ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS062

ZGOUBI STATUS: IMPROVED PERFORMANCE, FEATURES, AND GRAPHICAL INTERFACE*

D. T. Abell†, P. Moeller, R. Nagler, B. Nash, I. V. Pogorelov, RadiaSoft LLC, Boulder, CO, USA
F. Méot, Brookhaven National Lab, Upton, NY, USA,
I. B. Beekman, ParaTools, Inc., Eugene, OR, USA
D.W. I. Rouson, Sourcery Institute, Oakland, CA, USA

Abstract

The particle tracking code Zgoubi [1,2] is used for a broad array of accelerator design studies, including FFAs [3] and EICs [4,5]. Zgoubi is currently being used to evaluate the spin polarization performance of proposed designs for both JLEIC [6] and eRHIC [7], and to prepare for commissioning the CBETA BNL-Cornell FFA return loop ERL [8]. We describe our on-going work on several fronts, including efforts to parallelize Zgoubi using new features of Fortran 2018 [9], and a new implementation of Zgoubi's particle update algorithm. We also describe a new, web-based graphical interface for Zgoubi.

ZGOUBI'S HISTORY AND FEATURES

Zgoubi was originally developed in the 1970s as a spectrometer code. This heritage explains it's focus on, and capabilities for, detailed particle integration in spatially-varying magnetic fields. Zgoubi's particle update algorithm integrates the Lorentz force equation, $\mathrm{d}\vec{p}/\mathrm{d}t = q(\vec{E} + \vec{v} \times \vec{B})$, for a charged particle in electric field $\vec{E}$ and magnetic field $\vec{B}$. For the independent variable, however, it uses *distance s along the particle trajectory*. Using a prime ($'$) to denote differentiation with respect to s, defining the *normalized velocity* $\vec{u} = \vec{v}/v$, and expressing the particle momentum as

$$\vec{p} = m\gamma\vec{v} = m\gamma v\vec{u} = q(B\rho)\vec{u}, \tag{1}$$

where $(B\rho)$ denotes the usual magnetic rigidity, Zgoubi writes the Lorentz force law in the form

$$\frac{\mathrm{d}}{\mathrm{d}s}(B\rho)\vec{u} = (B\rho)'\vec{u} + (B\rho)\vec{u}' = \frac{1}{v}\vec{E} + \vec{u} \times \vec{B}. \tag{2}$$

Using this equation together with derivatives of the known electric and magnetic fields, Zgoubi can construct the sequence of derivatives $(B\rho)'$, $\vec{u}'$, $(B\rho)''$, $\vec{u}''$, *etc.* Zgoubi then uses these derivatives to update both position $\vec{r}$, and velocity $\vec{u}$ according to the Taylor series approximation

$$\vec{r}^{\,f} \approx \vec{r} + \Delta s\,\vec{u} + \frac{\Delta s^2}{2!}\vec{u}' + \cdots + \frac{\Delta s^6}{6!}\vec{u}^{(5)}, \tag{3a}$$

$$\vec{u}^{\,f} \approx \vec{u} + \Delta s\,\vec{u}' + \frac{\Delta s^2}{2!}\vec{u}'' + \cdots + \frac{\Delta s^5}{5!}\vec{u}^{(5)}. \tag{3b}$$

Since it's original application to spectrometer design, Zgoubi's capabilities have been extended to include cyclic optics (*i.e.* tracking in circular accelerators), spin dynamics [10], electric fields [11], radiation damping effects [12], as well as sophisticated longitudinal dynamics and in-flight particle decay. [13]. Here, for example, we give some brief specifics concerning the oldest and newest features available to Zgoubi users:

Field maps. A particularly valued member of Zgoubi's feature set is its ability to track particles through magnet field maps. This capability, available from Zgoubi's earliest days, has seen especially wide use since the renaissance of fixed-field alternating-gradient accelerators (FFAs). With their wide dynamic range and large aperture-to-length-ratio magnets, FFAs require detailed integration of particle trajectories to validate their design.

A very recent application of Zgoubi's field map capability is to the Cornell-BNL Energy-recovery-linac Test Accelerator (CBETA) [14]. This machine is a four-pass, 150 MeV, 40 mA energy recovery linac, with a pair of four spreader arcs that match each end of the 36 MeV linac to the FFA return arc. The principal CBETA FFA cell features a closely-spaced pair of Halbach magnets with aperture-to-aspect ratios ~ 0.7. Figure 1 shows a Zgoubi energy scan of the periodic orbits crossing this cell [8].

