PROCEEDINGS OF SCIENCE

PoS(Vertex 2011)026

The LiC Detector Toy fast simulation program

Manfred Valentan*, **Rudolf Frühwirth, Winfried Mitaroff, Meinhard Regler**
Institute of High Energy Physics (HEPHY), Austrian Academy of Sciences
Nikolsdorfer Gasse 18, 1050 Vienna, Austria
E-mail: manfred.valentan@oeaw.ac.at

The “LiC Detector Toy” (LDT) is a fast single-track simulation and reconstruction tool, aiming at the optimization of tracking detector design, i.e. geometric layout and amount of material. Its implementation is based on the MATLAB system, with a user-friendly graphic user interface. The simulation can handle cylindrical and planar detector layers, where the detector setup is defined in human-readable text files. Simulated helix tracks are reconstructed with a Kalman filter including statistical tests. A key feature is an intuitive possibility to repeat simulation runs, yielding curves of the detector performance as function of the polar angle, the momentum, and different detector setups. An interface exists for passing the reconstructed tracks to a vertex reconstruction toolkit. LDT has successfully been validated against two well-tested full simulation programs based on GEANT4. The tool has contributed to the optimization of detectors at the International Linear Collider (ILC) and at the Compact Linear Collider (CLIC).

PACS: 07.05.Kf, 07.05.Tp

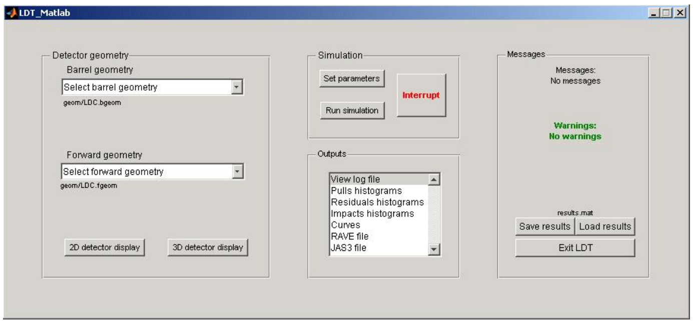
The 20th Anniversary International Workshop on Vertex Detectors – VERTEX 2011
June 19 - 24, 2011
Rust, Lake Neusiedl, Austria

*Speaker.

1. Introduction

The “LiC Detector Toy” (LDT)[1] – a simple but powerful and flexible software tool for detector design, modification and upgrade studies – aims at investigating the resolution of reconstructed charged tracks for the purpose of comparing and optimizing the track-sensitive devices and amounts of material of various detector layouts. This is achieved by a simplified simulation of the investigated setup, followed by full single track reconstruction. At present it supports silicon strip detectors (single or double sided, with any stereo angle α), pixel detectors and a time projection chamber.

The algorithms used in the tool are well established and field-proven. The software is based on MATLAB® [1], a commercial high-level matrix algebra language; usage is supported by a graphic user interface (GUI, see figure 1). Zoomable 2D and 3D displays of the setup (see figures 2 and 3) and flexible data presentation of the results are easily at hand. The core program has recently been refactored to run also in command line mode without GUI under OCTAVE [2], an open-source product which is source-code compatible.



Messages:
No messages

Warnings:
No warnings

Figure 1: Main window

The tool is deliberately kept simple and can be adapted to individual needs with little effort. Its main purpose, however, is to supply a tool for non-experts without algorithmic knowledge. Once the detector description (“input sheet”) has been set up, individual detector layers can be added, moved, modified or removed easily, and the result of these changes can be evaluated quickly. For more information see the “LiC Detector Toy User’s Guide” [3]. A thorough discussion on detector performance optimization can be found in [4].

2. Detector model

The detector model assumes a collider experiment with a solenoid magnet; it is rotationally symmetric w.r.t. the beam axis (denoted z), but not necessarily mirror symmetric w.r.t. $z = 0$. Detector surfaces are either finite cylinders (“barrel region”), or rings of planar disks (“forward/rear region”). The magnetic field $\vec{B} = (0, 0, B_z)$ is homogeneous and parallel to the z-axis, thus implying a helix track model (radius r_{H}). All material causing multiple scattering is assumed to be concentrated within thin layers; azimuthal variations must be averaged over the surface.

Note that small deviations from the surface geometry and/or from the magnetic field homogeneity (e.g. fringe field or detector-integrated dipole) influence the absolute results. However, these deviations do not present a restriction for detector optimization, since the purpose of this program is to evaluate the differential change of the detector performance when introducing a change in the detector setup.

