

**EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENE POUR LA RECHERCHE NUCLEAIRE**

CERN – PS DIVISION

CERN/PS 2000-070 (OP)

**BEAMOPTICS: A SYMBOLIC PLATFORM FOR MODELING AND
THE SOLUTION OF BEAM OPTICS SYSTEM**

Y. Chao*

Abstract

BeamOptics [1] is a Mathematica-based computing platform devoted to the following objectives;

- Structured representation and manipulation of particle beam optics systems with symbolic capabilities
- Analytical and numerical modelling of beam optics system behaviours
- Solution to specific beam optical or general accelerator system problems, in algebraic form in certain cases, through customized algorithms.

Taking advantage of and conforming to the highly formal and self-contained structure of Mathematica, BeamOptics provides a unique platform for developing accelerator design and analysis programs. The feature of symbolic computation and the ability to manipulate the beam optics system at the programming language level enable the user to solve or optimize his system with considerably more efficiency, rigour and insight than can be easily achieved with passive modelling or numerical simulation methods. BeamOptics is developed with continuous evolution in mind. New features and algorithms from diverse sources can be incorporated without major modification, due to its formal and generic structure. In this report, a survey is given of the basic structure and methodology of BeamOptics, as well as a demonstration of some of its more specialized applications, and possible direction of evolution.

**Thomas Jefferson National Accelerator Facility, Newport News, Virginia., USA.*

ICAP 2000 Conference, 10-14 September 2000, Darmstadt, Germany

Geneva, Switzerland
2 November 2000

***BeamOptics*: A Symbolic Platform for Modeling and the Solution of Beam Optics System**

Y. Chao
CERN, Geneva, Switzerland

Abstract

BeamOptics [1] is a *Mathematica*-based computing platform devoted to the following objectives;

- Structured representation and manipulation of particle beam optics systems with symbolic capabilities,
- Analytical and numerical modeling of beam optics system behaviors,
- Solution to specific beam optical or general accelerator system problems, in algebraic form in certain cases, through customized algorithms.

Taking advantage of and conforming to the highly formal and self-contained structure of *Mathematica*, *BeamOptics* provides a unique platform for developing accelerator design and analysis programs. The feature of symbolic computation and the ability to manipulate the beam optics system at the programming language level enable the user to solve or optimize his system with considerably more efficiency, rigour and insight than can be easily achieved with passive modeling or numerical simulation methods. *BeamOptics* is developed with continuous evolution in mind. New features and algorithms from diverse sources can be incorporated without major modification, due to its formal and generic structure.

In this report, a survey is given of the basic structure and methodology of *BeamOptics*, as well as a demonstration of some of its more specialized applications, and possible direction of evolution.

I. INTRODUCTION

A symbolic computing platform for beam optics can promise a degree of efficiency, flexibility, rigour and insight not easily matched by passive numerical modeling or simulation. Such a platform has become feasible with the arrival of modern mathematical software tools, notably *Mathematica* [1], featuring sophisticated symbolic and numerical capabilities and programming syntax optimized for concise scientific computing. *BeamOptics* was developed in this spirit. Beam optics and accelerator physics models are constructed and analyzed in *BeamOptics* using the syntactic structure and functionality of *Mathematica*, most notably the features of symbolic computation and list-directed operation, and the ability to manipulate an optical model at the programming language level.

In this report we start with the difficulties encountered in following the conventional procedure for solving beam optics and accelerator physics problems, which, although time-tested, is ripe for improvement given the availability of advanced software tools. The design principles of *BeamOptics* are then introduced, the emphasis being on addressing these difficulties. This is followed by a delineation of the structure and existing functionality of *BeamOptics*, with examples demonstrating typical model representation, manipulation and analysis using *BeamOptics*, showing where it is most advantageous. This report is concluded by outlining current and planned activities on expanding the scope and community participation, with emphasis on the open-system nature of *BeamOptics*.

II. MOTIVATION FOR *BeamOptics*

II.1. Conventional approach to solving optics problems

BeamOptics was developed to provide an alternative paradigm for solving optics-related design and optimization problems. Traditionally such problems, be it the design of a matching section or minimization of radiation-induced emittance growth, are carried out in these typical steps:

Algebraic Manipulation: Algebraic manipulation is done by hand based on a known or derived mathematical model, leading to a set of initial conditions. When the algebra involved becomes too complicated, either a simplified system has to be substituted or a special-purpose computer program created to handle the numerical aspect of the problem. Flexibility of such programs is usually limited.

Numerical Modeling: Initial conditions are used in a numerical modeling program, examples of which are the multitude of beam optics and particle tracking packages, which incorporate finer details not accountable by simple algebraic methods. These packages may have the ability to “scan” the parameter space for insight on the system, usually done by ad hoc meta-programs, at a penalty of time.

Iteration: Numerical results are examined and compared to the design goal. The simple algebraic model is invoked again for interpretation of outcome and improvement on parameters. The entire process is iterated until convergence is achieved onto the goal, within tolerance.

II.2. Difficulties encountered in the conventional approach

Some counter-productive aspects of the process described above are easily recognized:

Low degree of integration of methods: The algebraic manipulation is separate from the numerical modeling, making parameter exploration awkward, inefficient and prone to human error.

Limitation on parameter exploration: Parameters are usually explored through minimization of multi-variable functions, the result of which is not necessarily the absolute optimum. Convergence towards optimal solutions may require a number of iterations, exploring different parts of the parameter space.

Limited flexibility and adaptability: Tailoring standard packages to a user’s specific needs at source code level is difficult. New features, such as a physical process not conceived at inception, can be difficult to retrofit without altering the syntax or even structure of the package.

Bulky input format: Numerical packages usually depend on element-directed input format, leading to bulky files or procedures. Creating and editing input files can be time-consuming and error-prone.

Lack of physical insight: It is difficult to extract insight on qualitative behaviors based

on numerical simulations, as opposed to insight derived from analytical expressions.

Need for multiple computing platforms: For large design projects, there may be a need to employ several different packages with incompatible data formats, running on different platforms.

Given the limitation on available software tools in the past, this mode of solving beam optics problems has been the best approach available. However, the possibility of addressing most of these difficulties by developing a package that is *Mathematica*-based, yet self-contained, was noticed and seriously explored. *BeamOptics* is the outcome of this effort. At the current phase of development, *BeamOptics* already provides a platform for performing optics and accelerator physics related design and analysis in a manner free of many of the above problems, as will be shown in the next section. The important point is, however, that given its flexible structure and ease to evolve, with *Mathematica* or independent thereof, it is much better poised to address challenges, whether existing or anticipated, than the traditional paradigm.

