  - much quicker jobs with lower disk space requirements

# More thoughts on DELPHES

- Am currently using Pythia and DELPHES to reproduce an ATLAS diphoton search
  - we reproduce the ATLAS exclusion limit at the truth level (with appropriate scaling)
  - we get the wrong shape when we use DELPHES
- The problem is almost certainly due to the photon isolation
  - our DELPHES “photons” currently include lots of photons from neutral meson decays
  - need to remove jet/photon overlaps

- ATLAS do this:

Photon candidates were required to be within  $|\eta| < 1.81$ , the value being chosen by an optimisation of the signal acceptance versus background rejection, and to be outside the transition region  $1.37 < |\eta| < 1.52$  between the barrel and the end-cap calorimeters. The analysis used “loose” and “tight” photon selections [42]. The loose photon selection includes a limit on the fraction of the energy deposit in the hadronic calorimeter as well as a requirement that the transverse width of the shower, measured in the middle layer of the EM calorimeter, be consistent with the narrow shape expected for an EM shower. The tight photon selection additionally uses shape information from the first layer to distinguish between isolated photons and photons from the decay of neutral mesons.

- Have an additional isolation cut:

ET deposit within  $\Delta R=0.2$  of the candidate cluster minus the candidate energy must be less than 5 GeV

- Am still defining a workaround for this, but:
  - DELPHES does not include isolation information for photons
  - a simple cone-based overlap removal may not work here



# General thoughts for the future

- Lots of ATLAS object identification cuts cannot be done in any public simulation
  - “tight” and “loose” criteria using shower shape information
  - et cone based isolation information (sensitive to pileup)
  - complicated trigger strategies (i.e. going below a turn on curve, or a plateau efficiency that isn't 100% but is not specified)
- No general public simulation should aim to get these things right
  - I support your call for efficiency functions or validated simulations from ATLAS/CMS
- Multivariate analyses are becoming more popular within collaborations
  - e.g. use a neural net, boosted decision tree, etc, to classify signal and background events
  - gives a highly non-trivial cut that would be hard to implement outside ATLAS/CMS
  - not really a simulation issue, but it might be worth thinking about how these should be handled in terms of published information