SIMULATIONS AND MEASUREMENTS OF BEAM BREAKUP IN DIELECTRIC WAKEFIELD STRUCTURES*

P. Schoessow#, A. Kanareykin, C. Jing, Euclid Techlabs, 5900 Harper Rd, Solon, OH 44139
A. Kustov, Dynamics Software, Helsinki, Finland
W. Gai, J. G. Power, ANL, Argonne, IL 60439, U.S.A.
Alexander Altmark (LETI, St Petersburg Russia)

Abstract

Beam breakup (BBU) effects resulting from parasitic wakefields are a serious limitation to the performance of dielectric structure based accelerators. We report here on numerical studies and experimental investigations of BBU and its mitigation. An experimental program is underway at the Argonne Wakefield Accelerator facility that will focus on BBU measurements in dielectric wakefield devices. We examine the use of external FODO channels for control of the beam in the presence of strong transverse wakefields. We present calculations based on a particle-Green's function beam dynamics code (BBU-3000) that we are developing. We will report on new features of the code including the ability to treat space charge. The BBU code is being incorporated into a software framework that will significantly increase its utility (Beam Dynamics Simulation Platform). The platform is based on the very flexible Boinc [1] software environment developed originally at Berkeley for the SETI@home project. The package can handle both task farming on a heterogeneous cluster of networked computers and computing on a local grid. User access to the platform is through a web browser.

INTRODUCTION

Beam breakup effects resulting from parasitic wakefields provide a potentially serious limitation to the performance of dielectric structure based wakefield accelerators. We report here on the status and recent results from the project "Beam Breakup Instability in Dielectric Structures", an experimental and numerical investigation of BBU and its mitigation. The numerical part of this research is based on a particle-Green's function beam breakup code (BBU-3000). The code (fig. 1) is a flexible 2D and 3D code based on analytic Green's functions for single particle fields in axisymmetric dielectric loaded structures. The development and features of the code have been described elsewhere [2, 3]. In this paper we focus on recent efforts at porting the code to a workstation cluster, providing access via the web using the Boinc environment, and implementing a tree algorithm for space charge calculations.

The experimental program focuses on BBU measurements at the AWA facility in a number of high gradient and high transformer ratio wakefield devices.

*Work supported by US DoE SBIR Grant, Contract #DE-FG02-07ER84823
#paul.schoessow@euclidtechlabs.com

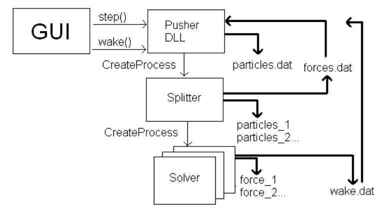

Figure 1: Structure of the BBU-3000 code.

SIMULATION PLATFORM

The Beam Dynamics Simulation Platform development continued and the decision was made to develop on top of existing Boinc framework. For that, the following key elements were designed:

- Experiment data input from the Boinc web-interface
- Enhanced Graphics
- Multitasking in Boinc: both multiple users and projects
- "Sandbox" or providing users the possibility to customize the solver code in the Web GUI.

Boinc was originally based on the concept of volunteer computing, when people from around the world donate their unused CPU resources for the Boinc project. So Boinc is designed for high throughput: millions of volunteer hosts, millions of jobs per day. Its server runs on Linux.

The support for the client side in Boinc is very well though-out and very convenient. Downloading a Boinc client and its use is easy, well-productized and tested. Adding a new client to an existing project requires minimum effort and does not present any problems to the user.

On a given host computer, each project has a separate project directory (fig. 2). All the files for that project - application files, input files, output files - are stored in the project directory.

Each job runs in a separate slot directory, which contains links to all the files needed by that job. The names of the link files are called logical names, and the files they point to have physical names.

The server side in Boinc is less productized, and every project requires considerable effort to set up. The processing cycle of a job is rather complicated:

- A *work generator* creates the job and its associated input files.
- Boinc creates one or more instances of the job.
- Boinc dispatches the instances to different hosts.
- Each host downloads the input files.

