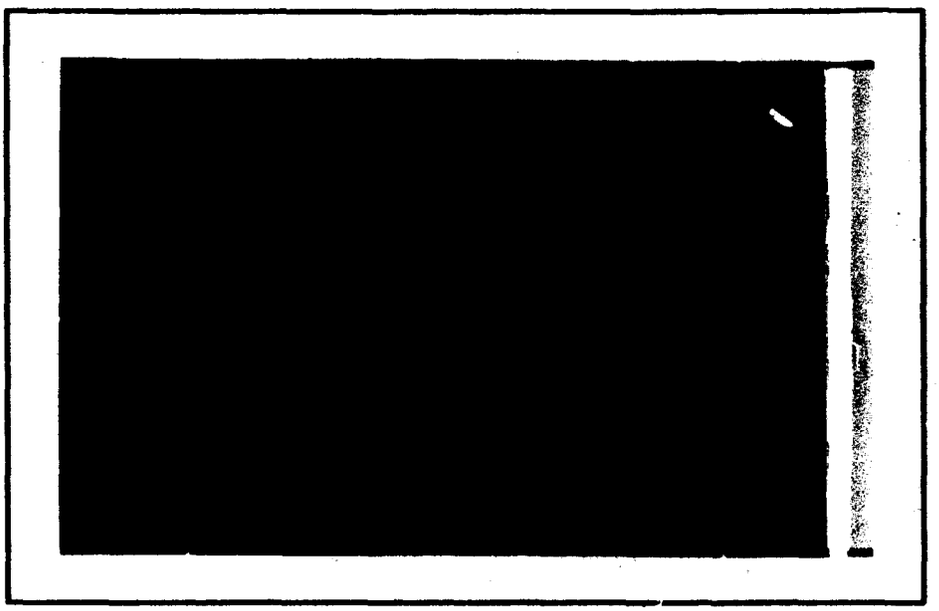
CH9400182 -
CH9400185



C R P P

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE - SUISSE

LRP 494/94 April 1994

Papers presented at the 6th Joint EPS-APS International Conference on Physics Computing

Lugano, Switzerland, August 1994

LIST OF CONTENTS

DAPHNE, A 2D Axisymmetric Electron Gun Simulation Code

T.M. Tran,[a] D.R. Whaley,[a] S. Merazzi,[b,c] R. Gruber,[c]

[a] *CRPP, Association Euratom-Confédération Suisse, Ecole Polytechnique Fédérale de Lausanne, CH-1007 Lausanne, Switzerland*
[b] *DGC, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*
[c] *Centro Svizzero di Calcolo Scientifico, Via Cantonale, Galleria 2, CH-6928 Manno Switzerland*

ABSTRACT

The 2D axisymmetric code DAPHNE has been developed for the design of the magnetron injection guns used in gyrotrons. It is based on the domain decomposition technique and the finite element method to solve the Poisson equation in 2D cylindrical (z,r) geometry. This phase of the calculations is carried out entirely by the existing modules of the ASTRID programming environment. The particle trajectories are then determined by using the standard Particle In Cell (PIC) scheme. Several aspects of the numerical model used in DAPHNE are detailed in this paper.

1. Introduction

The determination of the steady-state electron trajectories can be described as an iterative procedure which involves the following two steps:

1. The electrons are pushed, using the static electric and magnetic fields defined on a spatial mesh as in the standard Particle In Cell (PIC) method[1]. At the end of this step, all of the electron trajectories in the gun are determined.
2. From these trajectories, a charge density, defined on the mesh, can be calculated, using a PIC charge deposition scheme. The Poisson equation can then be solved to yield the electric field. The self-magnetic field from the electron beam is neglected, so that only the external static magnetic field is considered here.

These two steps are repeated until convergence of the electron phase space distribution is obtained. A Laplace electric field is used to start the iterations.

The DAPHNE code is embedded in the ASTRID[2] programming environment, utilizing the existing modules to define the geometry (MiniM module), create the adaptive mesh (Mesh module) and solve the Poisson equation (Solve module). The particle pusher is implemented in a separate module. All of these program modules communicate by means of a data base (see Fig. 1.1) which can then be post-processed by the BASPL graphics program[2]. An example of output of this graphics system is shown in Fig. 1.2. The details of the numerical model employed by DAPHNE are given in section 2 while the comparison with another code is presented in section 3.

2. Numerical Model

2.1. *Geometry and mesh*

The 2D computational domain is subdivided into subdomains in a structured way as shown in Fig. 2.1. Each subdomain is defined by its four faces, each of which is defined by a set of linear segments connected by points. The mesh of quadrilateral cells is then built locally on the subdomain. It is possible to adapt the mesh by specifying a mesh density function. In Fig. 2.2, the electron density was taken as the mesh density function. As we

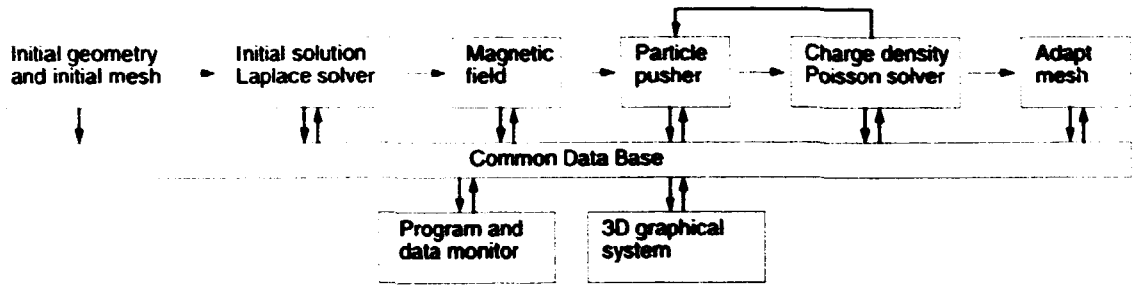


Figure 1.1: Flowchart of the DAPHNE code. Only the Particle Pusher and the Poisson Solver are invoked during the iteration loop.

