

# Development and improvement of the guinea-pig beam-beam simulation

G. Le Meur, F. Touze

M. Alabau, Ph. Bambade, O. Dadoun,

C. Rimbault, D. Schulte

LAL, ORSAY

# summary

- What is guineapig++ ?
- Development environment
- Distribution
- New features
- Tests
- On the computing grid
- Next developments

# What is guineapig++ ?

- An **object oriented** version (**C++**) of the beam-beam simulation program **guineapig** (Daniel Schulte, 1996)
- Advantages of OO :
  - **safety**, modularity, versatility
  - **Easy evolution** : implementation of new features
- Use of **Standard Template Library** : strings, containers (vectors, lists...)



# What is guineapig++ ?

- C-structures become C++ classes
- Original algorithms are kept
- Some new features
- Ready for specific developments for future ILC simulations : new classes, new algorithms

# Development environment : CMT

- CMT is a **Configuration management environment** (<http://www.cmtsite.org>) :
  - Notion of *package* (automatic management of libraries, tools...)
  - *Automatic implementation* on different platforms
  - *Automatic generation of makefiles* (from a description file : requirements )
- CMT is used in large experiments : ATLAS, LHCb, Planck...

# Development environment : SVN

- A **collaborating version manager** (successor of CVS)
- *History* of all operations
- *Several developers* with conflict management
- Easy web interface
- Easy visualization of differences between 2 versions (with TRAC, see below)




# Development environment : TRAC

- Web open source collaborative development tool
- *Wiki* interface
- *Bug reporting* tool (tickets)
- SVN interface
- *Blog*
- <https://trac.lal.in2p3.fr/GuineaPig>

GuineaPig - Trac

https://trac.lal.in2p3.fr/GuineaPig

Google



Search

Login | Settings | Help/Guide | About Trac

Wiki | Roadmap | Browse Source | View Tickets | New Ticket | Search | Tags

[Start Page](#) | [Title Index](#) | [Recent Changes](#) | [Page History](#)

## Welcome to the Lal guineapig++ project

---

### The GuineaPig++ project

Guineapig is a beam-beam simulation code written by D. Schulte (CERN). The aim of the project guineapig++ at LAL is to provide an reliable, versatile, modular object oriented (C++) guineapig code. The expected benefit is the ability to easy implement new features and fonctionnalities.

### People

- Developpers: Guy Le Meur, François Touze
- Contributors: Cécile Rimbault, Olivier Dadoun...

### Download

[au LAL](#)

[download standalone kits](#)

[download from the SVN Repository](#)

### Compile the software

[from tarball with CMT included](#)

[from SVN checkout](#)

### documentation

see [main features](#)

⇒ [doxygen documentation](#)

⇒ [on the the GRID](#)

### configuration management

Terminé

trac.lal.in2p3.fr



# Distribution

- The user has only to connect to the guineapig **TRAC** :  
<https://trac.lal.in2p3.fr/GuineaPig>
- Click on 'download standalone kits' for choosing a **tar file** (a version of guineapig++)
- Untar, see the README, **no one has to know anything of CMT, SVN, TRAC**, if no developing.
- Currently : Linux, Darwin (PowerMacIntosh) are available

# New features : **bhabhas**

- **Bhabha deflection** : beam-beam effect on bhabha event previously prepared in an *input file* : *bhabha.ini*
- *Output files* with same format as for pairs : *bhabhas0.dat (before tracking)* and *bhabhas.dat (after tracking)*
- Keywords : *do\_bhabhas*, *track\_bhabhas*, *store\_bhabhas*

# New features : random number generation

- A new keyword **rndm\_seed** allows on to choose a seed (long int)
- The original implementation (rndm7) of assumes a **32-bits computer** :

$$x_{n+1} = (a \cdot x_n + c) \bmod 2^{32} \text{ (with suitable values for } a \text{ and } c)$$

