

SGV - a fast detector simulation

Mikael Berggren, LPNHE Paris

The *Simulation a Grande Vitesse* is:

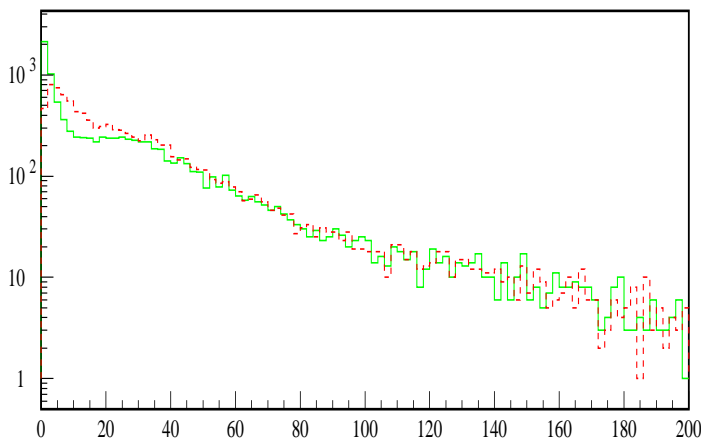
- **Simple** - First events generated minutes after down-load
- **General** - Easy to change detector configuration
- **Documented** - UG and Reference manual on the Web (500+ pages in printed version)
- **Much used for existing experiment** - DELPHI
- **Tested** - Billions of events produced without problems within the DELPHI SUSY searches.
- **Fault tolerant** - Error-trapping and re-start
- **Portable** - to VMS, HP-UX, AIX, Digital UNIX, Linux,...
- **Complete**: Tracking with full error matrix, calorimetry with shower merging, particle identification, impact parameters,...
- **Compares** well with real data

And

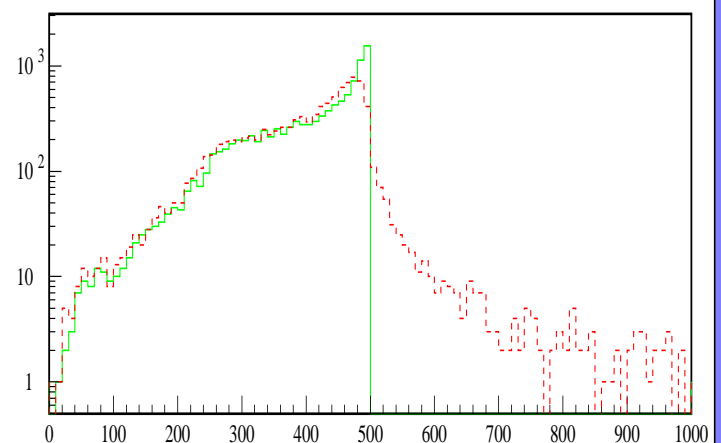
- **FAST** - 240 ee \rightarrow stau stau events/s at LEP200 or 20 ee \rightarrow stop stop events/s at the LC (TESLA at 500 GeV) on an Alpha.
- Fully implemented for **TESLA** - used for physics studies.
- **Needed** for Physics studies :

Solid green: Generated, Dashed red : SGV

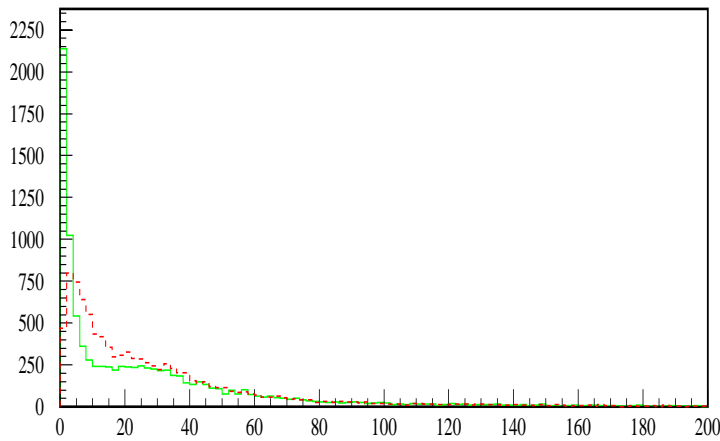
Solid green: Generated, Dashed red : SGV



