EMBEDDING FINITE ELEMENT RESULTS FOR ACCELERATOR COMPONENTS IN A MOMENT APPROACH BEAM DYNAMICS CODE [*] [†]

T. Roggen, B. Masschaele, H. De Gersem,
K.U.Leuven Kulak,
Wave Propagation and Signal Processing Research Group, Kortrijk, Belgium
W. Ackermann, S. Franke, T. Weiland
TU Darmstadt, Theorie Elektromagnetischer Felder (TEMF), Darmstadt, Germany

Abstract

A moment based beam dynamics code has particular advantages, i.e. accuracy and efficiency, over macro-particle tracking and full particle-in-cell (PIC) codes respectively. Instead of embedding analytical descriptions of the accelerator components in the beam dynamics model, it is proposed to insert a surrogate model obtained from the finite element model of individual accelerator components. We apply the V-Code, which accepts moments up to the sixth order and accounts for space charge effects. We construct and calculate finite element and finite difference time domain models using the CST Studio Suite 2011 software package. An interface is implemented using VBA and MATLAB. As an example of the accuracy of this cascadic simulation approach, we compare the beam dynamics of an S-DALINAC quadrupole obtained by directly tracking particles to the calculated fields with the results for the cascadic approach with the V-Code.

INTRODUCTION

Start-to-end beam dynamics simulations become more and more a strategic tool to verify proper operation and implementation of new state of the art components into a particle accelerator, either for new accelerator designs as well as within refurbishing projects [1].

The V-Code is a beam dynamics simulation code based on the moment approach [2, 3]. A particle bunch is characterised in a six dimensional phase space by its longitudinal and transverse dynamics. The average bunch coordinates and momenta, and their higher order correlation parameters (up to the sixth order) are described by a phase distribution function $f(\tau, \vec{r}, \vec{p})$, with $\tau = c.t$ the equivalent time, c the speed of light, $\vec{r}$ the space coordinates and $\vec{p}$ the normalized momentum ($\vec{p} = \vec{P}/mc$). Solving the Vlasov equation (Eq.1), a bunch described by $f(\tau, \vec{r}, \vec{p})$ can be tracked throughout the accelerator using time integration, with the option to account for space charge forces.

$$\frac{\partial f}{\partial \tau} + \frac{\partial f}{\partial \vec{r}}\frac{\vec{p}}{\gamma} + \frac{\partial f}{\partial \vec{p}}\frac{\vec{F}}{mc^2} = 0 \qquad (1)$$

with $\vec{F}$ the force exerted by the particle accelerator component and $\gamma = E/mc^2$ the normalized energy, with energy E and mass m. Instead of modeling the magnetic or electric field distribution of the beam line element (BLE) analytically, the V-Code uses a multipole expansion to reproduce the field distribution using the field data from finite element or finite difference time domain simulation results. This paraxial approach requires a field component in the longitudinal direction as a data input for the field reconstruction, which is obtained by appropriate postprocessing of the simulation results from CST Studio Suite 2011. This contribution deals with the software implementation in CST Studio Suite 2011 to readily produce the V-Code field data files for several BLE's, using MATLAB.

IMPLEMENTATION

General Approach

To generate the V-Code field data files, first a static FE simulation or FDTD simulation is carried out. Then, field data needs to be extracted at the relevant positions, and post processed to generate the necessary data for the V-Code. CST Studio Suite 2011 provides the possibility to create custom VBA macros enabling easy execution of scripts
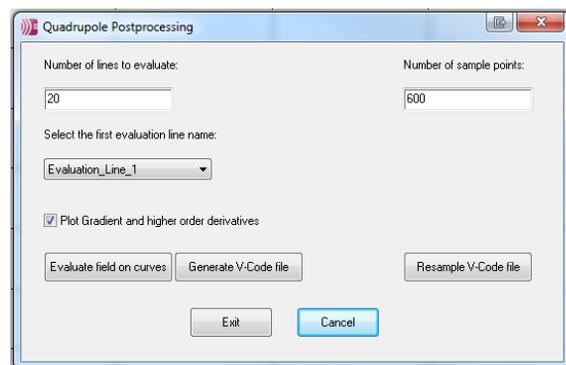

Figure 1: VBA dialog box popping up within CST for preparation of a data file for a quadrupole to be used in the V-Code.

containing parametrised operations, environment settings, solver executions, custom code and other tasks. Being COM based, it also permits CST to control (or to be controlled by) other applications, which allows us to use MATLAB (also COM based) for integrated numerical data post processing and for the generation of the V-Code data file.

After selecting a BLE in the drop down menu of the main window, a custom user dialog for this BLE pops up (Fig.1). Prior to forwarding the field data calculated with CST to MATLAB, it needs to be extracted from the field solution of CST, using the "Evaluate field on curves"-button. A spatially limited set of field components, typically along the particle reference track, is extracted and saved in a separate *sig file. This limits the necessary data transfer to MATLAB. Subsequently, a MATLAB server is invoked and all settings and parameters are forwarded to MATLAB's base workspace.

