elegantRingAnalysis: AN INTERFACE FOR HIGH-THROUGHPUT ANALYSIS OF STORAGE RING LATTICES USING elegant*

M. Borland †, ANL, Argonne, IL 60439, USA

Abstract

The code elegant is widely used for simulation of linacs for drivers for free-electron lasers. Less well known is that elegant is also a very capable code for simulation of storage rings. In this paper, we show a newly developed graphical user interface that allows the user to easily take advantage of these capabilities. The interface is designed for use on a Linux cluster, providing very high throughput. It can also be used on a single computer. Among the features it gives access to are basic calculations (Twiss parameters, radiation integrals), phase-space tracking, nonlinear dispersion, dynamic aperture (on- and off-momentum), frequency map analysis, and collective effects (IBS, bunch-lengthening). Using a cluster, it is easy to get highly detailed dynamic aperture and frequency map results in a surprisingly short time.

INTRODUCTION

Since there are many accelerator codes available, any code vying for attention of the user community must offer some special capabilities. In addition to differing in physics offerings, codes have various interfaces, including file-driven interfaces, scripting interfaces, and graphical user interfaces (GUIs). Typically, programs with file-driven or script features provide more capability, while GUI interfaces are easier for the beginner. elegant [1] is file-driven and uses the commandline SDDS Toolkit [2, 3] and common shell languages for scripting. This makes it well suited to automation and use on a Linux cluster, but presents challenges to the inexperienced user. Clearly, adding a GUI while keeping the flexibility and ability to parallelize and automate work would be highly desirable.

elegantRingAnalysis provides a GUI for elegant, allowing users to perform an extensive set of analyses for storage rings. Like elegant, elegantRingAnalysis is open source, so users can study the code and use it as a starting point for writing their own applications. The GUI allows the use of a Linux or other UNIX cluster, which can dramatically improve turn-around times.

FEATURES FOR STORAGE RING SIMULATION

Before describing elegantRingAnalysis in detail, we review some relevant features of elegant.

* Work supported by the U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

† borland@aps.anl.gov

Tracking codes typically belong to one of three types: those using matrix methods, those using traditional numerical integration (e.g., Runge-Kutta), and those using canonical integration. In elegant, the choice of methods is open to the user on a per-element basis for basic element types. For example, the QUAD and SEXT elements invoke a matrix implementation (up to third order), while the KQUAD and KSEXT elements invoke canonical integration with the exact Hamiltonian. This ability of elegant has the considerable advantage of allowing the user to fine-tune a simulation to, for example, improve speed or assess the importance of higher-order terms from a specific element.

For the APS ring, we find it acceptable to use canonical ("kick") elements for the quadrupoles and sextupoles while using a first-order matrix for the dipoles to increase speed. In smaller rings, the CSBEND element is recommended as it makes none of the large-radius approximations implicit in a matrix treatment. A limitation of the bending magnet simulation via matrices or CSBEND is that the fringe fields are not symplectic except to first order in transverse coordinates, a limitation of the thin-lens model.

elegant can impose random errors and perform correction of tunes, chromaticity, and orbit. Twiss parameters can be computed for the corrected machine, including the effects of the orbit and momentum offset. (However, coupled Twiss parameters are not provided.) In addition, higher-order tune shifts and spreads due to amplitude and momentum offset are computed.

elegant performs aperture determination, either due to dynamics, physical limitations, or both. Dynamic aperture (DA) is often computed without realistic apertures, which is potentially misleading for light source rings with errors and strong sextupoles, since these tend to have small vertical apertures that can limit the horizontal aperture due to nonlinear coupling. elegant provides a number of aperture types, including scrapers and one- and two-sided elliptical, hyper-elliptical, and rectangular collimators.

A related feature of elegant is frequency map analysis (FMA). Both DA and FMA can be very time intensive, making use of a cluster highly desirable.

elegant provides a variety of rf elements, including static, ramped, and modulated accelerating cavities and static deflecting cavities. Cavity modes, both monopole and dipole, may also be defined to simulate the long-range wakefield. Short-range wakes may be specified via a wake function or impedance function. If an instability is found, one can use elegant to simulate a simple single-bunch digital feedback system.

Although elegantRingAnalysis does not make use of this feature, elegant's optimization module can be used for matching. It permits the user to optimize not only Twiss parameters and matrix elements, but also chromaticities, tune spreads, radiation integrals (and related quantities such as the emittance), and beam properties obtained from tracking. `elegant`'s ability to save an optimized configuration to an SDDS file (called a "parameter file") and to load data from such files allows configurations to be manipulated using the SDDS toolkit. `elegantRingAnalysis` supports this by allowing the user to provide one or more parameter files to define the lattice to be analyzed. This does away with much tedious and error-prone editing of lattice files.