III. DESIGN PRINCIPLES FOR *BeamOptics*

BeamOptics offers a self-contained package, less prone to difficulties encountered in the traditional paradigm for solving beam optics problems. This can be attributed to the following design principles.

III.1. Self-contained infrastructure

BeamOptics allows the user to carry out the complete analysis of a problem without having to venture beyond the provided framework. It is not realistic for *BeamOptics* to include all conceivable high-level beam optics algorithms at inception. Instead it provides a standard infrastructure versatile enough for the user to build these algorithms on, without having to supplement it with the user's own overhead code or to channel it to another package on a different platform. In addition to specially-developed beam optical components, the entire algebraic and numerical infrastructure of *Mathematica* is available in *BeamOptics*. As a result, the user is free from the need of ad hoc overhead codes and can easily extend the scope of an ongoing project while staying within the same integrated platform. An additional advantage of building an optics analysis on the *BeamOptics* platform is that once a system is realized through its carefully planned framework, with internal consistency and exception-handling already worked out, the resulting representation of the system can be used for other analyses. This greatly enhances the option of code recycling.

III.2. Flexibility at low level

BeamOptics allows the user to manipulate the system being studied at a very low level, thus affording flexibility not possible with fixed, special-purpose packages. By casting the basic building blocks of *BeamOptics* as *Mathematica* entities, the same repertoire of *Mathematica* methods for object manipulation is available for *BeamOptics* objects. The representation of an optical system, instead of being a frozen set of input/output data beyond the user's control, can be interactively manipulated. Furthermore, these operations can be systematically programmed at the *Mathematica* language level, making possible the integration of multi-purpose design projects under a single scheme.

Connecting *BeamOptics* to the input and output ends of an independent *Mathematica* analysis can also be done seamlessly, taking advantage of the power of the latter at solving specialized physical problems.

III.3. Extensive use of list-directed structure and operations

The powerful feature of list-directed operations in *Mathematica* is maximally exploited in *BeamOptics*. Almost no optics or accelerator physics related problem can entirely avoid evaluation on a list structure when it comes to application to real machines. Many accelerator problems are indeed characterized mainly by effects accumulated over long lists of elements. By exploiting the powerful list-directed operations of *Mathematica* and generalizing the syntax as much as possible to be list-compatible as part of the design philosophy, a wide range of accelerator-related mathematical and database operations can be realized with compact, efficient and less-error-prone commands in *BeamOptics*.

III.4. Ability to perform algebraic computation

Algebraic (symbolic) computation, a natural alternative mode of computation to many numerical functions of *Mathematica*, is adopted in the same spirit by *BeamOptics*. At the level of generic building blocks, no distinction is made between symbolic, numerical, or mixed representations. Conversion from symbolic to numerical representation can happen at any stage. This offers the user freedom in attaching emphasis to maintaining rigour and gaining insight on the one hand, or solving problems efficiently and practically on the other, or anything in between. This again is accomplished without resorting to disjoint packages on different software platforms. When it is practical, the ability to carry out analysis in algebraic form in the complete or partial parameter space provides exact solutions without resorting to numerical techniques.

It should be noted that a well-defined approach to obtain solutions for optical modules [3] has been developed using *BeamOptics* when the model is not algebraically solvable. This typically involves the following steps:

- formulate the problem with an equal number of variables and constraints,
- obtain existence conditions for solutions,
- obtain exact solutions in the thin lens model,
- perform numerical optimization in the neighborhood of each thin lens solution to obtain corresponding thick lens solutions.

Notice that this approach explores all the theoretical local minima of the system, thus it does not miss the global minimum, as opposed to usual numerical minimization methods that strongly depend on the initial condition.

Another new *BeamOptics* function [4] provides a deterministic algorithm for converting a thin lens system of a specific type to the equivalent thick lens system.

An additional advantage of an algebraically described optical system is that all lattice and beam parameters can be readily calculated along the continuous longitudinal dimension. With a piece-wise numerically represented system this has to be done by modifying the element structure of the beamline.

III.5. Open system and constant evolution

An important design feature of *BeamOptics* is that, as a result of the self-contained infrastructure and flexibility at low level, it is an open system ready for straightforward and seamless incorporation of new algorithms. Development and incorporation of new modules require only basic knowledge in the *Mathematica* language and *BeamOptics* syntax. The current phase of scope extension involves contribution from diverse sources on many different subjects.

IV. STRUCTURE AND EXISTING FUNCTIONALITY OF *BeamOptics*

The *BeamOptics* package is made up of *Mathematica* functions organized by topics in beam optics and accelerator physics. The backbone that supports these functions and makes them mutually compatible is the scheme for representing optical elements and beam lines in terms of transfer matrices. This can be done algebraically, numerically, or as a combination of both. The optical transfer matrices representing individual elements or sections of the beam line are in full view of the user and open for manipulation. Beam lines, known as **Channels** in *BeamOptics*, take on the same structure as a **List** in *Mathematica*, allowing easy single-command manipulations, such as query by index, simultaneous operation on all elements, line inversion, splicing etc.. This also opens up the possibility for the user, taking advantage of the flexible and versatile **List** structure in *Mathematica*, to create new functions with more physical content that can be called as inline commands on a Channel. This is in fact the standard mode for incorporating new functions.

Existing functions can be grouped into the following categories.

IV.1. Basic tools for system building.

At the ground level, *BeamOptics* provides the user with tools for defining optical elements, geometry, and building Channels from them as described above.

IV.2. Optical modules.

These consist of the next level building blocks in *BeamOptics* for function-oriented modules. An extensive set of modules is available with more being developed. The possibility of algebraic computation makes these useful tools for system design and optimization.

IV.3. Standard analytical functions of an optical system.

Baseline geometrical properties, transfer matrices, envelope functions, etc. are standard built-in functions. At present only formulation based on decoupled transverse optics is implemented and fully tested. Extension into 4-dimensional coupled transverse optics is planned in a future upgrade.

IV.4. Perturbative calculations

Functions are developed to track perturbation around the central orbit due to linear elements, including dispersive effects, compounded effects in tracking with symbolic option, and response matrices describing trajectory and beam envelope responses to control elements for a given Channel.

