ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEPWA059

RF PLASMA-BASED ION SOURCE MODELING ON UNSTRUCTURED MESHES *

S. A. Veitzer†, Tech-X Corporation, Boulder, CO, 80303 USA
K. R.C. Beckwith, Tech-X Corporation, Boulder, CO, 80303 USA
M. Kundrapu, Tech-X Corporation, Boulder, CO, 80303 USA

Abstract

Ion source performance for accelerators and industrial applications can be improved through detailed numerical modeling and simulation. There are a number of technical complexities with developing robust models, including a natural separation of important time scales (rf, electron and ion motion), inclusion of plasma chemistry, and surface effects such as secondary electron emission and sputtering. Due to these computational requirements, it is typically difficult to simulate ion sources with PIC codes.

An alternative is to use fluid-based codes coupled with electromagnetics in order to model ion sources. These types of models can simulate plasma evolution and rf-driven flows while maintaining good performance. We show here recent results on modeling the H- ion source for the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL) using the fluid plasma modeling code USim. We present new meshing capabilities for generating and parallelizing unstructured computational meshes that have increased our parallel code performance and enabled us to model inductively coupled plasmas for long periods of operation.

ION SOURCE MODELING CHALLENGES

Ion sources are used in a large variety of research and industrial applications, including front-end uses for particle accelerators. A large class of sources produce ions through inductive coupling of rf energy with a plasma. For example, the internal antenna H- source currently in use, and the next-generation external antenna source at the Spallation Neutron Source (SNS) are both inductively coupled sources.

Accurate numerical models of ion sources can provide insights into techniques for both predicting source performance, as well as optimizing design features and understanding failure mechanisms and failure mitigation. However, in these kinds of systems there are typically physical processes that are operating on different spatial and temporal scales, which makes developing models that are both accurate and can be executed in a reasonable amount of time difficult. For instance, to explicitly capture the physics of plasma motion, a model must resolve the Debye length of the plasma over device lengths of many tens of cm, and resolve rf frequencies of tens of MHz over many hundreds or thousands of rf periods. In addition, ion motions are thousands of times slower than electron motions due to mass differences. The result is that numerical models require solving coupled equations on large physical domains with a very large number of computational cells, short time steps, and long duration simulations.

Due to the computational requirements, it is typically not possible to execute straight-forward explicit Particle-In-Cell (PIC) models for ion sources. Alternatively, models that ease time step restrictions over PIC models, such as fully implicit PIC models for both fields and particles (e.g. divergence-preserving ADI methods), some electrostatic methods, and fluid models that do not explicitly resolve electron motions, such as various magnetohydrodynamic (MHD) models can give results that are accurate and computationally efficient.

For PIC and fluid models, plasma and particle motions and electromagnetic and hydrodynamic fields are solved for on a computational grid, or mesh, that covers the domain of the simulation. Algorithms to efficiently solve these equations have been developed over the last 50 years and this continues to be an active area of research. Algorithms for solvers differ depending on the kind of mesh that is being used. For instance, on a structured mesh such as a uniform Cartesian mesh, one may employ the Yee algorithm in a Finite Difference Time Domain solver to compute electromagnetic fields to 2^{nd}-order accuracy. However, on an unstructured mesh, where mesh elements are comprised of tetrahedrons and hexahedrons, finite element or finite volume solvers are needed to gain 2^{nd}-order accuracy. Parallel computing, in the form of domain decomposition, is required to be able to model ion sources using any of the types of models described above. This is because in order to resolve the smallest physical processes, models can consist of meshes that have tens of millions of computational cells or more, meaning that the memory required to perform the computations are too large to fit into memory on a single computer, even with multiple cores. Parallel computation on large domain-decomposed meshes can provide sufficient performance in time-domain codes. Coupling distributed parallelism with reduced physical models that do not explicitly follow electron motions make it possible to simulate inductively coupled ion sources accurately while still being computationally efficient enough to perform long timescale simulations in a reasonable amount of wall-clock time.

* This work was performed under the auspices of the Department of Energy, Office of Basic Energy Sciences Award #DE-SC0009585. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.
† veitzer@txcorp.com

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEPWA059

USIM MESHING

We are in the process of developing a fluids-based simulation tool specifically designed for modeling ion sources, based on the simulation code USim [1]. USim is a parallel code for solving a variety of fluids, plasmas, and electromagnetics equations using finite volume and discontinuous Galerkin techniques. USim supports a number of different models including compressible resistive, Hall, two-temperature, gas-dynamic, and multi-fluid magnetohydrodynamics (MHD), Navier Stokes, Maxwell, and electromagnetic multi-fluid models. USim solvers operate on both structured and unstructured meshes, providing flexibility for solving a large variety of different types of problems. USim also has the capability to model radiation transfer and plasma spectral characteristics via PROPACEOS tables, developed by Prism Computational Sciences [2].

