 CMES, vol.20, no.3, pp.203-220, 2007

Recent Evolution of the Simulation Tools for Computer Aided Design of Electron-optical Systems for Powerful Gyrotrons

S. Sabchevski[1], I. Zhelyazkov[2], M. Thumm[3], S. Illy[4], B. Piosczyk[5], T.-M. Tran[6], J.-Ph. Hogge[7] and J. Gr. Pagonakis[8]

Abstract: Computer aided design of powerful gyrotrons for electron cyclotron resonance heating and current drive of fusion plasmas requires adequate physical models and efficient software packages for analysis, comparison and optimization of their electron-optical systems through numerical experiments. In this paper, we present and discuss the current status of the simulation tools available to the researchers involved in the development of multi-megawatt gyrotrons for the ITER project, review some of their recent upgrades and formulate directions for further modifications and improvements. Illustrative examples used represent results from recent numerical investigations of real constructions. Some physical problems that are outside of the capabilities of the existing computer programs and call for development of novel generation of codes are also examined. The ongoing work in this direction as well as the most characteristic features of the codes under development are briefly reviewed too.

Keyword: Gyrotron, CAD modeling, DAPHNE, ESRAY, ARIADNE.

1 Introduction

Computer aided design (CAD) has always been an indispensable tool for development of gyrotrons for various applications, including those for electron cyclotron resonance heating (ECRH) and electron cyclotron current drive (ECCD) of magnetically confined plasmas in thermonuclear fusion reactors such as tokamaks and stellarators. In recent years, a steady progress has been demonstrated in the development of high performance gyrotron tubes that are characterized by record levels of the output power, efficiency and pulse duration [Thumm (1997, 2003a, 2003b), Piosczyk at al. (2003), Denisov, Zapevalov, Litvak, and Myasnikov (2003), Dumbrajs and Nusinovich (2004), Lievin et al. (2005)]. This remarkable breakthrough has been facilitated significantly by the use of efficient computer codes for simulation of various components of the device, notably: electron-optical system (EOS), electrodynamic system (cavity with uptaper), quasi-optical mode converter, depressed collector etc. Among them, the EOS which incorporates a magnetron injection gun (MIG), magnetic coils, and beam tunnel is one of the most complex structures responsible for the formation of high-quality helical electron beams with parameters optimized for a stable and efficient operation [Piosczyk at al. (2006)]. Since the beam quality parameters (such as velocity ratio, velocity spread, azimuthal symmetry of spatial distributions) are crucial for the overall performance of the entire device the development of the EOS represents one of the most important and demanding tasks. This is particularly relevant to the next generation of multi-megawatt gyrotrons with EOS based on coaxial magnetron injection guns (CMIG) that are being currently developed and investigated by IHM-FZK, CRPP-EPFL and Thales. Currently, the targeted parameters of these tubes (output power 2 MW, stable long-pulse and eventually CW operation) represent a challenging task. The basic

[1] Institute of Electronics, Bulgarian Academy of Sciences, BG-1784 Sofia, Bulgaria.
[2] Faculty of Physics, Sofia University, BG-1164 Sofia, Bulgaria.
[3] IHM, FZK, D-76021 Karlsruhe, Germany.
[4] IHM, FZK, D-76021 Karlsruhe, Germany.
[5] IHM, FZK, D-76021 Karlsruhe, Germany.
[6] CRPP, EPFL, CH-1015 Lausanne, Switzerland.
[7] CRPP, EPFL, CH-1015 Lausanne, Switzerland.
[8] CRPP, EPFL, CH-1015 Lausanne, Switzerland.

(initial) design of such advanced gyrotrons has been done utilizing the same computer programs as those used for the previous less powerful (up to 1 MW) of their cousins. Although the results of the proof of principle experiments as a whole agree well with the predictions of the simulations and demonstrate the technical feasibility of the construction, some of the problems that are being encountered on this initial stage of operation and during first attempts for refurbishment of the engineering solutions definitely point out to some essential limitations of the available physical models and simulation codes. This situation motivated several attempts including an international research project for development of improved versions of the available software packages as well as for composition of novel computer programs using both modified legacy codes and new computational modules. The paper is organized as follows. In Sec. 2, we present briefly the current status and recent modifications of the simulation codes used for CAD of powerful gyrotrons. Some illustrative examples from numerical experiments carried out using them are discussed in Sec. 3 in order to point out their more characteristic features and compare their visualization capabilities. The limitations of the present generation of software tools are considered in Sec. 4. Also discussed in this section are several important physical phenomena that must be taken into account in the next generation of models and simulation codes. The requirements and major milestones that are being envisaged in the process of the forthcoming development of the prospective generation of simulation tools are also presented there. Several other important aspects of the gyrotron simulation related to such subsystems as the resonant cavity internal mode converter and launcher, output window, collector etc. will be addressed in a sequel to this paper and published elsewhere.

2 Software packages for simulation of electron-optical systems for gyrotrons

Over the years a great number of researchers and institutions worldwide have been involved in development of physical models and computer codes for simulation of EOS in various free electron devices. Most of them can be classified as 2-1/2 dimensional (i.e. based on a model formulated in a 5 dimensional phase space with 2 spatial coordinates and 3 coordinates in the momentum space) ray-tracing codes also known as trajectory analysis programs. While some of them (e.g. the most famous EGUN program [Hermannsfeldt (1988)]) are universal, other have been developed as problem oriented tools specialized to the specific problems related to the helical electron beams in different gyro-devices with magnetron injection guns (MIG) [Manuilov, Raiskii, and Tsimring (1992), Kuftin, Lygin, Tsimring, and Zapevalov (1992), Kuftin, Lygin, Manuilov, Raisky, Solujanova, and Tsimring (1993), Raiskii and Tsimring (1993), Tran, Whaley, Merazzi, and Gruber (1994), Lygin (1995), Borie, Grüber, and Westermann (1995), Borie, Illy, and Westermann (1997), Sabchevski, Idehara, Ogawa, Glyavin, Mitsudo, Ohashi, and Kobayasi (2000)]. All these codes are based on self-consistent physical models where the dynamics of beam electrons in static external electric and magnetic fields (over which the self magnetic and self electric fields of the space charge flow are superimposed) is governed by relativistic equations of motion. The self-consistent electrostatic field is found as an approximation to which the solution of the boundary value problem (with appropriate boundary conditions) for the electrostatic potential given by the Poisson equation converges in an iterative process. During each iteration the source term for the Poisson equation is inferred from the space charge distribution that corresponds to electron trajectories integrated using the fields obtained on the previous iteration. Recently, the same approach has been extended to 3D in some of the most advanced codes that represent the state-of-the-art in the field [Ives, Bui, Remacle, Shephard, Beall, Kapraun, and Tran (2001), Ives (2005), de Loos and van der Geer (1996), van der Geer and de Loos (2001), Petillo et al. (2005), Nelson, Petillo, and Levush (2005), Petillo, Nelson, DeFord, Dionne, and Levush (2005)].

It should be noted that among the codes which are in constant and steady developments are also the Russian codes EPOSR [Lygin, Manuilov, and

Tsimring (1987)] and EPOS-V [Lygin, (1995)] which have been used successfully for CAD of a series of high-performance gyro devices. The latest developments were thoroughly presented in [Manuilov (2007)]. In this section, however, we present only the current status and main characteristic of the codes available to our research team.

2.1 DAPHNE: a programming environment for optimization of gyrotrons

Most of the simulation packages mentioned above are stand-alone codes that rely on their own (developed ad hoc) computational modules or at most on standard mathematical libraries. Another more advanced approach is to use one or another problem solving environment (PSE), which provides the basic programming infrastructure and state-of-the-art mathematical tools (solvers, mesh generators, means for data manipulation and visualization and so on) for code development. Although developed in the distant late eighties, the DAPHNE code [Gruber, Merazzi, and Tran (1991), Tran, Whaley, Merazzi, and Gruber (1994)] still remains one of the most prominent realizations of such modern programming technology.

