FEATURES AND APPLICATIONS OF THE PROGRAM ELEGANT *

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

Abstract

`elegant` [1] is an open-source accelerator design and simulation code that has been in use and development for nearly three decades. In that time, it has evolved into a fairly general code for the design and modeling of linacs and storage rings, due in no small measure to suggestions and feedback from users world-wide. The code is best known for modeling of linacs for free-electron lasers and particularly its relatively fast and straightforward modeling of coherent synchrotron radiation in magnetic bunch compression systems. This capability led to the discovery of a microbunching instability in such systems, thus helping to seed a new field of research. `elegant`'s capabilities are enhanced by the use of self-describing data files and the Self-Describing Data Sets (SDDS) Toolkit. In this paper, we briefly review the features and capabilities of the code, then give a series of application examples from simulation of linear accelerators and storage rings.

OVERVIEW AND HISTORY

This paper begins with a brief sketch of the history of `elegant`. Following this, we provide an overview of features and capabilities. A somewhat more detailed overview is given of the use of self-describing files and external tools, an important and unusual feature of `elegant`. Following this, we describe three significant contributions involving `elegant`, namely: start-to-end simulation and discovery of the coherent-synchrotron-radiation-driven microbunching instability; top-up safety tracking; and direct optimization of storage ring nonlinear beam dynamics. We end with a summary and acknowledgments.

The code `elegant` began as a graduate student project while the author was working at the Stanford Synchrotron Radiation Laboratory (SSRL) on the design of the optics and bunch compression system for the SSRL injector [2], which was to be supplied with electrons by a thermionic rf gun. Out of frustration with existing codes, which proved difficult modify, the author began work on something that he hoped would be more elegant. The original meaning of the name `elegant` was "ELEctron Generation ANd Tracking," reflecting the code's somewhat limited initial purpose.

In 1991 development of `elegant` moved to the Advanced Photon Source (APS), where it was adapted for storage rings. In the late 1990's, `elegant` was upgraded to address the needs of free-electron lasers (FELs), particularly the Low-Energy Undulator Test Line (LEUTL) [3] project at the APS and the Linac Coherent Light Source (LCLS) [4] project at SLAC. As described in more detail below, this led to discovery [5, 6] of a microbunching instability driven by coherent synchrotron radiation (CSR) [7].

In the mid-2000's, work began on parallelization of `elegant` [8]. Presently, more than 90% of the beamline elements are parallelized. Parallelization also covers input/output operations [9], optimization, dynamic and momentum aperture searches, and frequency map analysis.

OVERVIEW OF CAPABILITIES

`elegant` is an open-source code for design and modeling of single- and multi-pass accelerators. It is written in C/C++ and runs on all major platforms. A new version is released about twice a year, including source code and executables, with automated regression testing to reduce the chance of introducing errors. An extensive on-line manual and a large set of examples are available for download. Additional help is available from an on-line forum.

A minimal run of `elegant` requires two input files: the command file and the lattice file. The former consists of a series of namelist-like structures that set up the problem and execute it. The latter is similar to the MAD8 [10] lattice format and defines a lumped-element beamline. The lattice parser is driven by an element dictionary linked to data structures, making it quite simple to add new features to existing elements or to add entirely new elements.

Tracking is performed in 6D phase space with over 100 element types. Various methods are used, including symplectic integration, matrices (up to third order), and traditional numerical integration, allowing the user to choose methods that provide the preferred combination of accuracy and performance. For example, symplectic methods are required in many ring simulations, but are typically unnecessary and slow for linac simulations. `elegant` can output particle coordinates or moments at any specified location, or as a function of position. Particle coordinate output can be read back into `elegant` for further tracking.

In addition to tracking, `elegant` provides calculation of coupled and uncoupled lattice functions; lattice parameters such as tune, chromaticity, and resonance driving terms; radiation integrals; transfer matrices; trajectories; and equilibrium or non-equilibrium beam moments in the presence of synchrotron radiation.

Prior to tracking and other computations, `elegant` can add errors to the lattice, either from internally generated distributions or from data files, then apply correction schemes (e.g., orbit, tune, chromaticity). Error and correction settings may be saved to disk for later use in post-processing or new simulations.

`elegant` can perform optimization of tracking results,

* Work supported by the U.S. Department of Energy, Office of Science, under Contract No. DE-AC02-06CH11357.

