Improvements in MAD in view of LHC Design

F. Christoph Iselin
CERN
SL Division, AP Group
CH-1211 Geneva
Switzerland

Abstract For the design of the LHC (CERN Large Hadron Collider) it is important to perform extensive checks on long-time stability. Comparison of MAD (Methodical Accelerator Design program) with specialized tracking programs showed that tracking speed of MAD was insufficient. Therefore time-critical routines have been tuned so as to take advantage of vectorization on both Cray and IBM vector computers.

The superconducting magnets used in LHC contain various high-order multipole errors. Definition of field error components has been made easier with a new command.

Among other effects the multipole errors introduce strong coupling. In LHC it is removed by a few families of skewed quadrupoles. This design requires matching over the whole ring, and forces MAD to recompute a large number of transfer maps over and over again. This process has been speeded up by orders of magnitude since MAD now precomputes transfer maps for single elements and for large pieces of the ring ("lumps"), and recalculates them only when really needed. It detects maps which depend on variable parameters and updates them automatically.

The effects of octupoles and decapoles have been included in the second- and third-order chromaticity calculation of HARMON, and it is possible to match second- and third-order chromaticity by adjusting those elements.

1 INTRODUCTION

The MAD program [1, 2] has been developed at CERN to provide an input language and a data structure suited for computations on circular accelerators. The program is portable across a variety of computers (Cray XMP under UNICOS, IBM under VM/CMS, VAX under UNIX or VMS, Apollo under Aegis or UNIX), but portability will not affect efficiency. Sections 2 and 4 address some issues where these requirements seem contradictory.

Sections 3 and 5 expose new features introduced to handle multipole errors occurring in large accelerators with superconducting magnets.

0-7803-0135-8/91$01.00 ©IEEE

2 TRACKING SPEED

Particle tracking serves mainly to determine long-time stability of the accelerator. In a typical tracking run some 20 to 100 particles must be tracked over 10^4 to 10^7 turns. For LHC (CERN Large Hadron Collider) [3] which contains more than two thousand magnets (including correcting elements), particles must sometimes be tracked through a sequence of up to 10^{10} magnets. It is estimated that tracking for LHC will require several months of Cray-XMP CPU time. For tracking the main issue is obviously speed, but Liouville's theorem must be obeyed.

Advantage should be taken of vectorization wherever possible. Unfortunately the vectorizing FORTRAN compilers on Cray and IBM systems work in different and sometimes contradictory ways. On the Cray the compiler never vectorizes an outer loop, and the size of vector registers is 64. On the IBM the compiler often chooses to vectorize an outer loop, and the size of vector registers varies with the computer model used, it is normally 128. Vectorization on the IBM is further complicated by the need to avoid frequent reloads of the cache store. On both systems loops using recurrence relations should be avoided. The vector stride must stay away from certain small powers of two, and the trip count of vectorized loops should be as large as possible. If it does not exceed the size of the vector registers, and the compiler knows this fact (short vector loop), the object code becomes faster.

MAD was carefully tuned to identify the bottlenecks, and the most time-consuming subroutines where tuned for speed. Where possible the code was written such as to vectorize advantageously on both systems. Four problem areas were identified in the original MAD program when tracking with thin multipoles:

1. In long tracking runs magnets are normally all simulated as thin multipoles. Originally tracking consisted of two steps:
   - All multipole components were copied from the dynamic store of MAD into a local array. During the copy process they were transformed from the amplitude-phase form into normal and skewed components.
   - The algorithm then used a recursive complex Horner scheme to accumulate the kicks for each particle.

The routine for thin multipoles accounted for more than 90% of CPU time. It had clearly to be improved.

2. Transfer of control, like subroutine calls and IF-tests affect running time adversely. Drift spaces are the most frequent elements in the machine. They are now treated by inline code, instead of calling subroutines over two levels. This saves about 10 % of the running time remaining after the multipole tracking was optimized.

3. In MAD particle positions could optionally be listed after each element. The tests for print requests involved function calls to unpack the print flags after each element in the machine. Taking out this facility resulted in another 5 % saving of CPU time.

4. After each turn particle positions are kept in a dynamically allocated table. This table is used later for plotting and written out on an ASCII file after tracking has finished. ASCII writing accounts for some 10 % of the tracking run time, and the central memory size limits the number of turns which can be saved. A new option in MAD, instead of saving this table, now immediately writes particle positions on a binary file after a turn is completed. The file format used is portable among various computers. This allows post-processing of the results in a variety of ways. A post-processor has been written by F. Galluccio [4].

The thin multipole routine has been rewritten in two versions. Both versions transform the multipole coefficients the first time each multipole is seen and save the transformed coefficients for later re-use.