[1] LiC stands for “Linear Collider”.

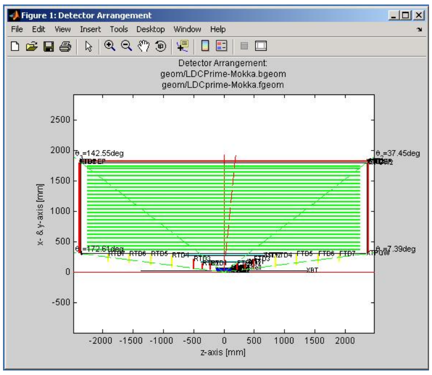


Figure 1: Detector Arrangement
File Edit View Insert Tools Desktop Window Help
Detector Arrangement:
geom/LDCPrime-Mokka.bgeom
geom/LDCprime-Mokka.fgeom
x- & y-axis [mm]
z-axis [mm]
θ =142.55deg
θ =37.45deg
θ =172.61deg
θ =7.39deg

Figure 2: Detector setup in 2D

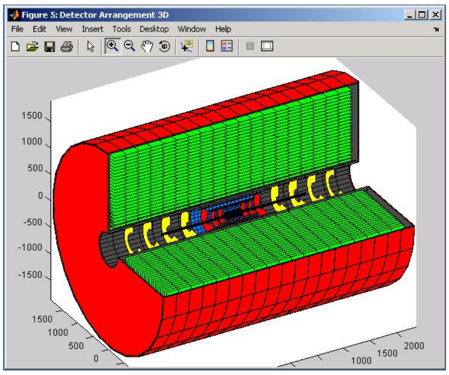


Figure 5: Detector Arrangement 3D
File Edit View Insert Tools Desktop Window Help

Figure 3: Detector setup in 3D

2.1 The input sheet

Two text files called “input sheets” – one for the barrel and one for the forward (including rear) regions – contain all information in human readable format that the program needs for building the detector model: the geometry and amount of material of the detector layers; the accuracy, type of the error distribution and efficiency of the sensor elements; and the geometry and amount of material of passive scattering layers (e.g. the beam tube and support structures).

The following example snippet defines two asymmetric passive layers (support frames), two detector layers (double-sided microstrips), and another passive layer (inner wall of a TPC). Among others, these layers can be seen in the zoomed display in figure 4.

```
Silicon Inner Tracker (SIT)

Number of layers:             5
Description (optional):     |support rings|-inner tracker-|TPC inner wall|
Names of the layers (opt.):   XSR1,  XSR2,  SIT1,    SIT2,    XTPCW
Radii [mm]:                   90,    90,    160,     290,     340
Upper limit in z [mm]:        110,  -90,    300,     550,     2160
Lower limit in z [mm]:        90,   -110,  -300,    -550,    -2160
Efficiency RPhi:              0,     0,     0.95,    0.95,    0
Efficiency 2nd layer (eg. z):-1,    -1,     0.95,    0.95,   -1
Stereo angle alpha [Rad]:     10*pi/180
Thickness [rad. lengths]:     0.07,  0.07,  0.0175,  0.0175,  0.14
error distribution:           1
0 normal-sigma(RPhi) [1e-6m]:
         sigma(z)    [1e-6m]:
1 uniform-d(RPhi) [1e-6m]:    50
          d(z)    [1e-6m]:    50
```

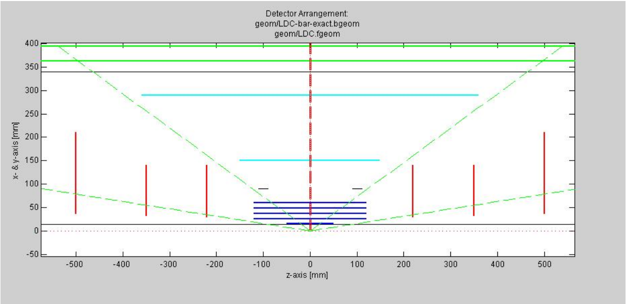
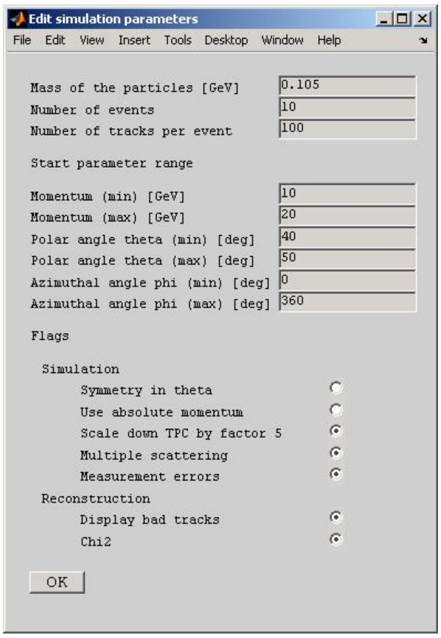


