11

Grid Computing for Fusion Research

Francisco Castejón and Antonio Gómez-Iglesias
Laboratorio Nacional de fusion. CIEMAT
Spain

1. Introduction

Fusion could be an environmentally and socially acceptable source of energy for the future, since it does not emit greenhouse gases or creates long term radioactive wastes. Moreover, fusion reactors will be intrinsically safe, since the losing of the optimum operational parameters will cause the stopping of the fusion reactor.

Nevertheless, fusion still presents several open problems that need to be overcome before the commercial fusion is available. So, fusion research needs to give answers to those open questions and experiments and theory developments must be carried out. The latter require a large computing capacity.

Since the needed temperatures for fusion to occur are of the order of hundred millions of degrees the matter will be in plasma state in fusion reactors. And plasmas are very complex systems to study, especially if they suffer the effect of the confining magnetic field of the device. Fusion research is concentrated on studying the properties of plasmas, which require a large computing effort. The problem is that plasma physics includes in fact almost all the physics since a lot of disciplines are playing a role in understanding plasmas. Plasmas are self-organised and complex systems that exhibit a non-linear behaviour and that require kinetic theory and fluid theory, both including electromagnetic equations, to be understood. So it is possible to find a wide diversity of applications and code to be run.

Specifically, the ITER [1] (International Tokamak Experimental Reactor) project will show a new range of plasma parameters that are outside the present experiments and simulations and will also present some new phenomena that have never been observed. Especially relevant are those related to burning plasmas, i. e., to the plasmas heated by alpha particles that are born in the fusion reactions. But ITER will not be the only device relevant for fusion that will be built in the next future. The large superconductor stellarator Wendelstein 7-X [2] that will be in operation by the end of 2014 in the IPP-Max Planck Institute of Greifswald, a city in the North of Germany, will need also a large computational effort to understand its plasmas. Both supercomputers and computing grids are needed for these activities.

The long term objective in fusion research modelling is to “build” a numerical tokamak and a numerical stellarator, which implies the full knowledge of the relevant physical phenomena that happens in a fusion device, including plasma physics itself, the properties of plasma waves, and plasma-wall interaction (PWI). These numerical fusion reactors could help to save a lot of money since all the scenarios and exploitation regimes that can be expected in fusion reactors can be simulated before doing the experiments. In this way, it would be also possible to teach future fusion reactor operators by the use of numerical simulators. The achievement of this knowledge is a challenging task that needs the understanding of all basic plasma phenomena and of all the physical phenomena that will happen in the reactor. Beyond this knowledge, it is necessary to have enough computation power to describe all these processes that can interact one another. Other important simulation field is the research on fusion materials. The materials needed for fusion reactors are still under discussion and a wide range of properties must be fulfilled. Especial attention must be paid to the effects of the neutron radiation on the material properties.

Hence, the most relevant tasks for the fusion modelling can be summarized as: first of all, it is necessary a full understanding of the physically relevant phenomena that will happen in a reactor, inside the plasma but also in the walls and in all the complex systems that will be installed. Second, the necessary tools for ITER and Wendelstein 7-X exploitation must be developed. Third, the quest for a numerical fusion reactor needs a large effort in the fields of software and hardware development.

With these tasks in mind, all the large scale computing tools available are necessary: computing grids and high performance computers (HPC). The latter have been customarily used for plasma modelling by fusion community from a long time ago, while grids are used in fusion only recently. In fact, grid activities for fusion research started as a pilot experience in 2004, in the frame of EGEE project [3]. After the EGEE projects work, EUFORIA project [4], which bridges the fusion, the grid and HPC communities, appears as a logic prolongation of these activities.

Beyond the use of computing grids and HPCs, it is necessary to establish workflows among applications that can run on heterogeneous computational environments and can deal with different plasma models and phenomena.