Closed-orbit correction. Zgoubi's newest capability is closed-orbit correction using the method of Singular Value Decomposition (SVD). To use this feature, one inserts into

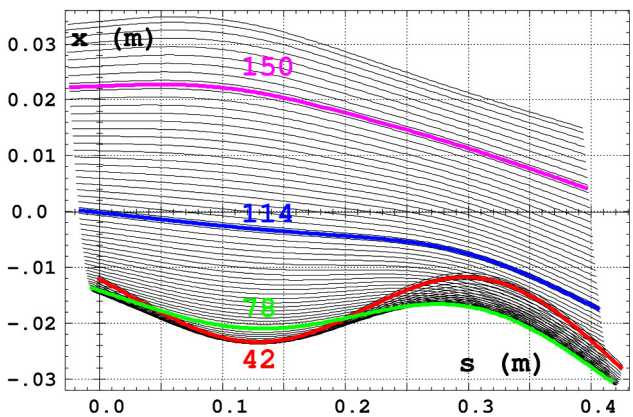

Figure 1: Zgoubi computation of periodic orbits across the 5° Halbach cell for the CBETA ERL, modeled using OPERA field maps. There are 64 different energies from 40 MeV to 166 MeV, with the four design energies, 36 MeV apart, highlighted.

* Work supported by the US Department of Energy, Office of Science, Office of Nuclear Physics, including grant No. DE-SC0017181.
† dabell@radiasoft.net

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS062

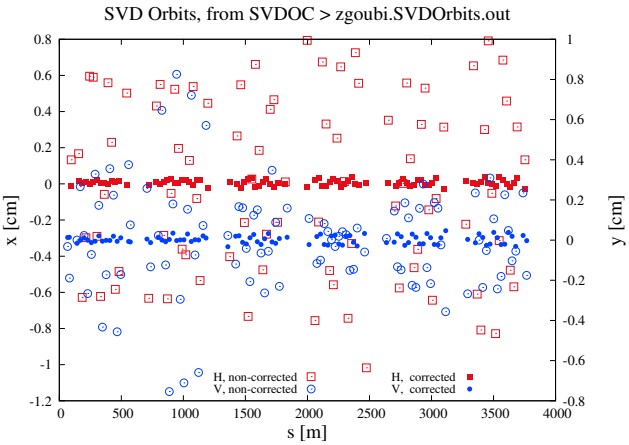

Figure 2: Orbit correction in the RHIC lattice using SVDOC. Red and blue marks correspond, respectively, to horizontal and vertical orbits. Open and filled marks correspond, respectively, to initial and corrected orbits.

the lattice both pickup and corrector elements. (This does *not* happen automatically!) Then orbit correction is activated via the new Zgoubi keyword SVDOC, which assumes that one has already computed a closed orbit or other reference trajectory (via the FIT keyword). Among other inputs, this command requires the names of the pickup and corrector elements to be used for orbit correction. Then, "under the hood", an orbit response matrix (ORM) is computed by applying angular orbit kicks at each corrector and measuring the resulting orbit error at each pick-up. The resulting response matrix (actually its inverse) can be saved to a file. This procedure can be carried out for as many as twenty different reference trajectories, which allows one to correct energy-dependent orbits in a multi-pass ERL or FFA.

After the ORM has been computed, one can apply ERRORs to the lattice and use the computed response matrix to correct the resulting distorted orbit. This allows one to test if the correctors placed in the lattice suffice to adequately correct against the possible range of machine errors one expects to see (and can properly simulate). Indeed, SVDOC can generate, in a single run, an arbitrary number of corrected random orbits, and thereby develop statistics concerning closed-orbit correction of a given lattice. Figure 2 shows an application to the 3.8 km RHIC polarized proton lattice.

IMPROVING PERFORMANCE

There are two avenues for improving Zgoubi's performance: speeding up the fundamental algorithms, and taking advantage of modern Fortran [9] to parallelize the code. To speed up the algorithms, we use the fact that the hierarchy of computations implemented in Zgoubi is very much like that used in the algorithms of truncated power series algebra (TPSA). This fact has allowed us to reimplement Zgoubi's particle update algorithm in a manner that reduces its memory footprint by a factor of about three, and the arithmetic involved by a similar factor.

We also make use of Modern Fortran, which provides facilities for *fine-* and *coarse-grain* levels of parallelism: Fine-grain—or loop level—parallelism requires the use of *pure functions*, which means no side effects are allowed. For example, the new loop modifier `concurrent`, as in

```
do concurrent(iord = 0:nord)
  B(iord + 1) = derivB(iord)
end do
```

constitutes a promise to the compiler that all iterations of the loop are independent of one another. This promise frees the compiler to perform optimizations it might not otherwise perform. In return, the compiler will do you the favor of issuing an error when you appear to break your promise!