Figure 4: Detector display (zoomed). Passive layers are black, active layers are colored. The definition of the light blue sensor layers can be found in the code snippet.

3. Simulation

The simulation generates charged tracks with user-defined start parameters (vertex position, momentum p or its transverse component p_t, and polar angle ϑ), performs helix propagation with breakpoints due to multiple scattering according to Rossi-Greisen-Highland [5], and simulates detector measurements including correlated or stochastic inefficiencies, and uniform or Gaussian observation errors, *but no other degradation of data*.[2]

Simulation parameters: The start parameters (momentum and direction) can be chosen randomly between user-defined minimum and maximum values. These start parameters do not obey any physics, the tracks of one event do not have any relations among each other, except that they start at the same vertex. If the minimum and maximum values of a range coincide, the corresponding starting value is fixed. In the window “Edit simulation parameters” (figure 5) the following parameters can be set:



Edit simulation parameters

Mass of the particles [GeV]	0.105
Number of events	10
Number of tracks per event	100

Start parameter range

Momentum (min) [GeV]	10
Momentum (max) [GeV]	20
Polar angle theta (min) [deg]	40
Polar angle theta (max) [deg]	50
Azimuthal angle phi (min) [deg]	0
Azimuthal angle phi (max) [deg]	360

Flags

Simulation

Symmetry in theta	○
Use absolute momentum	○
Scale down TPC by factor 5	●
Multiple scattering	●
Measurement errors	●

Reconstruction

Display bad tracks	●
Chi2	●

OK

Figure 5: ’Edit simulation parameters’

- Mass of the particles in [GeV/c^2]. If zero mass is entered, all particles are treated relativistically in terms of multiple scattering ($\beta p \approx p$);
- Number of events and tracks. In the present version each event contains the same number of tracks;
- Transverse or absolute momentum in [GeV/c];
- Polar direction angle ϑ in [^{o}]. Values from 0 to 180 are accepted. ϑ is chosen randomly from a uniform distribution of $\cos(\vartheta)$ between the limits $\cos(\vartheta_{min})$ and $\cos(\vartheta_{max})$. Attention: ϑ should not be mixed up with the dip angle $\lambda = 90^o - \vartheta$;
- Azimuthal direction φ in [^{o}]. At present the whole detector is assumed to be rotationally symmetric, so changing this parameter won’t affect the results. In a future version of LDT detector layers can be segmented and do not have to be rotational symmetric anymore.

Alternatively, one can define the start parameters in a separate text file. Use of this feature allows to process groups of tracks that were generated by an external event generator.

4. Reconstruction

Reconstruction is done separately for each track from all its simulated measurements *without pattern recognition*,[3] by fitting the five parameters and 5×5 covariance matrix at a pre-defined real

[2] Energy loss due to bremsstrahlung of $e^{\pm}$ may be included in a future release.

[3] The correct hit associations are kept and passed from the simulation stage.

or virtual reference cylinder of radius R_C, e.g. the inner surface of the beam tube.

Standard parameters are $[\Phi, z, \vartheta, \beta = \varphi - \Phi, \kappa = \pm 1/r_{\mathrm{H}}]$ at R_C, with Φ and φ denoting the azimuth angles of position and direction, respectively. The method used is an inwards-running Kalman filter [6, 7], with linear expansion from a "reference track" being defined by the simulation; process noise is calculated from the amount of material as seen by that track. Sensitive tests of goodness of the fits (χ^2 distributions and pull quantities) are calculated.

The resolutions of the track parameters are drawn from the distribution of many reconstructed tracks. Moreover, the point of closest approach in the $R - \Phi$ plane (projected impact parameter) and in space (3D impact parameter) are calculated, together with the corresponding resolutions. From the resolution of the curvature κ the momentum resolutions $\sigma\left(\frac{\Delta p_{\mathrm{t}}}{p_{\mathrm{t}}}\right)$, $\sigma\left(\frac{\Delta p_{\mathrm{t}}}{p_{\mathrm{t}}^2}\right)$ and $\sigma\left(\frac{\Delta p}{p}\right)$ are calculated.