The DAPHNE code is embedded in the ASTRID programming environment [Bonomi, Fluck, Gruber, Herbin, Merazzi, Richner, Schmid, and Tran (1990), Bonomi, Fluck, Gruber, Herbin, Richner, Schmid, Tran, and Merazzi (1991), Merazzi, Bonomi, Fluck, Gruber, and Herbin (1990)] which includes a data base management system MEMCOM for memory and data handling, 3D structured adaptive mesh generator, the ASTRID command language, the three-dimensional graphics system for visualization, the ASTRID finite element solver and a few utility programs to interface user-defined modules to the ASTRID. The architecture and flowchart of DAPHNE are presented in Figs. 1 and 2, respectively.

Figure 1 shows that the DAPHNE code consists of two types of modules, namely ASTRID modules and DAPHNE specific modules. This makes the code easily extensible through addition of new components. In one of the earliest realizations of DAPHNE for example a beam collector module (not shown in Fig. 1), which computes the time-dependent heat transfer, has been included. The modules of DAPHNE communicate with each other and with ASTRID modules through the data management system MEMCOM. The initial data for the simulation that describe the geometrical configuration of the EOS, potentials on the electrodes, currents in the magnetic coils and so forth are specified in data sets for the ASTRID module CASE. These data-sets are interpreted by a special conceptive command language. The 2D computational domain is subdivided into sub-domains in a structured way using the MiniM module. The sub-domains are defined by four faces, each of which consists of a set of linear segments. The mesh of quadrilateral cells is then built over the sub-domains by the module MESH. It can be refined adaptively by specifying a mesh density function. Using a density function related to the beam space charge distribution allows one to improve locally the discretization in the most important region, namely the one traversed by the electron beam.

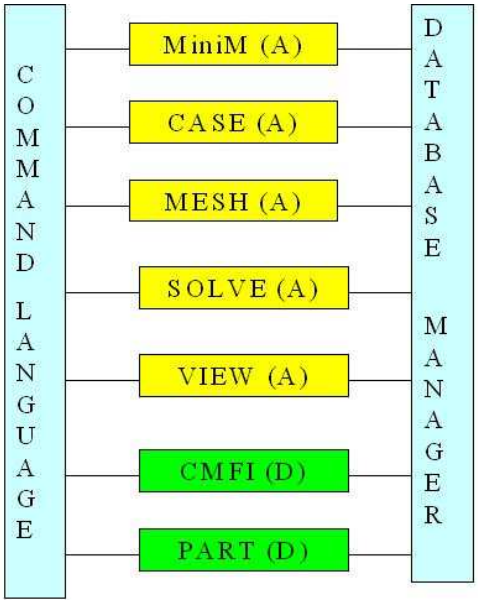


Figure 1: Structure of DAPHNE (modules denoted by A and B belong to ASTRID and DAPHNE, respectively)

The boundary value problem for the Poisson

equation with Dirichlet boundary conditions (set by CASE) is solved by the SOLVE module using the finite element method (FEM). The corresponding linear set of equations is solved using a direct LU factorization method.

In each mesh point, the magnetic field distribution is calculated from the Biot–Savart law by the program CMFI that is in fact a user-defined module.

The electron trajectories in the current electric and magnetic fields (obtained on the previous iteration) and corresponding space charge distribution are calculated in the PART module. For pushing the electrons, the relativistic version of the classic Boris–Buneman algorithm is implemented [Birdsall and Langdon (1985)]. For localization of the electrons inside the mesh cells an efficient tracking algorithm is used [Milgram (1989)]. It implies computing four geometrical quantities that depend only on the geometry of the quadrilateral cell and the particle coordinates.

As shown in Fig. 2, the particle pusher and Poisson solver are invoked in an iterative loop. In standard situations (provided appropriate parameters and fine grids are specified) the process converges in a few iterations. A typical run of DAPHNE on a single node of the PLEIADES cluster at EPFL takes less than 10 s CPU time.

The visualization of the simulation results is performed by the TRAJ2 program, which uses the Xgraphic library. The results of the simulation are stored in a database, which can be accessed by the MEMCOM monitor. The monitor is a command language driven program which offers most of the library functions realized in the MEMCOM database manager and table manager in an interactive environment. It provides means for: (i) inquiring the contents of MEMCOM data files; (ii) modifying, merging and altering the contents of data files; (iii) data conversion and file manipulation.

Recent modifications of the DAPHNE have been initiated by one of its developers, T.-M. Tran, who implemented a module that converts the data base file produced by DAPHNE from a format readable only by the MEMCOM monitor to the more versatile and more widely used HDF5 format [HDF5 (2005)].

One of the limitations of the DAPHNE code originates from the license of MEMCOM, which is a proprietary code. Therefore DAPHNE could not be ported to the computer platforms in all interested institutions involved in CAD of gyrotrons if they do not posses MEMCOM. This situation motivated recent modifications of DAPHNE that are focused on the replacement of this data base management system. The benefits expected are: (i) portability of the DAPHNE code; (ii) usage of the more advanced data storage and manipulation hierarchical data format HDF5; (iii) that modification is considered as mandatory for the planed transition to a 3D parallel version of the code, and, additionally, (iv) newly developed components of FUTILS (a module of Fortran 90 subroutines build on HDF5 and implemented in CRPP-EPFL) can be used in other applications including in prospective novel codes for modeling and simulation of gyrotrons. Additionally, the data base files in HDF5 format are more appropriate for storage, manipulation and organization of simulation results and can be viewed by several advanced visualization and analysis tools like the browsers HDFView, Visan and so on.

In the original version of the DAPHNE code there was no GUI for input of the geometry data and all the data are specified manually in the corresponding files read by the CASE module. For many users this is a laborious and error prone procedure. In order to solve this problem, several alternatives have been considered recently. Finally, a powerful software package, notably Gmsh [Geuzaine and Remacle (2006)], which has been selected as an appropriate tool for pre-, post-processing and visualization of the simulation results of the novel codes under development was recognized as a suitable tool for imputing the coordinates of the points along the boundary contour in DAPHNE as well. The geometry description in the Gmsh format is then converted to one in MiniM format by a simple parser code.

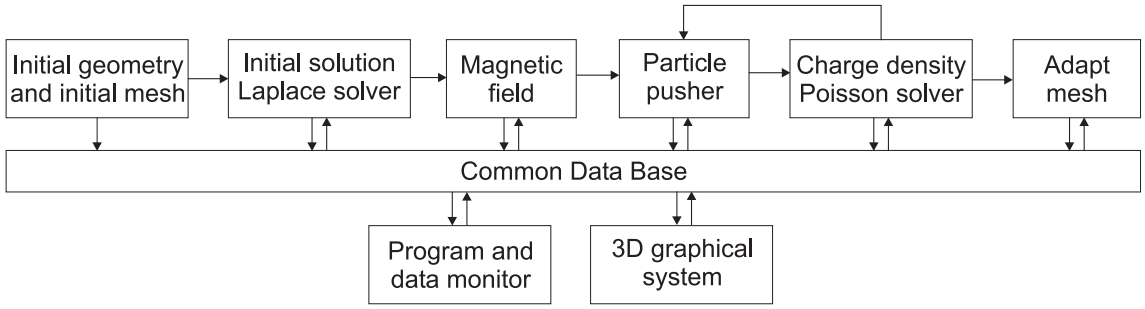


Figure 2: Flowchart of the DAPHNE code

2.2 *ESRAY: a software package for simulation of EOS of gyrotrons*

ESRAY [Illy (1997), Illy and Borie (1999] is the latest and the most advanced computer code from a series of similar simulation tools developed at the Forschungszentrum Karlsruhe (FZK). It makes use of the concepts and approaches implemented previously in the BFCPIC [Westermann (1994)] and BFCRAY [Borie, Grüber, and Westermann (1995), Borie, Illy, and Westermann (1997)] programs, the latter being an adaptation of the BFCPIC to a ray tracing code for studying magnetron injection guns. Moreover the ESRAY code extends these approaches through a modern object-oriented programming implementation in C++ and adds versatile post-processing capabilities based on an informative and easily readable visualization of the results of simulations. The latter is based on the use of its own graphical library built using Tcl (Tool Command Language) and Tk (Toolkit) [Ousterhout (1994)]. In its current form ESRAY is a 2-1/2D ray tracing code, which is characterized by two advantageous features. The first one is its extensibility that stems from the modular structure and object-oriented paradigm. The second distinguishing characteristic of the code is its portability. Recent experiments on several platforms demonstrated that it is in fact platform independent and could be ported easily under Linux and Cygwin (a Unix environment for MS Windows). As in all other two-dimensional ray tracing codes, taking advantage of the axial symmetry of the gyrotron, all calculations are performed in a computational domain, which is a half of an arbitrary meridional cross section. The boundary of the region is determined by the contours of electrodes at various potentials (Dirichlet boundaries) connected by segments (one of which is the axis itself) with the Neumann boundary conditions along them. Then a mesh is generated on the computational domain and an iterative procedure is used to obtain a self-consistent solution for electron trajectories and potential distribution. The most characteristic feature of ESRAY, which distinguishes it favorably from other codes mentioned above, is the utilization of boundary fitted meshes [Westermann (1994)]. The interpolation scheme implemented in it has been taken over from BFCPIC and BFCRAY. This allows one to treat complicated geometries without the inherent errors that are characteristic for approximations based on uniform or non-uniform meshes with grid lines parallel to the coordinate axes. As an example, in Fig. 3 we present such a grid used to simulate a coaxial magnetron injection gun of the EU 170 GHz gyrotron for ITER.