† borland@aps.anl.gov

lattice functions, transfer matrices, radiation integrals, etc. Several optimization methods are available, including parallel methods such as particle swarm or parallel simplex. These can be very useful for difficult matching problems. One can also elect to use a serial optimization method while tracking in parallel. In addition to optimization, `elegant` can perform multi-dimensional parameter scans.

Computation of dynamic aperture, position-dependent momentum aperture [11], and frequency map analysis [12] are important techniques for understanding beam dynamics in rings. `elegant` provides parallel computation of all of these time-consuming calculations.

SDDS AND THE TOOL-BASED APPROACH TO MODELING

A special, long-standing feature of `elegant` is its use of self-describing files. The Self-Describing Data Sets (SDDS) protocol was developed for the APS control system and subsequently replaced an earlier self-describing format used by `elegant`. With SDDS files, data is accessed by name only, which is a remarkably simple but powerful idea. Programs can also confirm units and data type, leading to very robust interfaces between codes.

Types of data placed in SDDS files by `elegant` include: input and output phase space; Twiss parameters, beam moments, matrices, trajectories, etc. vs position along the beamline; input of wakes, impedances, and parasitic mode frequencies; input and output of errors and correction settings; input and output of parameters of beamline elements; input of data for time-dependent modulation of element parameters; and input of kicker waveforms.

The libraries that read and write SDDS files are open source and are distributed separately. Support is provided for C/C++, FORTRAN, Java, MATLAB, and Tcl [13]. There is also an MPI-based parallel library [9].

In addition to establishing a robust interface between physics codes, SDDS provides the possibility of sharing pre- and post-processing programs among simulations. This idea led to creation of the SDDS Toolkit [14, 15], a collection of generic programs for analysis, manipulation, and display of SDDS data. Because SDDS tools typically both read and write SDDS data, they can be used sequentially to perform complex, customized data processing.

For example, suppose one has phase-space data from multi-turn tracking of a series of particles with different initial amplitudes. One could perform an FFT (`sddsfft`) then display this as a function of initial amplitude (`sddscontour`), as shown in Fig. 1. One could determine the locations of the spectral peaks (`sddspeakfind`) and plot these vs initial amplitude (`sddsplot`). Computing the action J_x (`sddsprocess`) would allow fitting (`sddspfit`) the frequency vs action, giving the tune shifts with amplitude. This command sequence could be collected into a script for reuse in future simulations. Individual tools, e.g., `sddsfft`, `sddspeakfind`, etc., can of course be reused in many unrelated computations.

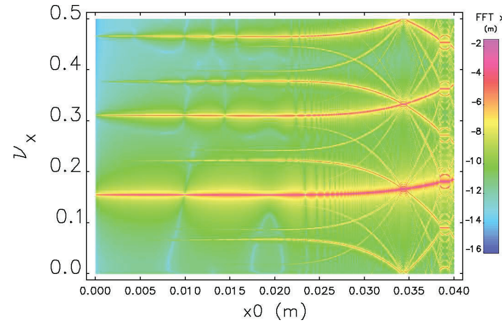

Figure 1: Illustration of using SDDS tools to display FFT of phase space tracking data.

The programs in the SDDS Toolkit have no accelerator- or `elegant`-specific features. There is also an `elegant`-specific toolkit [16] designed to supplement `elegant`, which provides calculation of x-ray brightness and flux, Touschek lifetime, intra-beam scattering, and potential well distortion, to name a few. This toolkit also provides conversion tools for non-SDDS-compliant programs, giving compatibility with SDDS postprocessing tools and, more importantly, allowing reasonably robust sequential use of multiple tracking codes. Among the codes for which such interfaces are provided are ASTRA [17], GENESIS [18], IMPACT-T [19], MARS [20], and TRACK [21]. Other APS-developed codes interface to SDDS directly, including the electron-gamma shower program `shower` [22] and the gun simulation program `spiffe` [23].

START-TO-END SIMULATION AND THE CSR MICROBUNCHING INSTABILITY

One benefit of interfacing many codes to SDDS is flexible, robust start-to-end (S2E) simulation, which has proved very valuable for S2E simulations of x-ray FELs [5, 24, 25]. (Note that S2E simulations were performed first for a visible-wavelength FEL using other codes [26].) A typical S2E simulation might begin by modeling the gun and the beginning of the linac with ASTRA. ASTRA's output is converted to SDDS using `astra2elegant`, allowing the remainder of the linac to be tracked with `elegant`. The final phase-space output might then be analyzed using `elegant2genesis`, followed by simulation of the FEL process using GENESIS. Since these tools are commandline-based, automation of the simulation and analysis is possible.

