Interfacing the BMAD Optics Library With Beam Instability Code

Tim Lambie-Hanson
Department of Physics, University of Richmond, Richmond, VA 23173
(Dated: August 13, 2004)

Beam instability code has been developed at Wilson Synchrotron Laboratory to simulate the effect of higher order modes on an electron beam. To use the beam instability code, information about the lattice must be imput. The BMAD optics library is a huge subroutine library that, along with other features, allows the user to create a lattice file, storing much information about each element and also the lattice as a whole. BMAD provides subroutines to generate transfer matrices and twiss parameters, track particles, and generally recover information about the lattice. To ease usage and increase efficiency of the beam instability code, a marriage between this code and BMAD has begun to be implemented.

I. INTRODUCTION

The purpose of this project is to use the BMAD optics library [1] to help make the beam instability software more efficient and userfriendly. The beam instability programs [2], bi and findbi, use an input file that consists of basic information about the higher order modes, the number of RFCavities in the lattice, the transfer matrices between cavities, and a particle's travel time between the cavities. These programs use this information to simulate the effect of the higher order modes both on the bunches in the beam and on the fields in the RF Cavities. A time-consuming part of the process of running the beam instability code was generating the transfer matrices between each RF Cavity. So a program, lattobbu.f90, was written to retrieve the number of RF Cavities, the transfer matrices between each cavity, and the time between each cavity from a BMAD lattice file, and then merge this information with a text file containing the information about the higher order modes. This information is output to a bi input file, which, with either bi or findbi, can be run (See Fig. 1). To ease usage, a script can connect these processes and allow the user to go from the two input files directly to bi or findbi output.

II. PROGRAMS AND USAGE

These programs were all developed while attempting to use the BMAD optics library to make bi more accessible. These programs all retrieve information from BMAD, manipulate this information in some fashion, and then output this information to a text file. In the case of lattobbu.f90 and erlbbuin.f90, this text file is in the form of a bi input file, in the case of alltransfermat.f90, it is not.

lattobbu.f90

The usage of lattobbu.f90 is relatively straightfoward. After compiling lattobbu.f90 with the BMAD makefile, simply run the executable with three command line arguments: 1)the

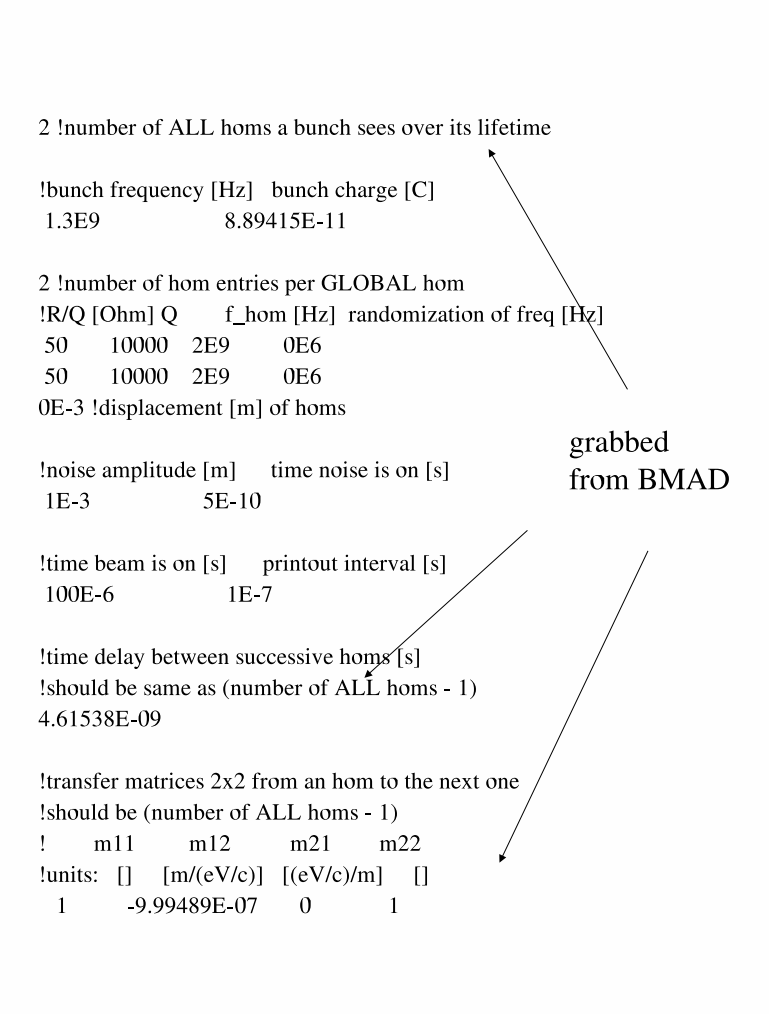

FIG. 1: A bi input file.