For tracking on the IBM the outer loop cycles over the particles, and the inner loop performs the Horner algorithm. The compiler vectorizes the outer loop without problems. The vectorization saves about a factor 3 over the scalar version.

On the Cray-XMP the outermost loop is replaced by a loop over batches of not more than 64 particles each, and the loop over the particles of a batch is moved into the recursive loop for the Horner scheme, and becomes a short vector loop. This results in a factor 10 improvement with respect to the scalar version.

After the above changes tracking through thin multipoles with the Cray version is slower by roughly a factor two compared to specialized tracking programs like SIXTRACK [5, 6]. It is felt that a general-purpose program like MAD, which has many more options, cannot easily be made more efficient than this. Tracking time on a Cray-XMP is about 5 microseconds per particle and thin multipole. This corresponds to about 10.6 milliseconds per particle and per turn in the LHC; or 4500 CPU seconds for 22 particles and 20000 turns.

3 FIELD ERRORS

Superconducting magnets tend to have large field errors compared to conventional magnets, and it is important to simulate them correctly. Normally the magnet designers produce error figures expressed as mean values and standard deviations for both normal and skewed multipole components. The existing error definition command EFIELD of MAD expresses field errors in terms of two random numbers for amplitude and rotation angle of each multipole. To simulate these distributions correctly, one needs two thin multipoles, one to represent the normal component, and one the skewed part of the error.

A new command EFCOMP has been introduced which expresses the errors in terms of normal and skewed components. With this command all errors of a magnet can be expressed with one thin multipole. This permits another cut in CPU time for tracking.

4 PRECOMPUTED TRANSFER MAPS

The skewed quadrupole components occurring in the main ring magnets of LHC introduce strong coupling. For cost reasons coupling is compensated with a few families of skewed quadrupoles placed in the insertions. This means that the whole ring must be expanded when these quadrupoles are matched. Many invariant transfer maps must be re-evaluated over and over again in such matching runs.

A big time saving is possible by using the LUMP command of MAD. It defines a "super-element" by concatenation of several transfer maps. Since the concatenation is made once and for all at definition time, it permits a significant time saving for all optical calculations, in particular for tracking and matching.

Element transfer maps were originally recalculated for each occurrence of the same magnet during optics and tracking calculations. A large fraction of running time was due to this recomputation. In particular for dipoles the transfer map must be built from three concatenated pieces, namely the magnet body and the entrance and exit fringe fields. Clearly time can be saved if the transfer map is prepared and saved the first time a magnet is seen, so that it can be re-used for all subsequent occurrences. The MAD data structure would make it easy to apply this method to all magnet types. However, this would consume a lot of memory, and would save little time for most elements. The production version of MAD uses it therefore for dipoles and thin multipoles only.

To make sure that precomputed maps always correspond to the state of element definitions, MAD contains an algorithm which detects whether a transfer map depends on any change of definitions. This algorithm marks all parameters and elements when they change, and follows all dependence chains before a command is executed. Any map which should be recomputed is automatically deleted. Even on a relatively slow workstation the CPU time used for this will not exceed 1 second.

5 HARMON SECTION

The large multipole errors in LHC cause large non-linear chromaticities, which must be corrected. The octupole and decapole components have been included in the HARMON computations for the first and second derivative of the dispersion with momentum error, and for the second- and third-order chromaticity. They are not yet used for the resonance coefficients. The HCELL command is able to match all three orders of chromaticity by varying a set of multipoles.

The matching procedure in HARMON has been speeded up drastically by switching to the LMDIF method which is also used in the matching module of MAD.

6 CONCLUSIONS

After having served for the design of LEP, MAD is now adapted to help in designing the LHC. Special needs for a proton machine had to be considered.

References

[1] D. C. Carey and F. C. Iselin, "Standard Input Language for Particle Beam and Accelerator Computer Programs". *1984 Summer Study on the Design and Utilization of the Superconducting Super Collider*, Snowmass, Colorado, 1984.

[2] H. Grote and F. C. Iselin, "The MAD Program (Methodical Accelerator Design) Version 8.1, User's Reference Manual". CERN/SL/90-13 (AP).

[3] G. Brianti et al., "The Large Hadron Collider (LHC) in the LEP Tunnel". Particle World, Vol 1., pp. 104–109, March 1990.

[4] F. Galluccio, private communication.

[5] F. Schmidt, "SIXTRACK Version 1, Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner, User's Reference Manual". CERN/SL/90-11 (AP).

[6] F. Schmidt, "SIXTRACK, A Single Particle Tracking Code", *Workshop on Non-Linear Problems in future Particle Accelerators*, Capri, Italy, 1990.