USim has the ability to generate structured, body-fitted meshes in 2- and 3-Dimensions, and also supports computational meshes generated by other software. Figure (1) shows an example of an unstructured mesh defining a computational domain for simulation of the SNS H- ion source. The mesh is very fine near the antenna surface, where important physical processes are occurring on small scales, and is more coarse in the volume and on the outer surface of the simulation domain, where plasma flows are smoother. In order to decompose this problem into sub-domains that can be processed in parallel, the mesh structures need to be partitioned into sub-meshes according to the number of cores that are to be used in the simulation. Once a partition has been computed, the simulation is executed in parallel on the sub-meshes.

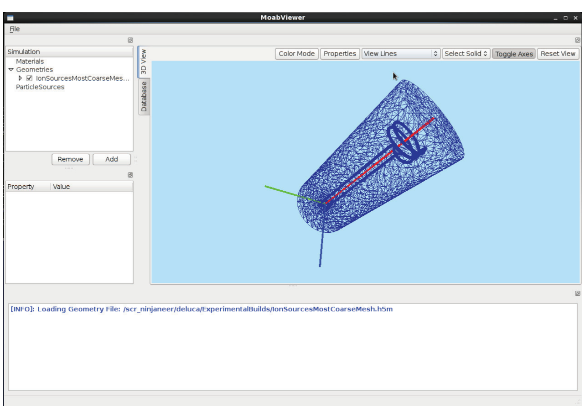

Figure 1: CUBIT-generated mesh of the SNS H- ion source internal antenna design, showing the unstructured mesh.

Generating a mesh partition suitable for parallel computation is difficult, and care must be taken in order to ensure that partitioned meshes are of high quality. If there are inconsistencies or other errors in the partitioning, then the physics solvers may introduce errors due to the mesh that may produce inaccurate or even unphysical solutions. This is especially important where processor boundaries intersect embedded geometry, because field solutions are particularly sensitive to errors and artificial discretizations in boundary conditions.

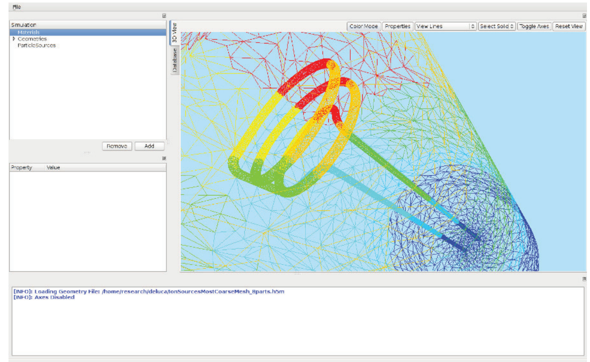

Figure 2: An example of the SNS H- internal antenna source mesh, partitioned for simulations with eight cores.

We have implemented parallel mesh importing and partitioning capabilities into USim, along with a new graphical user interface for controlling the mesh quality and partitioning, based on the Mesh Orientated Database (MOAB) [3] libraries, originally developed at Sandia National Laboratory, and currently maintained and developed by the Sigma team at Argonne National Laboratory [4]. USim engine capabilities have been refactored to utilize MOAB for mesh import, mesh connectivity calculation and mesh geometry calculations. By utilizing MOAB, USim is now able to import mesh files from a variety of commercial vendors including MSc and ANSYS, as well as utilize widely adopted mesh formats such as CGNS, ExodusII and GMSH.

Figure (2) shows an example of a parallel mesh partition, generated by USim using MOAB. In the figure, the mesh elements (cell edges) that lie on the embedded geometrical boundary of the rf antenna, as well as elements on the outer domain are colored according to the processor that they are on. Cells in the simulation volume are not shown in the figure.

Performance testing of MOAB-based mesh capabilities within the USim engine reveal significant improvements in code performance. For single core performance, memory usage has been reduced by a factor of more than 5x compared to previous meshing implementations, while simulation setup for a 3 million element mesh has been reduced to 80 seconds. When running the code in parallel, the total memory usage remains inline with that observed for the single core case, i.e. each parallel process requires an amount of memory given by (Memory Required for Single Core Run)/(# of Parallel Cores).