IV.5. Accelerator physics related phenomena and processes

This includes non-dissipative effects such as synchrotron radiation, intra-beam scattering, etc. An extended set of such functions is planned.

IV.6. Visualization tools

The powerful graphical capability of *Mathematica* is naturally available to *BeamOptics* as an organic component. Functions directing intermediate or final output to the graphics functions at the programming language level allow the user to visualize the calculation interactively without ad hoc add-on graphics software. The *Mathematica*-based package *Geometrica* can be loaded for rendering geometrical layout of beamlines and particle trajectories, with realistic element dimensions. In most cases, due to analytically-represented lattice and beam parameter functions, plots can be rendered with respect to a continuous longitudinal coordinate, as opposed to piece-wise plots typically generated by purely numerical packages.

IV.7. Graphical user interface

Various graphical user interface facilities are available, such as a clickable palette for generating syntactic templates for programmers of *BeamOptics*.

V. EXAMPLES

Some examples illustrate where *BeamOptics* can be advantageous in accelerator design and analysis.

V.1. Implementing FFAG Design in *BeamOptics* and Study of Chromatic Behavior of Closed Orbits

Machine description

The machine is an FFAG ring [5] with 8 periods. A period consists of two combined function dipoles separated by a drift space. One magnet is focusing and the other is defocusing in the horizontal plane. The two angles θ^F and θ^D are of opposite signs.

$$\{\text{thetaF}, \text{thetaD}\} = \text{Pi} / 180 \{33.75, -11.25\}$$

The reference dipole is a sector magnet whose length is the length of the central orbit. The radii r^F and r^D are

$$\{\text{rF}, \text{rD}\} = \{1.3, -2.535\}$$

The focusing strengths k^F and k^D are

$$\{\text{kF}, \text{kD}\} = \{0.348548, -0.0305544\}$$

Their edges are such that the property of similar orbits applies to that machine.

$$\{\text{upF}, \text{downF}, \text{upD}, \text{downD}\} = \\ \{0, \text{thetaF} - 0.15, 0.15 - \text{thetaF}, \text{Pi} / 8 - 0.25\}$$

For graphics representation, it is useful to include color options. Here F and D magnets are colored in Red and Blue respectively.

$$\{\text{colF}, \text{colD}\} = \{\text{RGBColor}[1, 0, 0], \text{RGBColor}[0, 0, 1]\}$$

The magnets b^F and b^D are defined by the function `Bend`. The arguments are their lengths, deflection angles, and focusing strengths, options to define the angle of the end faces with respect to the reference orbit, and graphical properties.

```

{bF, bD} =
  Bend[{rF, rD} {thetaF, thetaD},
    {thetaF, thetaD},
    {kF, kD},
    {EdgeUp → upF, EdgeUp → upD},
    {EdgeDown → downF, EdgeDown → downD},
    {OpticsStyle → {colF}, OpticsStyle → {colD}}]

```

Several properties of list processing have been used here to make the statement concise. The lists play the same role as single components. The first argument, which is the product of the list of radii and of the list of angles, computes to the list of lengths automatically. Then the attribute `Listable` attached to `Bend` in `BeamOptics` splits the single `Bend` function into a list of two `Bend` functions according to the rule

$$\mathbf{Bend}[\{a,b\},\{c,d\},\dots]=\{\mathbf{Bend}[a,c,\dots],\mathbf{Bend}[b,d,\dots]\}$$

A drift space is defined by the function `SS` (for Straight Section) and has the length as the unique argument:

$$\mathbf{s} = \mathbf{SS}[\mathbf{.75}]$$

The topology of a beam line is characterized by the function `Channel`. The half period of the FFAG machine is

$$\mathbf{ch} = \mathbf{Channel}[\mathbf{bF}, \mathbf{bD}, \mathbf{s}]$$

A full super-period results from the concatenation of `ch` with its mirror image.

$$\mathbf{period} = \mathbf{Mirror}[\mathbf{ch}]$$

The complete ring is made of eight periods

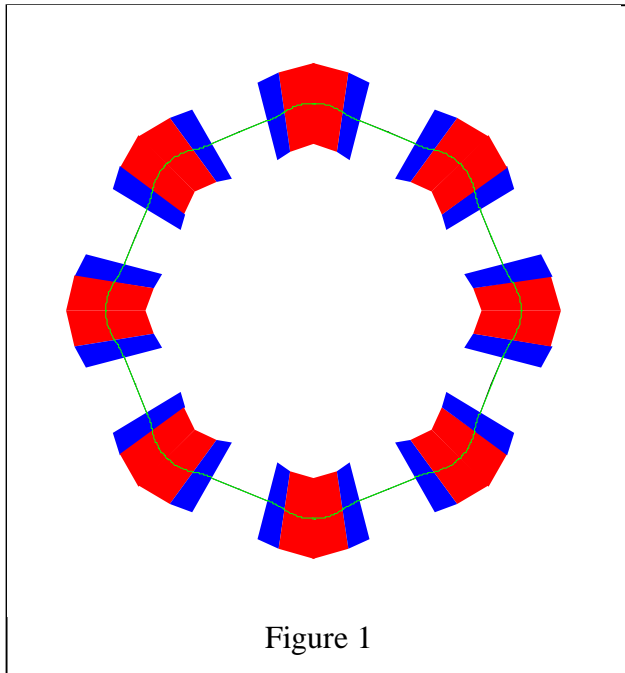
$$\mathbf{ring} = \mathbf{Repeat}[\mathbf{period}, \mathbf{7}]$$

It can be drawn using the function `Draw`.

