Cloud-based design of high average power traveling wave linacs

Cloud-based design of high average power traveling wave linacs

S V Kutsaev[1], Y Eidelman[2], D L Bruhwiler[2], P Moeller[2], R Nagler[2] and J Barbe Welzel[2]

[1]RadiaBeam Technologies LLC, 1717 Stewart St, Santa Monica, CA 90403, USA,
[2]RadiaSoft LLC, 1348 Redwood Ave, Boulder, CO 80304, USA

kutsaev@radiabeam.com

Abstract. The design of industrial high average power traveling wave linacs must accurately consider some specific effects. For example, acceleration of high current beam reduces power flow in the accelerating waveguide. Space charge may influence the stability of longitudinal or transverse beam dynamics. Accurate treatment of beam loading is central to the design of high-power TW accelerators, and it is especially difficult to model in the meter-scale region where the electrons are nonrelativistic. Currently, there are two types of available codes: tracking codes (e.g. PARMELA or ASTRA) that cannot solve self-consistent problems, and particle-in-cell codes (e.g. Magic 3D or CST Particle Studio) that can model the physics correctly but are very time-consuming and resource-demanding. Hellweg is a special tool for quick and accurate electron dynamics simulation in traveling wave accelerating structures. The underlying theory of this software is based on the differential equations of motion. The effects considered in this code include beam loading, space charge forces, and external magnetic fields. We present the current capabilities of the code, provide benchmarking results, and discuss future plans. We also describe the browser-based GUI for executing Hellweg in the cloud.

1. Introduction

High power travelling wave (TW) linear accelerators are the most promising solution for high-current applications that include: treatment of potable and waste water, removal of pollutants from stack gases, increased efficiency of material processing; and replacement of radioactive sources in sterilization application. The pulsed current in such linacs can be as high as tens of amperes. Such high currents can cause a number of specific effects, which must be accurately considered in the design stage. One of the most important of them is a beam loading effect, when a high current beam reduces the accelerating electric field amplitude and, hence, the output beam energy. Another important effect is the space charge forces acting in the bunch, which can be comparable to the single-particle electromagnetic forces at low energy. The space charge algorithm should also include the influence of neighbouring bunches, since in industrial linacs the beam typically consists of a train of bunches filling every RF period.

There is no commercially available software that provides fast, accurate calculations of beam loading effects in high power TW linear accelerators. For example, the commercial GPT code can only model beam loading for relativistic beams. The popular PARMELA code does not model beam loading at all, so the correct amplitude and phase in each accelerating cell must be calculated externally. Commercial codes like Magic 3D and CST Particle Studio can model the physics correctly;

however, they are expensive and the required run time is long, so multiple parameter optimization for system design is impractical.

Meanwhile, a rather simple and convenient method for intensive beam dynamics in TW accelerating structures analysis has been proposed [1] and realized in the Hellweg code [2]. This code is based on differential equations that can accurately simulate beam dynamics with beam loading, space charge, and external magnetic fields. We have recently enhanced the Hellweg code with improved algorithms and a cloud-based Graphical User Interface (GUI) to make this code available to the accelerator community.

2. Hellweg code overview

Hellweg was originally developed at the RF Technology Laboratory of Moscow Engineering-Physics Institute. Hellweg is a C++ application that runs on Microsoft Windows with a GUI. The kernel can be executed from the command line on GNU/Linux. The input data for this program is a plain text file that specifies the beam (initial phase, energy and phase distributions, input current, Twiss parameters), the traveling wave structure (input power, frequency and cell parameters), and computational parameters (number of macroparticles, mesh, etc.).

The numerical model is based on self-consistent equations describing electron motion in waveguide structures with variable dimensions. This system includes 2D dynamics in an axisymmetric structure; an equation for a self-consistent RF-field amplitude created by the beam; and an equation for a particle's phase in a self-consistent field. The code used the simplified ellipsoid-based single-bunch space charge model [3] and simple approximations of external magnetic fields.

Modern accelerator design, however, can require non-axially-symmetric field profiles such as multipole magnetic fields or accelerating structures with a cyclic variation of azimuthal asymmetries, which requires fully 3D dynamics. This then required the space charge model to be improved to the general 3D ellipsoid form for arbitrary aspect ratio, consideration of the space charge field for particles outside the beam, and adding fields from the neighboring bunches (which are not independently simulated but must be included approximately). Some applications (e.g. novel cargo inspection systems based on energy-ramping accelerators) require the consideration of realistic external magnetic fields, usually imported from pre-calculated models, and the ability to properly model the initial beam distribution including spherical cathodes and asymmetric 4D Twiss parameters. Some users require the ability to use Hellweg for only part of the accelerator system, for which robust import/export of multiple file formats is required.