Coarse-grain—or processor level—parallelism includes *collective operations*, which can operate on data spread across multiple processors. For example, one can find the min, max, and sum of such data using the Fortran intrinsics `co_min(a)`, `co_max(a)`, and `co_sum(a)`. These are specific data reduction operations. The *generic* version is `co_reduce(a, op)`, which performs the data reduction using any (possibly user-defined) binary operation `op`.

Coarse-grain parallelism also includes the use of *coarrays*, which "answer the question, 'What is the smallest change required to convert Fortran into a robust and efficient parallel language?'" [15]. Coarray syntax implements a Single Program Multiple Data (SPMD) model, in which a single program is replicated across units called *images* (usually processors), and the number of images may be chosen at run time. You must still devise appropriate parallel algorithms, but the case of non-interacting particles is essentially trivial.

Examples of declared coarrays include the following:

```
real :: a[*]
real, dimension(10) :: x[*], y[*]
real, dimension(0:21,6) :: m[*]
type(particle) :: ptcl(128)[*]
```

Here the tokens `[*]` denote that the given variable—scalar or array, intrinsic or derived type—may differ on the different images. The "smallest change" aspect of coarrays refers to the fact that any coarray variable lacking the square braces refers to the value *on the current image*. Only when one requires communication between images does one require the explicit use of square braces in the body of one's code. For example,

```
x(:) = y(:)[q]
```

asks that coarray `x` on the *local* image be given the value of coarray `y` on *remote* image `q`.

SIREPO INTERFACE FOR ZGOUBI

A browser-based user interface for Zgoubi has been implemented using RadiaSoft's open-source Sirepo software [16]. The current interface now allows a user to create and edit beamline elements and lattices, produce and view reports of the bunch and Twiss parameters, and execute Zgoubi

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS062

Figure 3: The *Lattice* tab of the Sirepo-Zgoubi interface.

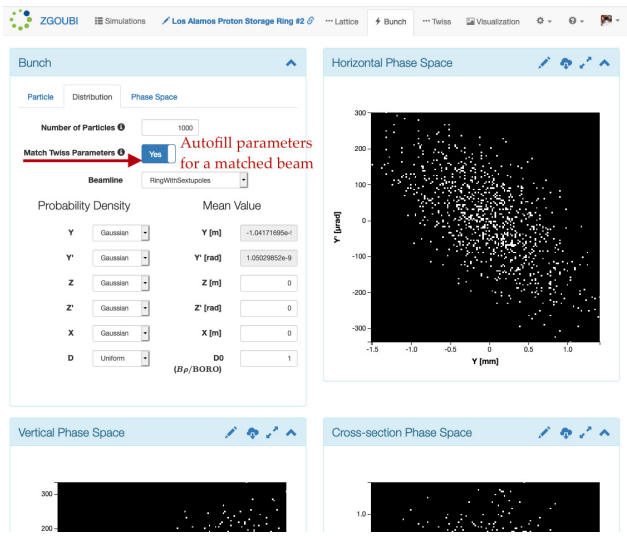

Figure 4: The *Bunch* tab of the Sirepo-Zgoubi interface.

simulations of the given lattice. Sirepo-Zgoubi (`https://beta.sirepo.com/#/zgoubi`) is now available as "beta" software. *Please provide us with feedback!* We have used several well-known machine lattices to test the implementation so far; these include the Los Alamos PSR, the AGS Booster, and the EMMA FFAG lattice. We have also done some preliminary work with Cornell's CBETA lattice, which is especially challenging because it includes multiple arc sections for different energies.

The Sirepo-Zgoubi *Lattice* tab, Fig. 3, allows a user to create and edit beamline elements, assemble those elements into a hierarchy of beamlines, and arrange those beamlines into a complete lattice. At the upper left of the tab, one sees a zoomable preview of the selected beamline or lattice—in this case the ten-cell lattice of the Los Alamos Proton Storage Ring. The *Bunch* tab, Fig. 4, includes an option that allows one to match the beam to the lattice. In this case,

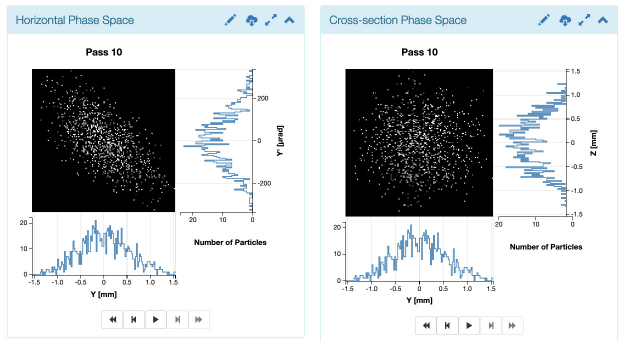

Figure 5: Phase-space plots in the *Visualization* tab of the Sirepo-Zgoubi interface.