The concept of the structured boundary-fitted coordinates implies the use of a logically rectangular grid and an appropriate mapping $\xi(r,z)$, $\eta(r,z)$ that transforms the physical area in the r–z-plane onto a logical area in (ξ,η) coordinates and vice versa. Thus, one solves the Poisson equation only in the logical grid but advances the particles integrating the equations of motion in the physical space. Therefore, whatever the complex boundary of the physical area is, it is mapped onto a single rectangle in the ξ–η-plane. In this rectangle an equidistant mesh is generated. The corresponding grid system in the physical space is computed

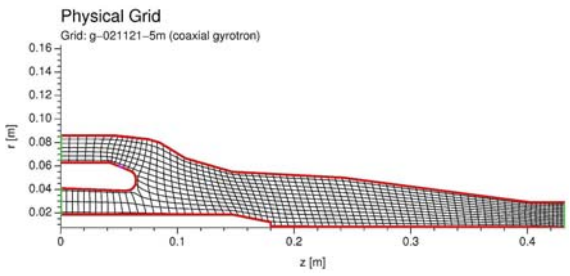


Figure 3: An example for a boundary fitted mesh (CMIG for a 170 GHz coaxial cavity gyrotron)

by solving a set of two elliptic partial differential equations

$$\Delta\xi(r,z) = P, \quad \Delta\eta(r,z) = Q.$$

After that one obtains the inverse map, $x(\xi,\eta)$, $y(\xi,\eta)$. Here, the functions P, Q control the density of the interior lines [Westermann (1994)]. The advantages of such elliptic meshes are that the resulting grids are inherently smooth, overlapping of grid lines is avoided and they can be easily adapted to arbitrary configurations.

The electric field at a particle position is calculated from the fields given at the neighboring mesh points of the corresponding quadrangle using a standard area-weighting method. Instead of a direct searching procedure for particle localization inside the cells a more effective interpolation scheme has been accomplished in ESRAY [Westermann (1994)]. Although the current version of ESRAY is stable and well validated against reference results obtained using other codes (mainly BFCRAY and EGUN) it is in a state of constant improvement focused on increasing its efficiency. Some data from time profiling of the code in a typical numerical experiment for the configuration shown in Fig. 3 are presented in Table 1.

It has already been mentioned that the predecessor of ESRAY, the BFCRAY was developed as a trajectory analysis program from the electrostatic PIC code BFCPIC in order to cope with the limited computing resources at that time. What makes the PIC simulations extremely computing intensive (i.e. requiring enormous amount of CPU time and memory) is the necessity to follow the motion of a great number of particles with extremely small time steps. The upper limit of the time step $\Delta t_{\max}$ is dictated by the gyro-frequency f_c and is given by the condition $\Delta t_{\max} \leq 1/20 f_{c\max}$. For instance, a simulation of the currently developed 170 GHz coaxial cavity gyrotrons requires $\Delta t \leq 3 \times 10^{-13}$ s. This means that for the simulation of the whole system from the CMIG till the collector (having a typical length of 2 m) an electron will require several ten thousand steps to traverse the gun and beam tunnel. The number of particles whose motion is traced must be sufficiently large, so that to insure smooth space charge density distribution over the computational grid and typically ranges to several thousands. Despite such tremendous computing resources both the increase of the computing power of the modern computers and application of efficient and optimized algorithms and codes makes such simulations possible. The latter observation motivated the developer of the ESRAY code S. Illy to write an electrostatic PIC version of the code called ESPIC. Some data from time-profiling that characterize the efficiency of the code are presented in Table 1. It should be noted that the ESPIC requires far more computational time but offers more possibilities and is a solid base for building an effective time-dependent code for simulation of non-stationary processes in the electron beam (including the transitional phenomena).

Table 2 presents a comparison between simulation results obtained using ESRAY and ESPIC. One can see that there is fairly good agreement between them.

2.3 ARIADNE: a 3D simulation package

The first version of ARIADNE code has been developed as a three-dimensional, parallelized, self-consistent, electrostatic trajectory code for simulation of gyrotron beam tunnel and collector assemblies [Pagonakis and Vomvoridis (2004)]. In its current form, however, it is a full-fledged code for trajectory analysis of the entire EOS [Pagonakis and Vomvoridis (2006)].

One of the most characteristic features of the ARIADNE code is that it can be executed on dis-

Table 1: Results from a time-profiling of ESRAY and ESPIC codes

Operation	Code			
	ESRAY		ESPIC	
	Time, s	%	Time, s	%
Initialization	0.170	0.962	0.350	0.003
Input	< 0.0005	0.000	< 0.0005	0.000
Output	9.500	2.830	2.760	0.024
Field solver	2.960	16.752	7315.110	62.384
E-field calculation from potential	0.110	0.623	238.270	2.032
Particle emission	< 0.0005	0.000	2.020	0.017
Particle interpolation	0.950	5.376	707.560	6.034
Particle pusher	3.840	21.732	1183.740	10.095
Particle localization	1.800	10.187	652.610	5.566
Particle killing	2.860	16.186	289.500	2.469
Particle (space charge) densities	1.460	8.263	487.720	4.159
Sum	14.650	82.909	10879.640	92.782
Missing	3.020	17.091	846.330	7.218
Total CPU	17.670	100.000	11725.870	100.000
Total elapsed (wall clock)	17.662	99.955	11728.857	100.25

Table 2: Comparison between simulation results of ESRAY and ESPIC codes

Operation	Simulation Code							
	ESRAY				ESPIC			
	Average	Spread %	Min value	Max value	Average	Spread %	Min value	Max value
Convergence reached on iteration No	11	–	–	–	33890	–	–	–
Number of particles for statistics	72	–	–	–	12	–	–	–
Energy of the electrons, keV	76.701	0.197	76.415	76.965	76.676	0.225	76.433	76.929
Velocity ratio (pitch factor)	1.793	10.168	1.380	2.022	1.793	10.294	1.392	2.017
Relative axial velocity	0.242	8.197	0.219	0.290	0.242	8.307	0.220	0.288
Relative perp. velocity	0.430	2.842	0.400	0.443	0.430	2.908	0.401	0.443
Beam radius, mm	9.621	1.259	9.388	9.842	9.613	1.008	9.472	9.860

tributed, parallel computer systems. The interaction between the program and the user is achieved through the *command line editor, interpreter* and *controller*.

A rich variety of commands is incorporated to make the introduction of three-dimensional geometries possible. The EOS configuration, usually defined along a longitudinal axis (a straight line but not necessarily coinciding with the axis of symmetry) by the description $r = r(z)$ of an axially-symmetric domain can be extended to one with a possible azimuthal dependence, i.e., $r = r(z, \theta)$. This is accomplished by making use of several elementary entities: (i) *curve*, which can be specified in a parametric form by a pair of functions or by transformation of an already defined curve; (ii) *contour*, which represents a transverse cross section of the system for two possible cylindrical configurations with only one (outer contour) or two (inner and outer) in the case of a coaxial structure of the tube; (iii) *segment*, which is generated by connecting several contours of the same type, that correspond to different axial positions (on using this entity one can specify deformed cylindrical or coaxial geometries); (iv) *interface*, which merges two or more adjacent segments of the geometry that can be used to describe complicated domains such as transitions from cylindrical to coaxial regions or abrupt changes of the radial position of the boundary, and (v) *geometry*, which is on the top of the hierarchy of geometrical entities and is a full set of segments and interfaces necessary to describe the entire three-dimensional computational domain. All geometrical entities are stored in the program's database.

