ASAP - A Symbolic Algebra Package for Accelerator Design

Eva Bozoki, Aharon Friedman and Ilan Ben-Zvi

National Synchrotron Light Source, BNL, Upton, NY. 11973

Abstract

We report the conceptual design and primary implementation steps of a symbolic algebra program based on MACSYMA [1] for the design of accelerators, storage rings and transport lines. The motivation for using symbolic algebra is discussed and a design case is presented that shows the advantage of this approach.

1. Introduction and motivation

The design of a modern accelerator is a complicated task that involves the integration of many devices. As a consequence many parameters must be optimized in order to achieve a satisfactory result. Even the design of a simple subsystem, such as a bending system, requires that the designer will pick a successful choice from a wide range of alternatives. Usually, the task is too large to allow an analytical design, and the designer has to use a computer code (such as MAD [2] or TRANSPORT [3]) to simulate the system and numerically find the desired conditions. The disadvantages of this numerical method are, that (i) the solution, i.e. the choice of the parameters may or may not be optimal and (ii) each change in a parameter requires to recalculate the whole system, thus a detailed design is lengthy and costly.

In this paper we present a method of solving the design problem by using a symbolic code instead of a numeric one. A set of beam parameters (tune, β- function, dispersion, beam size, etc.) are derived as a function of the properties of the physical elements (lengths of drift space, length and strength of dipole and quadrupole, etc. - referred to as variable parameters). The requirements are defined by a set of equations for the beam parameters, and at that point the designer can use fast numerical and intuitive methods to find proper working ranges. Instead of having to recalculate the beam parameters from scratch for every change in the variable parameters, the designer can now use a prepared set of equations and just plug-in the new parameter values. We believe that this method results in a great saving of design time and a better quality of design.

2. The physical model

An optical particle ray is described by a six dimensional phase space vector,

$$\vec{V} = \begin{bmatrix} X \\ X' \\ Y \\ Y' \\ \Delta\phi \\ \Delta E/E \end{bmatrix}, \text{ where } \begin{bmatrix} X, Y & \text{– location of the ray relative the beam center} \\ X', Y' & \text{– angle relative the beam axis} \\ \Delta\phi & \text{– path length difference between the ray and the center of the beam} \\ \Delta E/E & \text{– relative energy of the ray and the beam center} \end{bmatrix}$$

The whole beam can be described by a distribution function of these six elements. As the beam propagates through the optical elements its distribution function changes. the change of a single ray is described by a 6 x 6 matrix, A, which represents what happens to the ray as it passes through any particular element:

$$\vec{V}^{after} = A\ \vec{V}^{before}\ . \qquad (1)$$

The matrix of a compound system is obtained by multiplying the matrices of its elements. The matrices for the most common elements , drift space, dipole (wedge or not) and quadrupole are collected in the 'Element Library'. Other elements, in most cases, can be approximated as a sequence of the above simple elements (and their matrix as the product of the matrices of the simple elements). Note, that eq.(1) not only describes what happens to a single ray, but is actually a transformation between coordinate systems, hence it describes the new beam distribution function:

$$f'(\vec{X}') = f(A^{-1}\vec{X}'). \qquad (2)$$

Usually, constraints are placed on the system by requiring that some of the matrix elements or some functions of them satisfy certain conditions. There can be constraints on the beam size, dispersion, its derivate, the orientation of the phase space ellipse, isochronocity to list a few. The most commonly used conditions will be collected in a 'Constraint Library'.

In accelerator design there are typical operations performed on matrices or on their elements (concatenate, invert, reflect, etc.). There are also operations which use the matrix elements to calculate beam parameters. These operations will make a set of precompiled MACSYMA functions and subroutines and this set will be kept in an 'Operations

*Work performed under the auspices of the U.S. Department of Energy.

0-7803-0135-8/91$01.00 ©IEEE

Library'

3. The Computer Model

A block diagram of the structure of ASAP is shown in Fig. 1. The heart of the system is an existing symbolic algebra package, such as MACSYMA or any other existing package. The main requirements are:

1. Input format which lends itself easily to accelerator design language, i.e. which can be made to accept a command language similar to that of MAD,
2. Possibility to easily build and maintain optical elements library operations library and constraints library,
3. Can generate the expressions in a high level computer language, preferably FORTRAN or C, in addition to the usual algebraic form,
4. Can interact with the user in a user friendly environment,
5. Has some interactive capabilities, so the user can plot while working and then based on the information make decisions and continue the work.
6. Is computer efficient, since most of the applications will be computer intense.

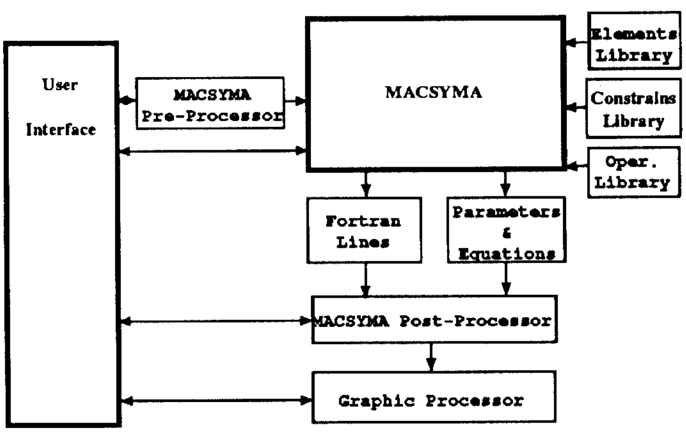

Fig. 1: The basic structure of ASAP

Most of the symbolic algebra packages, that exists today have the first five properties. However, only MACSYMA seems to have the ability to compile functions and subroutines, thus to be efficient.