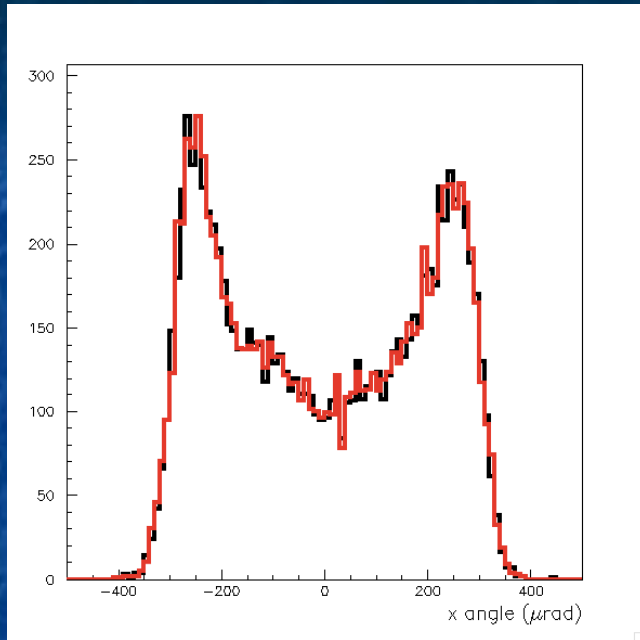
- A Haynes' algorithm has been implemented, for **64-bits computers** : analogue linear formula (mod  $2^{64}$ )
- The random generator is checked before computing.



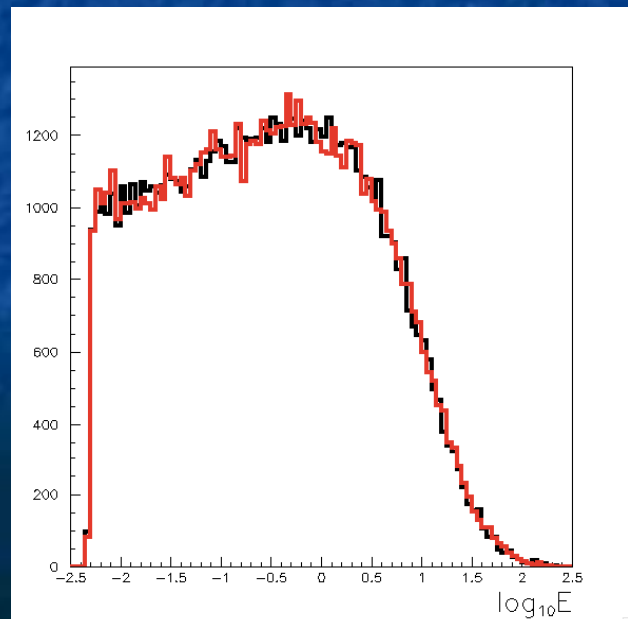
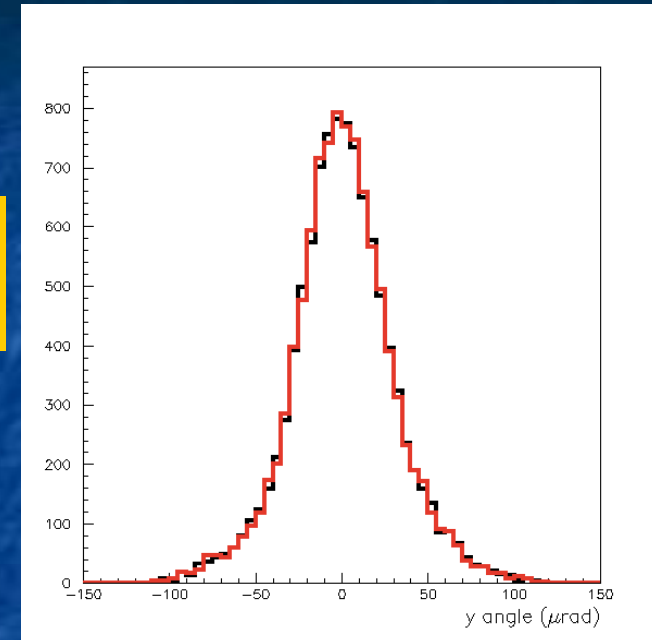
# New features : abstract I/O interface