A browser-based GUI for cloud computing is necessary to attract a significant user base in both the commercial/engineering and research communities. Scientific cloud computing is the only practical way to sustainably support software and users in the absence of significant funding. As described below, this approach also enables instantaneous collaboration and computational reproducibility.

We have recently made the following improvements to the code:

- converted the code to 3D model
- revised the beam initialization
- added new results for visualization
- enhanced the parametrization of solenoid fields and added fringe field effects
- improved the space charge model to 3D ellipsoid, including fields outside of the bunch
- added multi-bunch space charge model
- added import/export of beam to/from CST Particle Studio and ASTRA
- added import of 1D, 2D and 3D magnetic field maps
- added quadrupole elements with predefined 2D field maps
- developed a browser-based GUI, which exports Python script for command-line use

In the following section, we briefly discuss some of these enhancements.

3. New three-dimensional model

Hellweg was originally developed to solve beam dynamics problems with 2D cylindrical symmetry. This approach is sufficient for standard DLS-type accelerating structures with only the fundamental accelerating cavity mode; however, it is not adequate for the wide range of 3D systems, nor for problems with broken 2D symmetry: solenoids, quadrupole optical elements, higher order modes (HOM), asymmetrical structures etc.

We have derived from first principles the 3D differential equations for all velocity components (β_z, β_r and β_θ), where $\beta=v/c$. The equations include all six RF field components (E_z, E_r, E_θ ,H_z, H_r, H_θ) and three external magnetic field components (H_z^{ext}, H_r^{ext}, H_θ^{ext}):

$$\frac{d\beta_\zeta}{d\zeta}=\frac{1}{\gamma\beta_\zeta}\left[(1-\beta_\zeta^2)A_\zeta+\beta_\eta\left(H_\theta-\beta_\zeta A_\eta\right)-\beta_\theta\left(H_\eta+\beta_\zeta A_\theta\right)+\beta_\eta H_\theta^{ext}-\beta_\theta H_\eta^{ext}\right] \tag{1}$$

$$\frac{d\beta_\eta}{d\zeta}=\frac{1}{\gamma\beta_\zeta}\left[(1-\beta_\eta^2)A_\eta+\beta_\theta\left(H_\zeta-\beta_\eta A_\theta\right)-\beta_\zeta\left(H_\theta+\beta_\eta A_\zeta\right)+\beta_\theta H_\zeta^{ext}-\beta_\zeta H_\theta^{ext}\right]+\frac{\beta_\theta^2}{\eta\beta_\zeta} \tag{2}$$

$$\frac{d\beta_\theta}{d\zeta}=\frac{1}{\gamma\beta_\zeta}\left[(1-\beta_\theta^2)A_\theta+\beta_\zeta\left(H_\eta-\beta_\theta A_\zeta\right)-\beta_\eta\left(H_\zeta+\beta_\theta A_\eta\right)+\beta_\zeta H_\eta^{ext}-\beta_\eta H_\zeta^{ext}\right]-\frac{\beta_\theta\beta_\eta}{\eta\beta_\zeta} \tag{3}$$

Here all values are normalized by the wavelength λ or the particle rest energy W_0 as follows:

$$\zeta=\frac{z}{\lambda},\ \eta=\frac{r}{\lambda},\ \alpha=\frac{r}{\lambda}\dot{\theta}=\eta\dot{\theta}, A=\frac{E\lambda}{W_o}, H=c\frac{B\lambda}{W_o} \tag{4}$$

These equations were derived from the following fundamental equation of motion:

$$\frac{d\vec{v}}{dt}=\frac{e}{m_0}\sqrt{1-\frac{v^2}{c^2}}\left(\vec{E}+\frac{\vec{v}\times(\vec{B}+\overrightarrow{B_{ext}})}{c}-\frac{\vec{v}(\vec{v}\cdot\vec{E})}{c^2}\right) \tag{5}$$