`elegant` is a popular choice for linac modeling because it includes acceleration, wakes, and CSR [7], all important factors in FEL driver linacs. A bunch of electrons with rms length σ_z will radiate coherently at wavelengths $\lambda \gtrsim 2\pi\sigma_z$. Since electrons travel in a curved path through the bending magnet, they fall behind the radiation they emit. Thus, the tail of the bunch bathes the head in radiation, modulating its energy. Since this is happening in a dispersive system, the emittance will grow. CSR propagating into drift spaces between the dipoles of a magnetic chicane will also have a potentially significant effect [27].

The original LCLS design [28] featured double-chicane bunch compressors at 250 MeV and 4.5 GeV. The two chicanes of each compressor were separated by optical elements that imposed a $-I$ transform, which was intended to provide cancellation of emittance growth due to CSR. This was based on modeling with Gaussian bunches, rather than phase space from an injector simulation.

CSR was added to `elegant` using a relatively simple and fast line-charge model [29, 27, 30], which permitted using a very large number of particles, high longitudinal resolution, and arbitrary particle distributions. Thus, when PARMELA [31] and `elegant` were used for S2E simulation, we found a much different picture [5, 6], as show in Fig. 2. The phase space is modulated with an $\sim 3\ \mu$m period, to a degree that would severly impact FEL output.

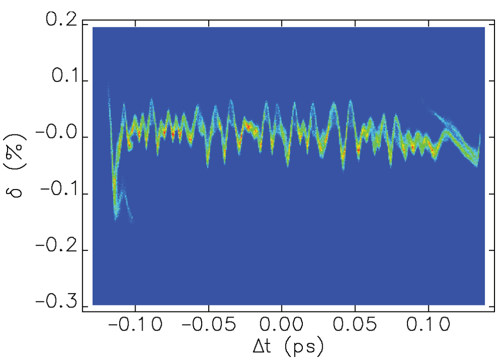

Figure 2: Final longitudinal phase space of the original LCLS design, showing the microbunching instability.

To explain the instability, we first note that any small longitudinal density clump will emit CSR, which causes acceleration of particles ahead of the clump and deceleration of particles behind the clump. Since this occurs in a dispersive system, the leading particles fall back while the trailing particles move forward, resulting in an enhancement of the density clump and a runaway condition. Hence, any nonuniformity (e.g., noise or ripples) in the initial longitudinal density can seed the instability. This instability is related to the "phase space fragmentation" seen earlier experimentally [32, 33, 34] and in simulation [35]. The essential difference is that fragmentation results from density spikes at a few locations in the bunch, whereas the microbunching instability can grow from noise or small modulations.

Following the discovery of this instabiilty, the LCLS design was revised to eliminate the double chicanes and add a superconducting wiggler to increase the slice energy spread, both of which helped reduce the instability. Shortly thereafter, the effect of longitudinal space charge (LSC) in the linac in amplifying this instability was postulated, along with use of a laser/undulator beam heater for improved suppression [36]. The impact of LSC and its mitigation were verified with `elegant` [37] for LCLS and other facilities.

In a related effort, PARMELA, `elegant`, and GENESIS were used for S2E jitter simulations of the LCLS [5]. These simulations illustrate the value of automated S2E, which is made easier by the use of SDDS. Several hundred sets of errors were added to both the PARMELA and `elegant` runs and kept track of using SDDS files. Since output from all three codes was in SDDS files, it was simple to perform correlation analysis between the errors and accelerator or FEL performance. This allowed determining which errors were the most harmful, leading to revision of some specifications to better meet requirements. These studies also showed that quadrupoles inside the chicane, although effective in reducing projected emittance growth, introduced transverse position jitter at the undulator entrance.

TOP-UP SAFETY TRACKING

Traditionally, storage ring light sources have operated in "decay mode," in which the beam is filled a few times a day and allowed to decay significantly between fills. This has several drawbacks, the most obvious being diminished integrated x-ray flux. In addition, heat load variations on x-ray optical elements result in less stable experimental conditions, as do intensity dependence of diagnostics and chamber temperatures. Finally, optimization of emittance, coupling, and bunch intensity are limited by the need to ensure reasonably long beam lifetime.