will see later, it is convenient to map each cell in the 2D cylindrical domain (z,r) into a square in the (u,v) space by the isoparametric transform

$$z(u,v) = \alpha_1 + \alpha_2 u + \alpha_3 v + \alpha_4 uv \tag{2.1a}$$

$$r(u,v) = \beta_1 + \beta_2 u + \beta_3 v + \beta_4 uv \tag{2.1b}$$

with u and v varying from -1 to 1. The relationship between the partial derivatives with respect to the (z,r) coordinates and the ones with respect to (u,v) can be found from the Jacobian J, derived from Eqs. (2.1):

$$J(u,v) = \begin{bmatrix} \partial z/\partial u & \partial r/\partial u \\ \partial z/\partial v & \partial r/\partial v \end{bmatrix} = \begin{bmatrix} \alpha_2 + \alpha_4 v & \beta_2 + \beta_4 v \\ \alpha_3 + \alpha_4 u & \beta_3 + \beta_4 u \end{bmatrix} \tag{2.2}$$

2.2. *Field Computation*

The Poisson equation in the cylindrical coordinates (z,r) can be expressed in the following variational form: find the potential ϕ such that

$$\iint \left(\frac{\partial \phi}{\partial z}\frac{\partial \eta}{\partial z} + \frac{\partial \phi}{\partial r}\frac{\partial \eta}{\partial r} \right) r dr dz = \frac{1}{\epsilon_0} \iint \rho \eta \, r dr dz \tag{2.3}$$

for all test-functions η, and with the potential ϕ satisfying the Dirichlet boundary conditions. The charge density of the beam ρ can be written simply as

$$\rho(r,z) = \sum_p q_p \frac{\delta(r-r_p)}{r}\delta(z-z_p) = -\frac{I\Delta t}{N_b}\sum_p \frac{\delta(r-r_p)}{r}\delta(z-z_p) \tag{2.4}$$

where the sum is over the "particles" p, I is the electron beam current, subdivided into N_b beamlets (or trajectories) and Δt is the time step used in the particle pusher. The contributions from each cell to the variational form Eq. (2.3) can be computed, using the transformation defined in Eqs. (2.1) and (2.2). Then choosing the test-functions $\eta(u,v)$ as bilinear finite elements[3] and expanding the potential $\phi(u,v)$ in the the same finite elements, we obtain a linear system of equations for the values of ϕ defined on the nodes of the mesh. The assembling of the matrix for the linear system together with its solution are dealt with by the ASTRID module SOLVE, using a direct LU factorization method. The construction of the right-hand-side of Eq. (2.3) is done in the particle pusher module as described later in this paper. Once the values of ϕ on the nodes of each cell are obtained, it is possible to compute the local electric field components

$$(E_u, E_v) = -(\partial\phi/\partial u, \partial\phi/\partial v) \tag{2.5}$$

which are defined on the edges of the cell, using the prescription that ϕ is piecewise linear in u along the lines $v = \pm 1$ and in v along the lines $u = \pm 1$. Note that in each cell, E_u is constant in u and linear in v while E_v is constant in v and linear in u. This property of the electric field will be used in the interpolation of the field on the particle positions that we need to push the particles.

2.3. *Particle Pusher*

The electron trajectories are traced by advancing in time the two particle space coordinates (z_p, r_p) and its three momenta (p_z, p_r, p_θ), using the relativistic Boris pusher[1]. In order to find the cell where the particle is located, we use a tracking algorithm[4] which consists of computing four geometrical quantities that are functions only of the geometry of the quadrilateral cell and the particle coordinates. Using the transform defined in Eq. (2.1) for the found cell, the particle isoparametric coordinates (u_p, v_p) can be determined. The electric field (E_u, E_v) at the locations of the particle is then obtained by interpolation (as specified in the previous section) on the transformed space. Finally, we transform the interpolated field (E_u, E_v) back to the real space, using the Jacobian J defined in Eq. (2.2) to obtain (E_z, E_r) required to accelerate the particle. We stop the particle pushing when all of the trajectories exit the computational domain.

At this point, the right-hand-side of the Poisson equation (2.3) with ρ defined in Eq. (2.4) can be constructed in the transformed space, using the bilinear function $\eta(u, v)$ and the coordinates (u_p, v_p) of the particle. This construct is strictly equivalent to the standard area-weighting charge deposition[1] on the mesh used in most PIC codes.

3. Comparison with EGUN code

A Magnetron Injection Gun of a gyrotron used for fusion plasma heating is considered for the comparison. The computational domain extending from the emitter to the microwave resonator (Fig. 3.1) is subdivided into 26 subdomains. The time integration step Δt is chosen to be about one sixth of the electron cyclotron period at the maximum magnetic compression and 11 representative electron trajectories are considered. In Fig. 3.2, we compare the steady-state electron distribution in velocity ratio α at the resonator, calculated from DAPHNE and EGUN[5] respectively. In this figure, each electron trajectory is labelled by its position at the emitter. For the case considered, it should be noted that in EGUN, squared meshes were used, implying a much larger number of mesh points than in DAPHNE which is based on an adaptive mesh technique.

Acknowledgements

The authors are grateful to B. Piosczyk from Kernforschungszentrum Karlsruhe for providing the EGUN results. This work was partially supported by the Fonds National Suisse pour la Recherche Scientifique.