The remainder of the paper is organised as follows. In Section 2 a general discussion on fusion applications porting is presented. Section 3 is devoted to the description of the serial applications that were ported in the beginning of the grid-fusion activities and to their scientific production. Section 4 is devoted to show the use of genetic algorithms in the grid for fusion research. The results of porting a Particle-in-Cell (PIC) code are shown in Section 5. Section 6 shows a complex workflow between an application running in a shared memory computer and grid applications. Finally, Conclusions come in Section 7.

2. Fusion on the grid: the strategy

From the beginning of fusion computation on the grid up to now, the range of plasma physics topics that are being investigated by means of grid technologies has been widened, and so have the techniques that have been developed to work on fusion on the grid [5]. The strategy that has been used in order that grid computing is extended in fusion community is to start porting those applications that can be easily gridifyed because of their distributed nature, and still can give physically relevant results. This is what we call “the demonstration effect”: it was necessary to show that grid computing is useful for the fusion community. With this objective, we have identified embarrassingly parallel applications that are composed of a huge number of identical processes, as a first step. The codes that are based on Monte Carlo techniques or on input parameter scans are clearly of this type: both are serial and do not need any communication between the CPUs, so they are perfectly suited to run on the grid. Once ported, the applications were exploited scientifically. After these two types of applications have been ported, others with more complexity have been identified. In these cases, more complicated workflows appear. Remarkably, a PIC code that requires more communication among CPUs has been ported. And, as another important example, an application to optimise the stellarator configuration based upon genetic algorithms has been also implemented.

Beyond considering the extension of the grid use to different types of applications, it is necessary to consider the application of these techniques to different research topics relevant for fusion reactors. This is why it could be necessary to make an especial effort in porting applications to the grid, although they could not be fully appropriated for this distributed architecture. From the core of the reactor to the edge, very different research fields can be identified. The plasma dynamics can be studied both by fluid and kinetic theories. The first ones consist of solving continuity-like and conservation equations in the presence of the three-dimensional background magnetic field, while the second consist of studying the properties of the individual plasma particles. The fluid equations must be solved using finite differences or even finite element techniques, so they are not, in principle, the most appropriated for the grid, while the kinetic approach problems can be easily ported to the grid, as has been done with the application ISDEP [6]. Another kinetic code that estimates collisional transport from a totally different point of view is DKES (Drift Kinetic Equation Solver) code, which solves the drift kinetic equation for the distribution function under several approximations [7].

Plasma heating can be performed experimentally by several methods, but there are two of them that can be modelled by means of grid technologies: electron heating by a microwave beam, which can be simulated by the estimate of a large number of rays as it is performed with the TRUBA code [8], and neutral beam injection (NBI) that can be simulated by means of the Monte Carlo code FAFNER [9]. The plasma wall interaction and the edge transport can be simulated by means of the use of a Monte Carlo code like EIRENE [10], a widely distributed code for plasma-wall interaction studies, or by a PIC code like BIT1 [11].

The Magnetohydrodynamic (MHD) equilibrium and stability are also important disciplines since they study the dynamics of the geometry of the magnetic trap. The main application that estimates the equilibrium is VMEC (Variational Moment Equilibrium Code), which has been also ported to the grid in the frame of the stellarator optimization. Finally, some material research codes should be considered in order to include the simulation of the reactor structure behaviour in the grid simulations. We are thinking of neutron Monte Carlo codes as a first option.

A key development is the building of complex workflows among applications that can run on different architectures including grid and HPCs. Several examples of those workflows are shown.

3. Embarrassingly parallel applications

As has been stated above, the serial applications of embarrassingly parallel nature were chosen as the first ones to be ported. The main of them are described below.