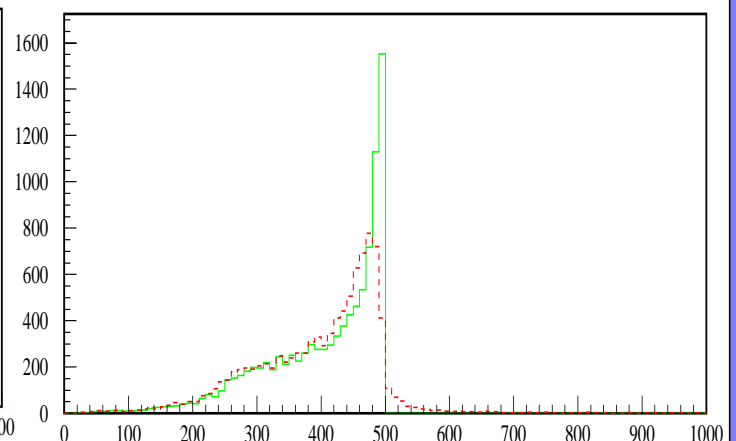
Missing P_T of detected particles



Total Energy of detected particles



Missing P_T of detected particles

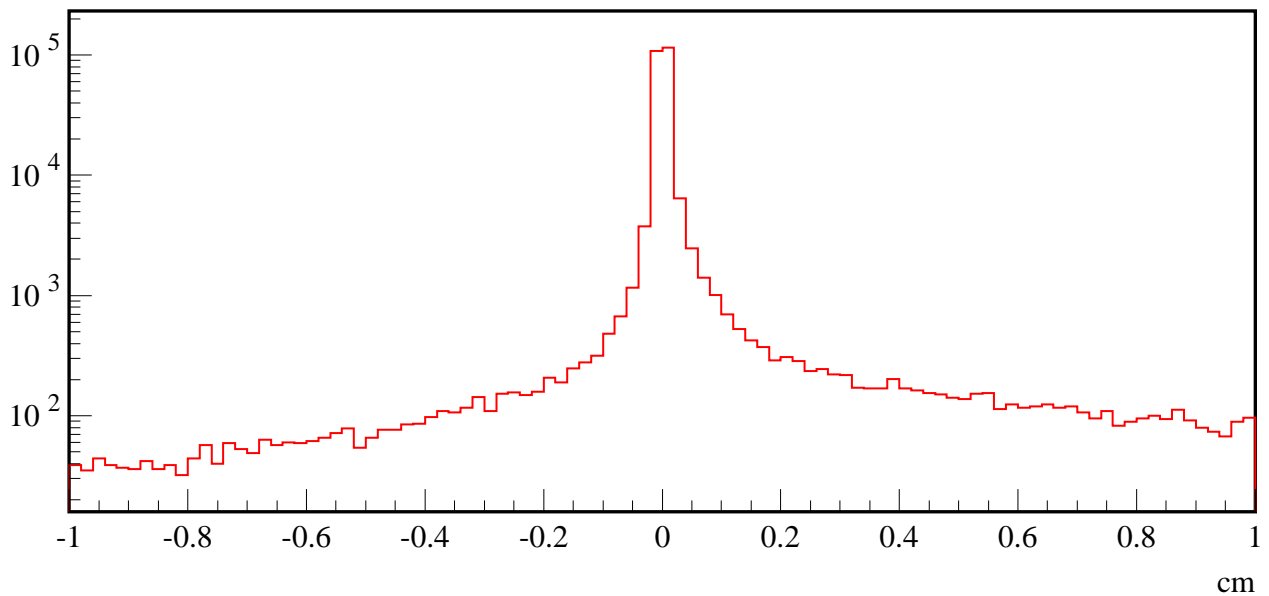


Total Energy of detected particles

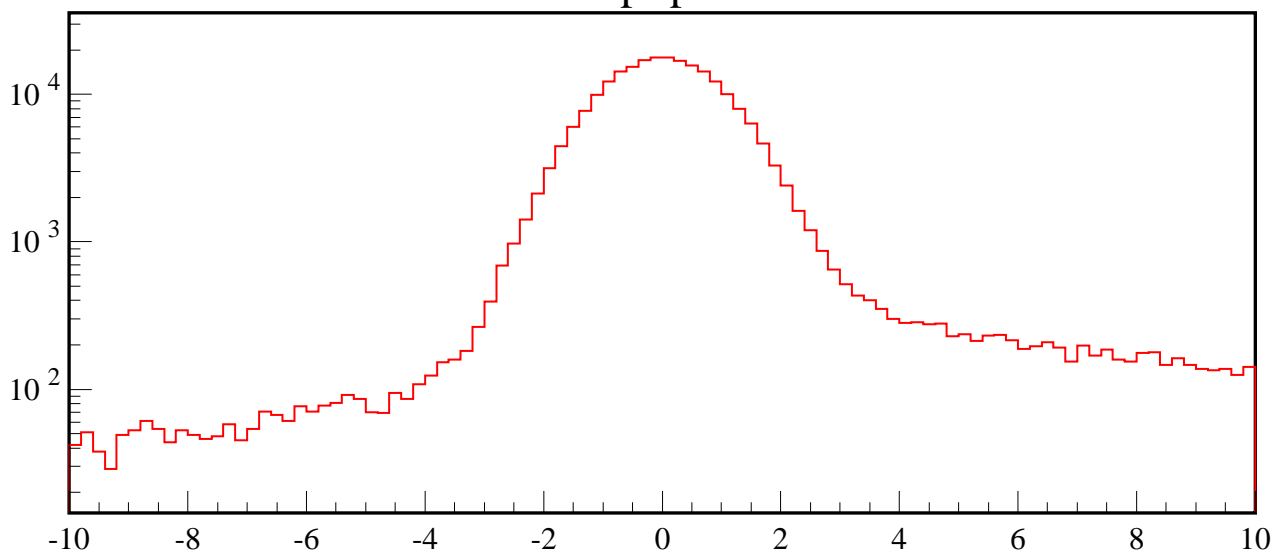
Impact parameters

Can't do these at all at generator level

SGV simulation of TESLA



$ip\ rphi$



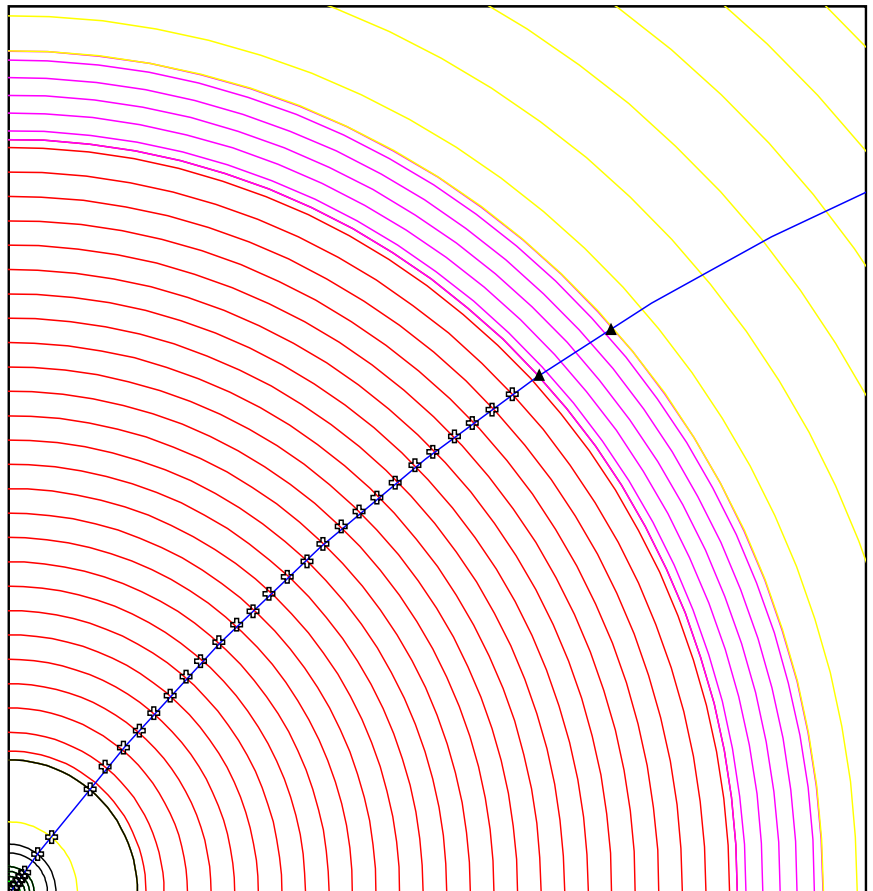
$ip\ rphi / \sigma$

How does it work?

SGV is a *machine to calculate covariance matrices*

Tracking:

- **Follow** track-helix through the detector, to find **what layers are hit** by the particle.



- From this, **calculate covariance** matrix at perigee, including effects of **material**, **measurement** errors and **extrapolation**.
- Smear perigee parameters accordingly, with Choleski decomposition (takes **all correlations** into account)
- Information on **hit-pattern** accessible to analysis. **Coordinates** of hits accessible.

