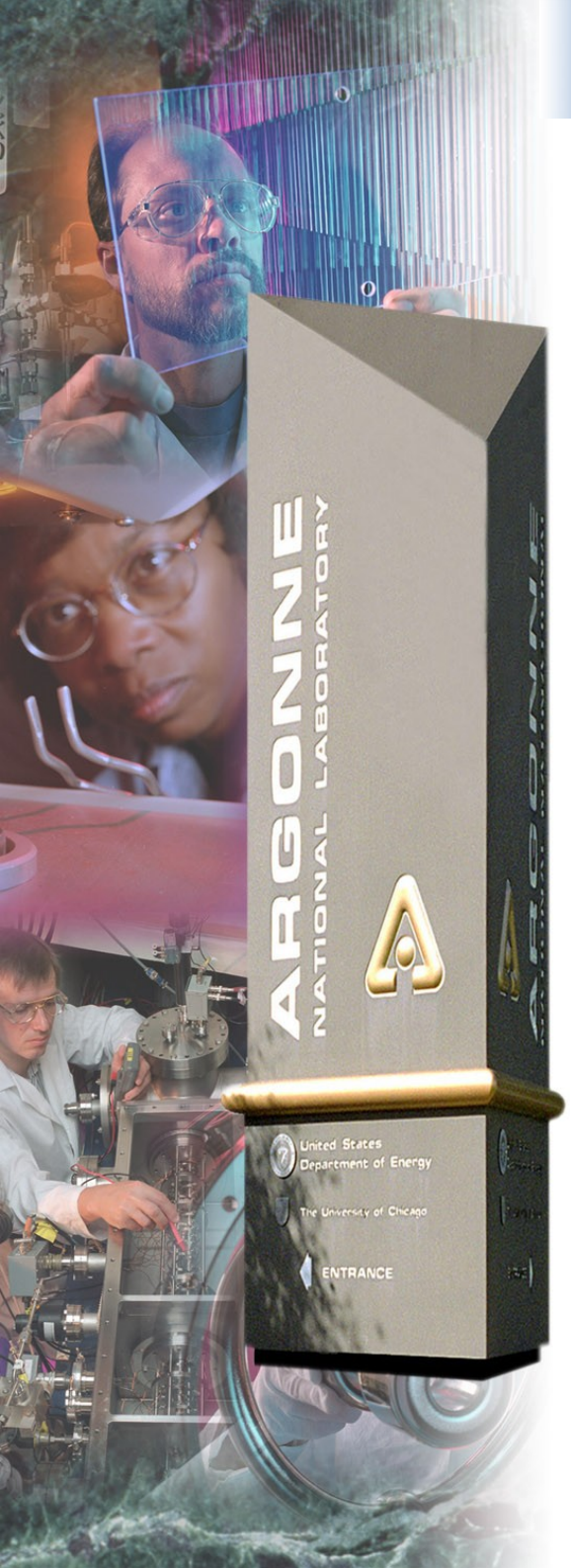


ARGONNE NATIONAL LABORATORY

Introduction to elegant

Michael Borland
Operations and Analysis Group
Accelerator Systems Division
Argonne National Laboratory

August 18, 2006

THE UNIVERSITY OF CHICAGO

Office of Science
U.S. DEPARTMENT OF ENERGY

Argonne National Laboratory is managed by
The University of Chicago for the U.S. Department of Energy

Outline

- Notable applications of **elegant**
- Program philosophy
- Details
  - Elements in **elegant**
  - Computations performed by **elegant**
- Related programs
- Examples
- Acknowledgements.

Notable Applications of elegant

- Beam transport lines for the SSRL pre-injector
- All APS accelerators now use **elegant**-designed optics
  - Design of the APS Positron Accumulator Ring and transport lines
  - Low-emittance optics development for the APS and APS booster
  - Design of bunch compressor and new linac optics for LEUTL
- APS top-up safety tracking
  - Ushered in a new mode of storage ring operation
- LCLS start-to-end and S2E jitter simulations
  - Discovered CSR-driven microbunching instability
- Used world-wide for FEL driver linac design, e.g.,
  - SLAC, DESY, BESSY, Sincrotrone Trieste, SPRing-8
- Used world-wide for ERL projects, e.g.,
  - Cornell, JLAB, BNL, Daresbury, JAERI
- Used to design the LTI Compact Light Source (commercial accelerator).

Quality Control

- **elegant** is important to APS operations and to users around the world
- Quality control is taken very seriously
- Source code is in CVS for version control and tracking
- Use extensive regression testing to "guarantee" that program updates don't break anything
  - When a feature is added, it is thoroughly tested
  - Selected test results are saved and used for checking of later versions
  - Program design allows us to largely automate this process.

Computer Science Philosophy

- Adopt a tool approach
  - Graphics and display functions external to program
  - Vastly simplifies the simulation code
  - Allows common pre- and post-processing tools for many codes
- External scripting required and supported
  - Program delivers data to user, not graphs of data
  - Data provided in a form that emphasizes scripting, not manual examination
  - Simplifies the program while empowering the user
  - Allows use of open-source scripting languages
  - Well suited to automated and cluster computing.

SDDS

- SDDS = Self-Describing Data Sets
  - A file protocol for data storage
  - A toolkit of programs that transform such files
  - A set of libraries for working with such files
    - *Support for C/C++, Tcl/Tk, Java, MATLAB, Python, FORTRAN*
  - A central component of the APS control system
- Knowing SDDS is key to using **elegant** effectively
  - Pre- and post-processing
  - Graphical and text output
  - Linking of multiple simulations and codes
  - Cluster computing.

Why Use SDDS Files?

- Programs that use SDDS can be made robust and flexible
  - Check existence, data-type, units of data instead of crashing or doing an incorrect calculation
  - Respond appropriately to the data provided
    - *Exit and warn user if required data is missing, has unknown units, etc.*
    - *Supply defaults for missing data (e.g., old data set)*
- Existing data doesn't become obsolete when the program is upgraded
- Self describing files make generic toolkits possible, which saves effort on writing pre- and post-processors
- SDDS-compliant programs are “operators” that transform data
  - Using pipes allows concatenating operators to make a complex transformation
  - UNIX-like philosophy: everything is a (self-describing) file.