The boundary value problem for the Poisson equation is solved by a finite element method (FEM) on a structured tetrahedral mesh. The mesh is generated in two steps. During the first one a background mesh is built in each cylindrical or coaxial segment separately. An internal automatic process implemented in the code resolves the possible incompatibilities between neighboring segments and specifies the parameters of the mesh. Finally, a computational tetrahedral mesh is generated in which each node possesses four indices that determine its position. In such a way a global definition of all elements is ensured and an excessive use of memory for storage of this information is avoided. The electrostatic potential is represented using second order tetrahedral elements, i.e., ten degrees of freedom are associated with four nodes and six midpoints of the edges of each tetrahedron and the corresponding electrostatic field varies linearly inside the elements. Since the sparse matrix to which the linear system for the Poisson equation is reduced is very large (for any realistic problem of practical interest) solving it in parallel is a must. The parallel solver of ARIADNE is based on the domain decomposition method. Therefore, the whole mesh is partitioned in a number of sub-domains that equals the number of available processors. It should be mentioned that the DAPHNE code (see Sec. 2.1) also uses a domain decomposition approach (in order to implement an adaptive mesh technique) but since it is a sequential code all parts of the mesh are treated by a single processor.

The external magnetic field produced by a set of solenoids is calculated as a superposition of the fields of individual windings of the coils given by the Biot–Savart law. Other possible inputs of the magnetic field are by specifying a map of the axial and radial components of the magnetic induction or by introducing functions that describe analytically these components. In any case, however, the magnetic field is specified in its own coordinate system, which may not necessarily coincide with those of the EOS. This allows one to study the influence of misalignment of magnetic system with respect to the optical axis.

A flexible input of the initial data for particle trajectories is implemented which allows one to realize different temperature limited emission models with arbitrary (including non-uniform) distributions of the extracted current and initial velocities. The relativistic equations of motion are integrated by the Runge–Kutta method.

Recent modifications of the ARIADNE code are focused on the improvement of the electric field calculation in the beam region. In particular, a dynamic curvilinear quadrilateral and automatic mesh generator, which can adapt its elements to

the beam shape, is incorporated in the code in order to separate the elements inside the beam and the elements outside the beam. This results in a more accurate calculation of the strong variation of the electric field at the boundary of the electron beam. On the other hand, the space charge of the beam is determined by continuous functions that are defined by the trajectories of two successive macro particles. This technique allows uncharged elements in the beam region to be excluded irrespectively of the density of the mesh.

Another more radical modification of the ARIADNE code, being in progress now, is to rewrite it completely in C++. The new code, ARIADNE++, will benefit from the object-oriented programming paradigm and will be characterized with increased extensibility and portability.

3 Illustrative examples from numerical experiments for the MIG's simulation of gyrotrons

The results of numerical experiments carried out using the DAPHNE code are stored in several output files in ASCII format and a binary database file and are visualized by a graphic system based on Xgraphic library in a set of widgets that plot: (i) the geometry of EOS and trajectories of the electrons in the r–z-plane; (ii) electron orbits in a transverse cross-section, i.e. in (x, y)-coordinates; (iii) emittance diagrams of beam electrons in the phase space of velocities (v_x–v_y-plane); (iv) axial velocities of electrons and magnetic field profile as a function of the axial coordinate z; and (v) analogous longitudinal dependence for the perpendicular velocity of electrons. A typical screen-shot from the trajectory plot is presented in Fig. 4. It is obtained from numerical experiments carried out to investigate the EU 170 GHz coaxial-cavity gyrotron. In some cases such visualization is not sufficiently informative and detailed, so that there is a need to use other plotting routines. Now, after the DAPHNE output is converted to the easily readable and more universal HDF5 format (see Fig. 5) one can extract any aspect of stored data and transfer them to other application. Figure 6 shows a plot of electron trajectories that has been produced exporting data from the data base file to the Gnuplot plotting program.

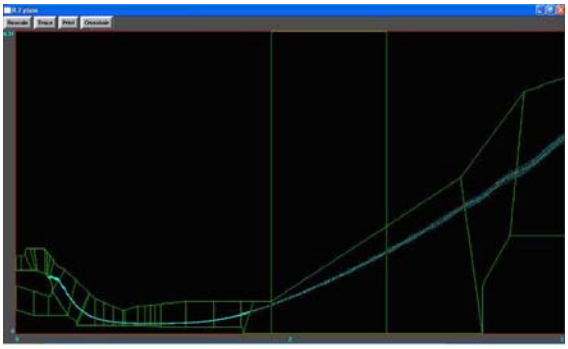


Figure 4: Screen-shot from trajectories plot produced by DAPHNE

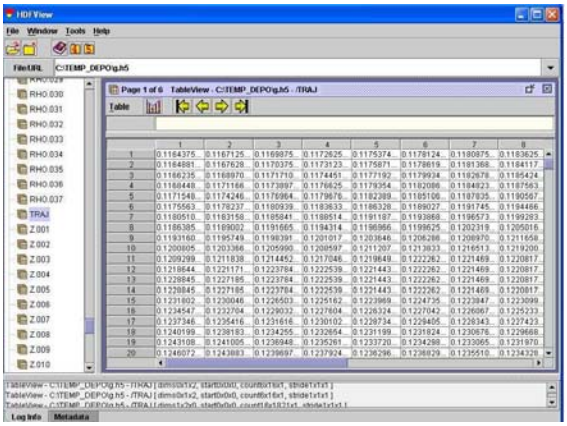


Figure 5: Screen-shot from HDFView showing the structure and content of the data base file in HDF5 format

Also available are MATLAB scripts that can be used for visualization of the simulation results. As an illustration, Fig. 7 shows electron trajectories and equipotential lines in a similar CMIG.

The visualization routines of the ESRAY code produce a successive set of graphical screens that present the simulation results in colorful images and diagrams that can be exported in print quality plots. It should be noted that the visualization could be performed in real time at specified sequence of iterations. Tools for manipulation of the graphical output (such as selective zooming, generation of line diagrams and so forth) are also provided. As an example, Fig. 8 presents the

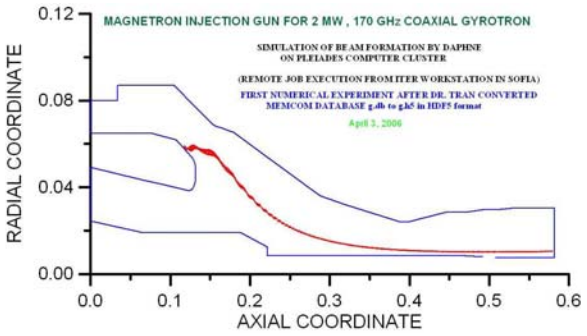


Figure 6: Plot of electron trajectories in a coaxial MIG. All lengths are in meters

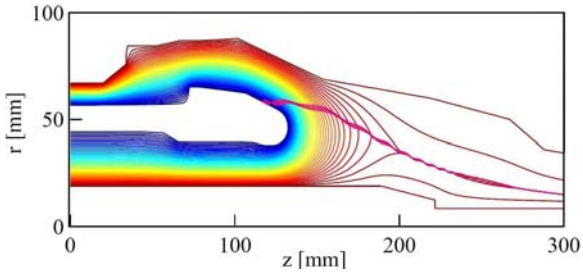


Figure 7: Trajectories and equipotential lines in a coaxial MIG

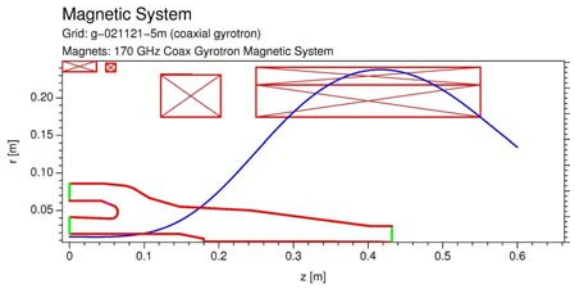


Figure 8: A plot yielded by the graphical output of ESRAY, which shows the CMIG's configuration and the coils of the magnetic system

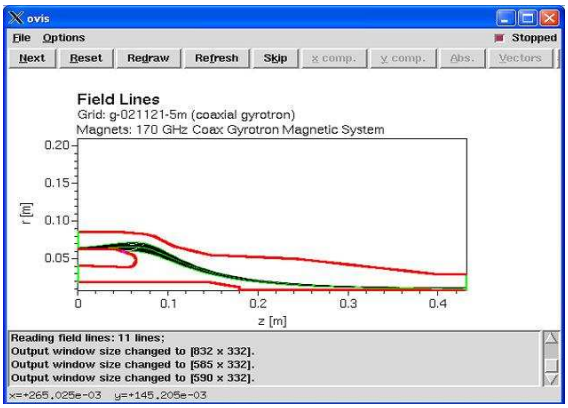


Figure 9: Magnetic field lines produced by the solenoids shown in Fig. 8

configuration of the coaxial MIG of a 170 GHz, 2 MW gyrotron together with the magnetic system, which consists of a set of solenoids in the gun region and main coils of the super-conducting magnet. On that plot the longitudinal magnetic field along the axis is also shown. Magnetic field lines produced by the coils in the beam region are drown in Fig. 9.