Calorimeters:

- **Follow** particle to intersection with **calorimeters**
- Decide **how the detectors will act**: MIP, EM-shower, hadronic shower, below threshold, etc.
- Simulate response from **parameters**.
- **Merge** close showers
- Easy to **plug in** other (more sophisticated) shower-simulation

Other stuff:

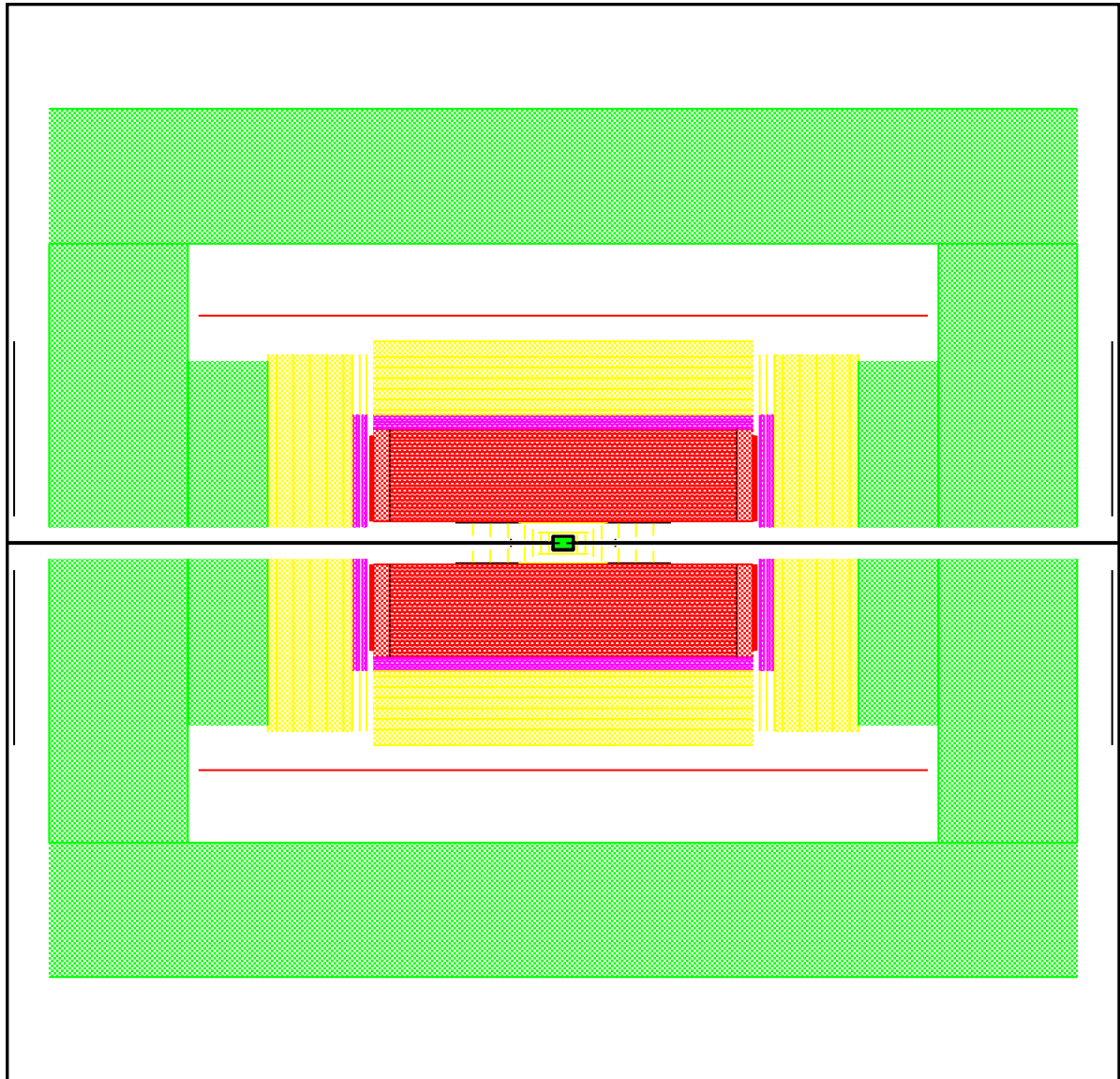
- **EM-interactions** in detector material simulated
- Plug-ins for **particle identification**, track-finding **efficiencies**,...
- **Scintillators** and **Taggers**

Some words on how to use **SGV**

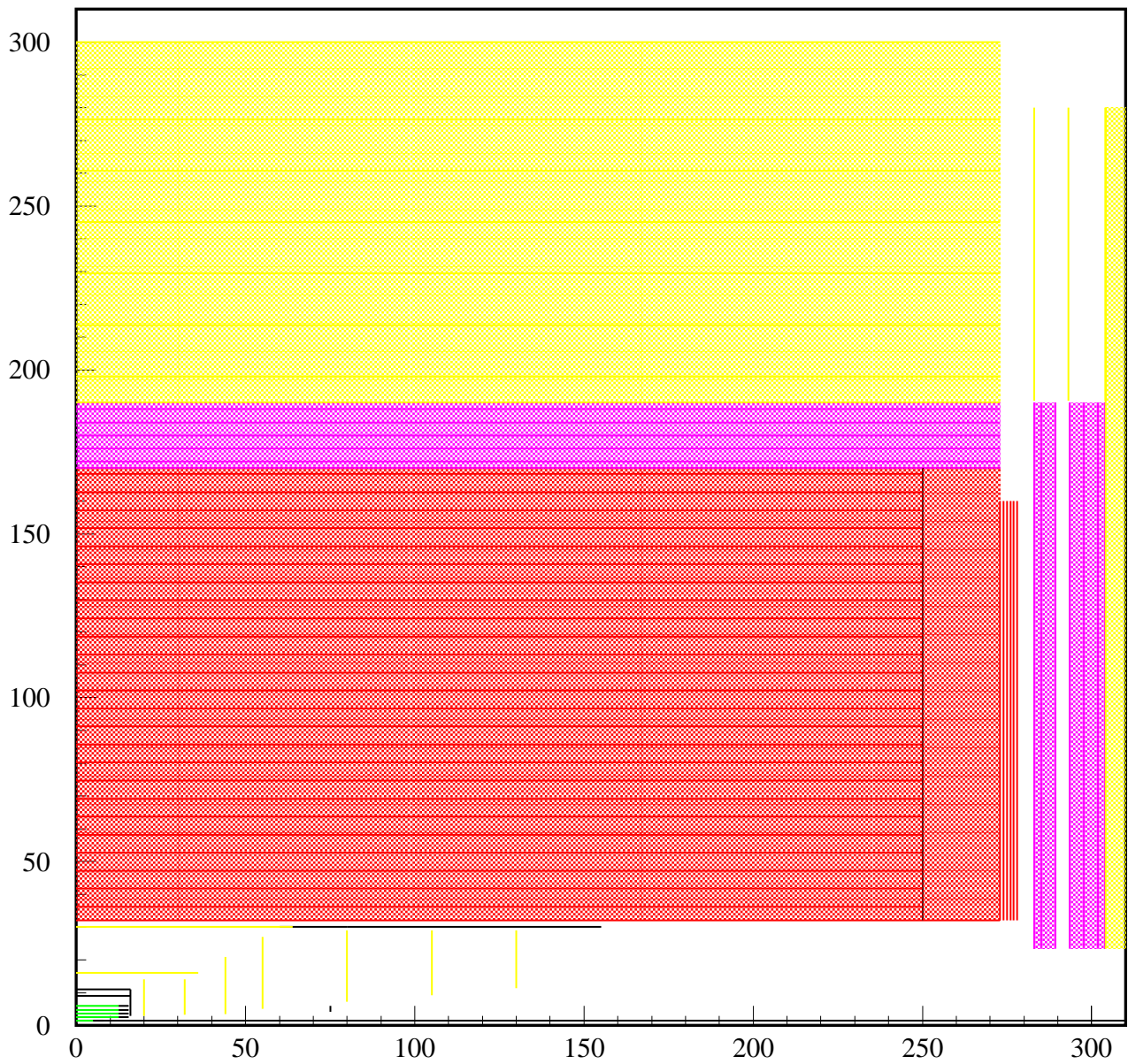
- **Generators**: Interfaces to **PYTHIA**, **JETSET** and **SUSYGEN** included. Interface to **PYTHIA 6** (double precision version) recently added. **Easy** to interface to others (but depends on how well-structured the generator is...).