References

1. C.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, McGraw-Hill Inc., 1985.
2. E. Bonomi, M. Flück, R. Gruber, R. Herbin, S. Merazzi, T. Richner, V. Schmid, C.T. Tran, "ASTRID: A Programming Environment for Scientific Applications on Parallel Vector Computers" in *Scientific Computing on Supercomputers II*, J.T. Devreese and P.E. van Camp, eds., Plenum Press, NY, 1990.
3. G. Strang, G. Fix, *An Analysis of the Finite Element Method*, 1973.
4. M.S. Milgram, J. Comp. Phys. 84, 134 (1989).
5. W.B. Herrmannsfeldt, *Electron trajectory program*, SLAC Rep. 226, Standford Univ., 1979.

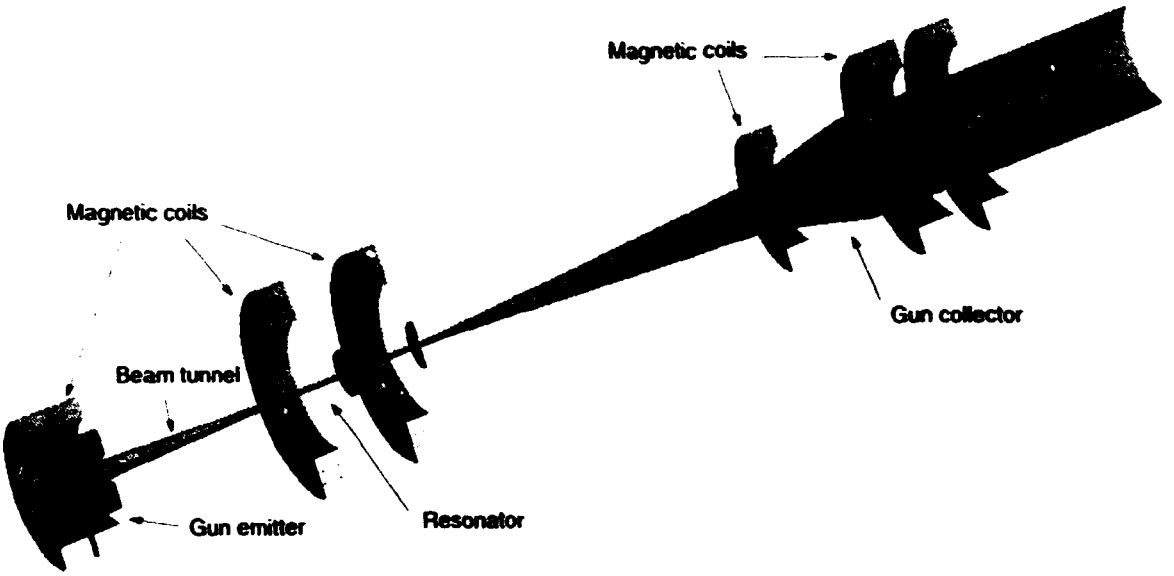


Figure 1.2: A 3D general view of the computational domain of the gyrotron.

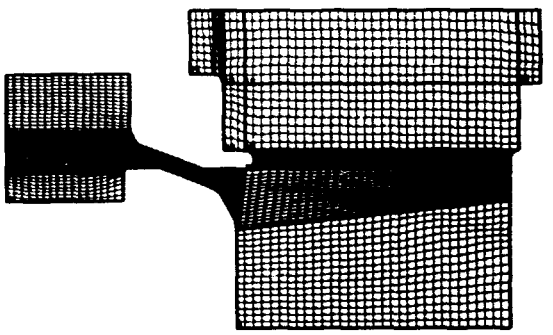


Figure 2.1: The geometry and the domain decomposition of the region near the gun emitter.

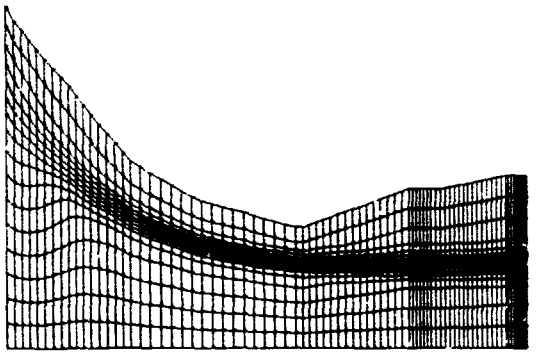


Figure 2.2: The adaptive mesh used in the beam tunnel (amplified vertically by a factor of five).

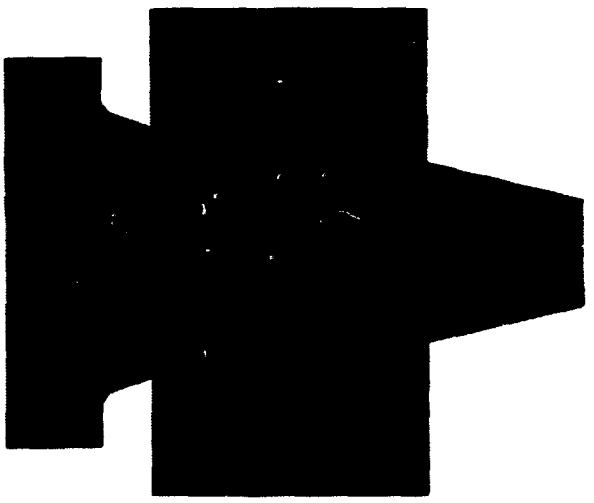


Figure 3.1: A 3D view of the gun considered in the comparison between DAPHNE and EGUN.

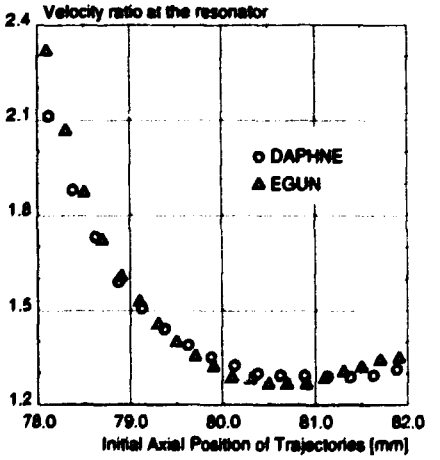


Figure 3.2: The distribution of the velocity ratio α at the resonator versus the trajectory initial position as calculated from DAPHNE (circles) and EGUN (triangles).