Clearly, maintaining nearly constant beam current would be highly beneficial. The process of doing this is called top-up, and was first performed during routine user operation at APS [38]. A significant concern for top-up operation is that injection occurs while shutters are open, bringing the dire possibility that injected electrons might escape down a user beamline.

Referring to Fig. 3, we see that if the first dipole were shorted, injected electrons entering the sector from the upstream insertion device (ID) straight would travel down the x-ray beamline. Of course, if nothing else happened, any stored beam would be lost on various apertures as the dipole field gradually dropped, which suggests that one can protect against an accident by insisting that top-up can only occur if beam remains stored [39]. Because the injected and stored beams have different trajectories, this may be insufficient if a partial dipole short ($F = B/B_0 < 1$) is accompanied by another change. For example, a strong corrector downstream of a partially shorted dipole might help maintain stored beam while injected beam escapes down the x-ray beamline. Thus, other changes can reduce the gap between the value of F_s that allows stored beam and the value F_i that allows injected beam to escape. When $F_s - F_i \leq 0$, the potential for an unsafe condition exists.

Determining this gap requires detailed tracking studies [40], which were first carried out using `elegant` in preparation for top-up operation at APS. The method involves backtracking hypothetical electrons that originate in the photon beamline. If backtracked electrons cannot enter the upstream ID straight section under conditions that allow stored beam, then no injected particles can reach the experimental hutch while beam is stored. In more detail, the method is to first choose a dipole to short and then perform

ISBN 978-3-95450-122-9

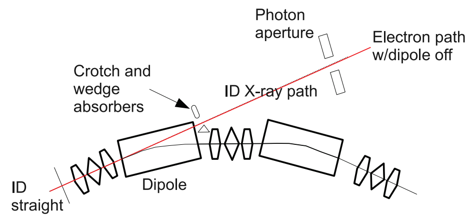

Figure 3: Diagram of a typical sector of a light source ring.

backtracking as a function of the degree of shorting. One also simulates the effect of additional failures or adjustments (e.g., steering correction with a downstream magnet) on the stored beam, allowing these to occur in a fashion that best compensates the effect on the orbit. This permits computation of the gap $F_s - F_i$ and determination of whether a problem exists.

For the APS, about 250 `elegant` runs are required for each of six unique beam line configurations. Each run involved 20 to 50 different conditions, giving a total of about 50,000 different physical situations. In spite of the complexity of the simulations, `elegant` required only minor modification, namely, addition of integration through field maps in order to properly model backtracked beam that is very far off-axis. Most of the complexity of the simulations was automated using scripts and the SDDS Toolkit.

Since APS first performed top-up safety tracking, several other facilities have performed similarly thorough studies (e.g., [41, 42, 43]) using several different codes.

DIRECT OPTIMIZATION OF STORAGE RING BEAM DYNAMICS

Another difficulty facing storage ring designers is nonlinear dynamics. This is particularly challenging for low-emittance electron rings, where strong focusing creates large negative natural chromaticity and small dispersion, leading to very strong sextupole magnets. Progress toward lower emittance is primarily impeded by the difficulty of ensuring sufficient dynamic acceptance (DA) for efficient injection and sufficient local momentum acceptance (LMA) [11] to deliver adequate Touschek lifetime.

Although a number of methods have been employed for this problem, solutions must always be evaluated by particle tracking. With modern computational resources, it is possible to instead *directly* optimize the results of tracking. Perhaps the first published work [44] used `elegant` and parallel simplex optimization. Several figures of merit were explored, the best being direct minimization of tune spread computed by tracking a group of particles.

Further direct optimization efforts [45] used `elegant` with two different parallel search techniques to directly optimize dynamic and momentum acceptance for the first time. The first technique was a simple parallel grid search and involved finding the conditions that maximized capture of an ensemble of particles that filled the desired transverse and longitudinal phase space. The method was effective but, as a grid search, limited in the number of variables.