The next three plots represent the self-consistent electrostatic field. The map of the electrostatic potential distribution is shown in Fig. 10. The corresponding axial and radial components of the electric field are visualized in Figs. 11 and 12, respectively. Electron trajectories integrated in the self-consistent electromagnetic field obtained after 11 iterations are plotted in Fig. 13.

The next examples illustrate the visualization capabilities of the ARIADNE code. As one can see from Fig. 14, in 2D this code produces plots analogous to those already discussed above. The advantage of ARIADNE is, however, that it can produce images of 3D objects. For this purpose, a program of ARIADNE, Plot3D, written in C++, and the Mesa 3D Graphics Library are used. This is demonstrated in Fig. 15, where the structured tetrahedral mesh used in the simulation is shown. In that case, the mesh consists of 3 542 400 tetrahedra and 610 368 nodes. On a Pentium 4, 1.7 GHz processor, it takes less than 15 s to generate this mesh, which clearly demonstrates the main advantage of the mesh generator—its high speed. Generally, the time complexity of the mesh generator is of the order $O(N)$, where N is the number of mesh points.

Since the ARIADNE code is a newer one, it was necessary to validate it against well-tested and thoroughly benchmarked codes such as DAPHNE

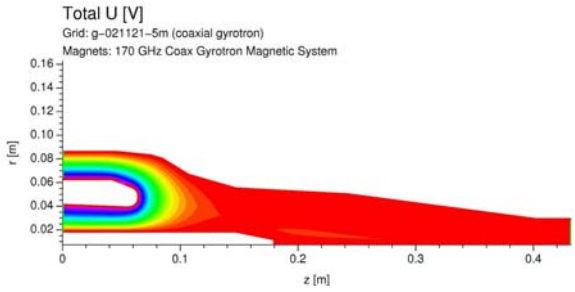


Figure 10: Visualization of the solution of the boundary value problem to the Poisson equation

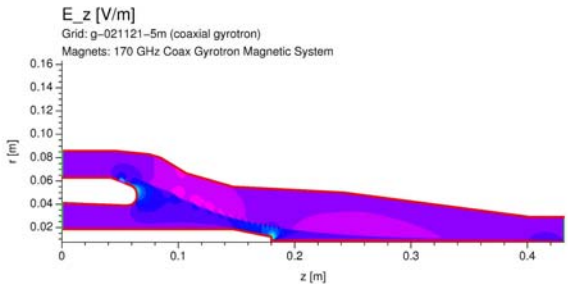


Figure 11: Visualization of the axial electric field of the CMIG

and ESRAY. The results of a series of numerical experiments carried out for different configurations but at the same input parameters (magnetic field at the cathode and resonator, beam voltage, beam current etc.) have demonstrated a fairly good agreement. As an illustration we present here only two examples. The first one compares the results of the simulation of the same coaxial system (of the 170 GHz, 2 MW gyrotron) as in the examples above. The basic beam quality parameters evaluated by ESRAY, EPOSR [Manuilov, Raiskii, and Tsimring (1992)] and ARIADNE are presented in Table 3. There $\beta_\perp$ is the the transverse electron velocity normalized to the speed of light.

In Fig. 16 an analogous comparison between results obtained by using the DAPHNE and ARIADNE codes is presented. In this case, a different EOS is used, notably the one that is based on a conventional MIG designed for a 140 GHz FZK gyrotron. The numerical experiments have been carried out for a beam current of 36 A at accelerating voltage of 78 kV. The agreement between the results is good and the discrepancy is of the order of inevitable numerical fluctuations.

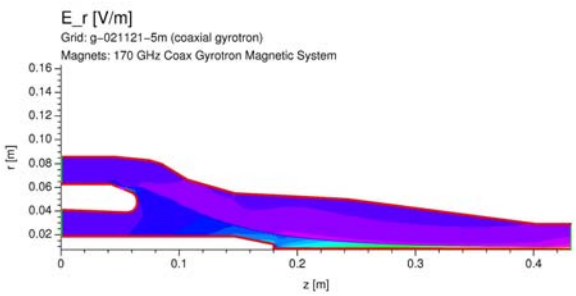


Figure 12: Visualization of the radial electric field of the CMIG

Table 3: Some beam quality parameters obtained using ESRAY, EPOSR and ARIADNE codes

Parameter	Computer Code		
	ESRAY	EPOSR	ARIADNE
Beam radius, mm	10.0	–	10.0
Pitch factor	1.31	1.42	1.37
Average relative perp. velocity, $\beta_\perp$	0.392	0.407	0.388
$\beta_{\perp\,\min}$	0.380	–	0.316
$\beta_{\perp\,\max}$	0.406	–	0.408
Spread of $\beta_\perp$, %	2.1	–	3.28

4 Outlook and future plans

In the preceding sections the current status and some recent modifications of the basic software packages for simulation of EOS for gyrotrons have been reviewed emphasizing briefly their capabilities and advantageous features. Here, on the contrary, we focus on some of their weak points that call for future improvements.

A common drawback of practically all packages is the absence of suitable GUI for preparation and pre-processing of the input data for simulations and for post-processing of the results of numerical experiments. This situation is quite typical for the

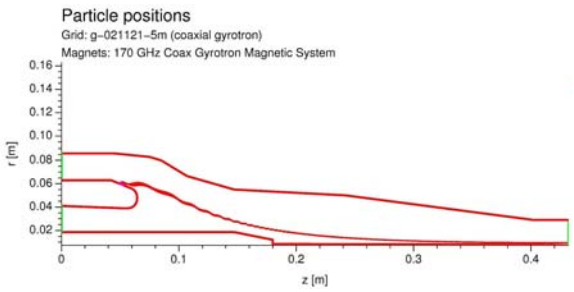


Figure 13: Ray tracing by ESRAY: Electron trajectories in the CMIG for a 170 GHz, 2 MW ITER gyrotron

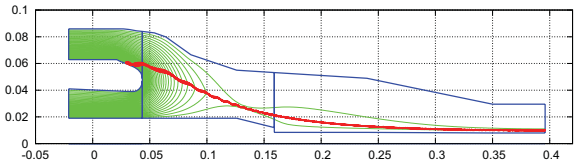


Figure 14: Electron trajectories and equipotential lines in the CMIG for a 170 GHz, 2 MW ITER gyrotron produced by the ARIADNE code. All lengths are in meters

so-called research codes (as differ from commercial ones) and together with the lack of detailed documentation and instructions for preparation of the initial data (especially geometry input) constitutes the main problem for the users. Therefore there is still room for improvements of the available codes supplying them with well documented computer modules for user-program interaction. Obviously ongoing work in this direction will be one of the main tasks. As mentioned above, an appropriate strategy for avoiding such problems in the novel simulation tools that are currently under development [Sabchevski et al. (2006a, 2006b)] is the utilization of advanced problem solving environments that provide pre-, post-processing and visualization capabilities. One such package, which is considered appropriate for the new generation of 3D codes, is the Gmsh [Geuzaine and Remacle (2006)]. It has its own CAD modeler and can be used for: (i) input and visualization of the geometry and simulation parameters through a versatile and convenient GUI; (ii) generation and visualization of finite element meshes; (iii) front-end for invoking different solvers, and (iv) post-processing and visualization of the simulation results. These functions will be used in a novel code, GUN-MIG_3D, which could be considered as a successor of the GUN-MIG/CUSP package [Sabchevski, Idehara, Ogawa, Glyavin, Mitsudo, Ohashi, and Kobayasi (2000)]. Another package combined with Gmsh is the GetDP (**G**eneral **e**nvironment for **t**reatment of **D**iscrete **P**roblems) [Dular and Geuzaine (2006)]. It is a scientific software environment for the numerical solution of integro-differential equations, open to the coupling of physical problems (electromagnetic, thermal, etc.) as well as of numerical methods (finite element method, integral methods and so on). It can deal with such problems of various dimensions (1D, 2D or 3D) and time states (static, transient or harmonic). The main feature of the GetDP is the closeness between its internal structure, the organization of data defining discrete problems and the symbolic mathematical expressions of these problems. Both the function-