Toolkit Approach to Physics

- **elegant** is a very complex SDDS "operator."
  E.g., it
  - Transforms phase space from beginning to end of a system
  - Transforms magnet parameters into, e.g., Twiss parameters, tunes
- All input to and output from **elegant** is in SDDS files *except*
  - Lattice structure
  - Command stream
- **elegant**'s capabilities are augmented by other operators
  - SDDS toolkit: a large collection of inter-operative data analysis, manipulation, and display programs
  - SDDS-compliant physics programs.

Capabilities of elegant: elements

- **elegant** has no assumed "best" approach to modeling
  - User can often select from expedient methods
- Matrix methods
  - Second-order matrices for drifts, solenoids, bends, and correctors
  - Third-order matrices for quads, sextupoles, and alpha magnets
  - User-supplied second-order matrix
- Symplectic methods:
  - Fourth-order Ruth integrator for bends, quads, sextupoles, higher multipoles.
  - Hamiltonian has exact energy dependence
  - Can invoke classical synchrotron radiation and quantum excitation for tracking.

Capabilities of elegant: elements

- Time-dependent elements
  - Kicker with user-specified waveform
  - Rf cavities with exact phase dependence
    - *Simple cavity with perfect source*
    - *Phase-, frequency-, and amplitude-modulated*
    - *Phase-, frequency-, and amplitude-ramped*
    - *User-specified on-axis field profile*
    - *Deflecting cavity with noise and modulation*
  - Momentum ramp
  - Traveling-wave accelerator.

Capabilities of elegant: elements

- Numerically-integrated elements
  - Planar undulator with co-propagating laser beam
  - Solenoid from user-supplied field map
  - Dipole with extended fringe fields
- Apertures/material
  - One- and two-sided rectangular, elliptical, and super-elliptical collimators
  - Scrapers with optional elastic scattering
  - Foil elastic scattering.

Capabilities of elegant: elements

- Collective effects
  - Intra-beam scattering in rings (L. Emery)
  - Short-range longitudinal and transverse wakes
  - Longitudinal- and transverse rf modes
  - Coherent synchrotron radiation in dipoles and drifts
  - Longitudinal space charge in drifts and cavities
  - Linear transverse space charge (A. Xiao)
- Diagnostics
  - Beam position monitors
  - Particle coordinate/property analysis points (to SDDS file)
  - Histogram analysis points (to SDDS file).

Capabilities of elegant: elements

- Miscellaneous
  - SCRIPT element will incorporate an external program as an element in an elegant lattice
  - User-specified scattering distribution
  - Lumped-element synchrotron radiation
  - Pick-up/driver elements for simulating transverse single-bunch digital feedback in rings.

Capabilities of elegant: Calculations

- Twiss parameter computation
  - Optionally-computed on-orbit and with errors
  - Includes radiation integrals
  - Includes chromaticity to third order and first-order tune shift with amplitude
  - Optional computation of coupled Twiss parameters (V. Sajaev)
  - SDDS output
- Transport matrix
  - Optionally computed on-orbit and with errors
  - SDDS and text output vs s (first- and second-order)
- Floor coordinates
  - Fully three-dimensional computation
  - SDDS output.

Capabilities of elegant: Calculations

- Variation
  - Nested sweeps of “any” parameters of any elements
  - Sweep using values supplied in an external SDDS file
- Errors
  - Errors for “any” parameter of any element
  - User-selectable distributions, amplitudes, cutoff, ...
  - Linking of errors between elements
  - Multiple error sets in one run
  - Loading of error sets from SDDS files
  - Saving of error values to SDDS files.

Capabilities of elegant: Calculations

- Saving and loading
  - Optimized lattice can be saved
    - *To a new (text) lattice file*
    - *To an SDDS lattice parameter file*
  - SDDS file can be manipulated with Toolkit, reloaded
  - SDDS file can be generated by another program to provide loadable custom error sets.

Capabilities of elegant: Calculations

- Closed orbit
  - Variable or fixed path length
  - On- and off-momentum
  - SDDS output
- Correction
  - Will correct tunes, chromaticities, and trajectory/orbit
  - Does these sequentially with optional iteration
  - SDDS output
    - *trajectory/orbit correction matrix*
    - *corrector strengths*
    - *correction statistics*
    - *quadrupole strengths*
    - *sextupole strengths.*

Capabilities of elegant: Calculations

- Optimization
  - Optimizes user-supplied penalty function depending on almost any calculated quantity
    - *Final or interior beam parameters from tracking*
    - *Final or interior Twiss parameters*
    - *Global values like equilibrium emittance, tunes, chromaticities*
    - *Final or interior matrix elements*
    - *Final or interior floor coordinates*
  - Uses Simplex by default, but has other methods.

Capabilities of elegant: Calculations

- Beam generation
  - Gaussian, hard-edge, and other distributions
  - Optional quiet-start sequences
  - Bunch train generation
  - Initial distribution can be saved to SDDS file
- Beam importation
  - Load beam from SDDS file
    - *Previously-generated and saved*
    - *Previously tracked*
  - Sequences of beams in one SDDS file.

Capabilities of elegant: Calculations

- Aperture finding (dynamic and physical)
  - Searches for aperture boundary with optional subdivision of search interval
  - Various search modes
    - *Single- and multi-particle search starting from large amplitudes*
    - *Search along one or more lines starting at zero amplitude*
- Particle losses
  - Optional output of locations and coordinates of lost particles
  - Optional output of initial coordinates of transmitted particles.