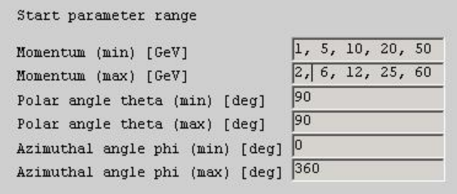
5. Simulation modes

The examples given below are inspired by the International Large Detector (ILD) [8], or from its predecessor Large Detector Concept (LDC), at the International Linear Collider (ILC) [9].

5.1 Automatic loops

The program is able to carry out several simulation runs in sequence. In each of these runs, it will use different pairs of start parameters. The result is the detector performance as a function of e.g. the momentum or the polar angle.

Instead of defining one range you simply put in multiple ranges for the desired start parameter. And it is *this* start parameter which is the variable that changes with every simulation run. Figures 6 and 7 show an example for multiple momentum ranges.



Start parameter range

Momentum (min) [GeV]	1, 5, 10, 20, 50
Momentum (max) [GeV]	2, 6, 12, 25, 60
Polar angle theta (min) [deg]	90
Polar angle theta (max) [deg]	90
Azimuthal angle phi (min) [deg]	0
Azimuthal angle phi (max) [deg]	360

Figure 6: Curve with ranges of p_{t}

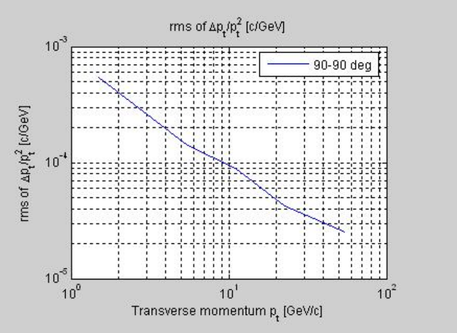
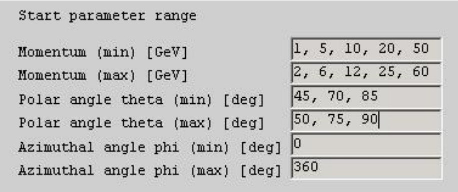


Figure 7: Example: *rms* $\left(\frac{\Delta p_{\mathrm{t}}}{p_{\mathrm{t}}^2}\right)$ as function of p_{t}

5.1.1 Multiple loops

You can compute multiple curves at once, for example as function of p_{t} with different values of ϑ. To do this, just put in multiple pairs of minimum and maximum values of ϑ as start parameters (see figure 8). Having carried out the simulation for the first range of ϑ and all ranges of p_{t}, the program continues the calculation with the next ϑ range and the first p_{t} range. Now, in the output plots, more curves will appear, displaying the results for the different ϑ ranges (see figure 9). You can easily do the same for multiple curves as function of ϑ.



Start parameter range

Momentum (min) [GeV]	1, 5, 10, 20, 50
Momentum (max) [GeV]	2, 6, 12, 25, 60
Polar angle theta (min) [deg]	45, 70, 85
Polar angle theta (max) [deg]	50, 75, 90
Azimuthal angle phi (min) [deg]	0
Azimuthal angle phi (max) [deg]	360

Figure 8: Input of curves with different ϑ ranges

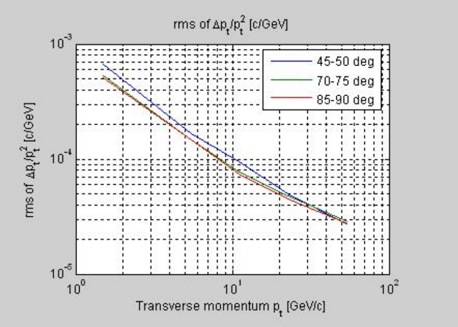
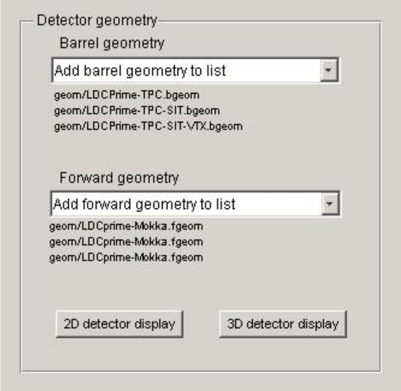


Figure 9: Curves with different ϑ ranges

5.1.2 Looping over detector setups

In literature, one often finds plots showing e.g. the relative momentum resolution $\sigma\left(\frac{\Delta p_t}{p_t^2}\right)$ as function of the transverse momentum p_t for different detector setups.