Graphical representation

The ring can be drawn using `Draw`. The magnets are then represented according to the `OpticsStyle` options and the reference orbit is superimposed, as shown in Figure 1.

$$\mathbf{g1} = \mathbf{Draw}[\mathbf{ring}] ;$$

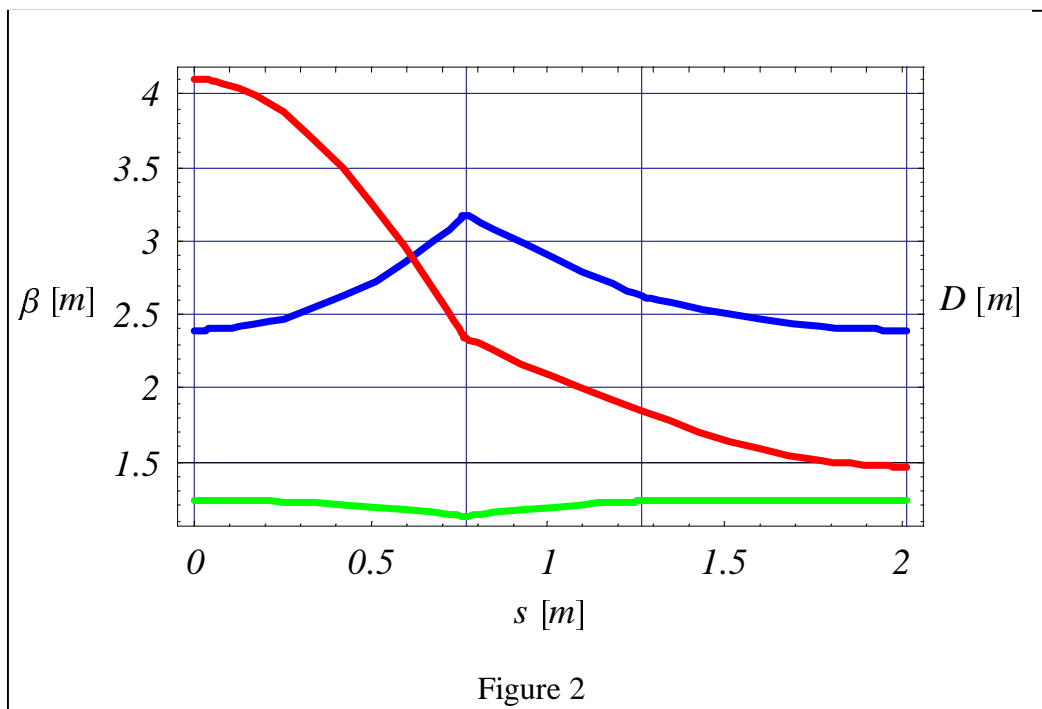


Characteristic functions

The β functions and the orbit dispersion can be calculated using the function `HalfPeriod` since the period of the machine is symmetric.

hp = HalfPeriod[ch]

```
{OpticsPlot → (- Graphics -),
 SigmaEnd → {Sigma [4.10126 , 0] , Sigma [2.39644 , 0] } ,
 Tunes → {0.14942 , 0.122071 } ,
 DVectorEnd → DVector [1.24016 , 0] , Path → 0.474349 }
```



The output is a list of replacement rules whose left hand sides are the names of the variables and the right hand sides the value of the variable. The graphical output of the lattice functions is shown in Figure 2. They can be used, for instance, to show the similarity of the orbits of an FFAG machine (Figure 3).

```

dv = DVectorEnd /. hp;
g2 = AbsoluteDPlot[ring, dv, {-.5, .5},
  PlotStyle -> RGBColor[0, 1, 0]];

```

It has to be noted that, in contrast with other optics programs, the orbit dispersion can be drawn about the *curved* reference orbit. This is useful in the present case and for isochronous machines in which the variations of the path length with momentum cannot be understood if the orbit dispersion is plotted about a straight reference orbit.

The following command combines the machine element layout with the closed orbits at different momentum offsets, shown in Figure 4:

```
show[g1, g2] ;
```

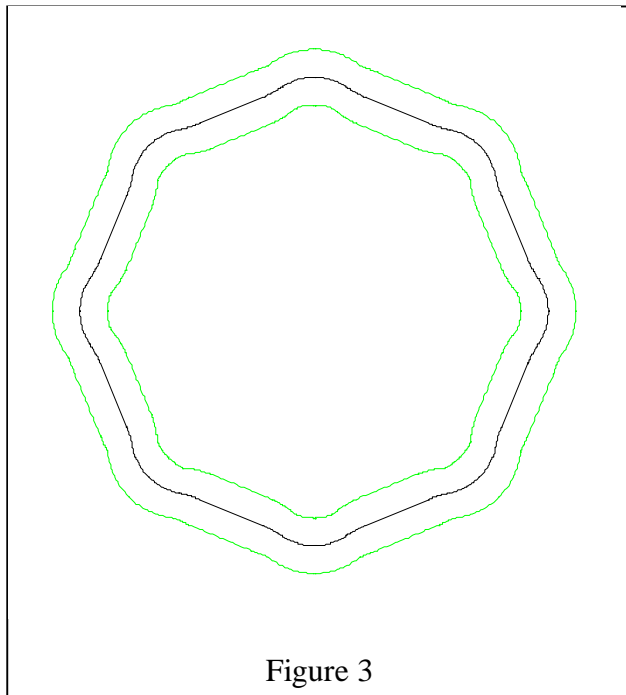


Figure 3

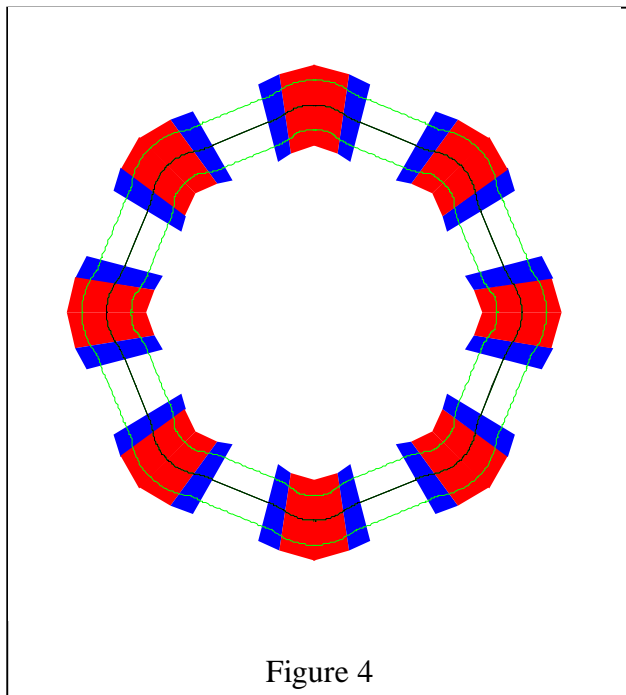


Figure 4

V.2. Analytical Study of the Matching Triplet

At the present time the FFAG triplet is being analyzed using *BeamOptics*. In the following we show a similar application of *BeamOptics* to a pure quadrupole triplet, which also helps demonstrate how a *BeamOptics* function is defined and used.