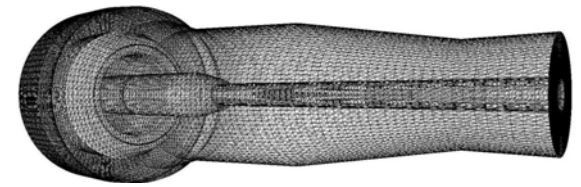
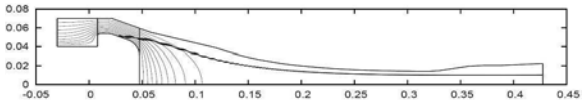


Figure 15: Three-dimensional presentation of the boundary mesh for the electron-optical system of the 170 GHz, 2 MW FZK's gyrotron



Simulation results	DAPHNE	ARIADNE
Beam radius, mm	10.00	10.02
Relative transverse velocity	0.386	0.378
Relative axial velocity	0.290	0.293
Velocity ratio	1.33	1.29
Spread of transv. velocity, %	0.4	0.8

Figure 16: Comparison of simulation results of DAPHNE and ARIADNE codes

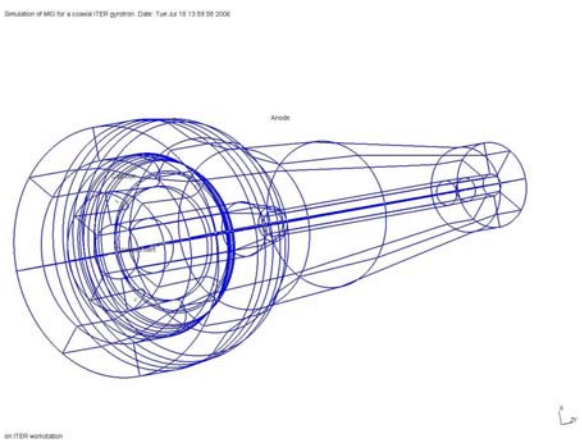


Figure 17: Three-dimensional geometry of a coaxial MIG for an ITER gyrotron specified and edited using the Gmsh package

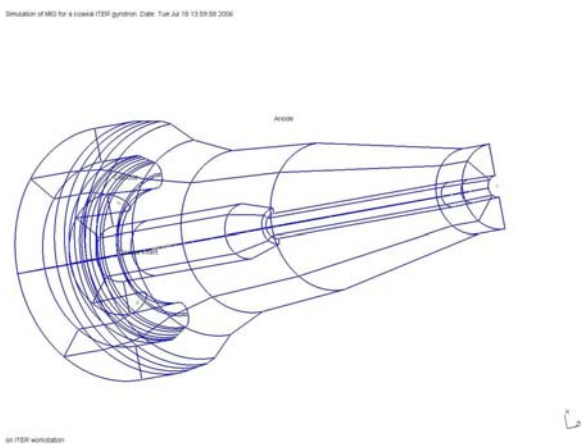


Figure 18: Three-dimensional geometry of a coaxial MIG for an ITER gyrotron specified and edited using the Gmsh package (another angle of view)

ality and the structure of this package are considered as appropriate for the problems related to gyrotron simulation. GetDP is extremely suitable especially for solution of electromagnetic problems since it makes use of an advanced formulation based on differential forms and finite elements (Whitney edge elements of various orders, Nedelec's elements and so forth) tailored specifically for integrating the Maxwell's equations [Dular and Geuzaine (2006), Bossavit (1998)].

As an illustration in Figs. 17 and 18 we present screen-shots that demonstrate the capabilities of the Gmsh to specify and visualize a three dimensional geometry of an EOS for the coaxial cavity gyrotron as well as to generate and plot a tetrahedral finite element mesh. Using the GetDP package the electrostatic problem for the Poisson equation is solved. Figures 19 and 20 show the potential distribution in selected cross sections together with the computational grid (note that the mesh is chosen too coarse in order to be viewed better in these illustrative figures). The development of a module that implements the relativistic Boris–Buneman algorithm for particle tracing on tetrahedral grids is in progress now.

It should be noted that even the most advanced 3D codes for electron gun design mentioned above have limitations inherent to the static nature of their physical models that neglect a great number of non-stationary processes (various instabilities, transient processes, dynamics of reflected electrons and neutralizing ion background and so forth) that affect (as a rule adversely) the quality of the electron beam and the operation of the whole tube. Thus, the analysis of the non-stationary processes accompanying beam formation requires time dependent physical models formulated in three spatial coordinates. The codes built following such an approach usually are different implementations of the particle-in-cell (PIC) method where the electric field is discretizated on a grid (mesh) and the electron beam is represented by an ensemble of particles moving self-consistently on the grid [Hockney and Eastwood (1981), Birdsall and Langdon (1985)]. A well-known code of this type is MAGIC [Smithe and Ludeking (1997)]. MAGIC is an electromagnetic, time-domain PIC code for simulation of plasma physics processes that involve interactions between space charge and electromagnetic fields. Starting from a specified initial state, the code simulates a physical process as it evolves in time. The full set of Maxwell's time-dependent equations is solved to obtain electromagnetic fields. Similarly, the Lorentz force equation is solved to obtain relativistic particle trajectories, and the

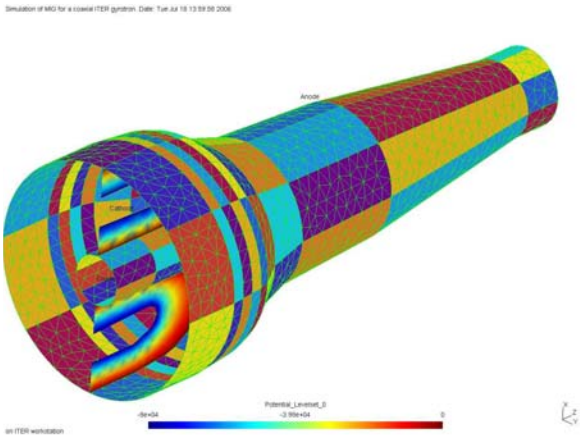


Figure 19: Computational mesh and potential distribution obtained solving the boundary value problem for the Poisson equation in a coaxial MIG for an ITER gyrotron