3.1 The ISDEP code

The ISDEP (Integrator of Stochastic Differential Equations for Plasmas) code is used to estimate transport properties of fusion devices by following independent particle trajectories in the plasma, according to the well-known movement equations in the guiding centre approximation. This problem is perfectly suited to the grid, since all the particle trajectories are independent and can be solved separately in the nodes of the computing grid. As a first step we solved these equations in the TJ-II stellarator, which has a very complicated geometry, without collisions [12]. The effects of collisions have been included by adding a stochastic term. Typically 10^7 ions must be followed to obtain representative results and Monte Carlo techniques are used: Particles are distributed randomly, according to the experimental density and ion temperature profiles, considering a Maxwellian distribution function in velocity space. The next step could be to add Langevin Equations for heating and turbulence. An example of orbit in the real 3D Stellarator geometry takes 10 s CPU time in a single processor, totalling 10^7 s for one million of particles. The total distribution function can be obtained at a given position and requires about 1 GB data and 24 h x 512 CPUs. An example of several trajectories, together with the coil structure of TJ-II, is shown in Figure 1. The use of the grid for these calculations has allowed us to obtain the 3D collisional ion fluxes without any approximation on their diffusive nature or on the orbit size of the trajectories. This is an excellent example of application to be run on the grid, since all the ions can be run independently and the accuracy of the results can be increased just by adding more calculated ions, distributed initially according to the density and temperature of the background plasma.

This application has been used to open a new line of research consisting of studying the particle and heat flux onto the vacuum chamber walls. Once the flux structure is known it is possible to develop strategies to minimize the flux and to reduce the possible hot spots in the chamber [13].

More recently, the problem has been converted in a non-linear one by allowing the background plasma to change by the influence of the test particles [14]. The non-linear version of the application elapses about 35 times more CPU time than the linear one, but allows the study of the plasma evolution. This task can be accomplished only due to the grid computing capabilities.

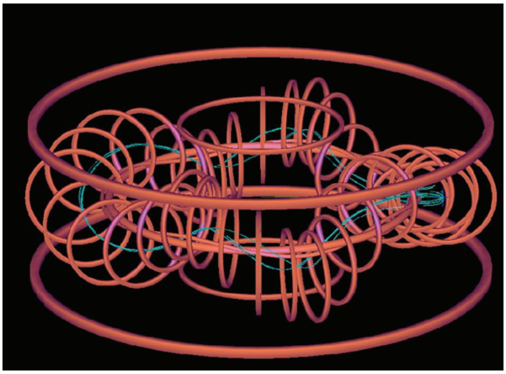


Fig. 1. Ion trajectories and TJ-II

3.2 Microwave heating: the MaRaTra application

The microwave beam for plasma heating can be simulated by a bunch of independent rays with different wave numbers. The trajectories of every ray are estimated by solving their Hamiltonian equations. A single weakly relativistic ray takes about 10 minutes in a 3D plasma confinement device. The microwave beam is simulated by 100-200 rays in the usual situations, where electromagnetic waves are used to heat the plasma. Nevertheless, there is a special case of plasma heating that requires the use of a much larger number of rays in order to simulate the microwave beam behaviour. In these cases about (100-200 rays) x (100-200 wave numbers) ~10^5 rays can be needed to have a good description of the plasma behaviour. This type of waves is known as electron Bernstein waves, which are characterised by being approximately polarised along the propagation direction. The ray tracing TRUBA [8] has been used to simulate the behaviour of this type of waves in a complex system like TJ-II.

The TRUBA code has been ported to the grid using the Gridway metascheduler [15] to perform massive ray tracing. The application that runs on the grid is called MaRaTra (Massive Ray Tracing) [16]. MaRaTra has been used, for instance, to design the hardware system for launching the waves in TJ-II stellarator and for more complicated works. Figure 2 shows examples of ray trajectories performed with MaRaTra.