PARALLEL SIMULATION OF RADIO-FREQUENCY PLASMA DISCHARGES

M. FIVAZ, B. BÄUMLE*, A. HOWLING, L. RUEGSEGGER, W. SCHWARZENBACH

Centre de Recherches en Physique des Plasmas, Association Euratom-Confédération Suisse, Ecole Polytechnique Fédérale de Lausanne, 21, av. des Bains, CH-1007 Lausanne, Switzerland
Email: Fivaz@crppsun.epfl.ch, bb@ife.ee.ethz.ch

**Electronics Laboratory, Swiss Federal Institute of Technology, 8092 Zürich, Switzerland*

ABSTRACT

The 1D Particle-In-Cell & Monte Carlo collision code XPDP1 is used to model radio-frequency argon plasma discharges. The code runs faster on a single-user parallel system called MUSIC than on a CRAY-YMP. The low cost of the MUSIC system allows a 24-hours-per-day use and the simulation results are available one to two orders of magnitude quicker than with a super computer shared with other users. The parallelization strategy and its implementation are discussed. Very good agreement is found between simulation results and measurements done in an experimental argon discharge.

1. Introduction

Plasma-assisted deposition is widely used in industrial applications such as the production of amorphous silicon for solar cells and other electronic components. In a deposition reactor, a weakly ionized plasma is produced and maintained through energization of electrons by a radio-frequency potential applied between two electrodes. In the stationary state, ions and radicals produced respectively by ionization and dissociation of the neutral gas flow continuously towards the electrodes where they are deposited.
The ion energy distribution at the electrode affects the quality of the deposited layer. Good modeling of the reactor is useful for the design of new reactors and for the optimization of existing ones. It requires a precise description of the kinetic and collisional processes of the reactor plasma. The Particle-In-Cell & Monte-Carlo collision code XPDP1 [1] allows such a modeling in slab (1D) geometry.
The characteristics of the stationary state can be obtained after a long evolution. The time step used in the simulation is limited by the inverse of the bulk electron plasma frequency, and the stationary state is obtained after the order of 10^6 steps. Electron numerical heating due mainly to Monte-Carlo collision processes precludes the use of less than 10^5 particles if the electron velocity distribution is to be obtained precisely. These two facts lead to prohibitive computing time, of the order of 100 hours of CRAY-YMP CPU time for each simulation.
A parallel implementation of XPDP1 code on a low-cost, single-user fast parallel system called MUSIC [2] provides about two simulations per week, allowing an exploration of parameter space unattainable with super computers shared with other users.

2. Simulation Model

We give here a short summary of the simulation model; the reader is referred to ref. [1] for a complete description. A 1D Particle-In-Cell (PIC) model provides a kinetic description of the argon ion and of the electrons. Various collisional effects are taken into account by a Monte-Carlo process which uses a model of collisional cross sections. The plasma is assumed to be homogeneous in two directions, the 1D simulation being performed along the electrode axis.
The neutral argon gas is treated as a uniform background. For each plasma species, i.e. electrons and argon ions, a set of N_p super particles is used as a discretization of phase space. These

particles are characterized by a spatial position X and three velocities Vx, Vy and Vz. The forces acting upon the particles are calculated after the resolution of the Poisson equation on a fixed spatial grid. In summary, one computational cycle involves mainly four computational steps:

1. The charge density produced by all particles is gathered on a fixed spatial grid.
2. The electric potential is computed through a finite difference resolution of discretized Poisson equation on the grid, using the charge density obtained in step one as source. The boundary conditions reflect the electrical characteristics of the electrodes. The electric field at grid points is then obtained by numerical differentiation.
3. The electric field is interpolated from the grid points to the particle positions. The particle velocities and positions are advanced in time using a leap-frog algorithm.
4. The particle velocities are updated according to Monte-Carlo collisional processes. The number of particles increases when neutral ionization occur.

3. Parallel Implementation

3.1 The MUSIC system

The MUSIC system [2], or Multiprocessor System with Intelligent Communication, is a low-cost distributed memory system with up to 63 processors (Motorola DSP-96000) connected through a fast communication ring. The peak performance of a full-size system is 3.8 GFlops. It is programmed in C language using Single Process Multiple Data paradigm. Single precision (32 bits) and extended precision (44 bits) floating points numbers are available. The communication between processors are to be programmed explicitly using a MUSIC-specific C library. The parallel system is connected to a host SPARC station for control and input-output.

3.2 Parallelization strategy

The computational time for steps 1, 3 and 4 defined above is proportional to the number of particles and, in the cases of interest to us, dominates the total computing time. To take advantage of this fact, the particles are distributed approximately evenly among processors. Each processor operates steps 1, 3 and 4 on the set of particles it was attributed. These steps are thus processed locally and in parallel.
Each processor owns a private copy of all grid quantities, e.g. charge density and electric field. After step one, each processor owns a copy of the charge density produced by the particles which are stored in its memory. Between steps one and two, all processors broadcast their private charge density, i.e. communicate it to all other processors. Each processor sums the contributions of all other processors to get the total charge density. The electric potential and the electric field are then solved simultaneously, and therefore redundantly, on all processors. Step 2 is therefore not parallelized.

This parallelization strategy is effective in the simulations of interest to us, where the computing time is dominated by particle operations and where the memory needed to store grid quantities is small enough to be stored on all processors.