All subsequent data processing on the *sig file is handled through BLE specific MATLAB functions, invoked with the "Generate V-Code File"-button. For every BLE-type the field distribution reconstruction method in the V-Code is different, resulting in type-specific data post processing functions in MATLAB. After normalization of the data, the V-Code data file is saved in a *dat format, which can be read by the V-Code.

Accurate V-Code beam dynamics calculations require smooth field transitions at borders and inside a BLE. This can be verified through the smoothness of higher order derivatives of the data provided in the V-Code data file. A feature for visual inspection of the derivatives up to the thirth order is implemented in the code. Related to this requirement, a cubic spline interpolation method is implemented to enhance the field transition smoothness (Fig.2).

At this moment V-Code field data file generation is available for RF cavities, dipoles, quadrupoles, sextupoles, octupoles, solenoids, steerer magnets and Wien filters.

Implementation for a Quadrupole

The magnetic field component on the axis of an ideal quadrupole, sextupole or octupole is zero, which implies a different approach to the field reconstruction method. The V-Code requests a dataset containing the variation of the gradient along the longitudinal axis. Differences with the general implementation are firstly the post processing step in CST, which is extended to multiple files containing spatially limited sets of field components, and secondly the MATLAB data processing steps. Special care has to be taken of the CST mesh settings when generating these files.

Implementation for a Dipole

When performing beam dynamics for dipoles, the V-Code also takes into account their fringe fields [4]. Therefore the magnetic field reconstruction is more complicated and more field information has to be provided. Since the field reconstruction is performed in the proximity of the bunch, the reference particle's path in the dipole
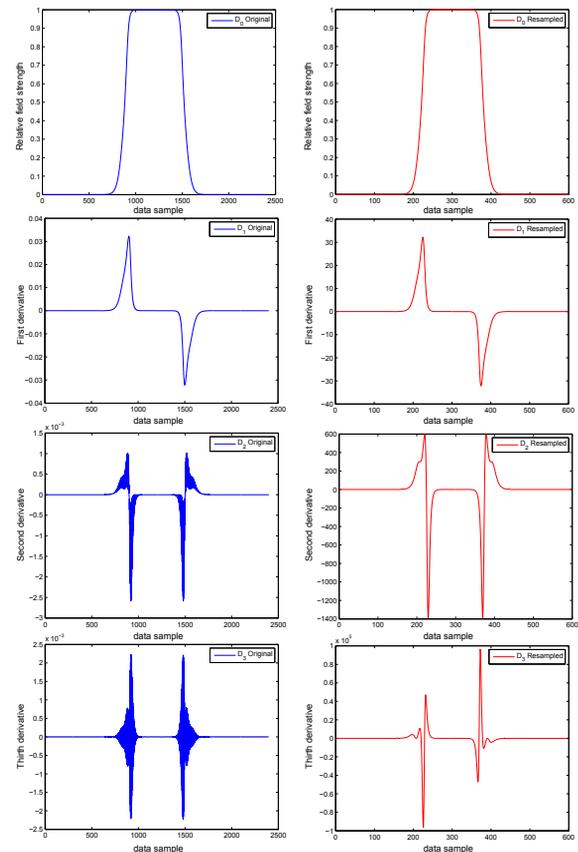

Figure 2: Verification of the field transition smoothness through higher order derivatives for a quadrupole V-Code data file. Left: Unfiltered data. Right: Filtered data.

needs to be imparted. This can be determined using the built-in CST particle tracker, and exported as two *dat files containing the x and z coordinates (X_Track_File.dat and Z_Track_File.dat), when we define z as the axis normal to the initial particle propagation in a orthogonal right handed coordinate system. The normalized field strength of the bending magnet along the reference path can be extracted in a Field_File.dat, analogue to the general approach. Less straightforward is the composition of the Derivative_File.dat and the Edge_Focussing_File.dat, containing respectively the gradient amplitude and the gradient direction (in rad) of the vertical field component relative to and along the reference path. These data can be retrieved evaluating the vertical field component on multiple auxiliary circles along the reference particle's path using the CST post processing tools, as proposed in [4]. However, the auxiliary circle construction and field evaluation in CST is CPU-intensive and can only be executed on a single core. Moreover, this procedure has to be iterated along the reference particle track, resulting in a lot of CPU time. Therefore, the complete 3D field will be exported to MATLAB, which is far more efficient in performing this data processing. The field is analysed on an analytical circle around every particle position, using MATLAB's Parallel
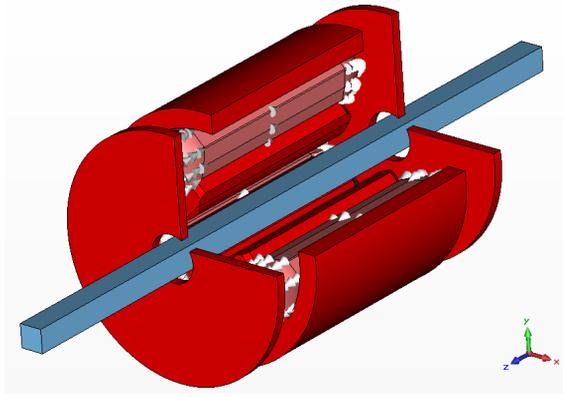
05 Beam Dynamics and Electromagnetic Fields

D06 Code Developments and Simulation Techniques

Figure 3: CST model of the Danfysik QSL040 quadrupole.