Symmetric matching triplet

A symmetric triplet consists of 3 quadrupoles with alternating focal lengths f_1, f_2 and f_1 , interleaved with inner drift lengths d and outer drift lengths l , and it is reflectively symmetric. In the thin lens approximation, a symmetric triplet has enough variables to match an arbitrary incoming Twiss parameter pair (β, α) to an arbitrary outgoing Twiss parameter pair. Tracking from input to output provides a system of four non-linear equations which, after successive elimination, gives a cubic equation for one variable [6]:

$$z^3 - \frac{2d}{c} z^2 + z + \frac{d^2 - c^2}{c(bc - ad)} = 0$$

$$a = \frac{t_{h,11} + t_{v,11} - 2}{4}$$

$$b = \frac{t_{h,11} - t_{v,11}}{4}$$

$$c = \frac{t_{h,21} + t_{v,21}}{4}$$

$$d = \frac{t_{h,21} - t_{v,21}}{4}$$

where $t_{h,mn}$ and $t_{v,mn}$ are the elements of the horizontal and vertical transfer matrices respectively. At most three real solutions can be found for z . The parameters f_1, f_2, l and d are functions of the variable z and the parameters a, b, c, d . A special solution occurs when two crossovers or a crossover and a waist are to be matched. In this case the focal lengths are equal in absolute value

$$f_1 = -f_2 = f$$

and the parameters of the triplet take the simple form:

$$f = \frac{\alpha_1 + \alpha_2}{\gamma_1 - \gamma_2}, \quad d = \left| f \right| \sqrt{1 + \frac{|\alpha_1 + \alpha_2|}{\alpha_0}}, \quad l = \frac{\alpha_0}{|\alpha_1 + \alpha_2|} d$$

where

$$\alpha_0 = \sqrt{\beta_1 \gamma_2 + \beta_2 \gamma_1 - 2(1 - \alpha_1 \alpha_2)}$$

with the other variables being Twiss parameters in the horizontal plane.

This result is implemented as a *BeamOptics* function with the syntax

MatchingTriplet $[\sigma_{1h}, \sigma_{1v}, \sigma_{2h}, \sigma_{2v}, \mathbf{opts}]$

where the σ 's represent the (β, α) pair in both planes at entry and exit of the triplet, and **opts** refers to graphics options.

Using the Optimum Optical Systems method [3], it is possible to build a triplet made of finite length quadrupoles. Each solution in the thin lens model is one initial condition for a numerical minimization convergent in the neighborhood of the thin lens solution. In the case of thick lenses, the syntax is the following:

MatchingTriplet $[\sigma_{1h}, \sigma_{1v}, \sigma_{2h}, \sigma_{2v}, \mathbf{l1}, \mathbf{l2}, \mathbf{opts}]$

with **l1** and **l2** being the finite lengths of the 2 types of quadrupoles. In this case **opts** as options can include parameters for the numerical minimization, such as maximum number of iterations.

Application of the function MatchingTriplet in the thin lens solution

One starts with thin lenses. For the following matching conditions imposed by Twiss parameters, only two real solutions exist, and the program returns both solutions numerically in the form of a 2 element-list.

MatchingTriplet $[\mathbf{Sigma}[5, 8], \mathbf{Sigma}[5, -8], \mathbf{Sigma}[2, 0], \mathbf{Sigma}[2, 0]]$

```
{OpticsPlot → {- Graphics -, - Graphics -},
 BeamLine → {Channel [SS [0.0946461 ],
                Q[-0.239198 ], SS [0.302262 ], Q[0.310575 ],
                SS [0.302262 ], Q[-0.239198 ], SS [0.0946461 ]],
 Channel [SS [0.658155 ], Q[-0.64 ], SS [1.02281 ],
                Q[0.64 ], SS [1.02281 ], Q[-0.64 ], SS [0.658155 ]]]}}
```

The plots of β_x and β_y for these 2 solutions are rendered as part of the output, as shown in Figures 5 and 6.