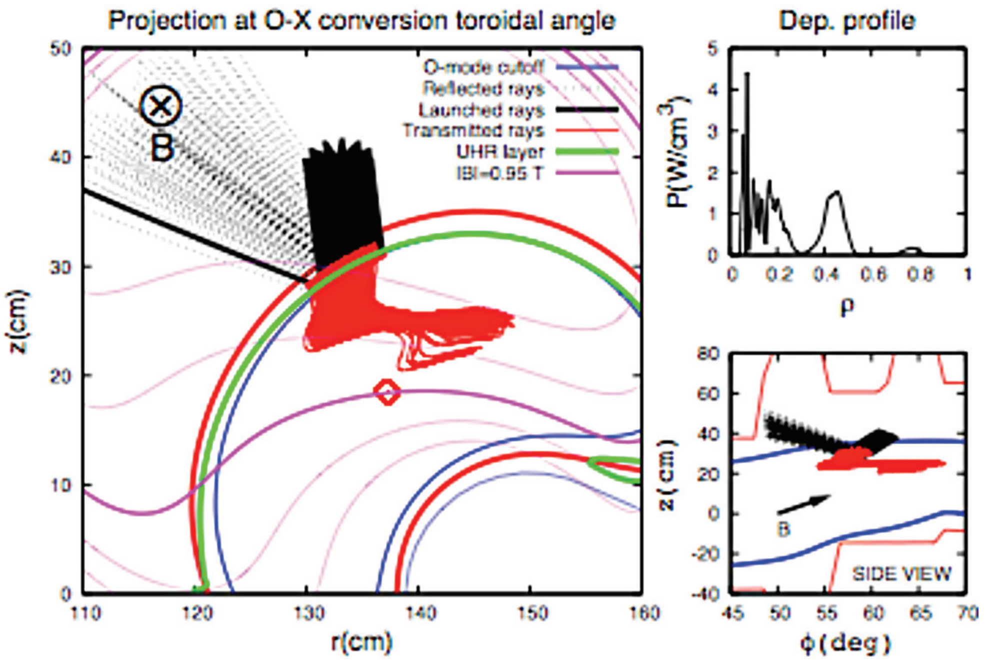


Fig. 2. Ray trajectories in TJ-II plasma (side and top views) and power deposition profile of 100 rays.

3.3 NBI heating: FAFNER2 on the grid

NBI (Neutral Beam Injection) heating system is commonly used in large and medium size plasma fusion devices. This heating system consists of launching energetic neutral particles that can penetrate into the magnetic field device colliding with the plasma species. The collisions will ionize the incoming hot neutrals that will deposit their energy in the plasma.

The properties of this heating system and the different phenomena that are produced inside the plasma must be estimated using a Monte Carlo code that takes into account the cross-sections of all the possible processes. FAFNER2 code is the common tool that is used in the

fusion community to perform this kind of calculations. Every neutral trajectory is estimated in a single CPU of a computing element of the grid and the birth points in the 5D space (3D in real space plus 2D in velocity space) of ions are calculated.
Putting all the results together, it is possible to study the fractions of heat that go to ions and to electrons, the direct losses, and the fraction of neutral particles that go through the plasma without colliding. It will be possible to establish connection between FAFNER2 and other codes described above like ISDEP, the ion kinetic transport code, which will allow us to study the confinement of fast ions. Figure 3 shows the escape points of the fast particles as coming from FAFNER2 calculations after followed their trajectories using ISDEP. We have observed similar advantages in porting FAFNER to the grid as the ones achieved with ISDEP.

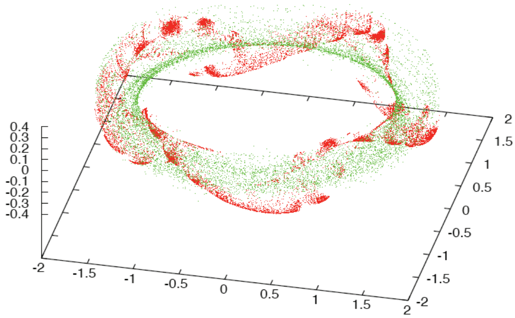


Fig. 3. Escape points of the fast ions as coming from FAFNER2 calculations after following their trajectories using ISDEP.