Figure 4: Beam envelope comparison between CST and V-Code for the Danfysik QSL040 quadrupole.

Computing Toolbox. This enables the user to speed up the data generation significantly using up to eight cores on a single machine [5].

BEAM DYNAMICS COMPARISON

To verify the correct implementation of our cascadic approach and to test our implementation, a beam dynamics comparison was set up between the V-Code and the CST particle tracker, and this for every BLE. Hereunder the procedure and results for the Danfysik QSL040 quadrupole (Fig.3), one of the quadrupoles used in the S-DALINAC, are discussed. This quadrupole, mirror plates included and in focussing configuration, was modeled in CST using the original technical drawings from the manufacturer and was extended at both sides with a drift tube.

For the magnetic field calculation and particle tracking, a mesh of 20 million cells with symmetry settings in three dimensions was used to track 21000 computational particles representing individual electrons with an energy of 10 MeV (energy spread 0 %). Since only a uniform particle distribution and no Gaussian distribution can be selected in CST's particle tracker, the comparison implies a verification of the beam envelope of both simulations. The beam envelope is defined as 3σ, with σ the variance of the Gaussian bunch in the horizontal (x) or vertical (y) direction. The initial electron bunch has a transverse radius of 1 mm (3σ) and no transverse momentum. In CST, the electrons are emitted from a source with an outer diameter of 2 mm and an inner diameter of 1.8 mm. This allows a dense particle distribution on the edge of the source with a limited number of particles, since we are interested in the envelope of the beam.

Figure 4 shows the beam dynamics comparison between CST and the V-Code. Both the horizontal and vertical beam envelopes of the bunch are plotted in function of the longitudinal propagation z of the particle in the BLE. The electron bunch enters a drift tube at 0 mm and arrives at the mirror plate of the focussing quadrupole at 166 mm. At 434 mm the bunch exits the quadrupole and enters a second drift tube of 166 mm. The procentual difference of
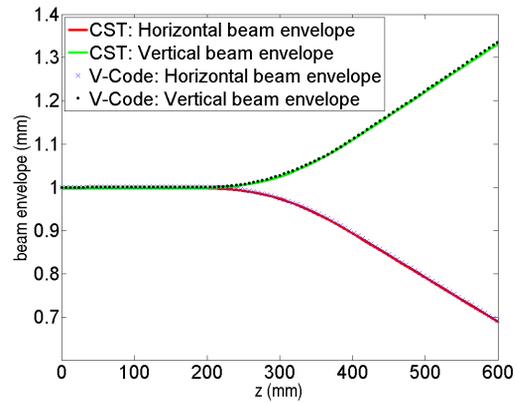
the beam envelope size is less than 1 %, which indicates a good agreement between both particle trackers and confirms proper implementation of our cascadic approach.

CONCLUSIONS

The implementation in CST Studio Suite of VBA and MATLAB based software to readily generate the V-Code field data files for the reconstruction of the magnetic or electric fields in the V-Code was succesful for RF cavities, dipoles, quadrupoles, sextupoles, octupoles, solenoids, steerer magnets and Wien filters. It enables the user to easily create and verify the quality of these files directly in the CST environment through custom BLE dialog screens.

A beam dynamics comparison calculated with the V-Code and with the CST particle tracker demonstrates a good coherence between both trackers, with a procentual difference of the beam envelope of only 0.5 %.

REFERENCES

[1] P. Spentzouris, J. Cary, L. C. McInnes, W. Mori, C. Ng, E. Ng, R. Ryne, "Advancing computational science for future accelerators and accelerator technologies",Journal of Physics: Conference Series, Vol.125 No.1, p.012005, 2008, IOP Publishing

[2] A. Novokhatski and T. Weiland, "Self-Consistent Model for the Beams in Accelerators", ICAP'98, Montery, September 1998, F-We21,

[3] S. Franke, W. Ackermann, T. Weiland, "A Fast and Universal Vlasov Solver for Beam Dynamics Simulations in 3D", ICAP'09, San Francisco, September 2009, TH1IODN01, pp. 208-211,

[4] S. Franke, W. Ackermann, B. Steiner, T. Weiland, "Implementation of Fringe Field Dipole Magnets into the V-Code Beam Dynamics Simulation Tool", EPAC'08, Genoa, June 2008,TUPP089, pp. 1738-1740

[5] MATLAB: Parallel Computing Toolbox, Available: http://www.mathworks.nl/products/parallel-computing/ index.html, [Accessed: August 29, 2011]