To take into account the space charge of the particle we have solved the Poisson equation for the ellipsoid beam and obtained the following adjustment component values for the electric field:

$$E_r=\frac{\rho}{2\varepsilon_0\gamma^2}\left(M_x(\lambda)\cdot x\cos\theta+M_y(\lambda)\cdot y\sin\theta\right) \tag{6}$$

$$E_\theta=\frac{\rho}{2\varepsilon_0\gamma^2}\left(M_y(\lambda)\cdot y\cos\theta-M_x(\lambda)\cdot x\sin\theta\right) \tag{7}$$

$$E_z=\frac{\rho}{2\varepsilon_0}M_z(\lambda)z \tag{8}$$

Here $M_{x,y,z}(\lambda)$ is a form factor of the beam defined as follows:

$$M_u(\lambda)=R_xR_yR_zu\int\limits_\lambda^\infty\frac{ds}{(R_u^2+s)\sqrt{(R_x^2+s)(R_y^2+s)(R_z^2+s)}} \tag{9}$$

In Eq. (9), R_x, R_y, and R_z are the bunch dimensions, u represents a coordinate x, y or z, and λ is a root of a characteristic equation:

$$\frac{x^2}{R_x^2+\lambda}+\frac{y^2}{R_y^2+\lambda}+\frac{z^2}{R_z^2+\lambda}=1 \tag{10}$$

A multi-bunch space charge effect can be included by solving the characteristic equation for two neighbouring bunches traveling ahead and behind of the reference bunch, at a distance L:

$$E_z=\frac{\rho}{2\varepsilon_0}\left[M_z(\lambda)z+M_z(\lambda_1)(z-L)+M_z(\lambda_2)(z+L)\right] \tag{11}$$

The detailed derivation and analysis of the formulas (1)-(11) will be presented elsewhere.

Support for 3D dynamics required major revision of the particle initialization algorithm. Previously, the beam's transverse coordinates could only be initialized via three Twiss parameters: α, β and ε. Moreover, the distribution was Gaussian in the x-coordinate, not in the radial variable, which can be technically incorrect for realistic beams. We have recently added the following options of beam initialization, which cover many cases of realistic 3D distributions:

- 4D Twiss parameters: α_x, β_x [cm/rad], ε_x [cm*rad] α_y, β_y [cm/rad], ε_y [cm*rad]
- Gaussian longitudinal distribution; W0 [MeV] ΔW [MeV] σ_W [MeV] φ_0 [deg] Δφ [deg] σ_φ [deg]

- Emission from a spherical cathode: R_{cath} [cm], R_{sph} [cm], kT [eV]
- Kapchinsky-Vladimisky distribution: a_x [cm], b_y [cm], φ [deg], h

Generic beam initialization does not always provide the required particle distribution, especially when the electrons come from an external simulation of a particle source, such as a gun or another accelerator. Also, users often need to track a single particle with defined parameters, or a beam with some exotic distribution. For these cases, we added the capability to import a sequence of particles from a text file.

4. Example of industrial linac design

We benchmarked the code on simple problems that allow for analytical calculation. Such tests included:

- Beam in a drift pipe with and without magnetic field
- Beam in RF accelerating structure
- Bunching
- Acceleration
- Beam loading
- Emittance conservation

Also, we have designed an industrial linac with Hellweg and compared the simulations with the PIC module of CST Particle Studio [4]. The linac is designed for a maximum power of 9 MeV, with a 2π/3 constant impedance travelling wave structure. The accelerator is powered by a klystron with 5 MW peak power at operating frequency of 2856 MHz. The accelerating structure consists of 25 cells, including lower-beta bunching cells. Beam focusing is performed with three 0.1T solenoids. The linac design is presented in Figure 1.