The second technique was a combined-objective genetic algorithm (COGA), inspired by multi-objective genetic algorithm (MOGA) work [46, 47, 48]. Here, the LMA was computed and its minimum value was maximized as a proxy for Touschek lifetime. The DA was characterized by its area, computed in a robust fashion that avoids pathological DA shapes. This method was used to optimize potential upgrade lattices for the APS, involving insertion of up to eight symmetrically placed long straight sections (LSSs). In addition to allowing the tunes to vary, 21 independent sextupole families were used. One interesting discovery was that, even though the linear optics was reflection symmetric, allowing the sextupoles around the LSS to break reflection symmetry was highly beneficial. A typical sextupole configuration is shown in Fig. 4.

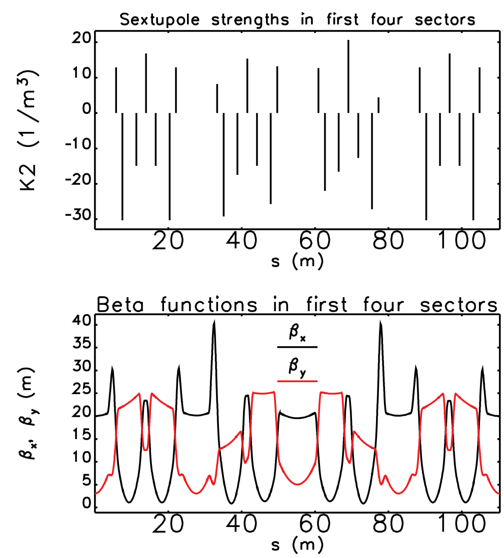

Figure 4: COGA-optmized sextupole distribution and beta functions for a section of the 8-LSS APS lattice.

Both techniques resulted in significantly improved configurations for APS operations, e.g., a 25% improvement in beam lifetime and the largest dynamic aperture measured at the time. The COGA technique was subsequently refined [49, 50] into a MOGA technique that included direct optimization of Touschek lifetime computed from the LMA. (Independent tracking-based optimization efforts by other groups [51, 52] appeared at about the same time.) The DA and LMA were computed using parallel methods in `elegant`, which permitted use of massively parallel resources on the IBM BlueGene architecture. These changes were driven by the increasingly difficult requirements of the APS upgrade, including the desire to have the LSSs placed in non-symmetric locations, the need to incorporate the short-pulse x-ray (SPX) system [53] with its highly modified sextupole configuration [54], and the need to provide two-fold reduced horizontal beam size in one straight section. Most recently, the optimization has been expanded

ISBN 978-3-95450-122-9

to include the SPX crab cavities in the DA and LMA tracking, as well as minimization of emittance dilution from the cavities.

CONCLUSIONS

The program `elegant` has been in use and development for nearly three decades. In that time, it has grown from a linac-only matrix tracking code to a flexible, capable tool for design and simulation of single- and multi-pass accelerators. One special feature is the thorough use of self-describing files, giving access to the SDDS Toolkit for pre- and post-processing, as well as providing robust integration with other codes. Several noteworthy applications were reviewed, including start-to-end simulation, discovery of the CSR-driving microbunching instability in bunch compressors, top-up safety tracking, and direct optimization of storage ring nonlinear dynamics. Application and development of `elegant` and related codes continues.

ACKNOWLEDGMENTS

The author wishes to express his appreciation to the many people who have contributed to `elegant`. Space, and the fallibility of memory, prevents acknowledging all the many contributions. The main authors of `elegant` are M. Borland, W. Guo, V. Sajaev, H. Shang, C.-X. Wang, Y. Wang, Y. Wu, and A. Xiao. Y. Wang led the parallelization effort with input from M. Borland and H. Shang. Contributors to the `elegant` toolkit include M. Borland, Y.-C. Chae, R. Dejus, X. Dong, L. Emery, H. Shang, R. Soliday, and A. Xiao. Contributors to SDDS include M. Borland, L. Emery, H. Shang, and R. Soliday. R. Soliday developed multi-language SDDS libraries, performed multi-platform ports, and provides on-going multi-platform support. Many users have helped to improve the code through their suggestions and bug reports. Ideally, this paper would have included a survey of the many interesting applications that have made use of `elegant`, but that would have taken far more space than is available. Special mention goes to P. Emma, who was the first person outside of APS to use `elegant`. Special mention also goes to H. Wiedemann and J. Galayda, for their advice, patience, and support.