IMPLEMENTATION

`elegantRingAnalysis` is a separate application and requires no modification of `elegant`. The application consists of source code in the Tcl/Tk scripting language, along with a set of template files. The template files are `elegant` input files with substitutable fields that are filled in by the `elegantRingAnalysis` script according to user entries in the GUI.

Many of the tasks performed by the script are very CPU intensive. Hence, use of a Linux cluster is a key, unique feature of the program. The APS cluster supports job submission using Sun's open source Grid Engine (SGE) software. Thus, to use the cluster, `elegantRingAnalysis` simply needs to prepare, submit, and monitor jobs on the queue. When jobs are completed, the script must collate the results and present them to the user. Use of SGE is a remarkably simple and straightforward way to make use of cluster resources and is compatible with simultaneous use of the cluster for true parallel computing [4].

Because Tcl/Tk is an event-driven language, it is possible to simultaneously initiate many tasks from `elegantRingAnalysis`, with each task consisting of many jobs on the queue. The event loop is used to monitor the status of each group of jobs and initiate postprocessing. Job status is monitored through use of the semaphore file feature of `elegant`, which causes `elegant` to create a designated file only when the run has successfully completed. `elegantRingAnalysis` merely watches for creation of these files. An improvement would involve use of information from the queue along with the semaphore mechanism to detect failed jobs.

At minimum, the user must provide the name of a lattice file. Typically, this is a file with kick elements only, or with a judicious mixture of kick and matrix-based elements, suitable for dynamic aperture and other studies. A purely matrix-based lattice may be provided in addition and is used for analyses that do not require kick elements. The user may also specify one or more parameter files, which (as mentiond above) are SDDS files containing values for element parameters. These are the easiest way to overlay matching results onto a lattice structure. The script has the ability to save these and other user entries in a configuration file that can be loaded at a later time.

EXAMPLES

The GUI for `elegantRingAnalysis` is organized into a series of tabs, each of which allows the user to perform a specific type of analysis. Rather than show screen-shots of the contents of the tabs, we instead give examples of results from `elegantRingAnalysis`. The lattice used is the APS storage ring low-emittance lattice [5], including lattice calibration [6].

The `PhsSpc` tab of `elegantRingAnalysis` allows performing phase-space tracking with a series of particles with equispaced initial x or y amplitudes. The other coordinate is typically set to a small, fixed value so the tune can be determined. A fixed momentum offset may also be supplied. Tracking 31 particles for 1024 turns each on 31 processors takes about 2 minutes. After tracking, `elegantRingAnalysis` displays phase-space diagrams, FFTs of particle motion in both planes, and tunes as a function of initial amplitude.

Like most accelerator codes, `elegant` computes the dispersion function. It also computes higher-order dispersion at the start of the lattice. The `HghrOrdrDsprsn` tab permits higher-order dispersion computation as a function of position in the lattice. This is done by having `elegant` compute closed orbits for a series of momentum offsets, then processing the data with the SDDS Toolkit to perform fits to $\lfloor n/2 \rfloor^{\mathrm{th}}$ order, where n is the number of off-momentum orbits computed.

`elegant` of course computes the tunes ν_x and ν_y as well as the chromaticities up to third order (i.e., $\partial^n \nu_x / \partial \delta^n$, where n is 1, 2, or 3). However, in lattices with strong focusing and strong sextupoles, it is often best to compute the tunes vs momentum offset from tracking. This can be done with the `OffMmntmTunes` tab. After running a series of particles at a sequence of momentum offsets, the data is analyzed to provide tunes vs momentum as well as tunes superimposed on a resonance diagram.

Dynamic aperture (DA) is an important property of any ring lattice and a time-consuming one to optimize and compute. We report elsewhere [7] in this conference on cluster-based tools to optimize DA. `elegantRingAnalysis` provides several methods for performing this analysis. The first, under the `DA` tab, simply uses `elegant`'s standard, single-processor algorithm to get a relatively coarse DA. The second, under the `OffMnmtmDA` tab, uses the same algorithm on multiple processors to get the DA for a series of momentum offsets. The third, under the `DA+Errors` tab, uses the same algorithm on multiple processors to get the DA including errors and orbit correction, for a single momentum offset. At this time, tune and chromaticity correction are not included, even though `elegant` has this capability.

The fourth and final DA-related tab is `FineDA`, which uses multiple processors to compute a high-resolution dynamic aperture. The algorithm in this case is different from the others and consists simply of tracking a set of lines of constant initial x, one line per processor. For the APS,

0-7803-8859-3/05/$20.00

tracking a single line of 101 particles 500 turns takes a few minutes. With 100 processors, a 201x101 grid is thus computed in about 10 minutes. An example of such a result is shown in Figure 1. The individual points are color-coded by the number of turns survived.