- After a possible queuing delay caused by other jobs in progress, the host executes the job, and then uploads its output files.
- The host reports the completed job, possibly after an additional delay (the purpose of which is to reduce the rate of scheduler requests).
- A *validator* checks the output files, perhaps comparing replicas.
- When a valid instance is found, an *assimilator* handles the results, e.g., by inserting them in a separate database.
- When all instances have been completed, a *file deleter* deletes the input and output files.
- A *DB purge* program deletes the database entries for the job and job instances.

In the sequence above, the work generator, the validator and the assimilator needed to be developed. This is a rather complicated task, requiring a deep understanding of data processing logic, even though examples are provided.

Integration with the existing BBU-3000 code presented a particular challenge—the task dispatching, which is implemented by calling a Debian script, needed to be invoked from Windows code running on a separate host. The task was solved by implementing a Web front end to the Boinc server and using HTTP POST requests as a transport.

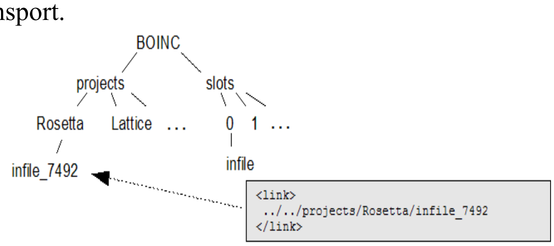

Figure 2: On a given host, each project has a separate project directory. All the files for that project - application files, input files, output files - are stored in the project directory.

Each project has a web page, where tasks and their statuses, users and groups, news etc. are visible. In the case of BBU-3000, the Boinc job is a combination of the Solver and its output data (`particles.dat`). The work generator is based on the pusher and the Splitter, which might divide the `particles.dat` file into set of smaller ones for each simulation step. The validator also uses the Splitter for assembling resulting partial `forces.dat` files into the cumulative final one.

TREE ALGORITHM FOR SPACE CHARGE

The present algorithm used in BBU-3000 computes pairwise particle interactions at each time step to determine the forces on each electron in the simulation. This algorithm scales as $O(N^2)$ where N is the number of particles used. While for small particle numbers this is not problematic, larger scale problems (particularly for the ramped bunch train and other multibunch experiments) require a large number of particles and hence can become very inefficient.

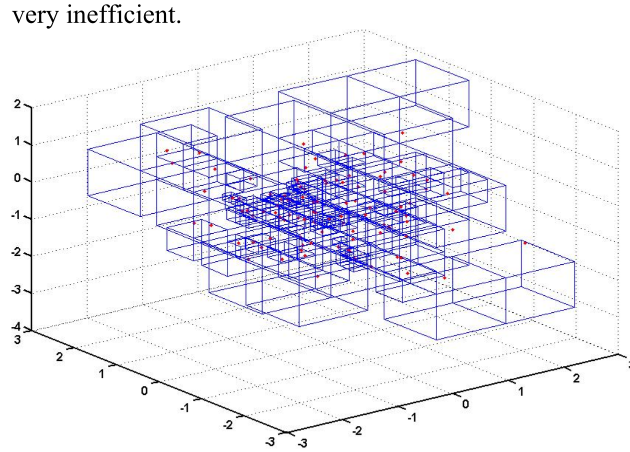

Figure 3: Test of the tree-code algorithm. Each electron (red) in the bunch is assigned a place in a hierarchy of cubic cells (blue). Only the final level of cells, each containing a single electron is shown. The total number of macroparticles in this example is 100.

For the space charge calculation the most promising approaches are based on a set of algorithms developed in recent years that factor particle-particle interactions into short and long range components [4]. Interactions between particles in close proximity are computed as in the existing code. Forces on a given electron resulting from clusters of electrons at longer separations are handled by replacing the individual cluster particles in the force calculation by a single particle with effective properties computed through the use of spatial averaging.

We are developing a new algorithm to speed up the force calculation. The basic idea is to surround the particle bunch with a small r-z computational grid. The macroparticles are interpolated onto the grid and the forces computed at each particle for each nonempty grid cell. This results in a scaling $O(N^b)$, b<2, since the outer loop over particles in the calculation is now replaced by a loop over the cells of the grid. The outline of the algorithm is as follows:

At each timestep

1. Boost particles into the rest frame of the bunch. We will assume that the relative velocities of the particles are small so that the forces in the rest frame are purely electrostatic.