Capabilities of elegant: Calculations

- Miscellaneous
  - Frequency map analysis
  - Position-dependent momentum aperture
  - Orbit/trajectory amplification factors, with correction
  - Slice analysis along a beamline
  - Approximate SASE FEL evaluation
  - Subdivision of elements
  - Subprocess execution for pre-, intermediate-, or post-processing
  - Set up semaphore files for flagging run status.

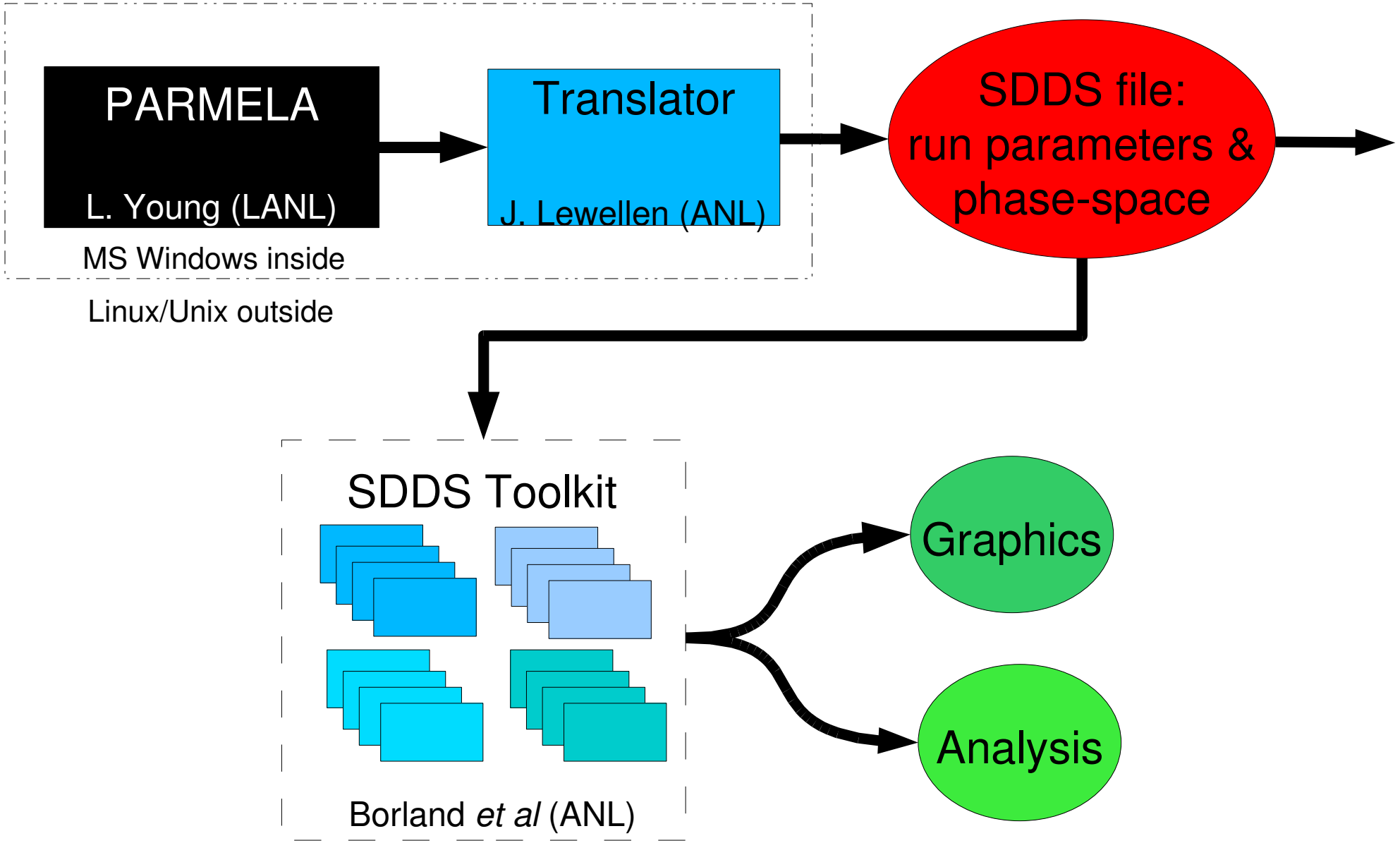
Related Programs

- SDDS files increase productivity by letting programs interface smoothly
- Programs that work with **elegant** include
  - **shower**, an interface to the EGS4 electron-gamma shower code (L. Emery)
  - **spiffe**, a PIC code for gun simulations (M. Borland)
  - **clinchor** computes single- and coupled-bunch growth rates due to HOMs (L. Emery)
  - URMEL/APS provides mode data in a form that **clinchor** and **elegant** accept (L. Emery)
  - ABCI/APS provides wake data that **elegant** accepts for tracking (L. Emery)
  - MAFIA/APS provides data that can be used in tracking with **elegant** (after a transformation) (L. Emery).