Such plots allow for resolving the effect of changing e.g. some geometric parameters for a wide range of momenta. LDT is able to produce such plots in a very easy way, similar to what was explained in the previous section. In the main window's section "Detector geometry" one can define a so called "geometry list" of barrel and forward input sheets which are processed in sequence (see figure 10). To establish such a list, one has to use the selections "Add barrel (forward) geometry to list" and "Remove barrel (forward) geometry from list" in the corresponding pull down menus.

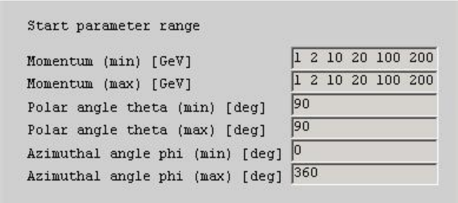


geom/LDCPrime-TPC.bgeom
geom/LDCPrime-TPC-SIT.bgeom
geom/LDCPrime-TPC-SIT-VTX.bgeom

geom/LDCprime-Mokka.fgeom
geom/LDCprime-Mokka.fgeom
geom/LDCprime-Mokka.fgeom

Figure 10: Example for geometry lists

Using the shown example geometry lists, evaluation at different p_t would result in three curves, one for each detector setup. A sample input of start parameters is shown in figure 11, the result of the simulation can be seen in figure 12.



Start parameter range

Momentum (min) [GeV]	1 2 10 20 100 200
Momentum (max) [GeV]	1 2 10 20 100 200
Polar angle theta (min) [deg]	90
Polar angle theta (max) [deg]	90
Azimuthal angle phi (min) [deg]	0
Azimuthal angle phi (max) [deg]	360

Figure 11: Sample start parameters (5 different momenta per detector setup)

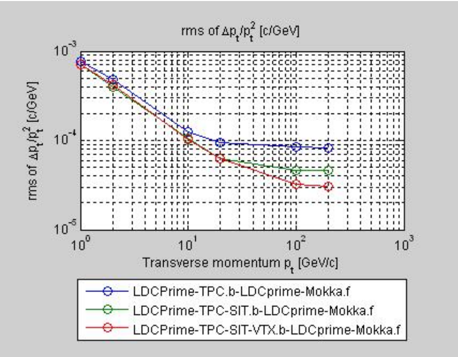


Figure 12: Results of a simulation run with different input sheets, as function of p_t

5.2 Fine-grained simulation mode

So far, the track parameter resolutions are determined by simulating and reconstructing many tracks, and subsequently calculating r.m.s. values from their distributions. The same values can in principle be extracted from the covariance matrix elements of only *one* reconstructed track. Thus, instead of simulating several tracks to get the track parameter resolutions at a certain momentum and angle, LDT in this mode simulates and reconstructs only one track. This makes it possible to compute curves consisting of a large number of points of the parameter space.

The fine-grained simulation mode can be used e.g. to perform a fine angular scan of a detector setup; this is of particular interest for identifying regions of reduced detector coverage. Figure 13 shows an example for this case. This simulation mode can also be used to calculate curves as function of the momentum, as shown in figure 14.

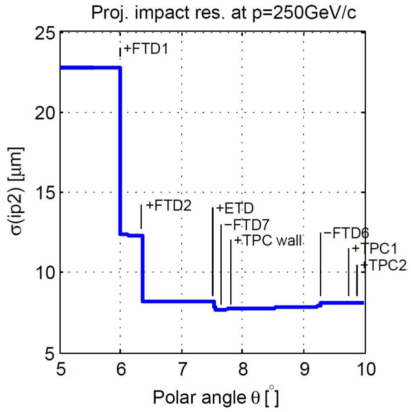


Figure 13: Fine angular scan: You can see detector layers start and stop contributing to the resolution