The user interface is the other important component of the system. It takes the structure of the accelerator and the constraints on the system from the user and transmits them to MACSYMA. The user interface will converse with the user in 'accelerator intelligent' language so that the user is not required to be a computer expert. Once MACSYMA produced the solution, the user interface will, while interacting with the user, build the appropriate Fortran code and run it. Finally, it will present the results with the aid of a graphic interface. Fig. 2 illustrates the actual steps of the ASAP aided design, showing that the User Interface plays the major role (at least for a beginning user, who is not very familiar with MACSYMA). However, as the user gains more experience, he/she will be able to bypass the User Interface and interact directly with MACSYMA and making use of the element, constraint and operations libraries, thus making the design process faster.

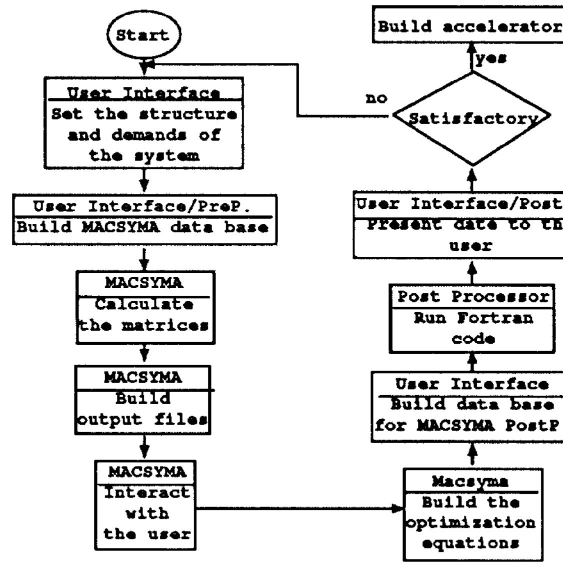

Fig. 2: Flow chart of ASAP

4. Libraries

The *Elements library* will consist of at least the following elements (keeping the MAD naming convention [2]): Drift, Sbend, Rbend and Quad. An additional element is Edge which is a description of the fringing field and rotation for dipoles.

The *Operations library* will include two types of operations. (i) Operations on matrices, such as concatenate the optical elements, invert a matrix, reflect a set of elements to create a reflex-symmetric beam line. (ii) Operations that construct the beam parameters from the matrix elements and the beam initial parameters. Such operations are: getting the dispersion parameter ($\eta=M_{16}$), construct the beam σ matrix, getting the phase space ellipse parameters, etc.

In many cases it is constructive to limit the dimension of the phase space to less then six. In such cases the library will supply operations to extract sub matrices from the 6x6 matrices.

In addition to the tailored Operations library, ASAP will use the built-in MACSYMA operation libraries to perform things such as solving equations, optimizing expressions with a given set of constrains, Taylor expansions, Integrations, etc.

The *Constraints library* will contain optical conditions (point-to-point imaging, parallel-to-point focus, etc.), achromaticity, isochronocity, conditions on phase space ellipse, beam matrix (orientation and limits on beam size and beam divergence), dispersion, etc.

5. Test Case

As a demonstration of the advantages of the described approach, we bring here a design case performed by one of the authors of this paper [4]. In this case, MACSYMA was used to design a large angle ($\Theta = 142°$) and a very short bending radius (ρ=.222m) composite bend (consisting of two combined function dipoles with drift spaces and a quadrupole between them). The constraints were (i) to match the dispersion vector (η, η') to a desired value at the end of the bend, (ii) to keep the beam size minimal and (iii) to minimize the length of the bend with (iv) realistically implementable quadrupole strength. The variable parameters were the lengths, field index and bend angle of the dipoles (with $\theta_1+\theta_2=\Theta$), the lengths and strength of the quadrupole and the length of the drift spaces.

A numerical design program, such as TRANSPORT would have the following disadvantages: (i) Depending on the initial parameters it finds a local optimum which is not necessarily the best working point. (ii) The program will not converge to a solution if the initial parameters are too far away from it. Thus, the designer must have an initial guess which is quite close to the best solution. (iii) When not converging to a solution, the program does not indicate whether it is because there is no solution or that the initial guess was to far. (iv) In order to map the dependence on different variable parameters, one has to run the program many times. This process is extremely time consuming.

Instead, MACSYMA was used to construct an analytical representation of the total composite bend matrix (M) from the elements' matrices and to express the dispersion vector dependence on the variable parameters. Even though the resulting formulae were too involved to see how the dispersion vector depend on the properties of the physical elements, with simplifying assumptions, it was possible to glean insight on the asymptotic behavior of the system. Furthermore, using the non-approximative form, it was possible to calculate and plot η and η' as a function of the different variable parameters and to choose reasonable working regions for those parameters. A detailed description of this design case is found in ref [5].

6. References

[1] "MACSYMA User's Guide", *Symbolics, Inc., 1988.*

[2] F.C. Iselin, "MAD User's Reference Manual", *CERN/SL/90-13, 1990.*

[3] K.L. Brown, D.C. Carey, F.C. Iselin and F. Rothacker, "TRANSPORT", *CERN/SPS/80-04, 1980.*

[4] E. Bozoki, "Beam Transport to the SXLS Ring" *Presented at this conference.*

[5] E. Bozoki, *BNL Report - to be published.*