BMAD lattice file, 2) a text file with the remaining information from a typical bi input file, and 3) a flag, -x or -y, indicating which transverse direction the user would like to investigate. For bi usage, the format of the input does not matter, only the order of the numbers with whitespace in between, but when using lattobbu.f90 the input must be put in a more precise arrangement (See Fig. 2).

alltransfermat.f90

This program is the part of lattobbu.f90 that generates the transfer matrices. This program was mainly used for debugging the lattobbu.f90. It takes as input a lattice file and then outputs a text file with the two by two transfer matrices between each RF Cavity. For future use, in the interest of programming style, it might be a good idea to change this program to a subroutine, allowing code generating bi input files to simply call this subroutine by passing it a lattice file.

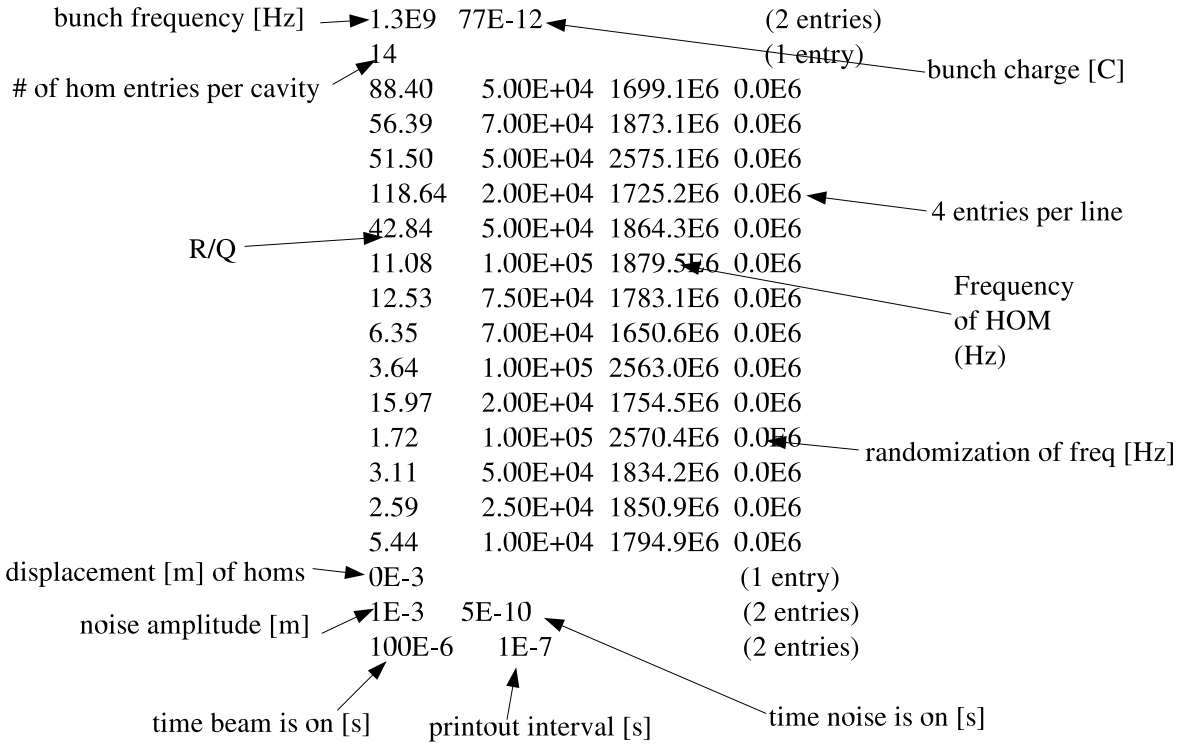

FIG. 2: The necessary format of the lattobbu.f90 input file with HOM information.

erlbbuin.f90

This program was generated to create the transfer matrices between higher order modes and the time between higher order modes input for a particular erl lattice. The reason the generic lattobbu.f90 program will not create these matrices is that in this erl between two known lattice files there is a generic arc that consists of unknown elements. Since no accelerating cavities would be in this arc, no higher modes exist, so a hypothetical transfer matrix is created using the twiss parameters before and after the arc, and an arbitrary distance is assigned the arc. The transfer matrix and time for a particle to travel the arc are placed in the proper spot of the input file, between the values of the two known lattices. To use this input with bi, the remaining information, the number of RF Cavities and the information describing the higher order modes, was simply cut and paste to this text file.