- **Detector geometry**: Given as **planes** and **cylinders**, with attributes attached (measurement, material, names,...). Read from a **human-readable** ASCII-file (**ex**). Simple **visualisation** of the detector included. **Up to three detectors** can be loaded simultaneously, and will be **looped over** event by event.

Detector example: TESLA TDR



A bit closer ...



- **User data** : Delivered in COMMON blocks as extended **4-vectors** , **track parameters** with correlations, **calorimetric clusters** . When relevant, **true values** also given. Auxilliary information on **particle history**, **detector-elements** used etc. Global variables also given.

- **Analysis tasks** : Information on **jets**, **event-shapes**, **secondary vertecies**, **impact parameters** and **b-tagging** filled by calls to routines, **included in SGV**. **Access routines** give an easy interface to the **detector geometry**.

General structure of **SGV**

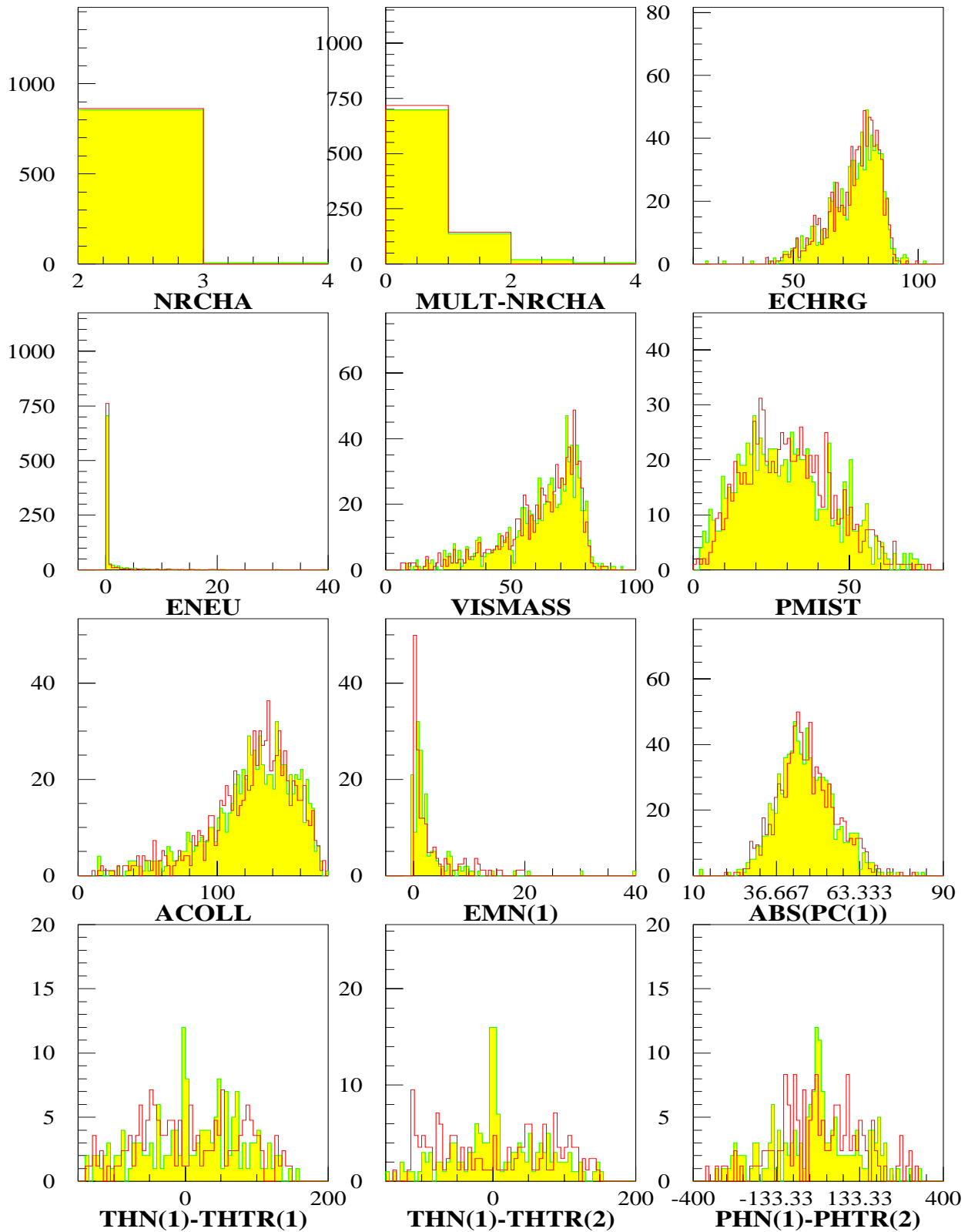
- **SGV** is made of **six** loosely connected parts:
 - The **Steering**, which takes care of initialisation and ending, and runs the event loop.
 - In the event loop,
 - The **Event Generator**
 - The **Detector Simulation**
 - The **Event Dispatcher**is called.
 - The **Detector Simulation** calls
 - The **Covariance Matrix Machine**
 - The **Event Dispatcher** usually simply trasferes the data to the **Event Analysis**, and sends a message when done to trigger the analysis of the generated event. However, instead the data can be written to an external file (RZ/FZ), or eg. be written to shared memory to be accessed by an independent process (eg. a presenter of some kind)
- The communication between the parts is by arguments only, except for the **Event Generator → Detector Simulation** (which is via the Lund common blocks), and the **Detector Simulation → Event Dispatcher**
- Also the steering is via arguments only, in the form of a control-vector.

SOME EXAMPLES.....