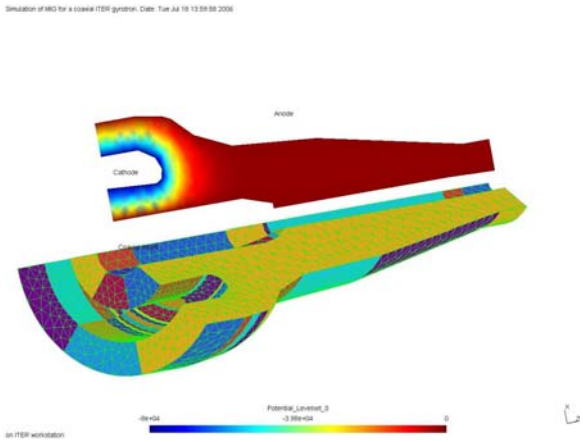


Figure 20: Computational mesh and potential distribution obtained solving the boundary value problem for the Poisson equation in a coaxial MIG for an ITER gyrotron (another angle of view)

solution to the continuity equation provides current and charge densities for Maxwell's equations. Even with such a powerful package it is not possible to perform high-resolution simulation of the entire EOS of the gyrotron. The reason for this is that despite the capabilities of the modern computers, the simulation of beam formation in real structures by 3D models and codes is computationally intensive and requires extremely huge computing resources. Among them the most critical one is the computation time. The estimates show that only the use of supercomputers (or clusters of computers, computing farms and Grid) in a combination with advanced and efficient programming technologies for parallel and/or distributed computing is the way to reduce it to an acceptable level. It is expected that ESRAY, ARIADNE and GUN-MIG/CUSP_3D will finally evolve to similar paradigms that allow transitional regimes of operation (e.g. switch on and off of the beam) and time-dependent processes to be simulated. Therefore it is important to use "building blocks" that are compatible with such prospective future evolution. A great number of different programming environments and libraries have been considered and tested with such task in view. The short list of them includes Cactus [Cactus Code (2006)], PETSc [PETSc (2006)], POOMA [Oldhami (2002)], Chombo [Chombo (2007)], HDF5 [HDF5 (2005)] and MPICH2 [MPICH2 (2007)]. We discuss these libraries and their relevance to the development of the new generation of computer codes for simulation of powerful gyrotrons in more details elsewhere [Sabchevski et al. (2006a), Sabchevski and Zhelyazkov (2007)]. It is expected that on a latter stage of the codes' development the following physical phenomena that affect the quality of the electron beams in powerful gyrotrons could be addressed: (i) various instabilities (e.g. negative mass instability, diochotron instability, global instability of the trapped electrons [Tsimring (2001)]); (ii) time dependent and non-uniform emission arising from various factors (e.g. distribution of the thermal field over the cathode, distribution of the work function of the emitting material, surface roughness, self-magnetic field of the filament); (iii) dynamics of the space-charge compensation; (iv) non-uniform distribution of electric and magnetic fields caused by the departure of the electrodes from axial symmetry and misalignment between the magnetic axis and optical axis; (v) vibrations and thermal expansion of the coaxial insert.

5 Conclusion

While the new generation of computer programs for simulation of gyrotrons is still in a stage of active development, the work focused on improvements and upgrade of the existing 2D programs will continue to exist in parallel to the creation of novel simulation codes and will be the basis for CAD and engineering of novel tools. Obviously the old codes, being fast and not requiring too much computational resources will continue to be useful for routine design work and for analysis and optimization of any EOS under consideration. It is expected, however, that soon new and more sophisticated programs that allow the implementation of more adequate physical models will be available and will be effectively used for more detailed analysis of the operational characteristics of high performance powerful gyrotrons. The most characteristic features of the next generation of codes are the circumstances that they (the codes) are:

- **based on adequate and informative physical models**;
- **portable** (developed on laptops, run on workstations, supercomputers, clusters, Grid);
- **extensible** (possessing flexibility in adding new physics);
- **efficient** (using optimal numerical methods and algorithms and utilizing parallel calculations for minimization of the required computational resources);
- **well validated** (being able to recover the results of the 2-l/2D numerical codes);
- **user friendly** (offering convenient pre- and post-processing and visualization, as well as comprehensive and detailed documentation).

Creation of such codes is a challenging task that requires joint efforts and well-organized collaboration and interaction between program developers and users. We believe that this can be achieved through international research projects in the framework of the European fusion program and through cooperation with other (non European) research institutions involved in the ITER project worldwide.

Acknowledgement: This work has been supported by Contract No FU06-CT-2004-00134 as a Shared Cost Project in the framework of the Euratom Key Action Controlled Thermonuclear Fusion entitled *Numerical Investigations of Selected Problems Associated with the Development of Powerful Gyrotrons for Fusion Research*.

References

Birdsall, C. K.; Langdon, A. B. (1985): *Plasma Physics via Computer Simulation*. McGraw-Hill, New York.

Bonomi, E.; Fluck, M.; Gruber, R.; Herbin, R.; Merazzi, S.; Richner, T.; Schmid, V.; Tran, C. T. (1990): ASTRID: A programming environment for scientific applications on parallel vector computers. In: J. T. Devreese and P. E. van Camp (eds) *Scientific Computing on Supercomputers II*, Plenum Press, New York, pp. 51-82.

Bonomi, E.; Fluck, M.; Gruber, R.; Herbin, R.; Richner, T.; Schmid, V.; Tran, C. T.; Merazzi, S. (1991): ASTRID: A programming environment for physicists. *Int. J. Mod. Phys.* C, vol. 2, pp. 21-29.

Borie, E.; Grüber, C.; Westermann, T. (1995): Calculation of MIG guns for gyrotrons using the BFCPIC code. *Int. J. Electronics*, vol. 78, pp. 789-807.

Borie, E.; Illy, S.; Westermann, T. (1997): Use of the BFCPIC and BFCRAY codes to describe space charge limited emission in electron guns for gyrotrons. *Int. J. Infrared and Millimeter Waves*, vol. 18, pp. 1-22.

Bossavit, A. (1998): *Computational Electromagnetism, Variational Formulations, Complementarity, Edge Elements*, Academic Press, San Diego, CA.

Cactus Code. (1998–2006): http://www.cactuscode.org/.

Chombo: Block-Structured Adaptive Mesh

Refinement Library. (2007): http://seesar.lbl.gov/anag/chombo/, *Version 2.0*.

Denisov, G. G.; Zapevalov, V. E.; Litvak, A. G.; Myasnikov, V. E. (2003): Megawatt gyrotrons for ECR heating and current-drive systems in controlled-fusion facilities. *Radiophysics and Quantum Electronics*, vol. 46, pp. 757-768.

Dular, P.; Geuzaine, Ch. (2006): GetDP: a General Environment for the Treatment of Discrete Problems. *Version 1.2*, http://www.geuz.org/getdp/.

Dumbrajs, O.; Nusinovich, G. S. (2004): Coaxial gyrotrons: Past, present, and future (Review). *IEEE Trans. Plasma Sci.*, vol. 32, pp. 934-946.

van den Geer, S. B.; de Loos, M. J. (2001): The General Particle Trace Code. Design, implementation and application. *PhD thesis*, TU Eindhoven.

Geuzaine, Ch.; Remacle, J.-F. (2006): Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Version 2.0*, http://www.geuz.org/gmesh/.

Gruber, R.; Merazzi, S.; Tran, T. M. (1991): DAPHNE: A programming environment for gyrotron optimization. In: M. R. Siegrist, M. Q. Tran, and T. M. Tran (eds) *Proc. 16th Int. Conf. on Infrared and Millimeter Waves* (August 1991, Lausanne, Switzerland), *SPIE*, vol. 1576, pp. 122-123.

Hermannsfeldt, W. B. (1988): EGUN—An Electron Optics and Gun Design Program. *Technical Report SLAC-0331* (Stanford Linear Accelerator Center, Stanford University).

HDF5 Library. (2005): http://www.hdfgroup.org/, *Version 1.6.5*.

Hockney, R. W.; Eastwood, J. W. (1981): *Computer Simulation Using Particles*. McGraw-Hill, New York.

Illy, S. (1997): Untersuchungen von Strahlinstabilitaten in der Kompressionszone von Gyrotron-Oszillatoren mit Hilfe der kinetischen Theorie und zeitabhaengiger Particle-in-Cell-Simulationen. *Dissertation*, Forschungszentrum Karlsruhe, Institut für Technische Physik, FZKA 6037.

Illy, S.; Borie, E. (1999): Investigation of beam instabilities in gyrotron oscillators using kinetic theory and particle-in-cell simulation. *J. Plasma Phys.*, vol. 62, pp. 95-115.

Ives, L.; Bui, T.; Vogler, W., Remacle, J.-F.; Shephard, M. S.; Beall, M. W.; Kapraun, D. F.; Tran, H. T. (2001): User-friendly, economical, 3D charged particle code with adaptive meshing. In: *28th IEEE Int. Conf. Plasma Sci. and 13th IEEE Int. Pulse Power Conf.* (17–22 June 2001, Las Vegas, NM, USA) (see also http://www.calcreek.com/software.html#boa).

Ives, R. L. (2005): Beam Optics Analysis – an advanced 3D trajectory code. In: *7th International High Density and High Power RF Workshop* (13–17 June 2005, Kalamata, Greece), Poster #7.