EXAMPLE: MESHING PERFORMANCE FOR THE SNS INTERNAL ANTENNA ION SOURCE

As an example of parallel partitioning and the effect on computational performance, we have performed scaling tests on mesh generation and partitioning, and on Poisson solvers,

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEPWA059

as implemented in USim. Strong scaling refers to scaling of the time when the total number of cells in a simulation is constant, while the number of computational cores increases. Perfect strong scaling means that the time to solution decreases by a factor of two when the number of cores used increases by a factor of two. Weak scaling refers to scaling of the solution time when the total number of simulation cells per core is constant while the number of cores increases. In this case, the number of cells in a simulation increases. Perfect weak scaling means that the time to solution increases by a factor of two when the number of computational cores increases by a factor of two. These tests were performed on the Cray XE6 cluster at NERSC (Hopper).

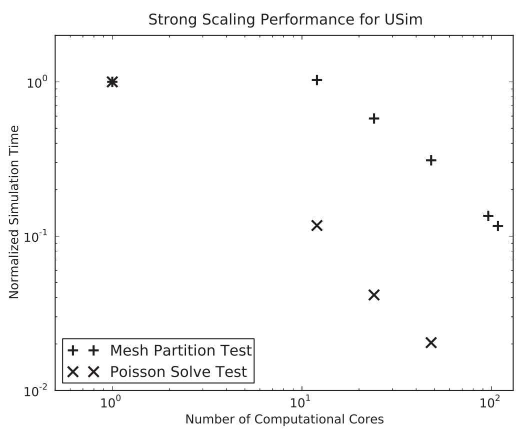

Figure 3: Strong scaling results for USim mesh partitioning and Poisson solves.

Strong scaling results are shown in Fig (3). We have tested two important aspects of parallel computation here: (1) mesh partitioning using the MOAB libraries, and (2) parallel computation, solving Poisson's equation on the parallel partitions. The former aspect is a measure of the ability of USim to decompose large ion source simulation problems into manageable domains for parallel computation, and the latter is a measure of USim's efficiency for performing parallel computation. As can be seen in the figure, there is very good strong scaling both for mesh partitioning and for Poisson solver performance. For mesh generation, there is a serial artifact that degrades the parallel performance compared to serial because the problem size (128x128x128 = 2,097,152 mesh elements) is relatively small, so serial decomposition of the mesh is efficient. However, this effect will not be seen in cases where the number of computational cells is large enough to not fit into memory for a single core.

Weak scaling results are shown in Fig (4). For weak scaling we only test mesh partitioning since the number of computational cells per core is constant in this case. We see very good weak scaling for USim in mesh partitioning for at least up to 8,429,568 cells (48 cores at 56x56x56 cells per core). These results indicate that USim partitioning of meshes for parallel computation is feasible for simulations that require

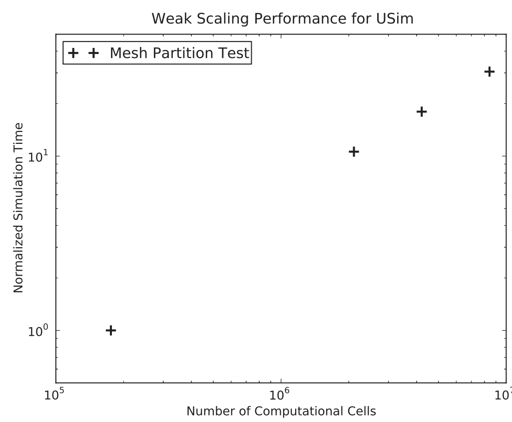

Figure 4: Weak scaling results for USim mesh partitioning.

a large number of computational cells in order to resolve plasma motions. This could be the case for sources that are larger, have more dense plasmas, or fine-scale geometric structures.

REFERENCES

[1] J. Loverich, A. Hakim, "Two-Dimensional Modeling of Ideal Merging Plasma Jets", *J. Fusion Energy*, **29**(6), 2010. 10.1007/s10894-010-9321-z.

[2] http://www.prism-cs.com/

[3] T.J. Tautges, R. Meyers, K. Merkley, C. Stimpson, C. Ernst, "MOAB: A Mesh-Oriented Database", *Tech. Report*, **SAND2004-1592**, Sandia National Laboratories April, 2004.

[4] http://sigma.mcs.anl.gov/team