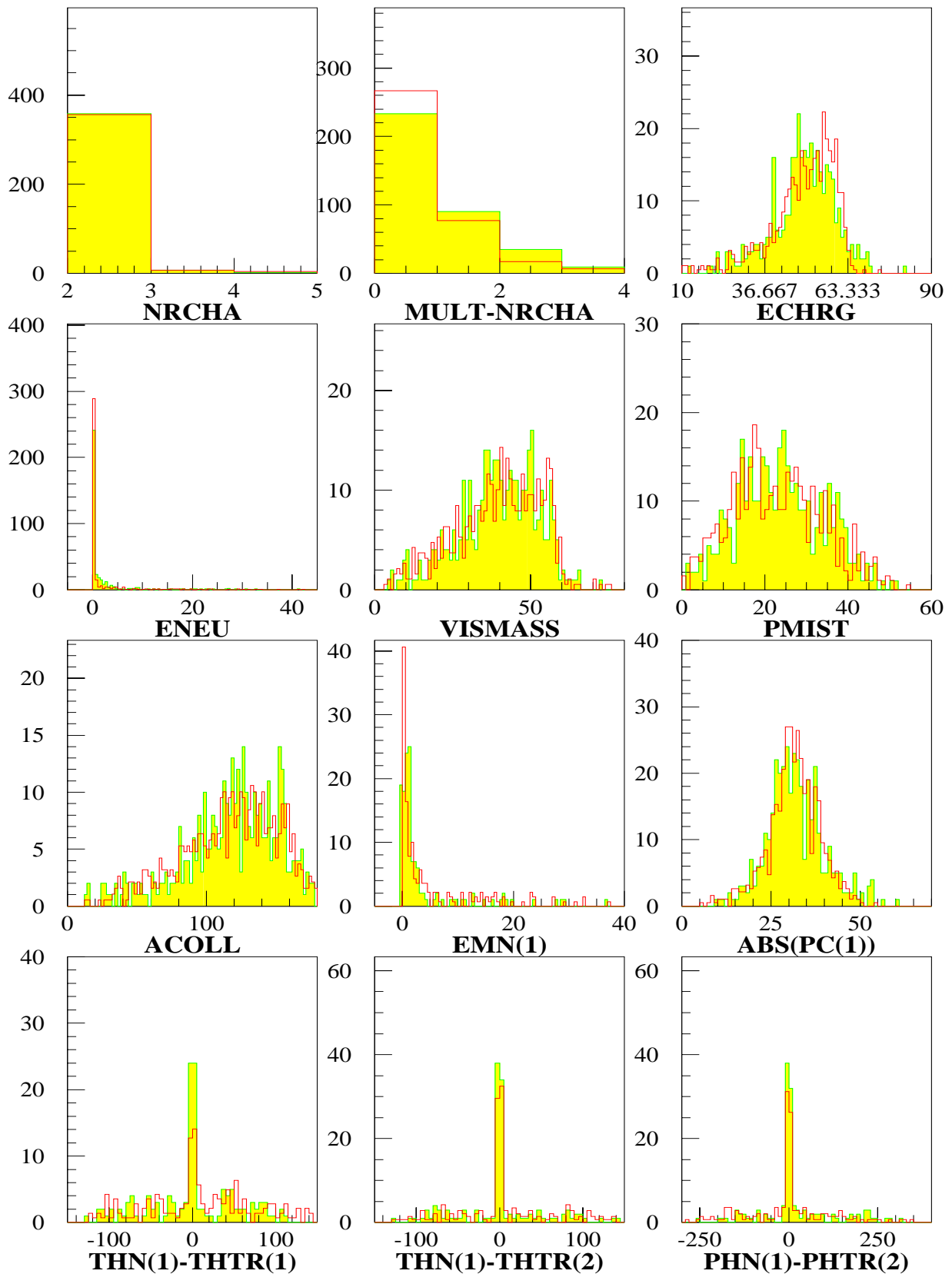
from the DELPHI neutralino analysis in different topologies.

Compare : **SGV (red)** with **DELPHI full simulation (yellow)**

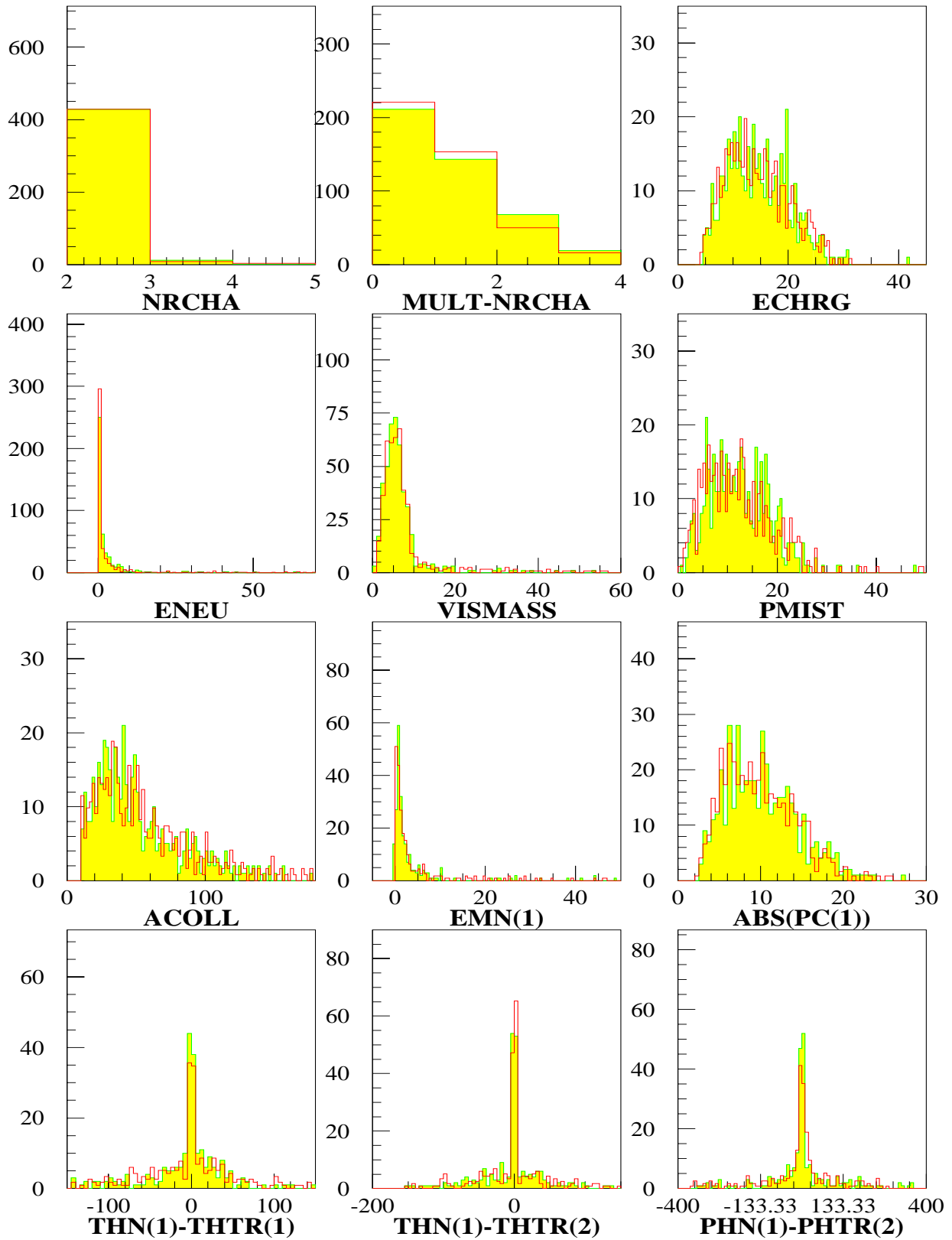
DELSIM vs SGV, $\mu\mu$, $M(\chi_{1,2})=40,120$