Kuftin, A. N.; Lygin, V. K.; Tsimring, Sh. E.; Zapevalov, V. E. (1992): Numerical simulation and experimental study of magnetron injection guns for powerful short wave gyrotrons. *Int. J. Electronics*, vol. 72, pp. 1145-1151.

Kuftin, A. N.; Lygin, V. K.; Manuilov, V. N.; Raysky, B. V.; Solujanova, E. A.; Tsimeing, Sh. E. (1993): Theory of helical electron beams in gyrotrons. *Int. J. Infrared and Millimeter Waves*, vol. 14, pp. 783-816.

Lievin, C. et al. (2005): Development of a 2-MW, CW coaxial gyrotron at 170 GHz for electron-cyclotron-resonance-heating in ITER. In: *6th IEEE International Vacuum Electronics Conference* (20–22 April 2005, Noordwijk, The Netherlands), Proceedings, pp. 21-24.

de Loos, M. J.; van der Geer, S. B. (1996): General Particle Tracer: A new 3D code for accelerator and beam line design. In: *Proc. 5th Eur. Particle Accelerator Conf.* (10–14 June 1996, Sitges, Spain), World Scientific, Singapore, pp. 1241-1243.

Lygin, V. K.; Manuilov, V. N.; Tsimring, Sh. E. (1987): On methods of integral equations and auxiliary sources in the trajectory analysis of intense electron beams. *Elektronika SVCh*, vol. 401, pp. 36-38.

Lygin, V. K. (1995): Numerical simulation of intense helical electron beam with calculation of the

velocity distribution function. *Int. J. Infrared and Millimeter Waves*, vol. 16, pp. 363-376.

Manuilov, V. N. (2007): State of the art of numerical simulation of the helical electron beams with high pitch-factor. In: *8th IEEE International Vacuum Electronics Conference* (15–17 May 2007, Kitakyushu, Japan), Proceedings, pp. 161-162.

Manuilov, V. N.; Raiskii, B. V.; Tsimring, Sh. E. (1992): On calculation of stationary electron beams in a space-charge limited current regime. *Izv. vuzov. Radiofizika*, vol. 35, pp. 846-855.

Merazzi, S.; Bonomi, E.; Fluck, M.; Gruber, R.; Herbin, R. (1990): ASTRID User Manual. *Rapport GASOV No 29*, EPFL.

Milgram, M. S. (1989): Does a point lie inside a polygon? *J. Comp. Phys.*, vol. 84, pp. 134-144.

MPICH2. (2007): http://www-unix.mcs.anl.gov/mpi/mpich/, *Version 1.0.5p4*.

Nelson, E. M.; Petillo, J. J; Levush, B. (2005): 3D self magnetic field calculations in the finite element gun code MICHELLE. In: *6th IEEE International Vacuum Electronics Conference* (20–22 April 2005, Noordwijk, The Netherlands), Proccedings, pp. 95-97.

Oldham, J. D. (2002): *POOMA, A C++ Toolkit for High-Performance Parallel Scientific Computing*. http://www.codesourcery.com/public/pooma/manual.pdf (see also http://www.nongnu.org/freepooma/).

Ousrehout, J. K. (1994): *Tcl and the Tk Toolkit*. Addison-Wesley, Boston, MA.

Pagonakis, J. Gr.; Vomvoridis, J. L. (2004): Evolution of an electron beam with azimuthal density nonuniformity in a cylindrical beam tunnel. *IEEE Trans. Plasma Sci.*, vol. 32, pp. 890-898.

Pagonakis, J. Gr.; Vomvoridis, J. L. (2006): Three-dimensional self-consistent electrostatic simulations of gyrotron beam tunnel assemblies. In: *Proc. 7th Workshop on High Energy Density and High Power RF* (13–17 June 2005, Kalamata, Greece), *AIP Conf. Proceedings*, vol. 807, pp. 271-278.

Petillo, J.; Eppley, K.; Panagos, D.; Nelson, E.; Dionne, N.; DeFord, J.; Held, B.; Chernyakova, L. (2005): The MICHELLE 2D/3D electron gun and collector modeling code: Advances and applications. In: *6th IEEE International Vacuum Electronics Conference* (20–22 April 2005, Noordwijk, The Netherlands), Proceedings, pp. 85-88.

Petillo, J. J.; Nelson, E. M.; DeFord, J. F.; Dionne, N. J.; Levush, B. (2005): Recent developments on the MICHELLE 2-D/3-D electron gun and collector modeling code. *IEEE Trans. Electron Devices*, vol. 52, pp. 742-748.

PETSc (Portable, Extensible Toolkit for Scientific computation). (2006): http://www-unix.mcs.anl.gov/pub/petsc/petsc-as/, *Version 2.4.1*.

Piosczyk, B.; Arnold, A.; Buding, H.; Dammertz, G.; Drumm, O.; Kartikeyan, M. V.; Kuntze, M.; Thumm, M.; Yang, X. (2003): Towards a 2 MW, CW, 170 GHz coaxial gyrotron for ITER. *Fusion Eng. Design*, vol. 66-68, pp. 481-485.

Piosczyk, B. et al. (2006): 170 GHz, 2 MW, CW coaxial cavity gyrotron for ITER – status and experimental results. In: *21st IAEA Fusion Energy Conference* (16–21 October, Chengdu, China), paper IT2-4Ra.

Raiskii, B. V.; Tsimring, Sh. E. (1993): On the design of systems for production of intense electron beams in the space-charge-limited emission regime. *Radiophys. Quantum Electron.*, vol. 36, pp. 735-738.

Sabchevski, S.; Idehara, T.; Ogawa, I.; Glyavin, M., Mitsudo, S.; Ohashi, K.; Kobayasi, H. (2000): Computer simulation of axis-encircling beams generated by an electron gun with a permanent magnet system. *Int. J. Infrared and Millimeter Waves*, vol. 21, pp. 1191-1209.

Sabchevski, S.; Zhelyazkov, I.; Benova, E.; Atanassov, V.; Dankov, P.; Thumm, M.; Dammertz, G.; Piosczyk, B.; Illy, S.; Tran, M. Q.; Alberti, S.; Hogge, J.-Ph. (2006a): Development of new generation software tools for simulation of electron beam formation in novel

high power gyrotrons. *J. Phys.: Conf. Series*, vol. 44, pp. 96-101.

Sabchevski, S.; Zhelyazkov, I.; Benova, E.; Atanassov, V.; Dankov, P.; Thumm, M.; Arnold, A.; Jin, J.; Rzesnicki, T. (2006b): Quasi-optical converters for high-power gyrotrons: a brief review of physical models, numerical methods and computer codes. *J. Phys.: Conf. Series*, vol. 44, pp. 102-109.

Sabchevski, S.; Zhelyazkov, I. (2007): Modelling and simulation of new generation powerful gyrotrons for the fusion research. *J. Phys.: Conf. Series*, vol. 63, paper 012003, doi:10.1088/1742-6596/63/1/012003.

Smithe, D.; Ludeking, L. (1997): An eigenmode solution algorithm based on high-order power iteration with fractally ordered shifts. *Computer Phys. Communications*, vol. 106, pp. 95-104 (see also http://www.mrcwdc.com/magic/description.html).

Thumm, M. (1997): Present developments and status of electron sources for high power gyrotron tubes and free electron masers. *Appl. Surface Sci.*, vol. 111, pp. 106-120.

Thumm, M. (2003a): Progress in gyrotron development. *Fusion Eng. Design*, vol. 66-68, pp. 69-90.

Thumm, M. (2003b): MW gyrotron development for fusion plasma applications. *Plasma Phys. Control. Fusion*, vol. 45, pp. A143-A161.

Tran, T. M.; Whaley, D. R.; Merazzi, S.; Gruber, R. (1994): DAPHNE, a 2D axisymmetric electron gun simulation code. In: *Proc. 6th Joint EPS-APS International Conference on Physics Computing* (22–26 August 1994, Lugano, Switzerland,) pp. 491-494.

Tsimring, Sh. E. (2001): Gyrotron electron beams: Velocity and energy spread and beam instabilities. *Int. J. Infrared and Millimeter Waves*, vol. 22, pp. 1433-1468.

Westermann, T. (1994): Numerical modelling of the stationary Maxwell–Lorentz system in technical devices. *Int. J. of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 7, pp. 43-67.