

Lucretia Developments

PT
SLAC



Remind me – what’s Lucretia?

- A Matlab-based toolkit for simulations of beam dynamics in single-pass electron beamlines
 - **Most components are Matlab scripts or functions**
 - **A handful of compiled C functions which run under Matlab (“mexfiles”)**
 - Compute intensive activities like tracking
 - Tracks pointlike rays with user-defined charge, not 4-D macroparticles
 - R-matrix calculation, Twiss propagation, lattice verification
 - Share a lot of code with Tracking
- Designed with the ILC in mind
 - **Somewhat useful for other applications – LCLS, XFEL**
 - **Designed from the ground up to be an all-in-one tool for LET studies**
 - **As “bulletproof” as we can make it**
 - Careful memory management
 - Careful error handling
 - Segmentation faults fixed as quickly as possible
 - Runs under Matlab
 - Matlab data (ie, beamline state, etc) preserved even if a fault is detected



What's in Lucretia?

- Fairly sophisticated beamline representation
 - Includes power supplies, klystrons, girders with or without movers, short range and long range wakefields, incoherent synchrotron radiation (several methods), crab cavities
- Beam representation
 - Multi-bunch representation
 - Several pre-defined distributions of rays, and users can define their own
- Instrumentation
 - Read-out of “observable” and “non-observable” stuff at each instrument
 - Bunch-by-bunch or train-by-train readback
 - BPM resolution and electrical offsets



Other Lucretia Features

An online manual in the form of a heavily cross-linked website:

<http://www.slac.stanford.edu/accel/ilc/codes/Lucretia/>

The **LUCRETIA** Project 

Welcome to **Lucretia**

Lucretia is a physics toolbox for the simulation of high-performance single-pulse electron beam transport systems. It is designed to support studies of linear collider Low-Energy Transport (bunch compressor, linear acceleration, final focus), and also the transport and acceleration systems of laser-driven free electron lasers.

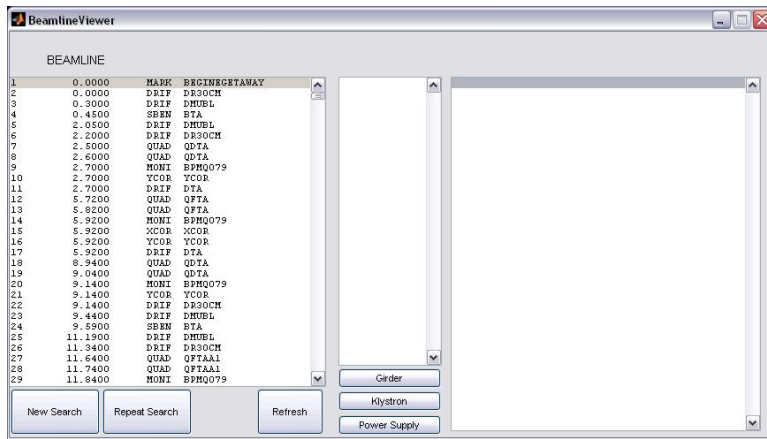
At this time, Lucretia is only available to run under **Matlab**. One could imagine releasing a version of Lucretia which runs under **Octave** as well. Current version dates for subfunctions (to see these, execute the function with 'version' as the only argument)

Component name	Version date
GetZroots	18-Apr-2005
GetZtwiss	03-Mar-2006
BeamAttnB	18-Apr-2005
TrackThru	05-Oct-2005
VerifyLattice	18-Apr-2005
LucretiaCommon	02-August-2007
LucretiaOptics	24-May-2007
LucretiaMatlab	08-Mar-2006

Lucretia Author List The authors and maintenance crew for Lucretia is:


[Steve McJoy](#), SLAC, Stanford University, USA
[Evan Fritzsche](#), SLAC, Stanford University, USA
[Glen White](#), SLAC, Stanford University, USA

The management also gratefully acknowledges the assistance of Paul Emma, The Budker-Institute, Andrei Terzile, and Andy Wolski. The authors regularly got the idea for a beam-dynamics code which was a library of functions rather than a self-contained program from Nick Walker, whom we mocked for a lack in 1994, but in the end, Nick got the last laugh.



A GUI for studying the beamline data structures

CVS control via a CVSWEB interface

Stanford Linear Accelerator Center 

SLAC Controls Software Engineering Group
CVSWEB [SLAC Detailed](#)
[SLAC Computing](#)
[Controls Software](#)

/afs/slac/g/ilc/cvs/Lucretia/

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display **diffs** between revisions.

Current directory: [ILC-main](#) / [Lucretia](#)

File	Rev.	Age	Author	Last log entry
/				
/nomdir/				
/tree/				
/web/				
/src/				
home-alt.html	1.1.1.1	2 months	whitegr	Imported source
home.html	1.2	2 months	whitegr	test update

Show only files with tag:

Please direct questions to CVS_Majordomo-list_cvs-ilc@slac.stanford.edu

What's Missing?

- At this point, there are two key features which are missing
 - **Insertion devices – wigglers and undulators**
 - **Wraparound elements – IR solenoids around other magnets, quads wrapped around accelerating structures, etc.**
- Plus at least a couple of other potentially useful features
 - **RF kicks from asymmetric couplers**
 - **Crab cavity LOMs (fundamental mode excitation)**
 - **?**

- Needed for e- linac
 - **Positron production undulator is seen by the beam which goes to the IP**
 - Helical undulator
- Which features are required?
 - **Optical effects (fringe field focusing)?**
 - **Path length change and/or momentum dependence of path length?**
 - **Mean energy loss?**
 - **Growth in energy spread?**
 - **Emittance growth from ISR?**
- Can this be “dummied up” with lots of tiny sector bends?
 - **Probably hopeless for a ring with an insertion device**
 - **Maybe okay for single-pass beamline**

Wraparound Devices

- An ongoing issue for accelerators in general
- My proposal: add a new FIELDMAP data structure to represent the wraparound element
 - **Zero length field maps with integrated E and B fields in an xy grid**
 - **Cross-referencing between the field maps and the elements they wrap around**
 - **Field maps can have power supplies, girders, movers**
 - **Tracking makes use of Matlab interpolation functions to find best approximation of actual field at each ray's location**

- Some questions
 - **Do we need the insertion device as a true element type, or can we fake it with a mess of sector bends?**
 - **Is the proposed general design of the FIELDMAP data structure, and its use, sensible?**
 - **Are there other important features which should be incorporated in the near future?**
 - Not all users are necessarily linear collider people!