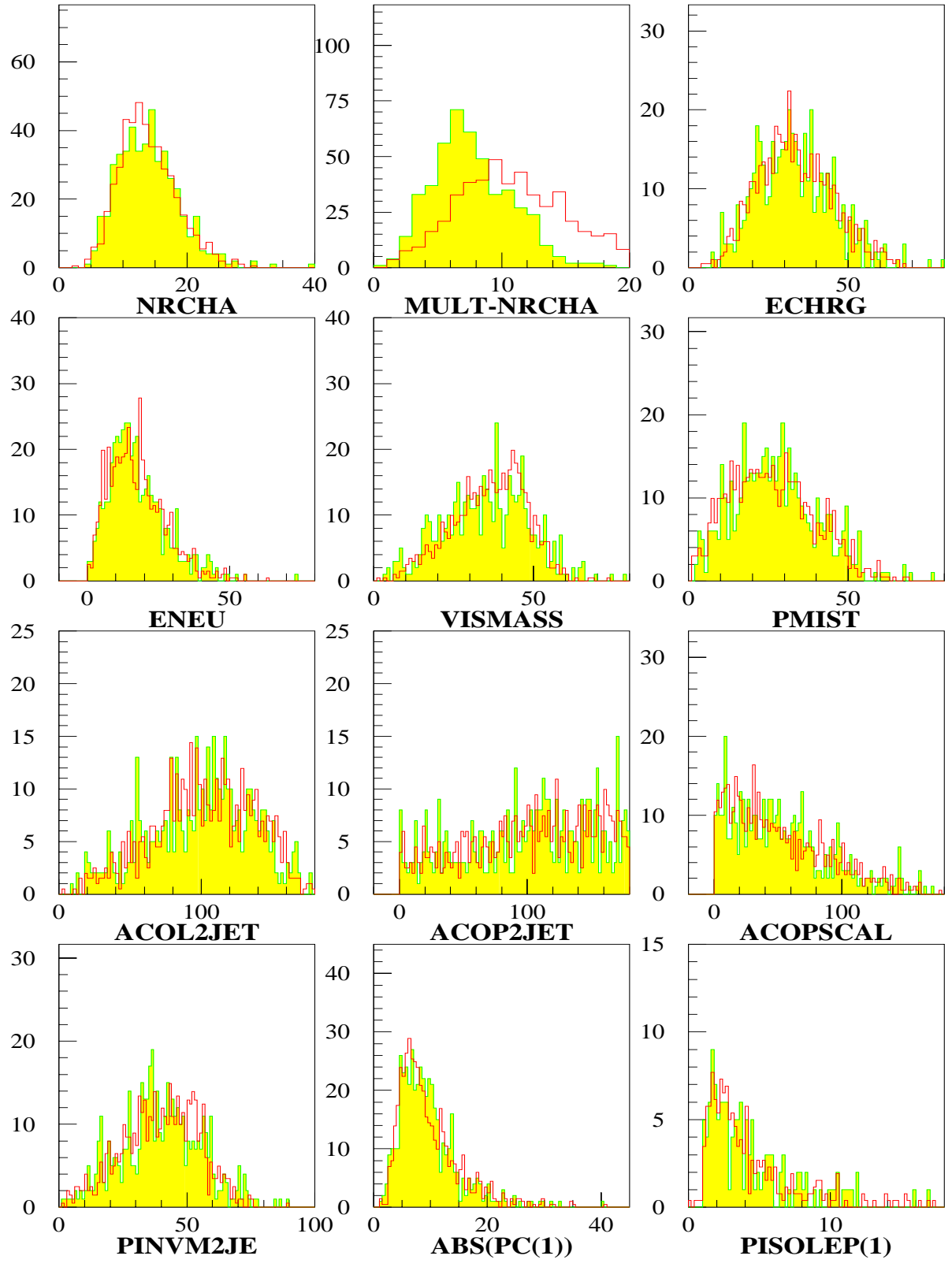
DELSIM vs SGV, ee, $M(\chi_{1,2})=60,120$