3.3 Implementation on the MUSIC system

The scheme allows simple implementation on the MUSIC system. We shall distinguish execution on the host and execution on the parallel system, referred to as the DSP system. The original program requires little code change and is broken in two parts:
- Computational routines. This part is compiled on both host and DSP system, and can be executed on both. The global arrays and variables on which these routines act keep the same name when compiled on host and DSPs, but arrays can have different lengths if they are distributed.
- File and graphics input-output, initialization: this is to be executed on the host only, as in the sequential version.

A MUSIC-specific part is added to the original code. This part takes charge of the transfer of initial values and parameters from host to the DSP system, of a control loop and of periodic communication of diagnostics from the DSP system to the host. After the initialization, the global

parameters, including array lengths, are then transferred to the DSP system. The global arrays are either:
- distributed among processors, in which case a local array length is computed on each processor and the content of the array is transferred accordingly, or
- reproduced on every processor, in which case each processor receives from the host an identical copy of the array content.

A control loop at DSP level controls the execution. It is similar to the control loop of the original program, with an additional call to the communication routine in charge of reconstructing the total charge density, as described above.

The code obtained can be compiled by any standard C compiler and runs on any sequential machine when a preprocessor directive is changed. About 1000 lines of specific coding had to be added to the original program. The computing routines require very little or no recoding. Changes requiring no new global parameters or arrays can be introduced in the code without modification of the MUSIC-specific part.

3.4 Performance

Extensive performance tests were done without using the Monte-Carlo processes, which are well parallelized and do not degrade the performance significantly. Computation time measurements yields a wall clock computing time t_s per step:

$$t_s = 5.2\ 10^{-5}\ n_c + 2.9\ 10^{-5}\ N_p/n_d + 2.9\ 10^{-6}\ n_c * n_d + 1.3\ 10^{-2}\ [s] \qquad (1)$$

Where N_p is the number of particles, n_c the number of grid points and n_d the number of processors. The four terms correspond respectively to field solving, parallel particle operations, communication of processor-private charge density arrays and overhead. In the conditions relevant to our simulations (10^5 to 10^6 particles) the speed-up is close to linear for 20 to 30 processors. The wall clock computing time on a 30 processor MUSIC system is then of 1 microsecond per particle and per step and is lower than the CPU computing time of a CRAY YMP implementation.
In simulations with fewer particles or more grid points, the time used in overhead and in communicating the grid charge density and in overhead is significant and can even dominate the total time. A better speed-up would then require a different parallelization strategy where grid quantities would be distributed among processors.

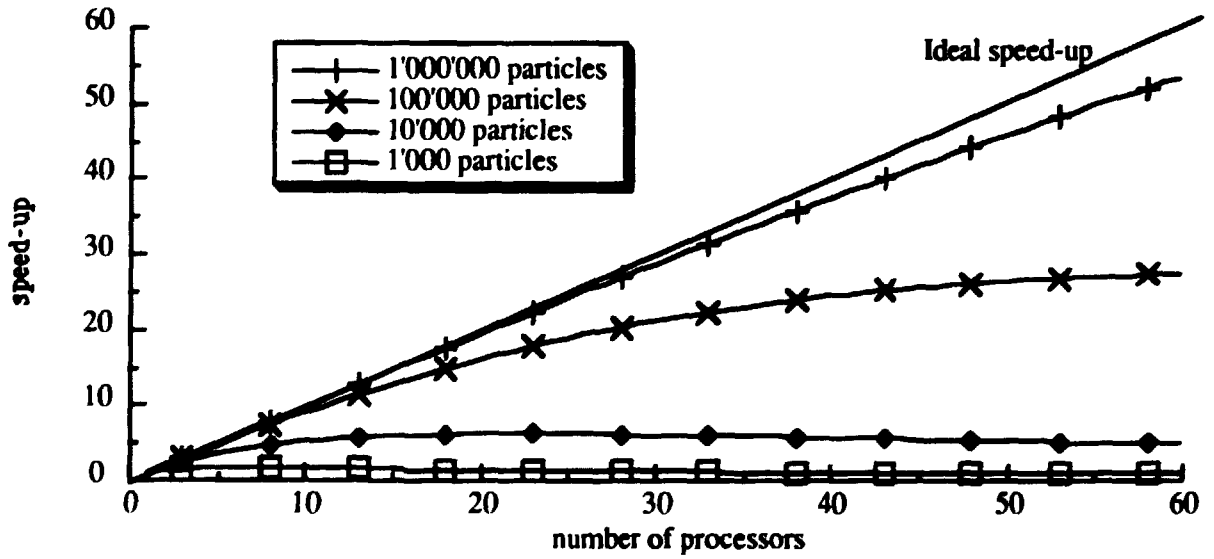


Figure 1: speed-up obtained in the MUSIC implementation, according to equation 1. n_c = 200

4. Results - Comparison with a Laboratory Experiment

Argon discharges are relatively easy to model, as they are free of chemical reactions and because cross sections of collisions involving argon neutrals or argon ions and electrons are available in the literature. Although no argon is deposited at the electrodes, the study of argon discharges can lead to better understanding of the basic physical phenomena occurring in deposition discharges.
Figure 2 shows the maximum energy of ions striking the electrodes as a function of the excitation frequency. The excitation amplitudes are adjusted so as to keep constant the total power deposited in the plasma [3]. Black dots are simulation results while white dots are obtained from mass spectroscopy measurements in a laboratory discharge. The simulations and the experimental measurements show a similar decrease of the maximum energy of ions striking the electrodes when the excitation frequency is increased, thus reducing the damage to the deposited layer from energetic ions. Each simulation took three days of computing time on a 30 processor system. The simulation can then provide relevant information not available in the actual experiment, such as the time-average electric potential or the electron energy distribution.