- Separate algorithms and I/O
- possibility of plugging *any I/O format*
  - Currently : ASCII
  - Future : HDF5, .... (binary formats, for intensive ILC simulation)
- Possibility of plugging different *graphical interfaces*

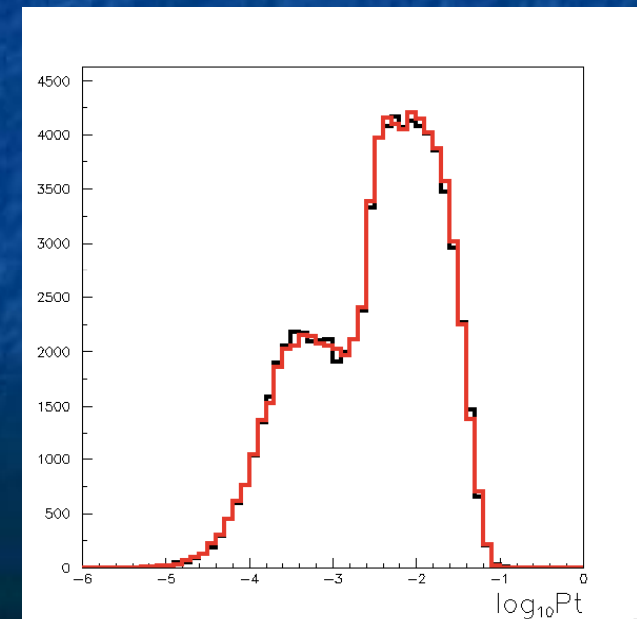
# Tests : comparison of guineapig and guineapig++