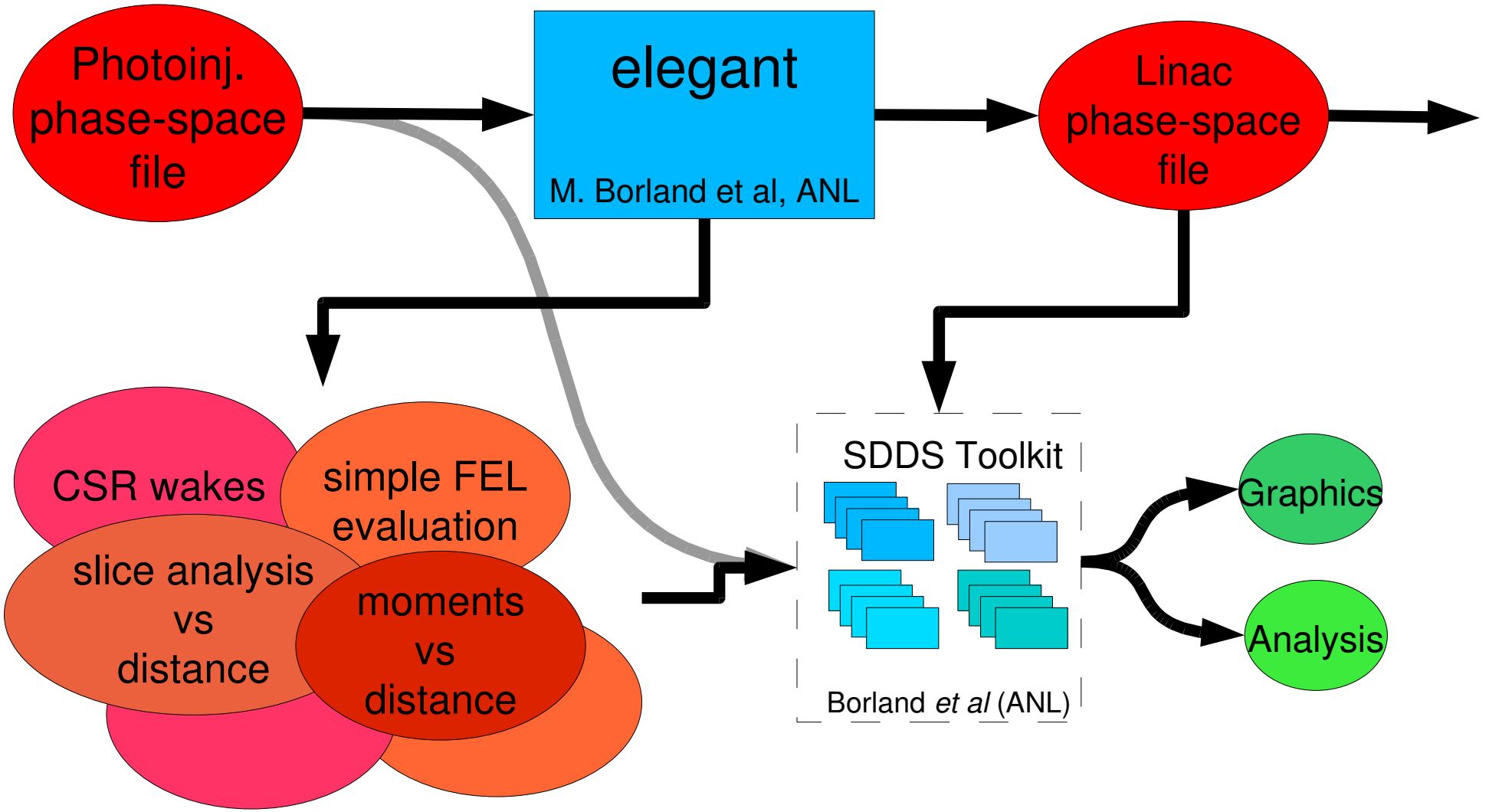
Related Programs

- **sddsrandmult** simulates random construction errors for multipoles and provides data for **elegant** tracking (M. Borland)
- **sddsfindresonances** finds resonances in frequency map data (H. Shang)
- **sddsemitproc** uses **elegant** output and experimental data to analyze emittance measurements (M. Borland)
- **sddsbrightness** uses **elegant** output to compute undulator brightness curves (H. Shang, R. Dejus)
- **sddsurgent** is an SDDS-compliant version the URGENT and US programs for calculation of undulator spectra and radiation patterns (H. Shang, X. Jiao)
- **sddsanalyzebeam** performs analysis of particle distributions (M. Borland)
- **sddsmatchtwiss** performs linear transformations of 6D phase space (M. Borland)
- **ibsEmittance** uses **elegant** output to compute intrabeam scattering effects (L. Emery, M. Borland)
- **haissinski** uses **elegant** output to compute bunch lengthening due to potential well distortion (L. Emery, M. Borland)
- etc. (See M. Borland et al., PAC 2001)