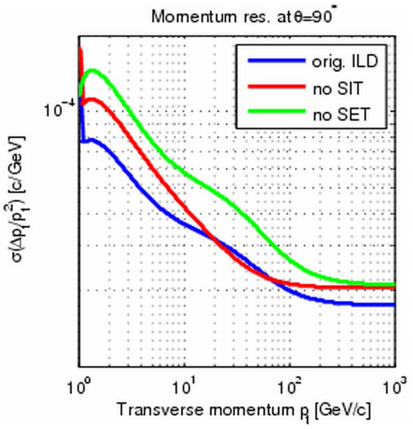


Figure 14: Fine p_t scan and comparison of different setups

6. Validation against full simulation studies

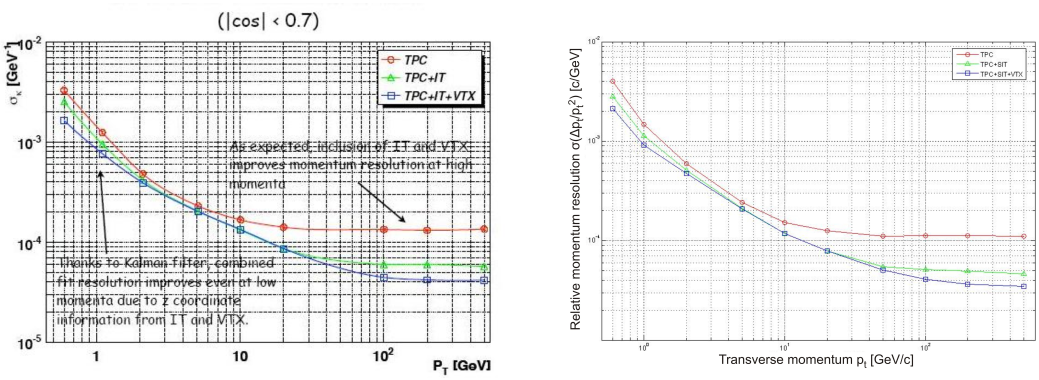
In order to gain confidence in the results given by a simulation tool, one has to perform cross-checks with already existing and tested ones. Despite known differences, e.g. by considered physical effects and the realism of the detector model, all simulation tools should give – within their approximations – the same results.

6.1 Validation against JUPITER / URANUS

In [10] a full simulation study shows the performance of barrel tracking in the Gaseous Large Detector (GLD) concept [11] at the ILC, containing a vertex detector, an inner silicon tracker, and a time projection chamber (TPC). The study was performed by the tool JUPITER[4] [12], which is a GEANT4 and ROOT [13] based full detector simulation. Reconstruction and data analysis used the tools URANUS / Satellites [12].

[4] Acronyms: JUPITER: "JLC Unified Particle Interaction and Tracking Emulator"; URANUS: "Unified Reconstruction and Analysis Utility Set"; GEANT: "Geometry and Tracking"; ROOT: "Root is an object oriented toolkit".

Comparison of that study (figure 15(a), on the left) with a corresponding one by LDT (figure 15(b), on the right) shows excellent agreement.



(a) $\sigma(\Delta p_\mathrm{t}/p_\mathrm{t}^2)$ vs. p_t (Jupiter).

(b) $\sigma(\Delta p_\mathrm{t}/p_\mathrm{t}^2)$ vs. p_t (LDT).

Figure 15: Validation of LDT (right plot) against Jupiter/Uranus (left plot). Different colors are different detector setups

6.2 Validation against MOKKA / MARLIN

The ILD concept [8] at the ILC resulted from the merger of a similar earlier concept (LDC) with GLD. LDC and ILD have GEANT4 based MOKKA[5] [14] as their full detector simulation, and MARLIN [14] as their reconstruction and analysis framework; LCIO [14] (not ROOT) serves as the event data model and persistency storage.