Delsim vs SGV, ee, $M(\chi_{1,2})=45,55$

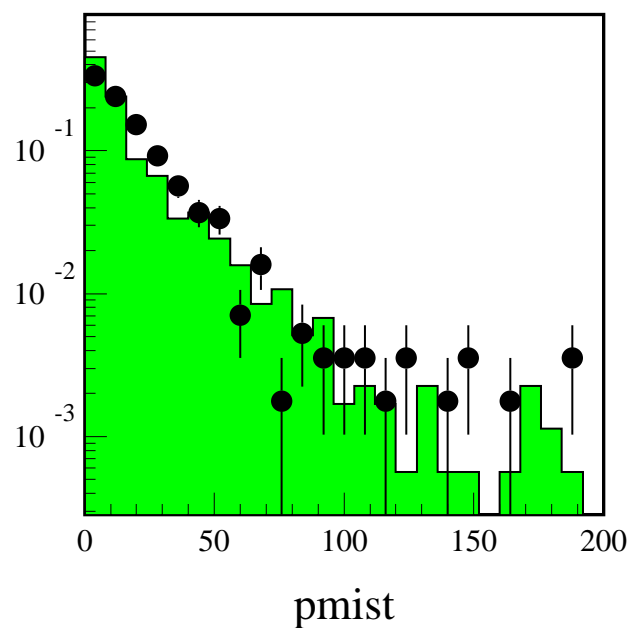
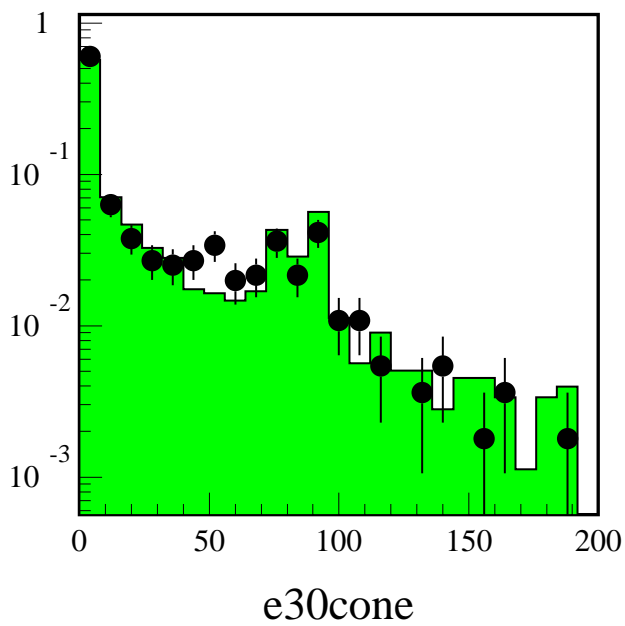
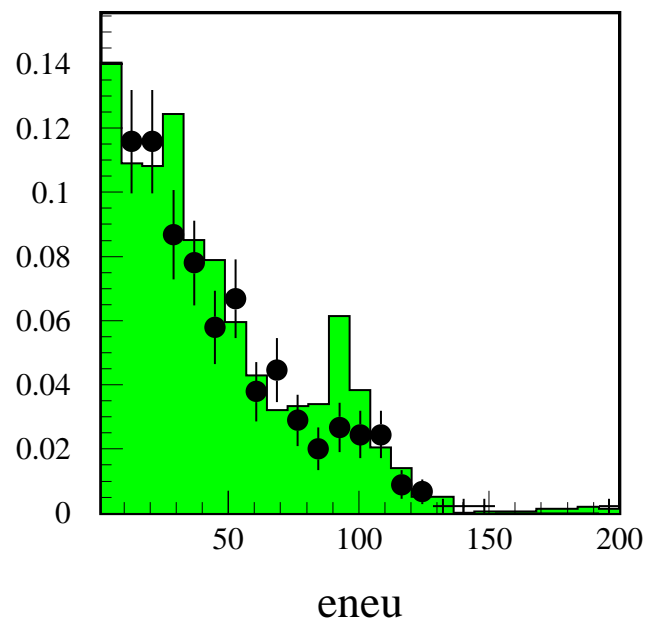
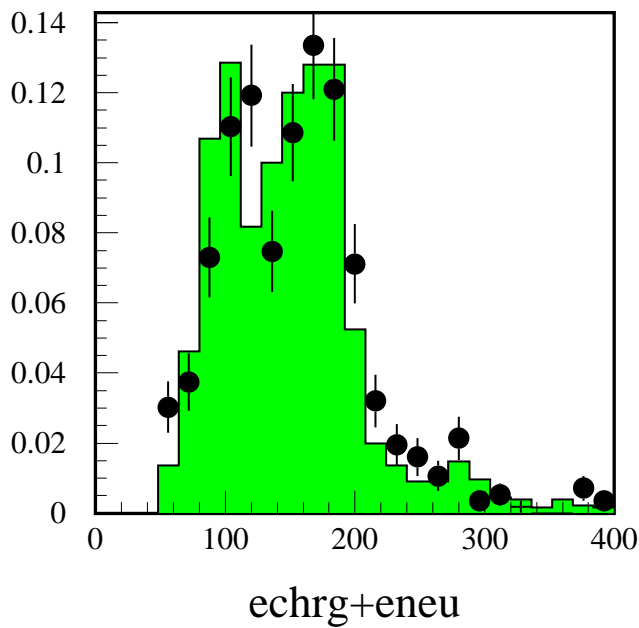


DELSIM vs SGV, qq, $M(\chi_{1,2})=40,100$



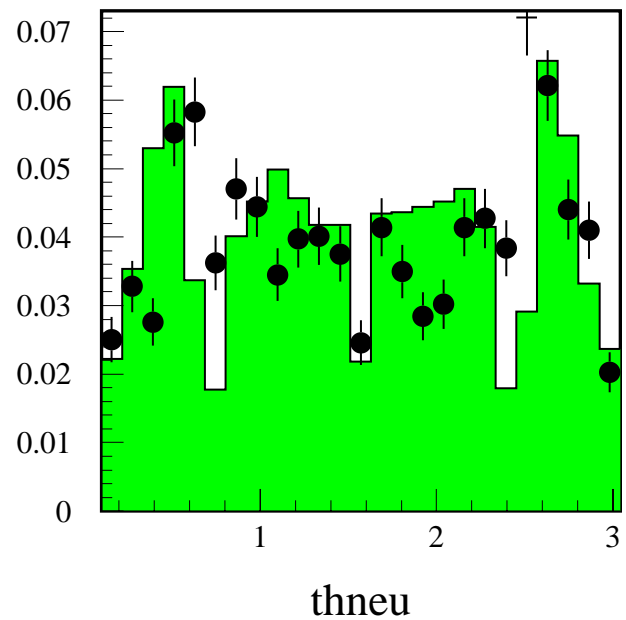
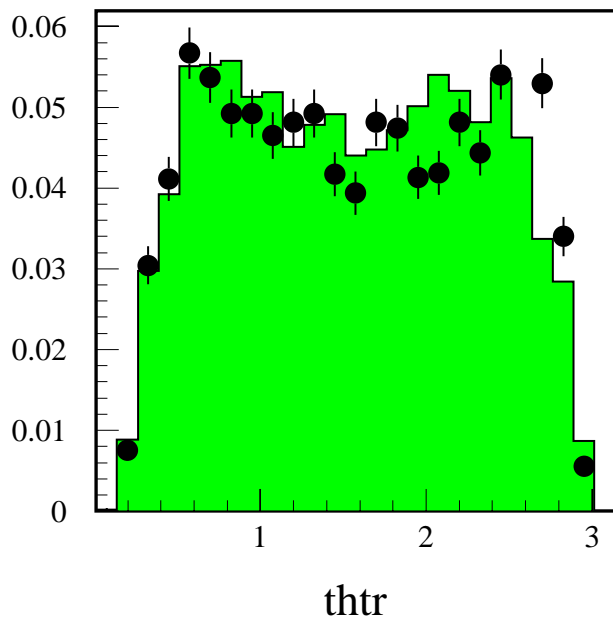
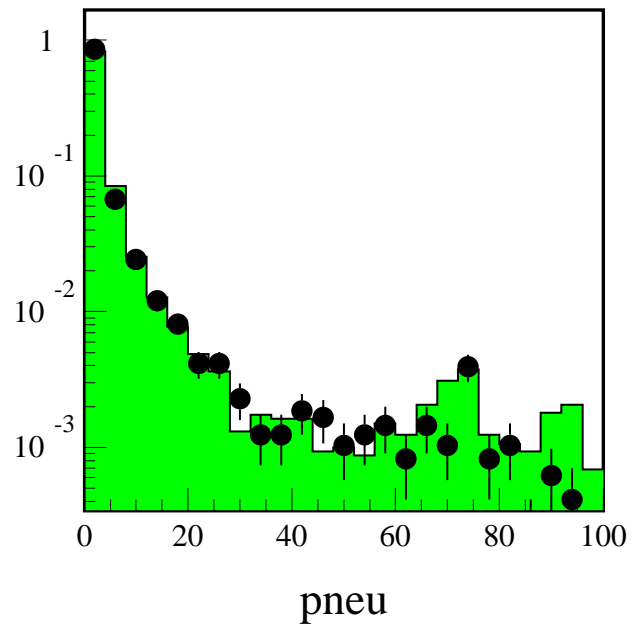
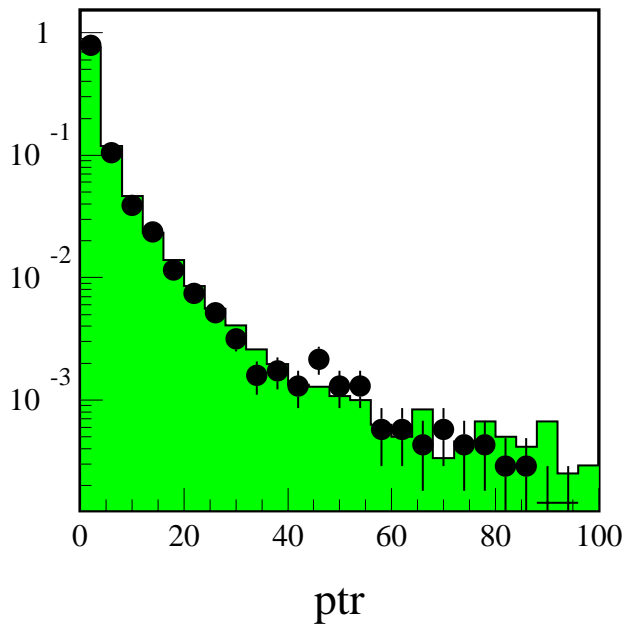
SGV compared to data: Global variables