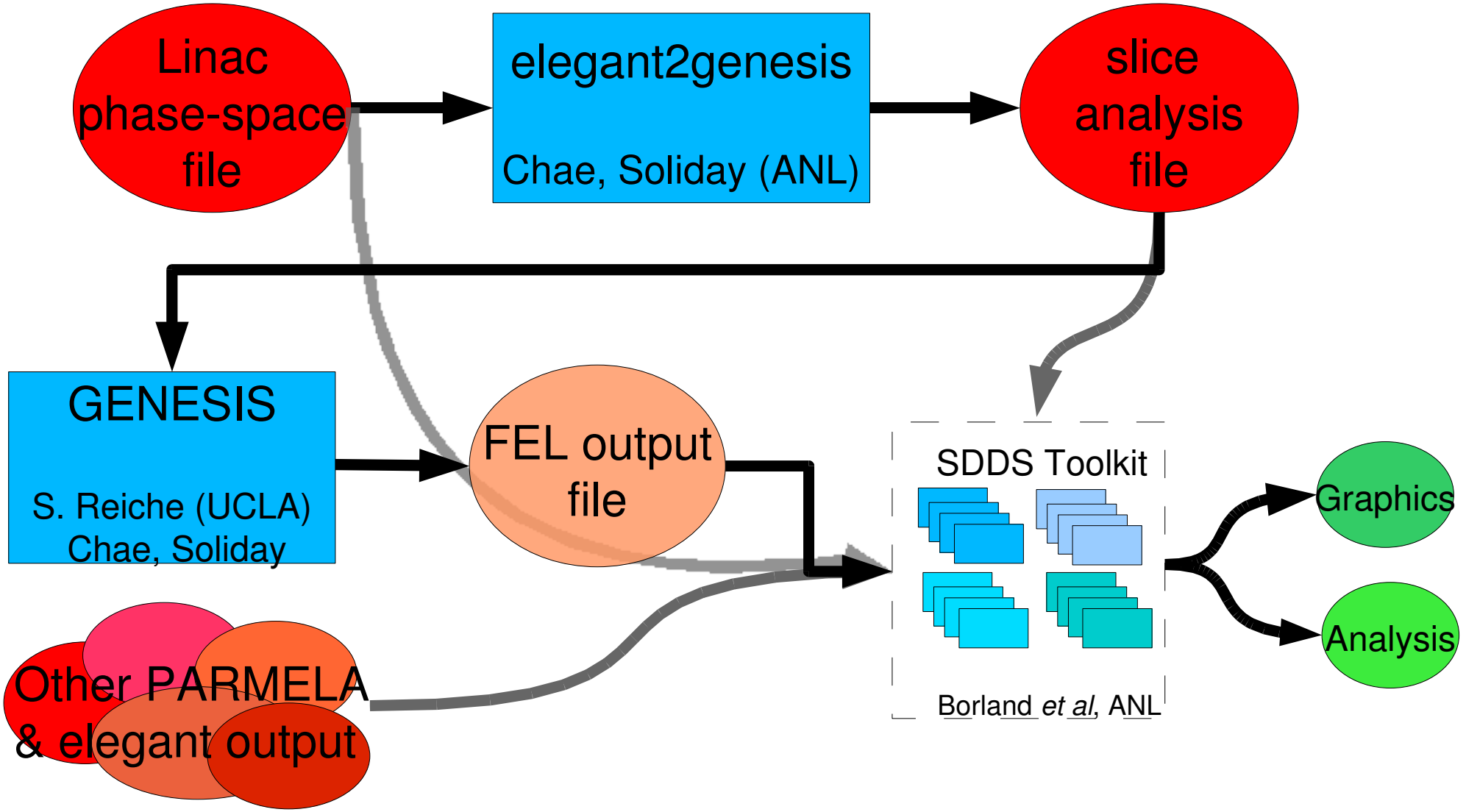
LCLS S2E Simulation Components



LCLS S2E Simulation Components

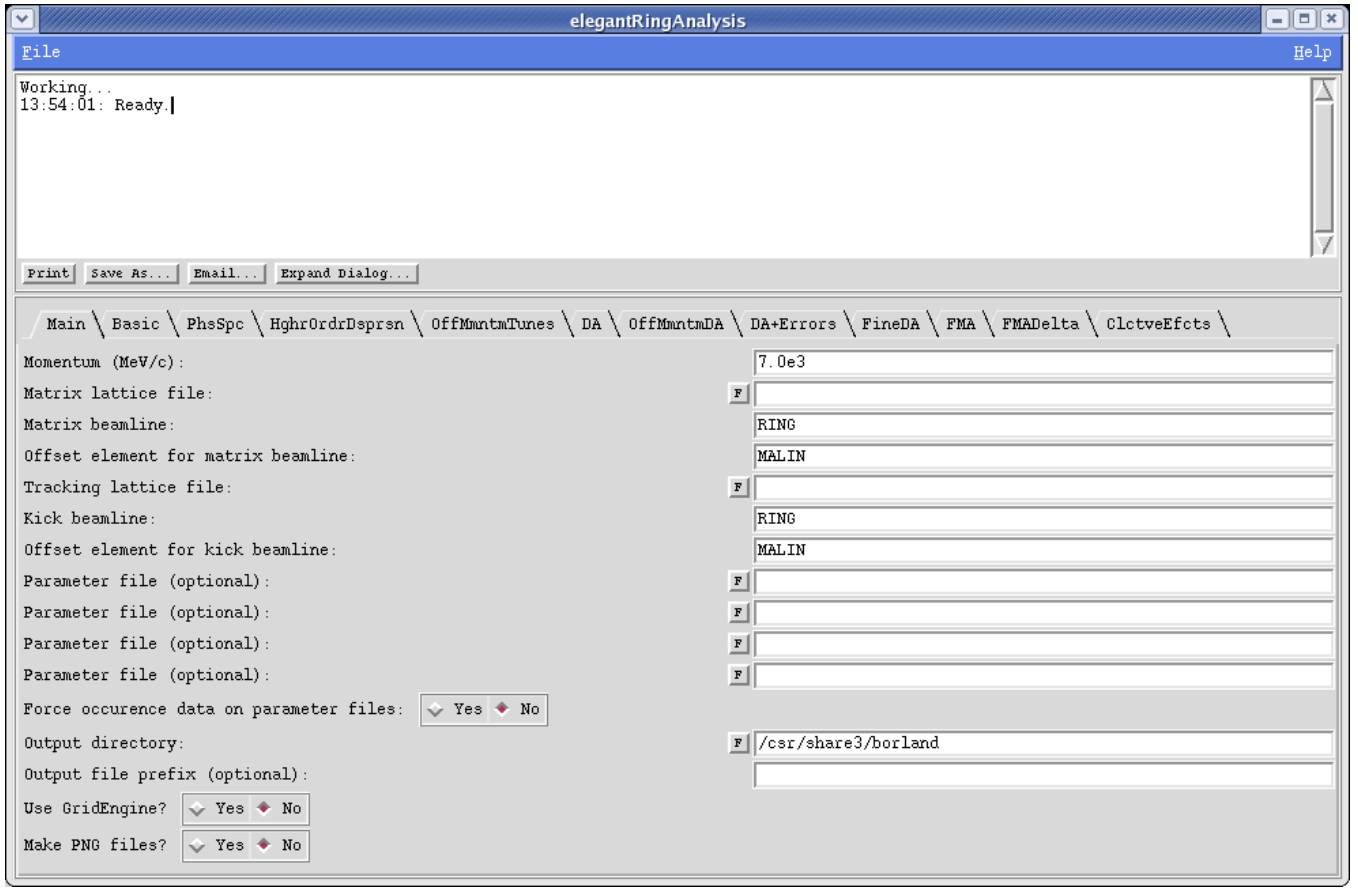


LCLS S2E Simulation Components



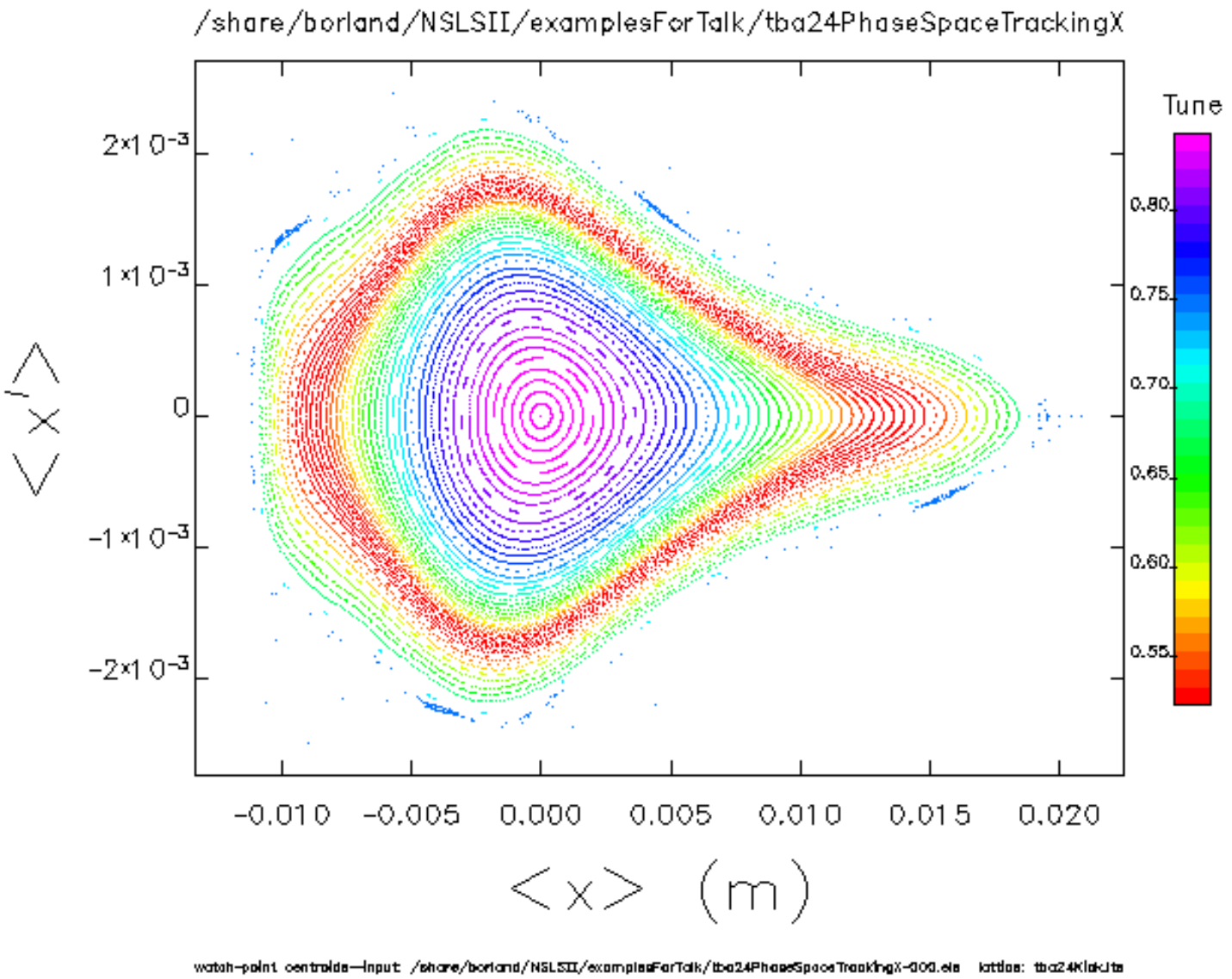
elegantRingAnalysis Application