Disrupted beam  
particle angles

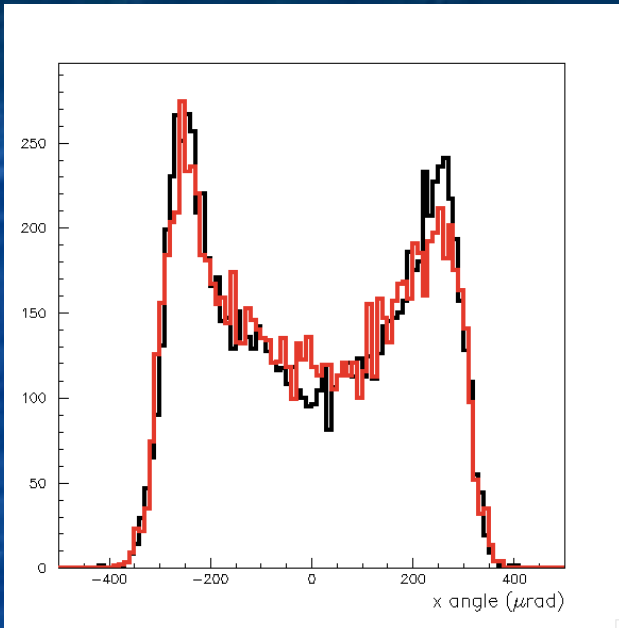


Incoherent pair  
particle  
E, Pt spectra

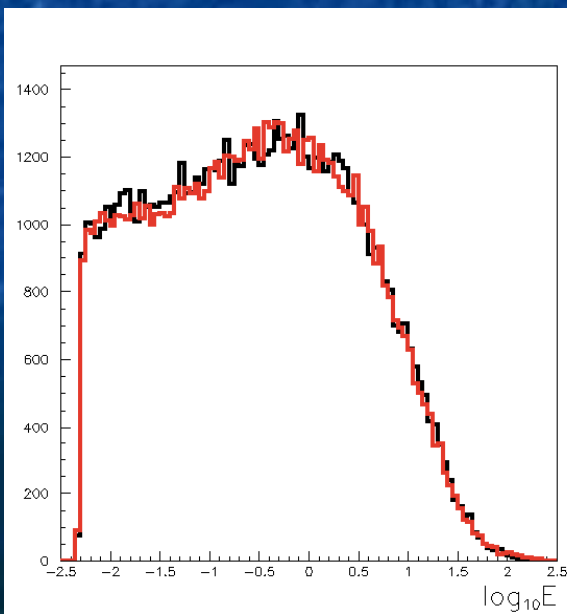
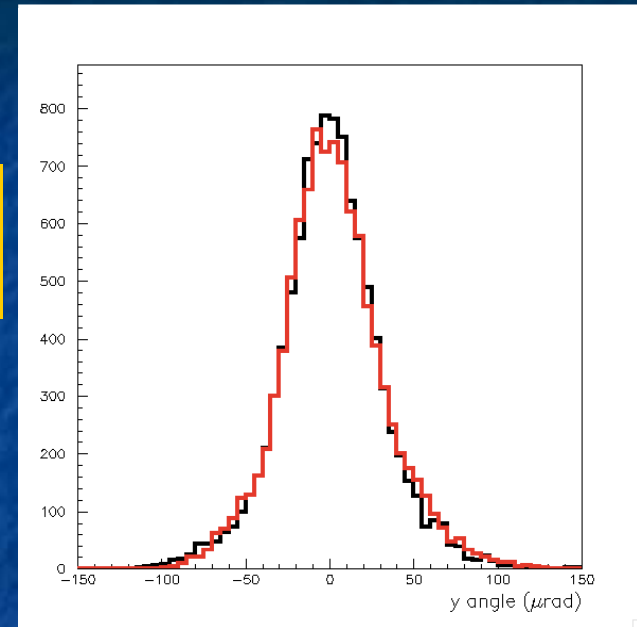


European LC Workshop,  
Daresbury Laboratory

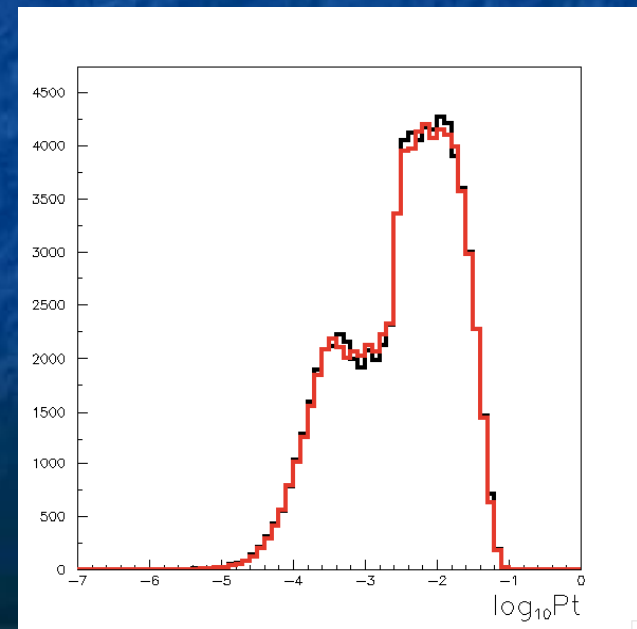
# Tests: guineapig++ on 32-bits and **on 64-** **bits** computers



Disrupted beam  
particle angles



Incoherent pair  
particle  
 $E, P_t$  spectra



European LC Workshop,  
Daresbury Laboratory



# Use on the computing Grid

- Guineapig++ has been used on the LCG-EGEE computing Grid (instructions are on the guinapig++ TRAC-site)...
- Thanks to the possibility of choosing **rndm\_seed** (in order to be sure to have different random realizations on each computer)
- **High statistics guineapig++** are being stored on the grid SE (Storage Elements) :  
`/grid/ilc/DADOUN/GuineaPig++_DATA`

# Next developments

- Complete implementation in C++ for features which are not yet available
- **Automatic** determination of the **grid dimensions and number of cells** (guess : use the disruption factor)
- **Depolarization effects**
- Complete the I/O abstract interface and implement **other formats** than ASCII (HDF5?)
- Parallelization and/or grid computation ?
- Other...