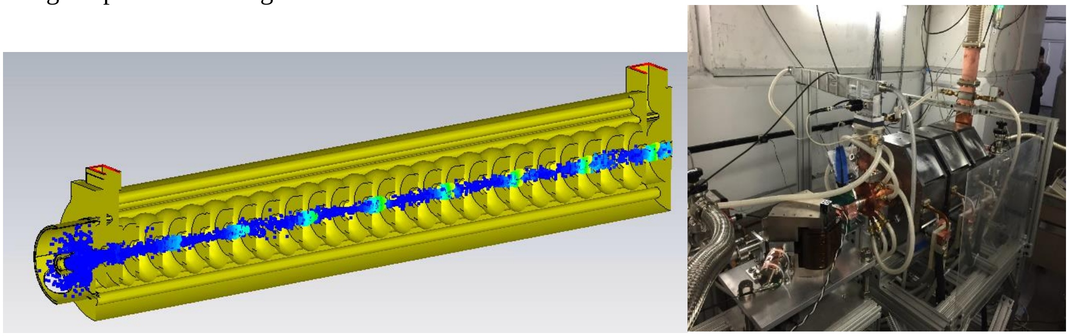
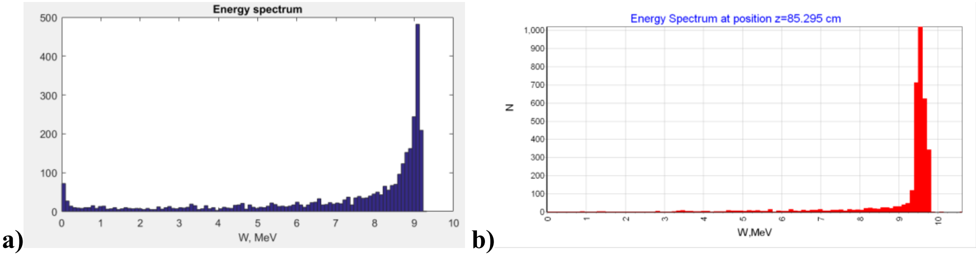


Figure 1. Left: 9 MeV accelerator RF design with the propagating beam (color represents the energy up to 9 MeV). Right: The actual picture of accelerator.

The comparison of the output beam parameters from Hellweg and CST are presented in Table 1 and Figure 2 and demonstrate a very good agreement.



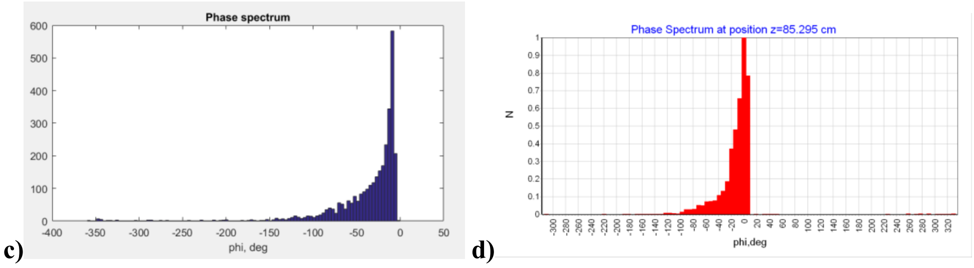


Figure 2. Simulated beam energy (a,b) and phase (c,d) spectra in CST Particle Studio (a,c) and Hellweg (b,d).

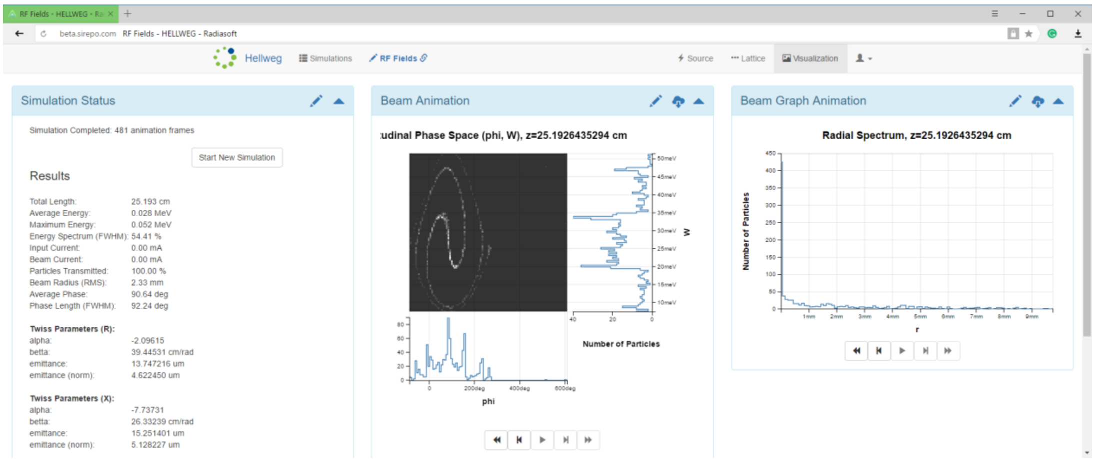
Table 1. Comparison of Hellweg and CST simulations.