A graphical user interface to many of the capabilities needed for storage ring evaluation.



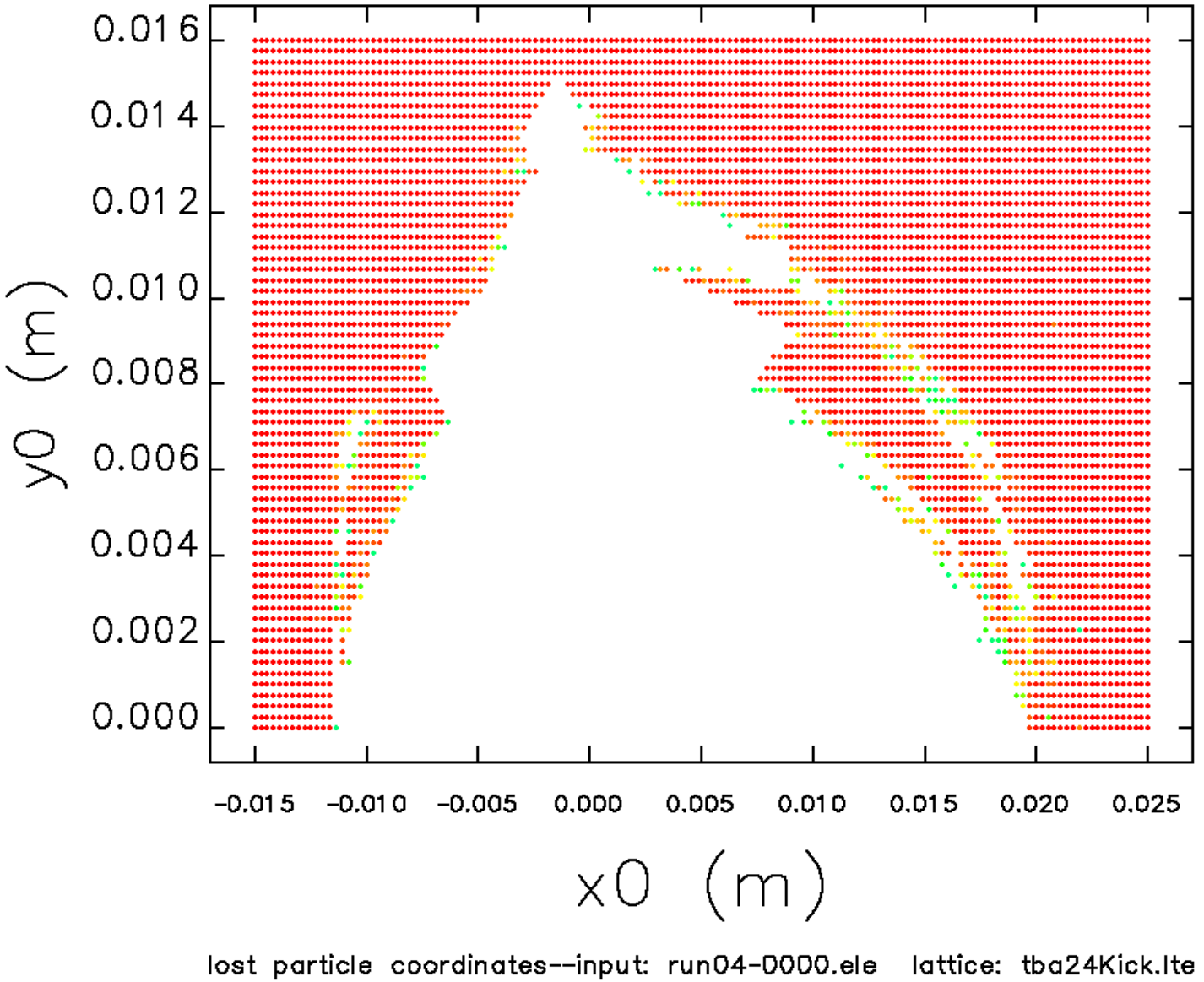
- Written in Tcl/Tk and SDDS
- Designed for use with a linux cluster for high throughput
- Mostly oriented toward tracking and dynamic aperture
- Also some collective effects calculations