REFERENCES

[1] M. Borland, LS-287, Advanced Photon Source (2000).
[2] H. Wiedemann et al., PAC 1991, 2688 (1991).
[3] S. V. Milton et al., Science 292, 2037 (2001).
[4] P. Emma et al., Nature Photonics 4, 641 (2010).
[5] M. Borland et al., PAC 2001, 2707 (2001).
[6] M. Borland, Y. Chae et al., NIM A 483(12), 268 (2002).
[7] B. E. Carlsten and T. O. Raubenheimer, Phys. Rev. E 51, 1453 (1995).
[8] Y. Wang, M. Borland et al., ICAP 2009, 355 (2009).
[9] H. Shang, Y. Wang et al., ICAP 2009, 347 (2009).
[10] H. Grote and F. C. Iselin, CERN/SL/90-13(AP), CERN (1991).
[11] M. Belgrounne et al., PAC 2003, 896 (2003).
[12] J. Laskar and D. Robin, Part. Accel. 54, 183 (1996).
[13] R. Soliday, ICALEPS 2001, 545 (2001).
[14] M. Borland, PAC 1995, 2184 (1996).
[15] R. Soliday, M. Borland et al., PAC 2003, 3473 (2003).
[16] M. Borland, L. Emery et al., PAC 2003, 3461–3463 (2003).
[17] K. Flöttmann, www.desy.de/ mpyflo/Astra_dokumentation.
[18] S. Reiche, NIM A 429 (1999).
[19] J. Qiang, R. D. Ryne et al., J. Comput. Phys. 163, 434 (2000).
[20] N. V. Mokhov, Fermilab-FN-628, FNAL (1995).
[21] V. N. Aseev et al., PAC 2005, 2053 (2005).
[22] L. Emery, PAC 1995, 2039 (1996).
[23] M. Borland, www.aps.anl.gov/asd/oag/oagSoftware.shtml.
[24] T. Limberg and P. Piot, PAC 2001, 2722 (2001).
[25] S. Reiche et al., PAC 2001, 2751 (2001).
[26] C. Parazzoli and B. E. C. Koltenbah, NIM A 449 (2000).
[27] M. Borland, PRSTAB 4, 070701 (2001).
[28] J. Arthur et al., SLAC-R-521, SLAC (1998).
[29] E. L. Saldin, E. A. Schneidmiller et al., NIM A 398, 373 (1997).
[30] G. V. Stupakov and P. Emma, LCLS-01-12, SLAC (2001).
[31] J. Billen, LA-UR-96-1835, LANL (1996).
[32] P. Piot, C. L. Bohn et al., EPAC 2000, 1546 (2000).
[33] R. Li, EPAC 2000, 1312 (2000).
[34] T. Limberg, P. Piot et al., NIM A 475, 353 (2001).
[35] M. Borland, PRSTAB 4, 074201 (2001).
[36] E. L. Saldin, TESLA-FEL-2003-02 (2003).
[37] Z. Huang, M. Borland et al., PRSTAB 7, 074401 (2004).
[38] L. Emery, PAC 2001, 2599 (2001).
[39] L. Emery and M. Borland, PAC 1999, 2939 (1999).
[40] M. Borland and L. Emery, PAC 1999, 2319 (1999).
[41] H. Nishimura, B. Bailey et al., NIM A 608(1), 2 (2009).
[42] I. P. S. Martin et al., EPAC 2008, 2085 (2008).
[43] Y. Li, L. Yang et al., PRSTAB 14, 033501 (2011).
[44] H. Shang and M. Borland, PAC 2005, 4232 (2005).
[45] M. Borland, V. Sajaev et al., PAC 2009, 3850 (2009).
[46] K. Deb, A. Pratap et al., IEEE TEC 6, 182 (2002).
[47] I. Bazarov and C. Sinclair, PRSTAB 8, 034202 (2005).
[48] L. Emery, PAC 2005, 2962 (2005).
[49] M. Borland, "Parallel tracking-based optimization of dynamic aperture and lifetime with application to the aps upgrade," FLS 2010, www-conf.slac.stanford.edu/icfa2010.
[50] M. Borland, V. Sajaev et al., ANL/APS/LS-319, APS (2010).
[51] L. Yang, "Multi-objective optimization of dynamic aperture," FLS 2010, www-conf.slac.stanford.edu/icfa2010.
[52] C. Steier and W. Wan, IPAC 2010, 4746 (2010).
[53] A. Nassiri et al., IPAC 2012, 2292 (2012).
[54] M. Borland and V. Sajaev, PAC 2005, 3886 (2005).

ISBN 978-3-95450-122-9