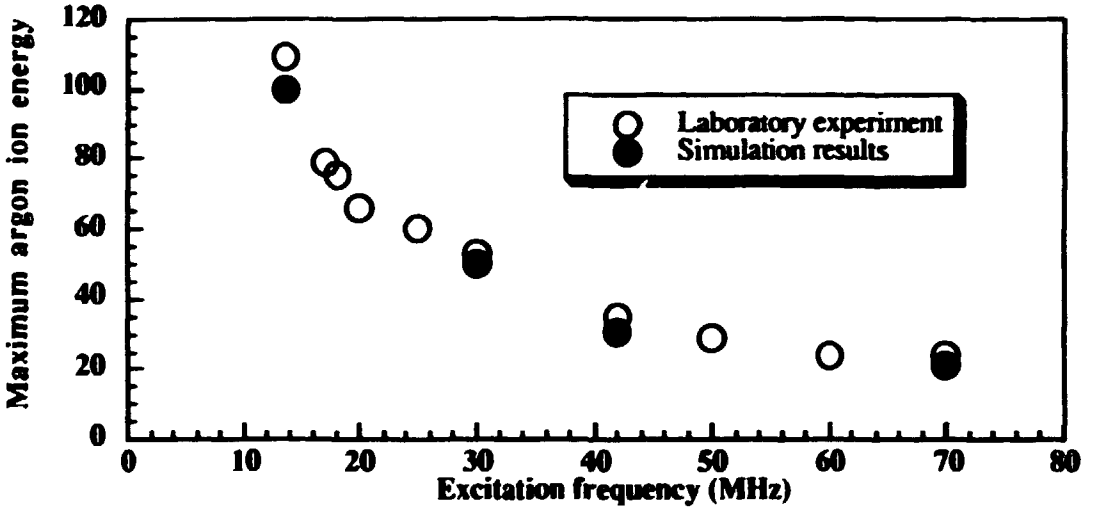


Figure 2: Maximal energy of argon ions striking the electrodes.

5. Conclusions

- The implementation of XPDP1 on the MUSIC parallel system is faster than the implementation on a CRAY YMP. The low cost of the MUSIC system allows a 24-hours-per-day use and the results of simulations are available two orders of magnitude quicker than on a super computer shared with other users. This implementation allows extensive simulations of radio-frequency plasma devices.
- Two man-months of programming effort and 1000 lines of additional code were needed to adapt XPDP1 to the MUSIC system. The program is still compatible with serial machines.
- The simulations performed show very good agreement with experimental data and should contribute to a better understanding of plasma reactor physics.

This work was partly supported by the Swiss National Science Foundation. We thank Prof. C.K. Birdsall for useful discussions and for providing us with the XPDP1 package. We are also grateful to Prof. A. Gunzinger for giving us full time access to a MUSIC system and to the members of the MUSIC group for their support.

References

1. C.K. Birdsall, *Particle-in-Cell Charged-Particle Simulations, Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC*, IEEE transactions on plasma science, 19, 1991
2. A. Gunzinger, U. Müller, W. Scott, B. Bäumle, P. Kohler, W. Guggenbühl, *Architecture and realization of a Multi Signalprocessor System*, Proc. Int. Conf. Application-Specific Array Processors, Berkeley, 1992
3. A.A. Howling, J.-L. Dorier, C. Hollenstein, U. Kroll and F. Finger, J. Vac. Sci. Technol. A, 10, 1080, (1992)

FINITE ELEMENT DISCRETIZATION OF HAMILTONIAN SYSTEMS: APPLICATION TO A DRIVEN PROBLEM WITH A REGULAR SINGULARITY

A. PLETZER
Centre de Recherches en Physique des Plasmas
Association Euratom - Confédération Suisse
Ecole Polytechnique Fédérale de Lausanne
E-mail: alexandre.pletzer@crpp.epfl.ch

ABSTRACT

The numerical advantages of decomposing a driven Sturm-Liouville equation into a symplectic (i.e. Hamiltonian) set of first order equations are investigated. By testing the convergence in energy norm for an exact problem with a regular singular point, it is found that for piecewise linear "tent" elements, the Hamiltonian finite element scheme is up to three orders of magnitude more accurate than the ordinary Galerkin-Ritz method.

1. Hamiltonian Decomposition

Consider the problem of solving the driven Sturm-Liouville equation,

$$Ly(x) \equiv [f(x)y'(x)]' - g(x)y(x) = r_1(x) - r_2'(x) \tag{1}$$

for $y(x)$, subject to boundary conditions $y'(0) + \beta_0 y(0) = \gamma_0$ and $y'(a) + \beta_a y(a) = \gamma_a$ at $x = 0$, and $x = a$ respectively. A Lagrangian density,

$$\mathcal{L} = \tfrac{1}{2} f y'^2 + \tfrac{1}{2} g y^2 + r_1 y + r_2 y', \tag{2}$$

can be constructed such that (1) is the Euler-Lagrange equation,

$$\left(\frac{\partial \mathcal{L}}{\partial y'}\right)' - \frac{\partial \mathcal{L}}{\partial y} = 0, \tag{3}$$

which leaves the action $\mathcal{S} \equiv \int dx \mathcal{L}$ invariant (assuming variations at the endpoint to vanish). It is also well known that one can define a conjugate momentum,

$$\pi \equiv \frac{\partial \mathcal{L}}{\partial y'} = f y' + r_2 \tag{4}$$

to the solution y, as well as a Hamiltonian density,

$$\mathcal{H} \equiv \pi y' - \mathcal{L} = -\tfrac{1}{2} f y'^2 - \tfrac{1}{2} g y^2 - r_1 y, \tag{5}$$

such that (1) can be rewritten,

$$\left.\begin{aligned} -\pi' &= \partial\mathcal{H}/\partial y &&= -gy - r_1 \\ y' &= \partial\mathcal{H}/\partial\pi &&= f^{-1}(\pi - r_2) \end{aligned}\right\} \tag{6}$$

as a system of two first order equations. A more compact but equivalent notation is

$$\mathsf{E}\cdot\mathbf{y}' - \mathsf{H}\cdot\mathbf{y} = \mathbf{s}, \tag{7}$$

where

$$\mathsf{E} \equiv \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{y} \equiv \begin{pmatrix} y \\ \pi \end{pmatrix}, \quad \mathsf{H} \equiv \begin{pmatrix} -g & 0 \\ 0 & f^{-1} \end{pmatrix}, \text{ and } \mathbf{s} \equiv \begin{pmatrix} -r_1 \\ -f^{-1}r_2 \end{pmatrix}. \tag{8}$$