Phase Space Tracking



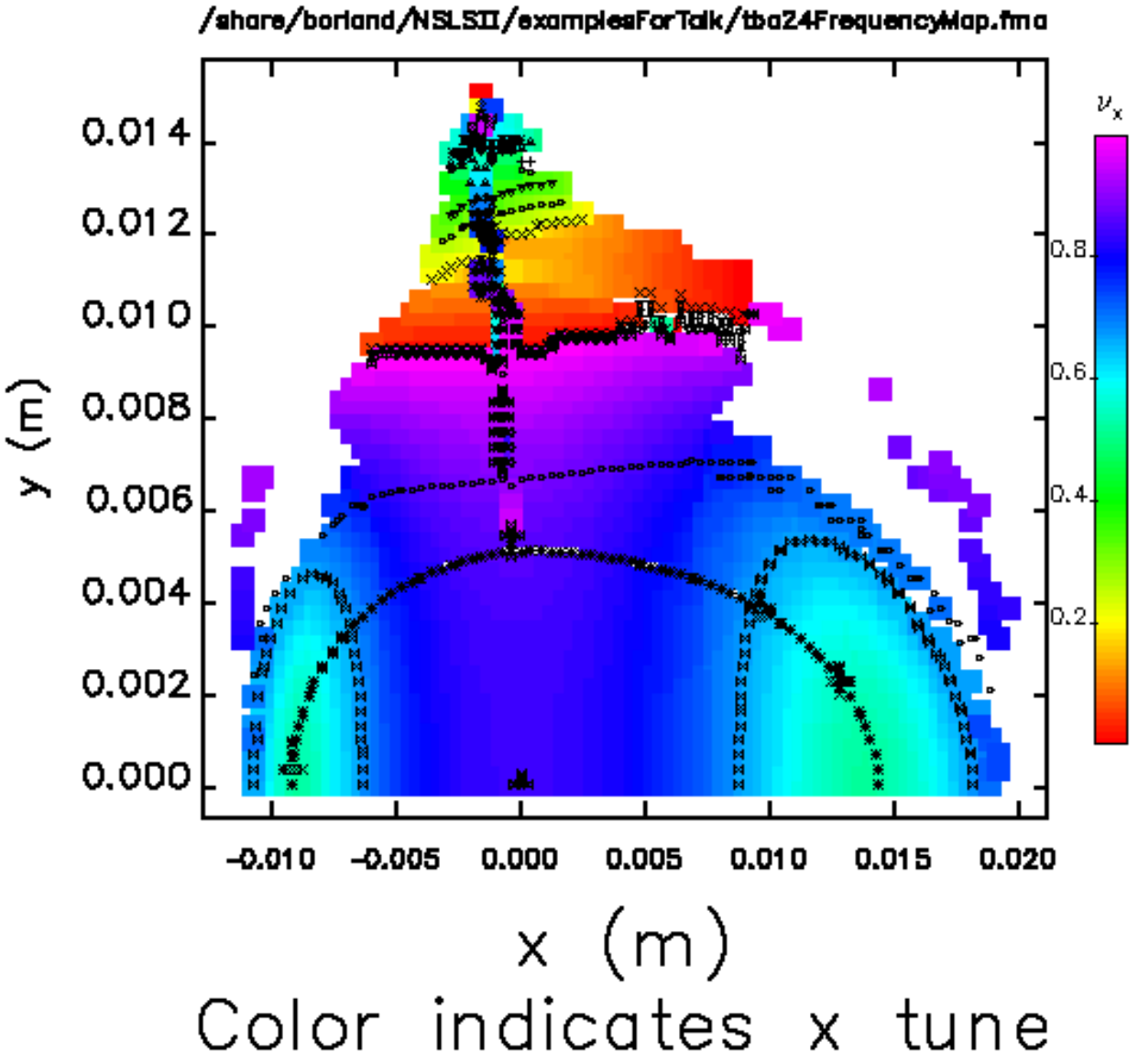
- 500-turn tracking, 15 kicks per element
- Color code shows tune
- 61 independent jobs
- Used ~60, 1.8 GHz CPUs simultaneously
- Took ~2 minutes