Code	**Hellweg**	**CST**
Input current	250 mA	250 mA
Output current	124 mA	115 mA
Beam energy	9.6 MeV	9.2 MeV
Input power	4.6 MW	4.6 MW
Output power	1.73 MW	1.72 MW
Simulation time	~30 sec	~8 hrs

5. Browser-based GUI for cloud computing

We have ported the Hellweg physics kernel from Windows to the gcc compiler on GNU/Linux, including an initial Python framework for unit and regression testing, while retaining the flexibility to compile with the original GUI for Windows. [5] The source structure is designed to enable delivery of the Hellweg kernel via PyPI [6] in the future, from the command line via ‘pip install rslinac’. Using the open source framework Sirepo [7], we have developed an initial browser-based GUI for Hellweg and made it publicly available for beta testing. [8] Sirepo allows easy integration of legacy, open source simulation codes. [9] The platform is designed to encourage reproducible research where sharing of simulations is crucial. [10] Figure 3 shows a screen shot of dynamics in an RF cavity.

Sirepo GUIs invoke JavaScript in the browser, while Python is invoked on a remote Linux server to execute the requested scientific kernel (e.g. Hellweg) within a Docker [11] container. Sirepo apps work on a tablet as well as a desktop computer, and they enable instantaneous collaboration – one can send a link to anyone, allowing them to immediately copy the full simulation and start working with it. One can also export a working Python script with all the associated input files, so the GUI does not prevent expert users from working on the command line.



Simulation Completed: 481 animation frames

Start New Simulation

Results

Total Length:	25.193 cm
Average Energy:	0.028 MeV
Maximum Energy:	0.052 MeV
Energy Spectrum (FWHM):	54.41 %
Input Current:	0.00 mA
Beam Current:	0.00 mA
Particles Transmitted:	100.00 %
Beam Radius (RMS):	2.33 mm
Average Phase:	90.64 deg
Phase Length (FWHM):	92.24 deg
Twiss Parameters (R):	
alpha:	-2.09615
betta:	39.44531 cm/rad
emittance:	13.747216 um
emittance (norm):	4.622450 um
Twiss Parameters (X):	
alpha:	-7.73731
betta:	26.33239 cm/rad
emittance:	15.251401 um
emittance (norm):	5.128227 um

Figure 3. Screenshot of browser-based Sirepo GUI, running an example Hellweg simulation.

6. Conclusion

The updated Hellweg presents the scientific and commercial opportunity to enable cloud-based design of high average power traveling wave linacs. The improved physics of Hellweg and the implementation it within Sirepo as an open source framework provides a fast, accurate and convenient tool for cloud computing of electron beam dynamics problems to accelerator physicists community.

References

[1] Masunov E.S., “Bunching and Acceleration of the Intensive Beam in the Waveguide Accelerating Structure with a Presence of an External Magnetic Field”, Journal of Technical Physics 49, pp. 1462-1463 (1979).

[2] Kutsaev S.V., “Electron dynamics simulations with Hellweg 2D code”, Nuclear Instruments and Methods in Physics Research A618, pp. 298–305 (2010).

[3] Lapostolle P. M., “Effets de la charge d'espace dans un accélérateur linéaire `a protons,” CERN, Geneva, Tech. Rep. CERN-AR-Int-SG-65-15, Jul 1965, in French

[4] Kutsaev S.V. et al., “Design, Simulations and Experimental Demonstration of an Intra-Pulse Ramped-Energy Travelling Wave Linac for Cargo Inspection”, Proc. of NAPAC’16 Conference, 2016

[5] The Hellweg source repository on GitHub, https://github.com/radiasoft/rslinac

[6] PyPI – the Python Package Index, https://pypi.python.org/pypi

[7] The Sirepo framework for cloud computing, http://sirepo.com

[8] Open beta test of Hellweg in the cloud, https://beta.sirepo.com/hellweg

[9] Nagler R., Moeller P. and Bruhwiler, D.L. “Sirepo: An Open Source Platform for Portable Reproducible Simulations,” https://doi.org/10.6084/m9.figshare.4688566.v1

[10] Nagler R., Bruhwiler D.L., Moeller P. and Webb S.D., “Sustainability and Reproducibility via Containerized Computing,” https://arxiv.org/abs/1509.08789

[11] What is Docker,” https://www.docker.com/what-docker

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Award Number DE-SC0015897.