Such a system is called *Hamiltonian*. It is of interest to explore all systems which conserve this property under transformations $\mathbf{y} \to \bar{\mathbf{y}} = \mathsf{U}\cdot\mathbf{y}$. Since the action $\mathcal{S} = \int dx \mathbf{y}^{\mathsf{T}}\cdot(\mathsf{E}\cdot\mathbf{y}' - \mathsf{H}\cdot\mathbf{y} - \mathbf{s})$ remains invariant, $\mathcal{S} = \bar{\mathcal{S}} = \int dx(\mathsf{U}\cdot\mathbf{y})^{\mathsf{T}}\cdot(\mathsf{E}\cdot\mathsf{U}\cdot\mathbf{y}' - [\mathsf{H}\cdot\mathsf{U} - \mathsf{E}\cdot\mathsf{U}']\cdot\mathbf{y} - \mathbf{s})$, one finds that

$$\mathsf{U}^{\mathsf{T}}\cdot\mathsf{E}\cdot\mathsf{U} = \text{const}\mathsf{E}, \tag{9}$$

that is U is a *symplectic* matrix [1]. Transformation (9) is also termed *canonical*. It is readily seen that the particular transformation

$$\mathsf{U} = \begin{pmatrix} 1 & 0 \\ e(x) & 1 \end{pmatrix} \tag{10}$$

keeping the solution y unchanged but transforming π into $\bar{\pi} = \pi + ey$ is symplectic, the corresponding transformed Hamiltonian and source term becoming

$$\mathsf{H} \to \begin{pmatrix} f^{-1}e^2 - \bar{g} & -f^{-1}e \\ -f^{-1}e & f^{-1} \end{pmatrix} \text{ and } \mathbf{s} \to \begin{pmatrix} f^{-1}er_2 - r_1 \\ -f^{-1}r_2 \end{pmatrix} \tag{11}$$

where $\bar{g} = g + e'$. Note that $\mathsf{H} = \mathsf{H}^{\mathsf{T}}$ is symmetric. The Hamiltonian decomposition is the sole decomposition of (1) into a first order system which conserves explicitely the self-adjointness of the system. Equations leading to a Hamiltonian and a source term of the form (11) may be considered as equivalent forms of the initial problem (1).

2. Test Problem

The following test case is taken from a problem arising in studying the stability of resistive ("tearing") magnetohydrodynamic modes in fusion plasmas. It is known that Eq.(1) with $r_1 = r_2 = 0$ approximates the equation governing the plasma displacement field ξ in the vicinity of a rational surface $x = 0$ where $f(x) \sim x^2$ and $g(x) = g_0 + g_1x + g_2x^2 + \cdots$. The singularity at $x = 0$ gives rise to two independent solutions, $\xi^{(b)} = x^{-\frac{1}{2}-\mu}\{\xi_0^{(b)} + \xi_1^{(b)}x + \xi_2^{(b)}x^2 + \cdots\}$ and $\xi^{(s)} = x^{-\frac{1}{2}+\mu}\{\xi_0^{(s)} + \xi_1^{(s)}x + \xi_2^{(s)}x^2 + \cdots\}$, indexed (b) for "big" and (s) for "small", respectively, with

$$\mu \equiv \sqrt{\tfrac{1}{4} + g_0} \tag{12}$$

assumed positive and real. The 'big' solution is always unbounded at $x = 0$ whereas the 'small' solution can either be bounded or unbounded, according to whether $\mu > \frac{1}{2}$ or $\mu < \frac{1}{2}$.

The plasma is unstable against 'tearing' modes if the ratio $\Delta' \equiv \xi_0^{(s)}/\xi_0^{(b)}$ is positive. The following numerical scheme is employed [2,3] to compute

$$\mu\Delta' = W(\check{\xi},\check{\xi}) + \tfrac{1}{2}\int_0^a dx\hat{\xi}(L\hat{\xi}) \tag{13}$$

using energy-like expressions. In (13), $\hat{\xi}$ is an arbitrary, prescribed function that approximates $\xi^{(b)}$ as $x \to 0$ and acts as a source term after applying $-L$ onto it, and $\check{\xi}$ is the solution of small solution behaviour we seek. We extract $\check{\xi}$ by solving the driven equation $L\check{\xi} = -L\hat{\xi}$ and applying natural boundary conditions – which are compatible with the small solution behaviour – at $x \to 0$, and homogeneous Dirichlet boundary conditions at $x = 1$.