2. Define a cubic cell that just encloses the (boosted) bunch. Subdivide into 8 equal cells. Discard empty cells; add nonempty cells to a list, flagging the number of macroparticles and the center of charge in each. Repeat until at the final level, all cells contain a single particle (Fig.3).

3. Loop over particles:

a. For all cells with a center of charge more than some predefined distance r_0 from particle k's position, compute the Coulomb force on k based on the center of charge and the number of particles in the cell.

b. Compute the pairwise force from the remaining particles on k individually.
c. Compute the new momentum of each particle based on the net force using a finite difference algorithm to solve Newton's law.
4. Boost macroparticles back to lab frame

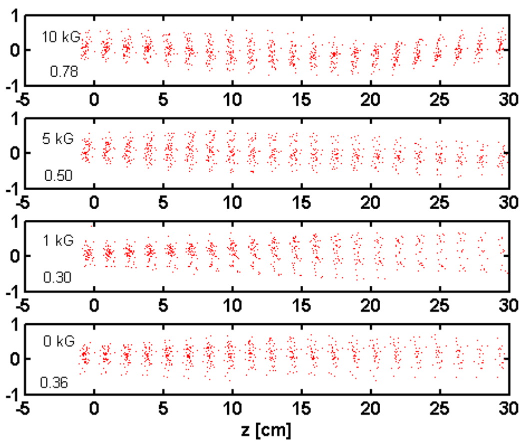

Figure 4: z-x beam profile snapshots for 1 mm initial offset. The solenoid field and fraction transmitted are as indicated.

BBU EXPERIMENT MODELING

Using the BBU-3000 code we have modeled a series of experiments that are planned at the AWA facility. Investigation of the beam dynamics, particularly for transverse instabilities, is an essential requirement for the project of developing practical wakefield devices because of the strong transverse wake forces generated from an offset high current beam.

In modeling the external focusing required to control beam breakup there are a number of considerations in specifying the optimal gradient and profile in a wakefield device. One is that the physical emittance of the bunch is growing as it loses energy so an increasing field profile is indicated. Furthermore, in the case of multiple drive bunches in a train, the energy profile of each bunch can differ along the structure. In the latter case it is important to also maintain a fixed ratio of bunch intensities for the correct functioning of the wakefield device.

Focusing channel design is critical for control of beam losses. A quadrupole FODO channel would be an ideal choice but the lattice period required to obtain BNS-like damping for that channel is nearly prohibitively small.

We investigated the use of a solenoid to control the beam in a dielectric wakefield decelerator/power extractor, having added the capability of including solenoid focusing in the BBU-3000 code. The dielectric structure geometry was taken as 2a=9.068 mm, 2b= 7 mm, L = 300 mm, permittivity=6.64. The beam parameters used in these simulations corresponded to the beam available at the AWA: 20 nC, 2 mm bunch length, $\varepsilon_x = \varepsilon_y = 20$ mm mr (rms).

Fig. 4 shows a comparison of snapshots of the beam spatial distribution for different values of the solenoid field as the beam propagates through the wakefield structure. Note that the beam nearly fills the vacuum channel transversely so that there is very little tolerance for collective deflection of the beam caused by HEM modes. The rather large solenoid field values needed for significant intensity preservation limit the utility of this approach.

SUMMARY

A project to study beam breakup in dielectric structures is underway. The software effort is based on development of the BBU-3000 code. Currently beam dynamics simulation tasks are rather CPU intensive. We are implementing a solution based on the Boinc software environment. Boinc is an established framework (or engine) that we are using to specify beams and simulation parameters, and also beam and EM field visualization, and to allow multiple users and projects to access the BBU-3000 software via the web and perform simulations on a cluster of machines. The ongoing project will result in Boinc providing a "sandbox" for easy addition of new projects on the fly, such as trying new waveguide geometries.

Other new capabilities like more efficient space charge computations and solenoidal focusing have been added. Control of beam breakup via external focusing remains a challenge.

REFERENCES

[1] http://boinc.berkeley.edu/trac/wiki/DesktopGrid
[2] W. Gai, et al., Phys. Rev. E, 55, 3481-3488 (1997)
[3] P. Schoessow et al., Proc.. 2009 Particle Accelerator Conference, paper WE6RFP063
[4] D.C. Rapaport, The Art of Molecular Dynamics Simulation, Cambridge University Press 2004, Chapter 13