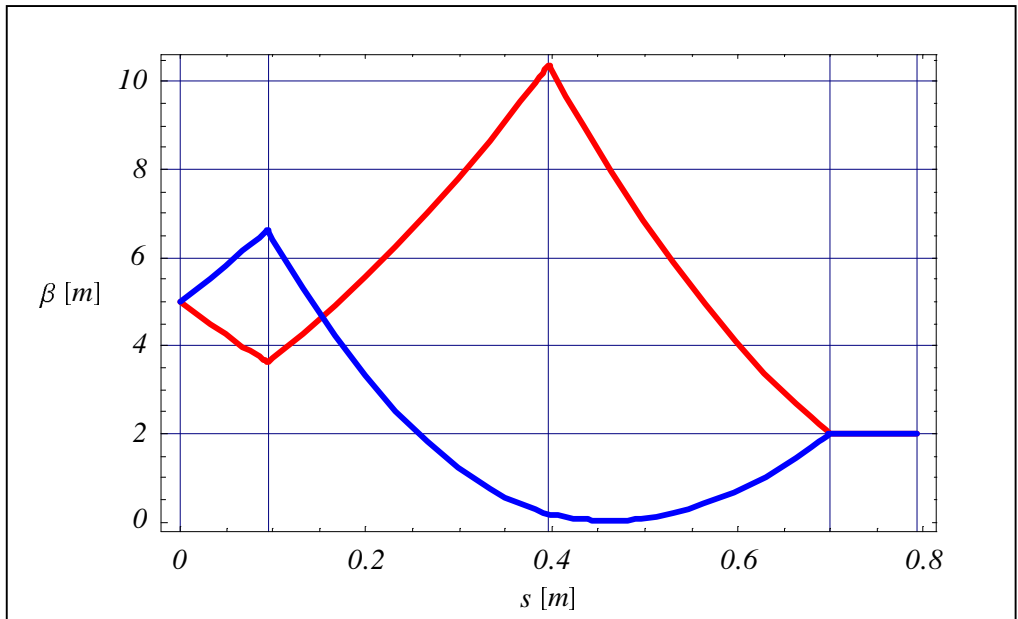


Figure 5

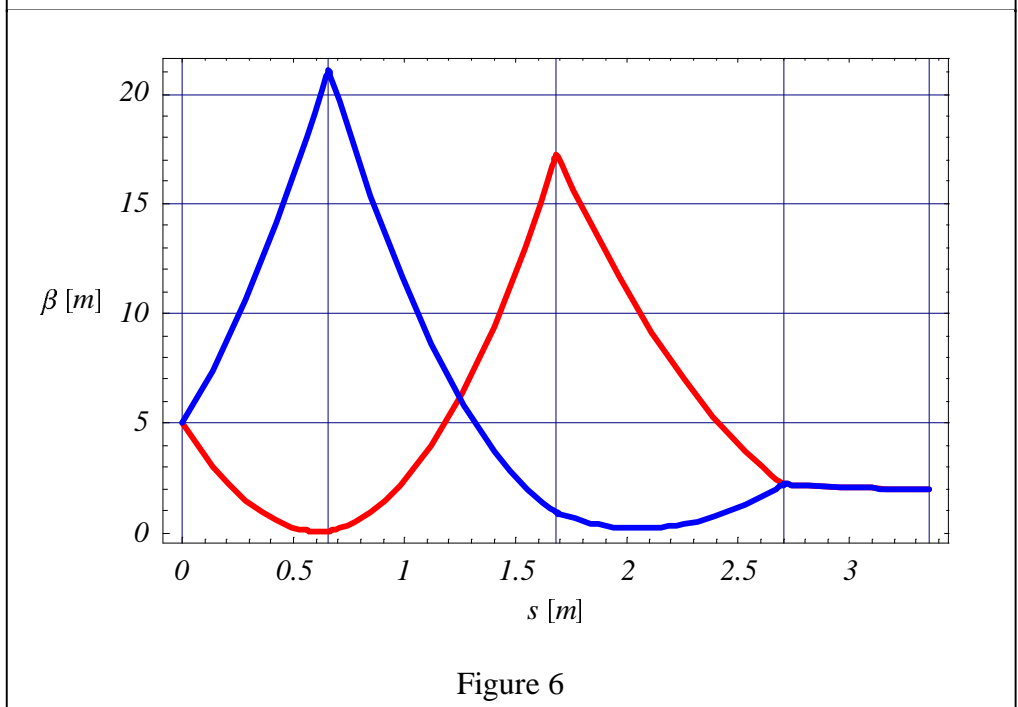


Figure 6

Application of the function MatchingTriplet in the thick lens solution

When switching to thick lenses, the quadrupole lengths have to be smaller than the spacing between the thin lenses. By simply adding these lengths in the input to the function *MatchingTriplet*, a perfect crossover is reached with an output sigma slightly different from the requested one.

Very thin quadrupoles are compatible with both of the previous solutions:

```
MatchingTriplet[Sigma[5, 8], Sigma[5, -8], Sigma[2, 0],
Sigma[2, 0], .05, .05]
```

The output now also includes final Twiss parameters matched by the solutions

```
{OpticsPlot → {- Graphics -, - Graphics -},
BeamLine →
{Channel [SS[0.0696461 ], Q[0.05, -86.1333 ],
SS[0.278258 ], Q[0.05, 62.6962 ], SS[0.278258 ],
Q[0.05, -86.1333 ], SS[0.0696461 ]],
Channel [SS[0.633155 ], Q[0.05, -31.2239 ],
SS[0.995709 ], Q[0.05, 31.2766 ], SS[0.995709 ],
Q[0.05, -31.2239 ], SS[0.633155 ]]}},
SigmaEnd → {{Sigma [3.0574, -2.52413 ],
Sigma [3.05742, 2.52413 ]},
{Sigma [1.80035, 0.0654468 ],
Sigma [1.80035, -0.0654468 ]}}}
```

Notice that some vertical lines in the plots (Figures 7 and 8) now represent the longitudinal extents of the thick quadrupoles.

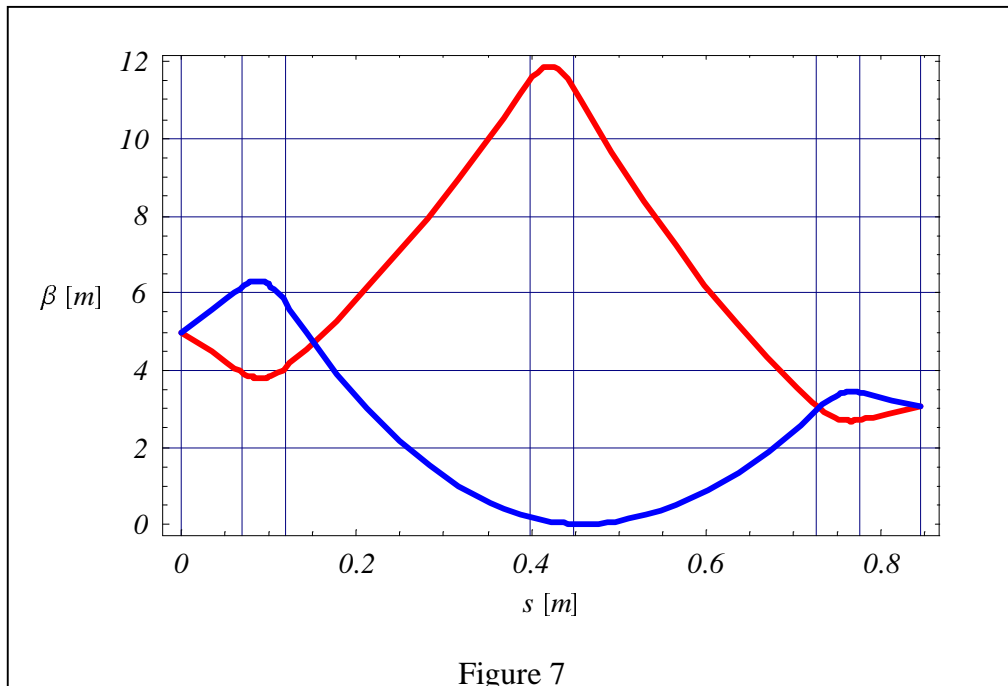
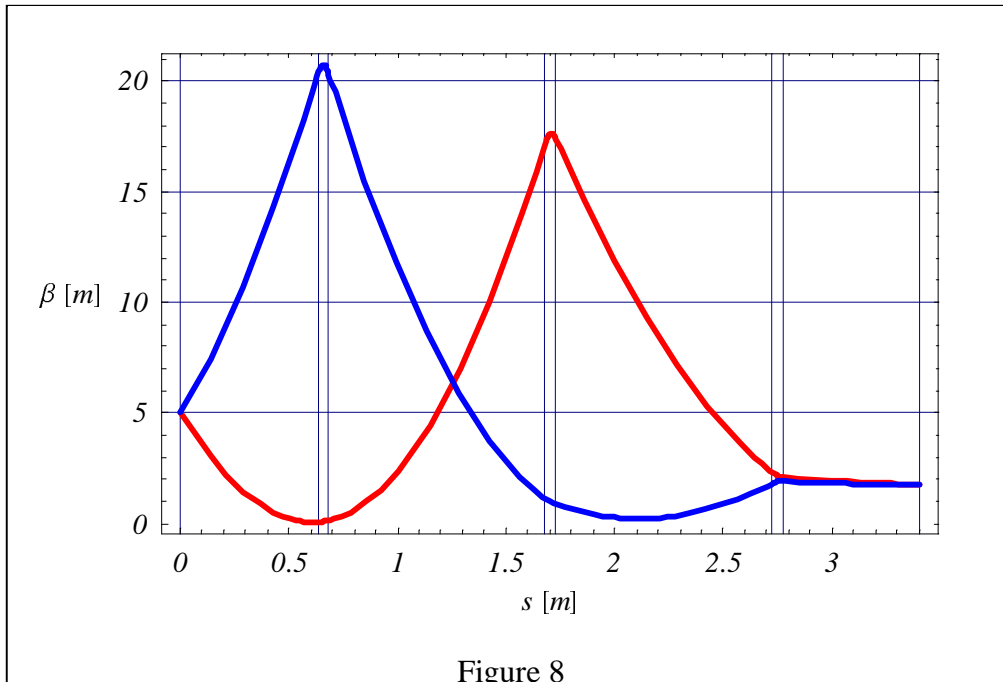