High-Resolution Dynamic Aperture



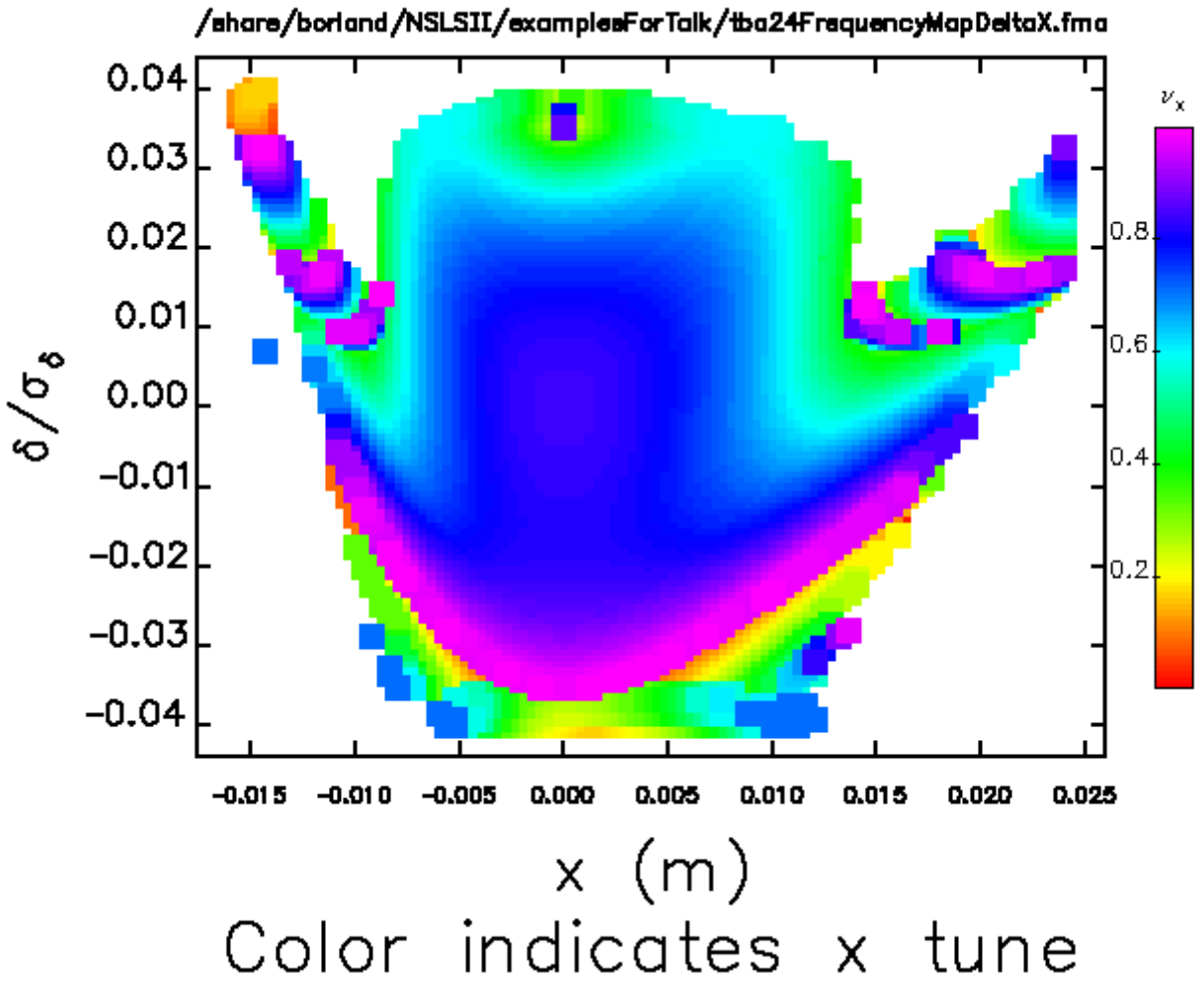
- 500-turn tracking, 15 kicks per element
- Color code shows turns survived
- 144 independent jobs
- Used ~50, 1.8 GHz CPUs simultaneously
- Took ~7 minutes

Frequency Map Analysis



- 512-turn tracking, 15 kicks per element
- Color code shows x tune
- Symbols show resonances
- 100 independent jobs
- Used ~80, 1.8 GHz CPUs simultaneously
- Took ~14 minutes

Frequency Map Analysis



- FMA in (x, δ) space
- 512-turn tracking, 15 kicks per element
- Color code shows x tune
- 100 independent jobs
- Used ~80, 1.8 GHz CPUs simultaneously
- Took ~14 minutes

Parallel elegant

- Have been parallelizing **elegant** for about 1 year
  - Approach allows gradual parallelization
  - Code switches to serial mode as needed
- All single-particle beam dynamics can now be done in parallel
  - Parallel version is in routine use at APS
  - Release expected within a month
  - Parallel efficiency of 95% with 512 processors on BlueGene/L
- Does load balancing to support heterogeneous cluster (e.g., weed)
- Regression tested against serial version.

Conclusions

- **elegant** is a capable, multi-purpose accelerator code
  - design
  - optimization
  - tracking
- **elegant** is at the center of an SDDS-compliant set of accelerator codes
  - Smooths the process of using many codes for complete accelerator design
- **elegant** is SDDS-compliant to allow users to automate simulations
  - LCLS start-to-end simulations
  - elegantRingAnalysis.

Acknowledgements

- Contributors to **elegant**:
  - M. Borland, L. Emery, V. Sajaev, Y. Wang, A. Xiao
  - Parallel version: Y. Wang, M. Borland
- Contributors to related tools:
  - M. Borland, R. Dejus, L. Emery, X. Jiao, H. Shang, R. Soliday
- SDDS Toolkit:
  - M. Borland, L. Emery, X. Jiao, H. Shang, R. Soliday
- Suggestions from users at APS and elsewhere are greatly appreciated.

Linux