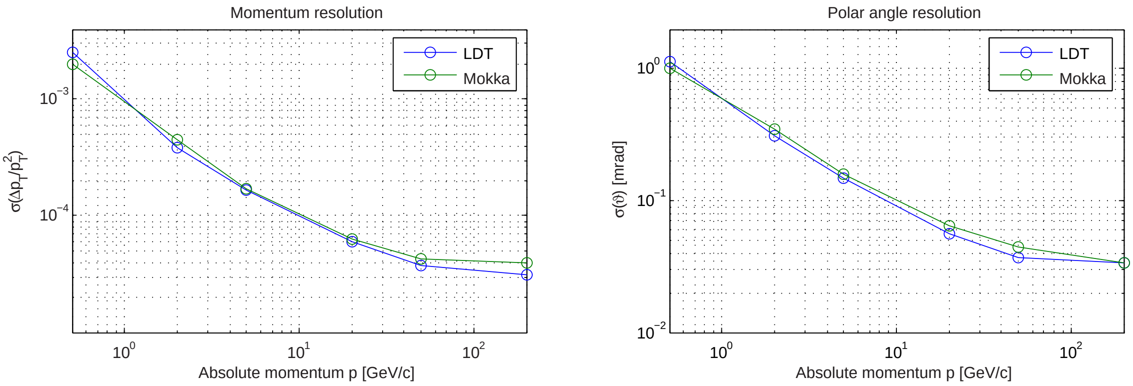


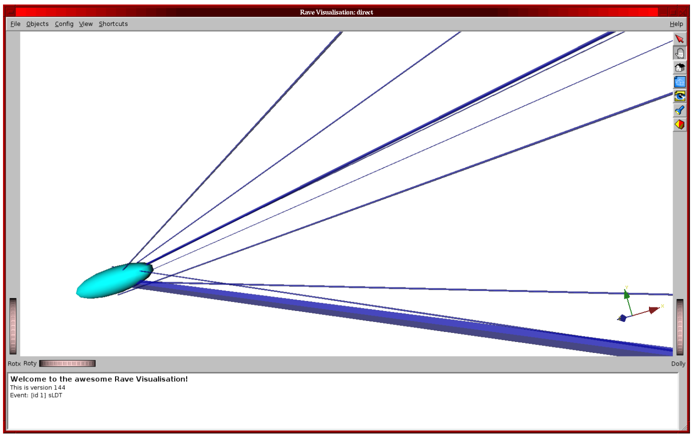
Figure 16: Validation of LDT (blue) against Mokka/Marlin (green). Left plot shows momentum resolution, right plot shows polar angle resolution.

In [15] a full simulation study presents the performance of barrel tracking in ILD (vertex detector, inner and outer barrel silicon tracker, and TPC). Figure 16 shows direct comparisons of that study (green) with a corresponding one by LDT (blue): the normalized relative momentum resolutions $\sigma(\Delta p_\mathrm{t}/p_\mathrm{t}^2)$ (left), and the polar angle resolutions $\sigma(\Delta\vartheta)$ (right) as functions of the absolute momentum p. The comparisons show good agreement.

[5] Acronyms: MOKKA: "Modellierung mit Objekten eines kompakten Kalorimeters"; MARLIN: "Modular Analysis and Reconstruction for the Linear collider"; LCIO: "Linear Collider Input/Output".

7. Subsequent vertex fit

LDT features optional text output of the fitted track parameters in CSV (comma-separated variables) format, which can be used as a simple interface to vertex reconstruction toolkits. This has been tested with the RAVE[6] toolkit and its auxiliary standalone framework VERTIGO [16]. LDT's CSV output is used to provide input data for the RAVE vertex fitter, which successfully reconstructs the simulated vertices using tracks simulated and reconstructed by LDT (see figure 17). This emphasises the reliability of both tools.



Welcome to the awesome Rave Visualisation!
This is version 144
Event: [id 1] sLDT

Figure 17: A vertex reconstructed by RAVE, using tracks from LDT

8. Summary

The LiC Detector Toy (LDT) is a fast simulation program for quick comparison and optimization of the track-sensitive devices and amounts of material of various detector layouts. Its main advantages are a simple detector description and the possibility to loop simulation runs and plot the detector performance as function of the momentum or the polar angle for different detector setups in one go. It is worth mentioning that LDT was used to validate analytic formulae for tracking performance, which have recently been generalized for more realistic detector layouts [17].

Further information on work done with LDT is given in [18]–[21]. Optimization studies also contributed to the ILD Letter of Intent [8], and to the CLIC Conceptual Design Report [22].

References

[1] D. Hanselmann and B. Littlefield: *Mastering* MATLAB *7*, Pearson Prentice-Hall (2005).
See also MATLAB *on the web*, `http://www.mathworks.com/`

[2] GNU OCTAVE v.3 *on the web*, `http://www.gnu.org/software/octave/`

[3] M. Valentan: *LDT repository*, `http://linearcollider.hephy.at/lictoy/`

[4] M. Krammer and W. Mitaroff: *Tracking Detectors*, in "Handbook of Particle Detection and Imaging" (ed. C. Grupen et al.), Springer, Berlin, ISBN 3-642-13270-7 (in print).