III. COMPUTING DETAILS

The first step in generating this information for the bi input file was reading in the lattice file and getting the raw numbers needed to create the input. To read in the lattice file, the subroutine bmad_parser is passed the name of the lattice file and the name for the ring_struct

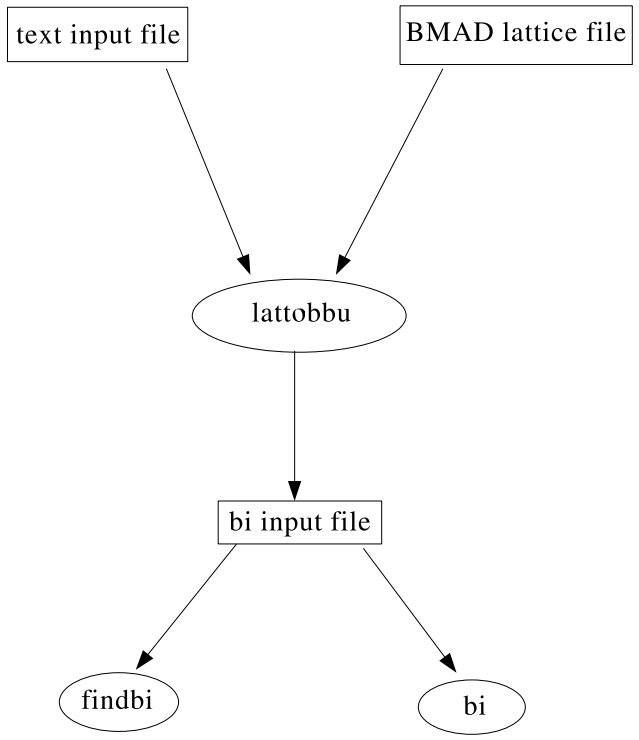

FIG. 3: Schematic of a script running bi or findbi from two input files.

that it is going to make. After this ring_struct is made, the subroutines twiss_propagate_all and ring_make_mat6 are passed the ring_struct to generate the twiss parameters at each element and transfer matrices for each element.

Time Between Higher Order Modes

The first step in generating the time between higher order modes is to loop over all the elements to see how many LCAVITY structures are in the ring_struct. This information is also necessary on its own for the bi input file. The program then loops over the ring_struct. At the first LCAVITY it saves the z position of the element. When it gets to the next LCAVITY, it subtracts this first distance from this current LCAVITY's z and divides by the speed of light, generating the time between these two cavities and writing it to the output file. It saves the newest cavity's z value, discarding the first cavity's z. This continues until the program has written one less time then the number of total cavities in the ring_struct.

Transfer Matrices

The most obvious method of generating transfer matrices between two elements is to multiply all of these matrices together, with the matrix furthest to the left modeling the element furthest around the ring. Since the transfer matrices used by bi have a momentum factor in them, this product matrix would have to be normalized by the energy. This method should work but in this effort was not able to implemented in such a way that gave accurate transfer matrices. The attempt to generate in such a way is in the program directtransfermat.f90.

So instead, a roundabout method of generating the transfer matrices was used. This method multiplied all the transfer matrices together throughout the entire ring_struct. It saved the information about first transfer matrix and when it got to to the second transfer cavity, used this information to calculate the matrix between the two. This matrix was normalized with the beam energy and then written to the output file. Then it would save the transfer matrix at the second cavity and discard the first transfer matrix and proceed this way until it propogated around the entire ring. This is as roundabout way of generating the matrices that takes more computing power and requires space for four more reals, but it generates accurate transfer matrices.

IV. RESULTS AND CONCLUSIONS

As a result of these new programs, using bi has become more efficient and easier. The capability to generate bi and findbi output by simply providing a BMAD lattice file and a text file with information characterizing the higher order modes increases the accessibility and potentially saves the user a lot of time generating the transfer matrices between higher order modes. Work remains to make bi more efficient and effective.

Currently, one program simulates the effect of transverse higher modes and another simulates the effect of longitudinal higher order modes. In doing so, the program assumes that these modes are decoupled. To achieve more complete results, these two programs should be merged to simulate the affect of higher order modes in both transverse directions and longitudinal higher order modes. Since computation time is bound to increase if the effect of all three modes is simulated during one run, a possible implimentation would include flags to indicate which modes (any combination of x-transverse, y-transverse, and longitudinal) should be simulated.

When generating the two by two transfer matrices, each number was simply taken from the BMAD structure and the two by two matrix multiplication was done manually. When working with the six by six transfer matrices, some sort of matrix structure should be used to ease the readability of the code. Experimentation with the BMAD mat6 structure indicates that some other matrix structure should be used to store these matrices while manipulating them.

The work documented in this paper is a step forward, but plenty of work remains to streamline bi and findbi usage. As more work is done, bi and findbi will become even more powerful tools in simulating beam instability due to higher order modes.

V. ACKNOWLEDGEMENTS

I would like to acknowledge Dave Sagan and Ivan Bazarov for proposing this project and guiding my efforts. I would also like to acknowledge Rich Galik for organizing the REU program and making it an enjoyable experience. This work was supported by the National Science Foundation REU grant PHY-0243687 and research cooperative agreement PHY-9809799.

---

[1] The BMAD library was developed by Dave Sagan at the Wilson Synchrotron Laboratory. Available at http://www.lns.cornell.edu/ dcs/bmad/

[2] Beam instability code was developed by Ivan Bazarov at Wilson Synchrotron Laboratory. Available at http://lepp.cornell.edu/ ib38/