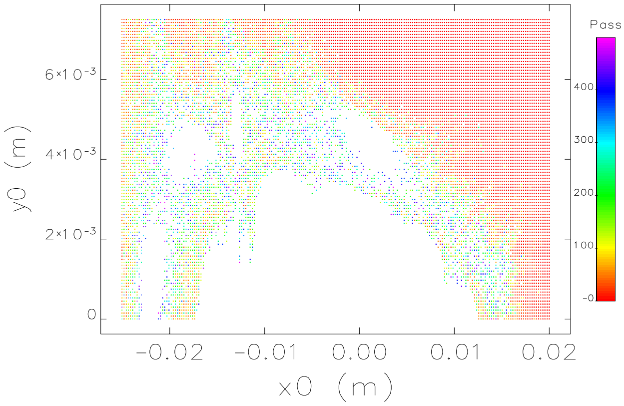

Figure 1: APS dynamic aperture (without physical apertures). The color indicates the number of turns survived before loss.

Frequency map analysis (FMA) is a useful technique for understanding and improving dynamic aperture. In this technique, a grid of particles is tracked and the particles' individual betatron tunes are determined by NAFF [8]. `elegantRingAnalysis` performs FMA in the same fashion as it performs fine DA tracking, by splitting the problem up into a series of lines of constant x. Unlike fine DA analysis, where we display the lost particles, in FMA we display the surviving particles. FMA is considerably more time-consuming than dynamic aperture calculations, since we have to not only track the particles, but also accurately determine the oscillation frequency for each particle. `elegantRingAnalysis` can display tune maps, in which the points are color-coded by the initial x or y amplitude. An example is shown in Figure 2. It can also display x-y maps, in which the points are color-coded by the x or y tune. An example is shown in Figure 3. Using approximately 40 processors with typical 1.5-GHz clock speeds, this analysis took about 30 minutes.

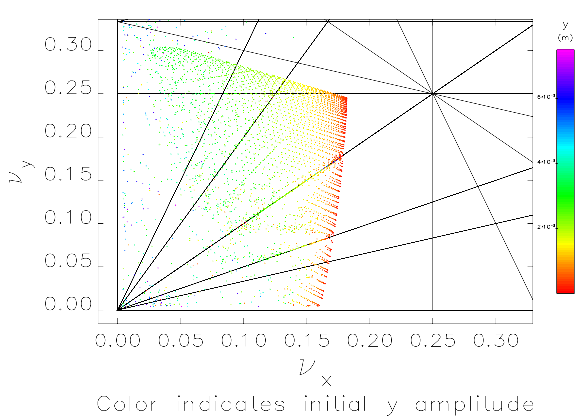

Figure 2: Frequency map analysis for the APS.

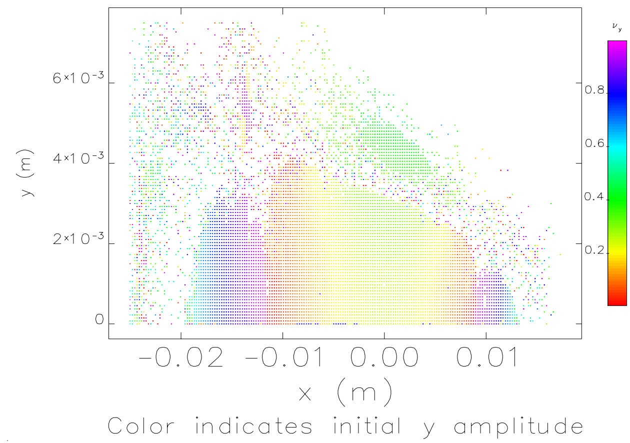

Figure 3: Frequency map analysis for the APS.

CONCLUSION

`elegant` is a powerful tool for simulation of accelerators, including storage rings. The `elegantRingAnalysis` script makes `elegant` easier to use, and in particular makes it simple to use a Linux cluster for high-throughput computing. Using `elegantRingAnalysis` and a Linux cluster, analyses such as high-resolution dynamic aperture and frequency map analysis can be completed in a matter of minutes.

ACKNOWLEDGEMENT

We thank V. Sajaev for providing input files for the APS including lattice correction, sextupole strengths, and coupling. We also thank H. Shang and R. Soliday for improvements to `sddsplot`, which greatly improved the usefulness of the FMA figures.

REFERENCES

[1] M. Borland, Advanced Photon Source LS-287, September 2000.

[2] R. Soliday et al., Proc. PAC 2003, 3473 (2003).

[3] M. Borland et al., Proc. PAC 2003, 3461 (2003).

[4] R. Soliday, private communication.

[5] L. Emery et al., Proc. EPAC 2002, 218 (2002).

[6] V. Sajaev et al., Proc. EPAC 2002, 742 (2002).

[7] H. Shang et al., "A Parallel Simplex Optimizer and its Application to High-Brightness Storage Ring Design," these proceedings.

[8] J. Laskar et al., Physica D 56 (1992) 253-269.