[6] Acronyms: RAVE: "Reconstruction of vertices in abstract versatile environments"; VERTIGO: "Vertex reconstruction toolkit and interface to generic objects".

[5] V. L. Highland: Nucl. Instr. Meth. **129** (1975) 497.

[6] R. Frühwirth et al. (ed. M. Regler, R. Frühwirth): *Data Analysis Techniques for High-Energy Physics*, 2nd edition, Cambridge University Press (2000), ISBN 0-521-63219-6, ISBN 0521-63548-9 (pbck).

[7] M. Regler, R. Frühwirth and W. Mitaroff: Int. J. Mod. Phys. **C7, 4** (1996) 521.
R. Frühwirth et al.: Nucl. Instr. Meth. **A 334** (1993) 528.

[8] H. Stoeck et al. (ILD Concept Group): The International Large Detector – Letter of Intent,
DESY / KEK / FNAL, Feb. 2010 (rev.), ISBN 978-3-935702-42-3, ISSN 0418-9833.
See also ILD *on the web*, `http://www.ilcild.org/`

[9] E. Elsen et al. (editors): International Linear Collider (ILC) – A Technical Progress Report,
Global Design Effort, FNAL, June 2011, ISBN 978-3-935702-56-0.
See also ILC on the web, `http://www.linearcollider.org/`

[10] A. Yamaguchi et. al.: presentation at the 8th ACFA Linear Collider Workshop 2005, Daegu, Korea:
`http://chep.knu.ac.kr/ACFA8/upload/7.12(Tue)/Afternoon/`
`Simulation_Reconstruction%20Session/acfa8talk_A_Yamaguchi2.pdf`

[11] GLD *on the web*, `http://ilcphys.kek.jp/gld/`

[12] JUPITER, URANUS and Satellites *on the web*,
`http://ilcphys.kek.jp/soft/index-gldsoft.html`

[13] GEANT4 and ROOT, `http://geant4.cern.ch/, http://root.cern.ch/`

[14] MOKKA, MARLIN, LCIO etc. *on the web*,
`http://polzope.in2p3.fr/MOKKA/`
`http://ilcsoft.desy.de/portal/software_packages/`

[15] A. Raspereza: presentation at the 5th ILC-ECFA Linear Collider Workshop 2008, Warsaw, Poland:
`http://ilcagenda.linearcollider.org/`
`contributionDisplay.py?contribId=52&sessionId=43&confId=2642`

[16] W. Waltenberger: IEEE Trans. Nucl. Sci. **58** (2011) 434.
See also RAVE *on the web*, `http://projects.hepforge.org/rave/`

[17] M. Valentan, M. Regler and R. Frühwirth: *Generalization of the Gluckstern formulas II: Multiple scattering and nonzero dip angles*, Nucl. Instr. Meth. **A 606** (2009) 728.

[18] M. Regler, M. Valentan and R. Frühwirth: *The LiC Detector Toy Program*, Proc. 11th Vienna Conf. on Instrumentation (VCI) 2007, Vienna, Austria: Nucl. Instr. Meth. **A 581** (2007) 553.

[19] W. Mitaroff, M. Regler, M. Valentan and R. Höfler: *Track Resolution Studies with the "LiC Detector Toy" Monte Carlo Tool*, Proc. 10th Int. Linear Collider Workshop (LCWS) 2007, Hamburg, Germany (ed. A. Frey, S. Riemann), p. 468: DESY, 2008, ISBN 978-3-935702-27-0, ISSN 1435-8077.

[20] M. Regler, W. Mitaroff, M. Valentan, R. Frühwirth, R. Höfler: *The "LiC Detector Toy" Program*, Proc. Int. Conf. on High Energy and Nuclear Physics (CHEP) 2007, Victoria, Canada: J. Physics Conf. Series **119** (2008) 032037.

[21] M. Valentan, M. Regler, W. Mitaroff, R. Frühwirth: *LiC Detector Toy - Tracking detector optimization with fast simulation and its application to the ILD design*, Proc. 11th Int. Linear Collider Workshop (LCWS) 2008, Chicago, USA: arXiv:0901-4183v1[physics.ins-det].

[22] A. Miyamoto et al. (editors): Compact Linear Collider (CLIC) – Conceptual Design Report, Vol. 2, CERN, Geneva, 2011 (in preparation).
See also CLIC *on the web*, `http://www.clic-study.org/`