Figure 6: The *Twiss* tab of the Sirepo-Zgoubi interface.

Sirepo-Zgoubi automatically uses Zgoubi's `FIT` procedure to compute the closed orbit, and then computes the periodic Twiss parameters. Those parameters are then used to define a matched distribution to initialize the desired beam. The *Visualization* tab, see Fig. 5, allows a user to perform multi-turn tracking of a defined bunch through a beamline, and then view the particle distribution either as a movie or turn-by-turn. And the *Twiss* tab, Fig. 6, allows one to compute and visualize, for a selected beamline, the reference trajectory (Optics panel at lower right) and the Twiss parameters (at upper right). The Twiss tab also displays a summary of the important beam and lattice parameters.

REFERENCES

[1] F. Méot, "Zgoubi", `http://sourceforge.net/projects/zgoubi/`.

[2] F. Méot, "Zgoubi users' guide", Tech. Rep. FERMILAB-TM-2010, Fermi National Accelerator Laboratory, Batavia, IL, Oct. 1997.

[3] F. Lemuet and F. Méot, "Developments in the ray-tracing code Zgoubi for 6-D multiturn tracking in FFAG rings", *Nucl.*

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS062

Instrum. Methods Phys. Res., Sect. A, vol. 547, pp. 638–651, Aug. 2005. doi:10.1016/j.nima.2005.03.156

[4] F. Méot, S. Brooks, V. Ptitsyn, D. Trbojevic, and N. Tsoupas, "Tracking studies in eRHIC energy recovery recirculator", Tech. Rep. BNL-108286-2015-IR (eRHIC/45), Brookhaven National Laboratory, Upton, NY, July 2015.

[5] F. Lin, Ya. S. Derbenev, V. S. Morozov, Y. Zhang, and D. P. Barber, "Study of electron polarization dynamics in the JLEIC at JLab", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, pp. 3218–3220. doi:10.18429/JACoW-IPAC2017-WEPIK114

[6] J. L. Martinez-Marin and B. Mustapha, "Spin dynamics in the JLEIC alternative pre-booster ring", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.–May 2018, pp. 87–90. doi:10.18429/JACoW-IPAC2018-MOPMF004

[7] V. H. Ranjbar, M. Blaskiewicz, J. M. Brennan, S. J. Brooks, D. M. Gassner, H.-C. Hseuh, I. Marneris, F. Méot, M. G. Minty, C. Montag, V. Ptitsyn, K. S. Smith, S. Tepikian, F. J. Willeke, H. Witte, B. P. Xiao, A. Zaltsman, and I. V. Pogorelov, "Progress on RCS eRHIC Injector Design", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.–May 2018, pp. 115–117. doi:10.18429/JACoW-IPAC2018-MOPMF016

[8] F. Méot, N. Tsoupas, S. Brooks, and D. Trbojevic, "Beam dynamics validation of the Halbach technology FFAG cell for Cornell-BNL energy recovery linac", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 896, pp. 60–67, July 2018. doi:10.1016/j.nima.2018.04.017

[9] J. Reid, "The new features of Fortran 2018", JTC1/SC22/WG5 N2161, Aug. 2018. https://wg5-fortran.org/documents.html

[10] F. Méot, "A numerical method for combined spin tracking and ray tracing of charged particles", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 313, pp. 492–500, Mar. 1992. doi:10.1016/0168-9002(92)90829-S

[11] F. Méot, "Generalization of the Zgoubi method for ray-tracing to include electric fields", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 340, pp. 594—604, Mar. 1994. doi:10.1016/0168-9002(94)90143-0

[12] F. Méot, "Simulation of radiation damping in rings, using stepwise ray-tracing methods", *J. Instrum.*, vol. 10, p. T06006, June 2015. doi:10.1088/1748-0221/10/06/T06006

[13] F. Méot, "The ray-tracing code Zgoubi—status", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 767, pp. 112–125, Dec. 2014. doi:10.1016/j.nima.2014.07.022

[14] G. H. Hoffstaetter, D. Trbojevic, C. Mayes, *et al.*, "CBETA design report, Cornell-BNL ERL Test Accelerator", June 2017. https://arxiv.org/abs/1706.04245

[15] J. Reid, "Coarrays in the next Fortran standard", Apr. 2010. https://wg5-fortran.org/N1801-N1850/N1824.pdf

[16] M. S. Rakitin, P. Moeller, R. Nagler, B. Nash, D. L. Bruhwiler, D. Smalyuk, M. Zhernenkov, and O. Chubar, "Sirepo: an open-source cloud-based software interface for x-ray source and optics simulations", *J. Synchrotron Rad.*, vol. 25, pp. 1877–1892, Nov. 2018. doi:10.1107/S1600577518010986