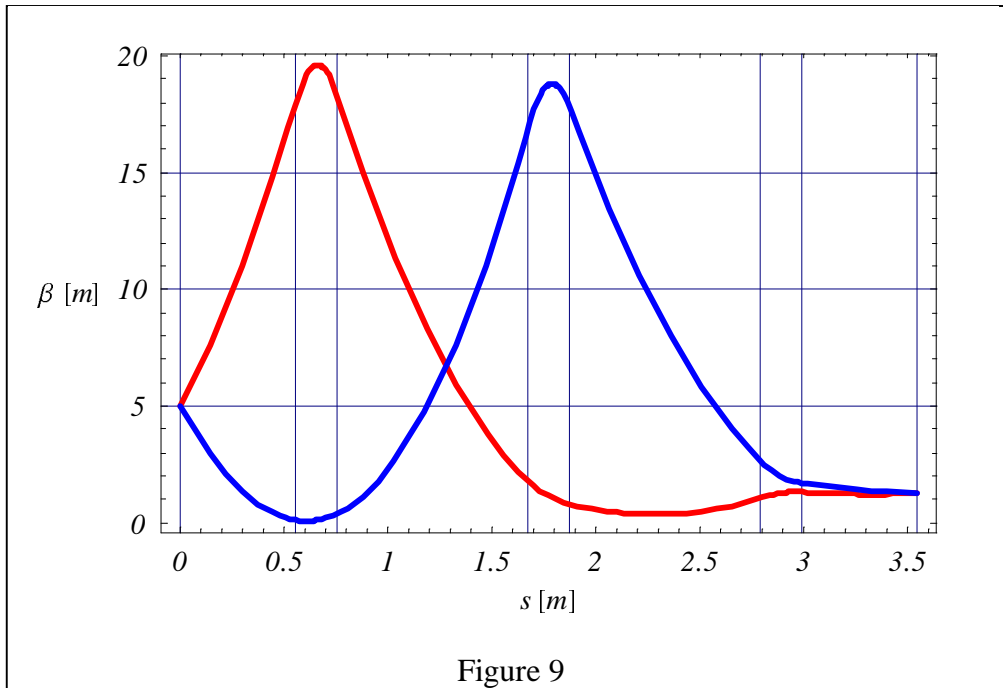
Figure 7



For longer quadrupoles, the first solution becomes unphysical, and is automatically discarded (Figure 9):

```
MatchingTriplet[Sigma[5, -8], Sigma[5, 8], Sigma[2, 0],  
Sigma[2, 0], .2, .2]
```

```
{OpticsPlot → {- Graphics - },  
BeamLine → {Channel [SS [0.558155 ],  
Q [0.2, 7.7992 ], SS [0.916689 ],  
Q [0.2, -7.83882 ], SS [0.916689 ],  
Q [0.2, 7.7992 ], SS [0.558155 ]]},  
SigmaEnd → {{Sigma [1.27448, -0.173455 ],  
Sigma [1.27448, 0.173454 ]}}}
```



One may notice that the final matched Twiss parameters are not very close to the target values. This is in large part due to the fact that the default triplet solution, with the option *Variable* set to 1, insists on perfect crossover of the Twiss parameters at the expense of matching accuracy. With this option *Variable* set to different values (2 to 4), the constraint on perfect crossover is relaxed to different degrees and a solution much closer to the target Twiss parameters can be reached (Figure 10).

```
Timing@MatchingTriplet[Sigma[5, -8], Sigma[5, 8],  
Sigma[2, 0], Sigma[2, 0], .2, .2, Variables -> 3]
```

```
{61.18 Second, {OpticsPlot -> {- Graphics -},  
BeamLine -> {Channel [SS[0.558155], Q[0.2, 8.52599],  
SS[0.774936], Q[0.2, -8.75977], SS[0.774936],  
Q[0.2, 8.52599], SS[0.558155]]},  
SigmaEnd -> {{Sigma[1.9867, 0.00311733],  
Sigma[1.92143, -0.188484]}}}}
```