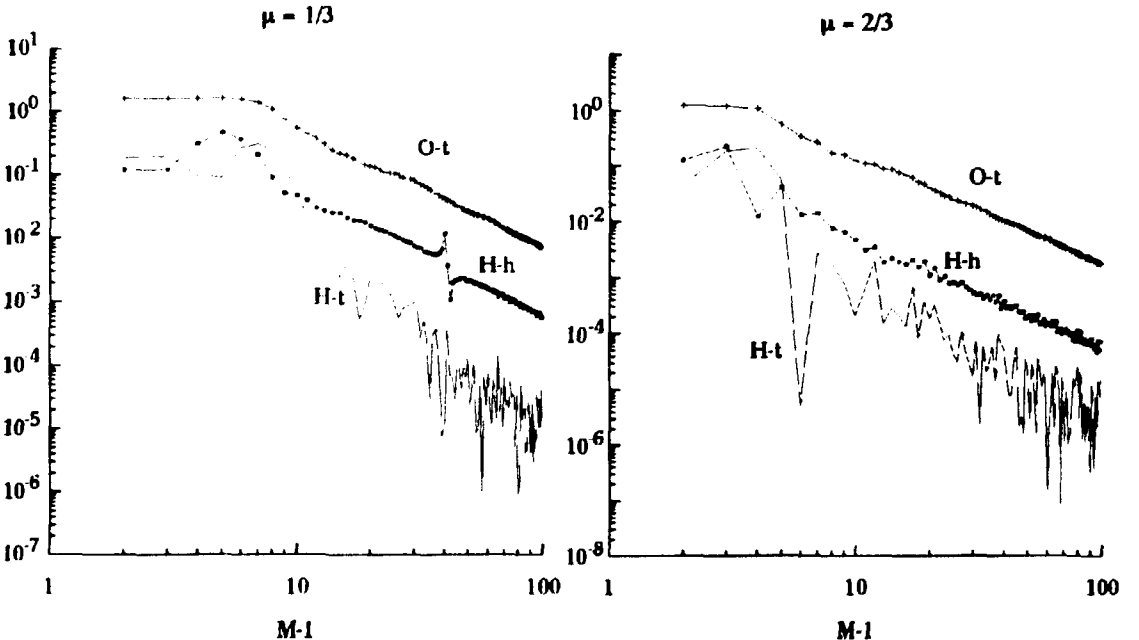


FIG. 1. Convergence of the relative error in absolute value of Δ' versus the number of node intervals for two different cases: $\mu = 1/3$ and $\mu = 2/3$. The mesh is scaled as $1/\mu$ to provide a higher mesh node density near $x = 0$. The curves represent one of the three applied finite element methods: the original problem which uses tent elements (O-t), the Hamitonian scheme using hybrid elements (H-h) and the Hamiltonian scheme using tent elements (H-t).

From (13), it is seen that the error in Δ' arises entirely from computing the "energy functional" $W(\check{\xi},\check{\xi})$,

$$W(u,v) \equiv \tfrac{1}{2}\int_0^a dx\,(u'fv' + ugv) - \tfrac{1}{2}\,[ufv']_0^a\,. \tag{14}$$

A similar expression to (13) can be obtained for the Hamiltonian system, but with a somewhat different form of 'energy functional', namely,

$$W_H(u,v) \equiv \tfrac{1}{2}\int_0^a dx\,(-\pi_u f \pi_v + u'\pi_v + \pi_u v' + ugv) - \tfrac{1}{2}[u\pi_v]_0^a \qquad (15)$$

where π_u and π_v are the congugate variables to u and v, respectively.

An exact test case is constructed by taking $f(x) = x^2$ and $g(x) = g_0 + x^2$. The solutions ($g_0 > -1/4$) are modified Bessel functions. By requiring $\xi^{(s)} + \Delta'\xi^{(b)}$ to vanish at $x = 1$, one obtains

$$\Delta' = \left(\tfrac{1}{2}\right)^{2\mu} \frac{\Gamma(-\mu)I_{-\mu}(1)}{\Gamma(\mu)I_{\mu}(1)}. \qquad (16)$$

Three approaches for computing Δ' are compared. First, a Galerkin-Ritz method based on linear tent elements is used to solve the problem in its original form (this will be referred to as the O-t scheme). Second, the equation is put in Hamiltonian form and the solution and its conjugate are expanded in tent elements (H-t scheme). Third, the solution is expanded in tent elements whereas its conjugate is expanded in piecewise constant elements which are centered on the half integer mesh (H-h scheme). The latter scheme can be regarded as a particular case of the hybrid element method [4], designed to avoid numerical pollution [5] due to the emergence of oscillatory modes generated by the mesh.

We have found that the Hamiltonian decomposition improves by at least one order of magnitude the accuracy achieved by the O-t scheme. One characteristic of the Hamiltonian schemes is that the corresponding W's are no more definite for $f(x)$ and $g(x)$ positive (or both negative), as can be noticed from (15). Thus, the finite element expansion allows the value of the error in W to switch from positive to negative values. This can be observed in curve H-h for the case $\mu = 1/3$. Numerical pollution is thought to be largely responsible for the rough behaviour of the convergence curve for the H-t scheme. It is however remarkable to notice that a polluting scheme can nevertheless lead to higher accuracy in the W norm. There are furthermore evidences that the overall slope of the H-t scheme is steeper that the otherwise observed M^{-2} convergence, as can be seen quite clearly in the case of $\mu = 1/3$, thus showing that pollution has some virtues. It is expected that one could avoid expensive convergence studies and extrapolations to infinite M by using the H-t scheme, which requires a modest number of mesh nodes (about 10) to reach a 1% accuracy, as compared to about 100 mesh nodes when using a O-t method. This point could turn out to be crucial for the resistive stability code PEST [6] when working near the ideal marginal stability point where convergence of the O-t scheme is steep.

1. H. Goldstein, *Classsical Mechanics*, Addisson-Wesley, London, 1980.

2. A.D. Miller and R.L. Dewar, J. Comp. Phys. **66**, 356 (1986).

3. A. Pletzer and R.L. Dewar, J. Plasma Phys. **45**, 427 (1991).

4. K. Appert, D. Berger, R. Gruber and J. Rappaz, J. Comp. Phys. **18**, 284 (1975).

5. X. Llobet, K. Appert, A. Bondeson and J. Vaclavik, Comput. Phys. Commun. **59**, 199 (1990).

6. A. Pletzer, A. Bondeson and R.L. Dewar, *Linear stability of resistive MHD modes: Axisymmetric toroidal computation of the outer matching data*, LRP 487/93, CRPP, 21 Av. des Bains, 1007 Lausanne, Switzerland.