Histogram: SGV, Points: DELPHI data



SGV compared to data: track variables

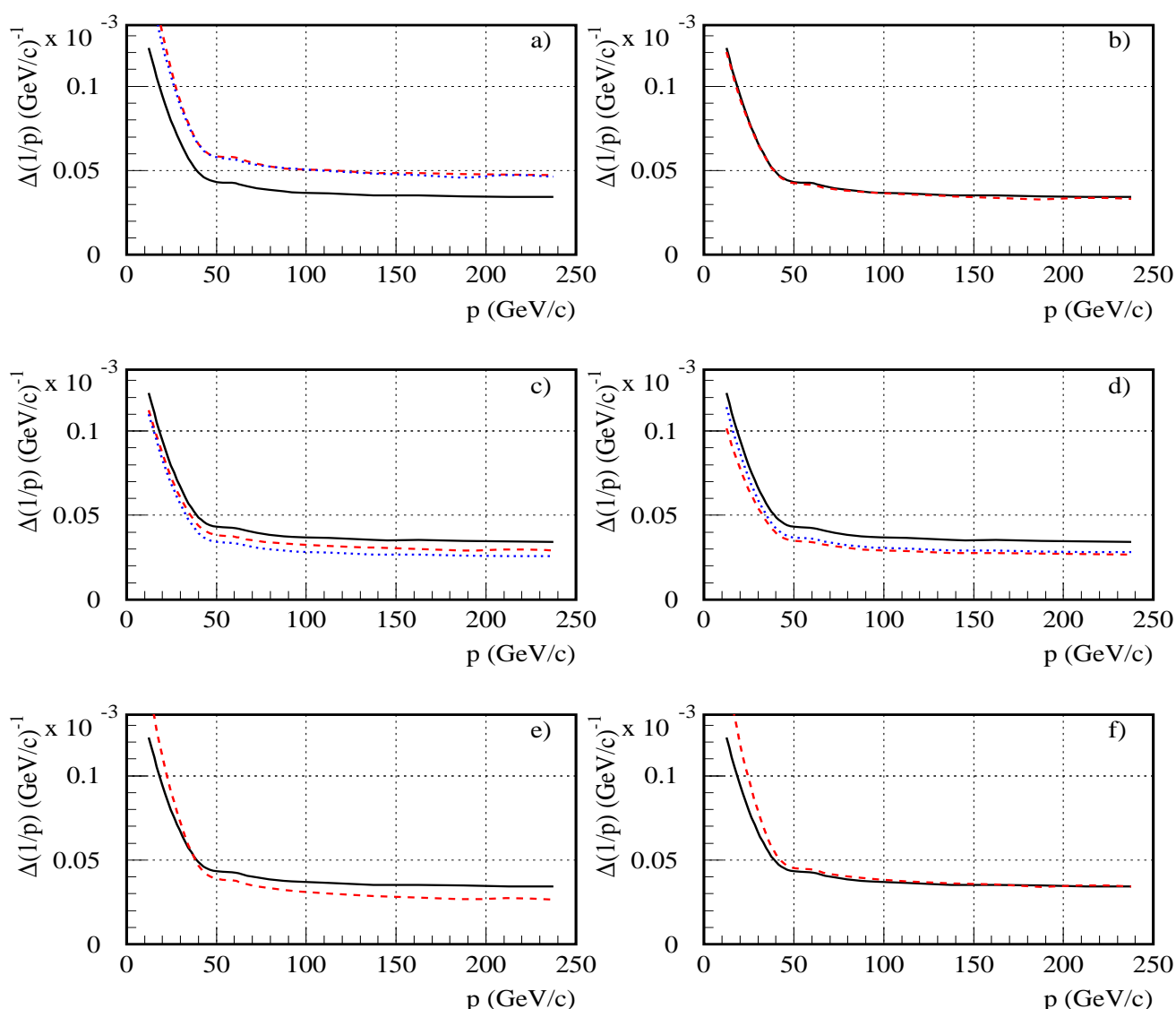
Histogram: SGV, Points: DELPHI data



SGV for detector studies: the TESLA SET

- How would the momentum resolution change if one adds an Silicon External Tracker outside the TPC? How many layers should one put? How should they be distributed?

Ask SGV !



Black is TDR, the other curves are 9 different SET options.

SGV: White paper

- Language: **FORTRAN77**
- Code management: **PATCHY**
- Depends on **CERNLIB**
- Distributed as: **Single compressed file** (Gzip)
- Installation procedure: **Self-installing** on UN*X and VMS
- Size: **Distribution-file 230 kB**. Expanded and **installed, 2.9 MB** is needed (including documentation, 1.1 MB). The PATCHY pam-file contains **35 000 lines**.
- Documentation: **Users Guide** and **Reference Manual included** in distribution-file, in the form of LaTeX. The installation procedure creates the corresponding PostScript files, if requested. The source is latex2htmlable.
- Manifestly runs on: **Linux, HP/UX, True64 (Digital UNIX), VMS**, with native compilers (=g77 in the case of Linux).
- Timing: **114 ms/PYTHIA qq event** on a Compaq 1655 notebook (**266 MHz I686**). **About 1 000 times faster than the DELPHI full simulation for the same channel.**

Getting hold of **SGV**

Down-load from

<http://home.cern.ch/berggren/sgv.html>

or use the pre-installed version in the cern.ch afs-cell (HP-UX, DigitalUnix, Aix and Linux) by executing

</afs/cern.ch/delphi/tasks/sgv/share/pam/sgv.sh>