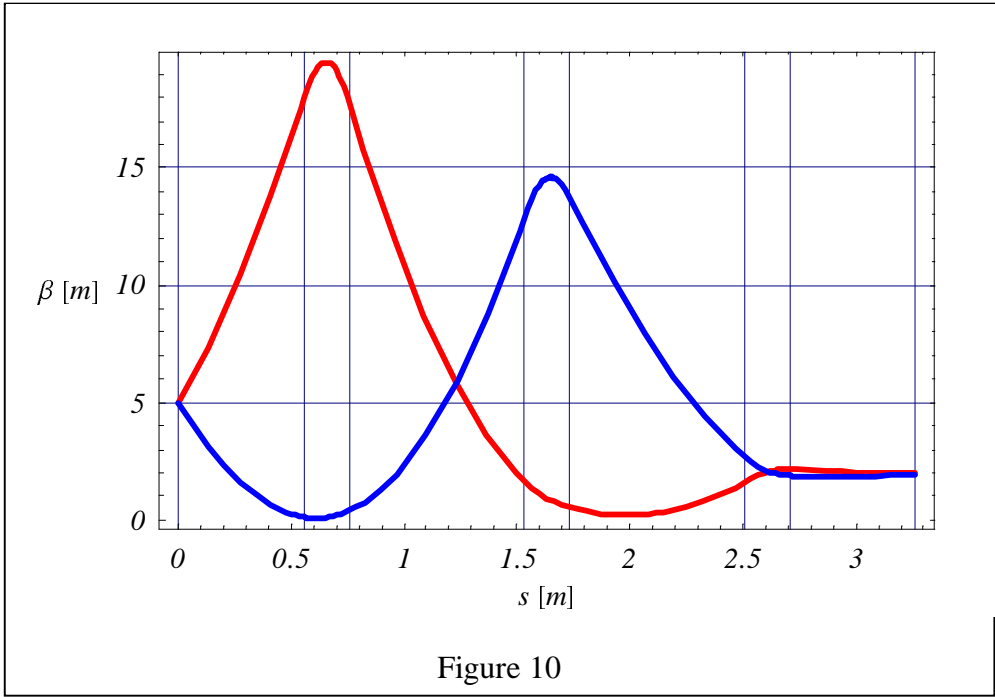


Figure 10

VI. PLANNED UPGRADES AND CURRENT ACTIVITIES

The content of all the categories of *BeamOptics* is being expanded at the moment, with contributions from diverse sources, and a few new directions of extension are presented below.

VI.1. Scope extension

Translators from standard optics programs: The models of many existing accelerator systems are stored in the format of traditional optics and other machine design packages. These are nonetheless very active files being constantly updated and processed for new information. We are certain that these model-describing files can be the basis on which to extract even more knowledge and performance enhancement if existing and planned *BeamOptics* functions can be brought to operate on them. Translator programs are important interfacing tools for realizing this.

Accelerator diagnostics and control algorithms: This is an area rich in both established and novel schemes for performing the very important task of operating and improving an accelerator system. Most of these schemes are well suited to the conversion into functions operating on a Channel object in *BeamOptics*. The latter actually corresponds to a full theoretical database of the machine, typically the object acted on by such algorithms. *BeamOptics*, in providing this theoretical database, can then serve as a very efficient test bench for accelerator diagnostics and control algorithms. Currently we are limiting the scope to transverse dimensions only.

An existing machine control architecture [7] that has been implemented online at CERN can serve as a model for future direct link between the above-mentioned diagnostics and control algorithms in *BeamOptics*, and accelerator control systems and database.

4-dimensional optics: Work on extending *BeamOptics* into the full 4-space in the transverse dimensions is under way. A beta version of 4D *BeamOptics* will be released when development of the planned 4D functions is complete.

Extended tracking functions: Various tracking functions, including nonlinear elements and special treatment for elements such as solenoids, will be added.

Longitudinal operations: In the initial stage, this will include transfer properties of elements having effects on longitudinal beam coordinates, aimed at studying longitudinal dynamics in transfer lines, and calculations of longitudinal phase space characteristics of circular machines.

VI.2. Infrastructure enhancement

The following possibilities of enhancing the infrastructure of *BeamOptics* are being studied.

Definition of interface between codes: Work is in progress to reach consensus on a standard of code interface, in view of convertibility issues between *BeamOptics* and other accelerator design packages, and anticipated incorporation of codes from

contributory sources.

Incorporation of compiled codes through *MathLink*: Work is in progress to Assess the need of and options for streamlining the “production versions” of some core algorithms via *MathLink* and compiled codes. This is expected to bring about substantial gain in speed when applied to large systems in real time, while still interfacing with *BeamOptics*.

VI.3. Evolution strategy: community participation and user/developer support

Concurrent with work on extending its scope, the evolutionary strategy for *BeamOptics* is being planned. The emphasis is on enhancing its appeal and usefulness to a wider audience in the accelerator community. To this end we are engaged in the following activities:

Community participation: We are actively soliciting collaboration and contribution from the accelerator community, possibly extending to a standing community-wide forum on *BeamOptics*.

User/developer support: Accompanying the release of *BeamOptics*, a web site [8] has been created for downloading updated versions, documentation, and dialogue with users and developers. A related supporting document under preparation is a future CERN Yellow Report, closely correlated to the feature and functionality of *BeamOptics*, providing theoretical bases for the latter. This report will also describe general strategies for designing new accelerators on these theoretical bases.

REFERENCES

- [1] *BeamOptics*, A Program for Analytical Beam Optics, B. Autin, (Ed.), C. Carli, T. D'Amico, O. Groebner, M. Martini, and E. Wildner, CERN Yellow Report 98-06, 1998.
- [2] S. Wolfram, The *Mathematica* Book, 4th Ed., Wolfram Media, 1999.
- [3] B. Autin and P. Royer, *Optimum Optical Systems*, CERN/PS 2000-013 (AE) and Proc. 7th European Particle Accelerator Conference, Vienna, Austria, June 2000.
- [4] Y. Chao, *A full-order, almost-deterministic optical matching algorithm*, to be submitted to Particle Accelerator Conference 2001, Argonne, Illinois, USA.
- [5] Shiji Machida, private communication.
- [6] T. d'Amico and G. Guignard, *Analysis of a symmetric triplet and its application to ring insertions*, CERN/PS 97-25 (LP) and Proc. 17th Particle Accelerator Conference, Vancouver, 1997.
- [7] A. Jansson and M. Lindroos, *"Intelligent" automatic beam steering and shaping*, CERN/PS 2000-019 (OP) and Proc. 7th European Particle Accelerator Conference, Vienna, Austria, June 2000.
- [8] <http://abs.web.cern.ch/ABS/beamoptics/index.htm>