3.4 Standard neoclassical calculations: the DKES code

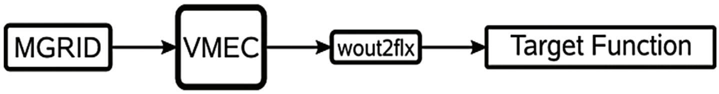
The ion kinetic transport code ISDEP is useful to estimate the ion collisional transport in different magnetic configurations without any assumption either on the diffusive nature of transport, on the energy conservation, or on the typical orbit size. But, presently, the electric field cannot be estimated by the code and must be supplied by experimental measurements. Therefore, it can be necessary to use a standard tool to estimate the transport in a way in which the electric field can be calculated self-consistently. This can be done using the standard neoclassical transport code DKES (Drift Kinetic Equation Solver), which is very common among the stellarator community. DKES estimates the mono-energetic transport coefficients, valid for a single particle of given energy, which must be convoluted with the Maxwellian distribution function of the particles (ions and electrons) in order to obtain the final coefficients for all the plasma species. Every mono-energetic coefficient must be estimated on a single node of the grid. Once a large number of values are obtained for different plasma parameters (namely electric potential and collisionality) the final transport coefficients can be estimated. The density and temperature gradients are ingredients to obtain the fluxes and, from the latter, the electric field can be calculated. In this way, it is possible to predict the electric field in a magnetic confinement device, which could be compared with the experimental results.
The use of the grid for running DKES code allows us to obtain a well calculated monoenergetic coefficients as a function of the input parameters, thus allowing a more precise estimation of the matrix of coefficients.

4. Improving the magnetic configuration by means of metaheuristics. The VMEC code

Although ITER will be a tokamak, the stellarator configuration must be taken into account as an alternative for the fusion reactor. Stellarators are steady state devices and are free from disruptions, which are the main caveats of the tokamak configuration. Nevertheless, the problem is that there is not a unique stellarator configuration that can be proposed as a candidate for the reactor. On the opposite, there exist a lot of different stellarator concepts (different magnetic configurations) available nowadays. The optimization based on the knowledge of stellarator physics is mandatory in order that stellarator community can propose a single candidate for a reactor. The optimization can be performed numerically by variation of the magnetic field parameters. The magnetic configuration, given by the flux surfaces and the magnetic field structure, is described by Fourier series. The strategy is to vary the Fourier coefficients and compute the so obtained "new stellarator" on a separate processor, which takes about 40 minutes, using the customary code VMEC (Variational Momentum Equilibrium Code). The outermost magnetic surface can be described by about 120 Fourier modes. The optimization criteria can be:

- Minimizing Neoclassical Transport and Bootstrap current.
- Equilibrium and plasma stability at high plasma pressure.

Metaheuristics can be used to select the optimum configuration for given target functions. An important point is to carefully select the target function or to include more than one in order to have a stellarator optimized under several criteria. The optimization process was performed in a supercomputer in this case. As a first step, we minimise the drift velocity of the particles, which in principle would imply the improvement of the confinement of the device. The target function that describes such drifts depends on the magnetic field structure and must be estimated using the VMEC output. This one will be the fitness function for all our algorithms. To obtain the magnetic field, it is mandatory to execute the following workflow:



This workflow takes 45 minutes for optimal configurations and 1'5 hours on average when we are performing our optimisation process.

4.1 Tested genetic algorithms

Several genetic algorithms (GA) are being investigated to perform this task. The first GA uses recombination to form new populations. All the population elements are chosen by using a tournament selection of a size of two and the worst one of the pair is randomly crossed with values of the best individual.

In our case, every individual is coded as a vector of floating point numbers and each element is forced to be within the desired range. With crossover operator this can be easily done with the proper initial random generation. Each chromosome represents a VMEC input parameter and an individual is, in fact, a configuration for the device (a single configuration which will be evaluated). The first execution of the genetic algorithm generates a random population of 1,000 individuals where the initial values of every parameter are into some predefined values for each of them.

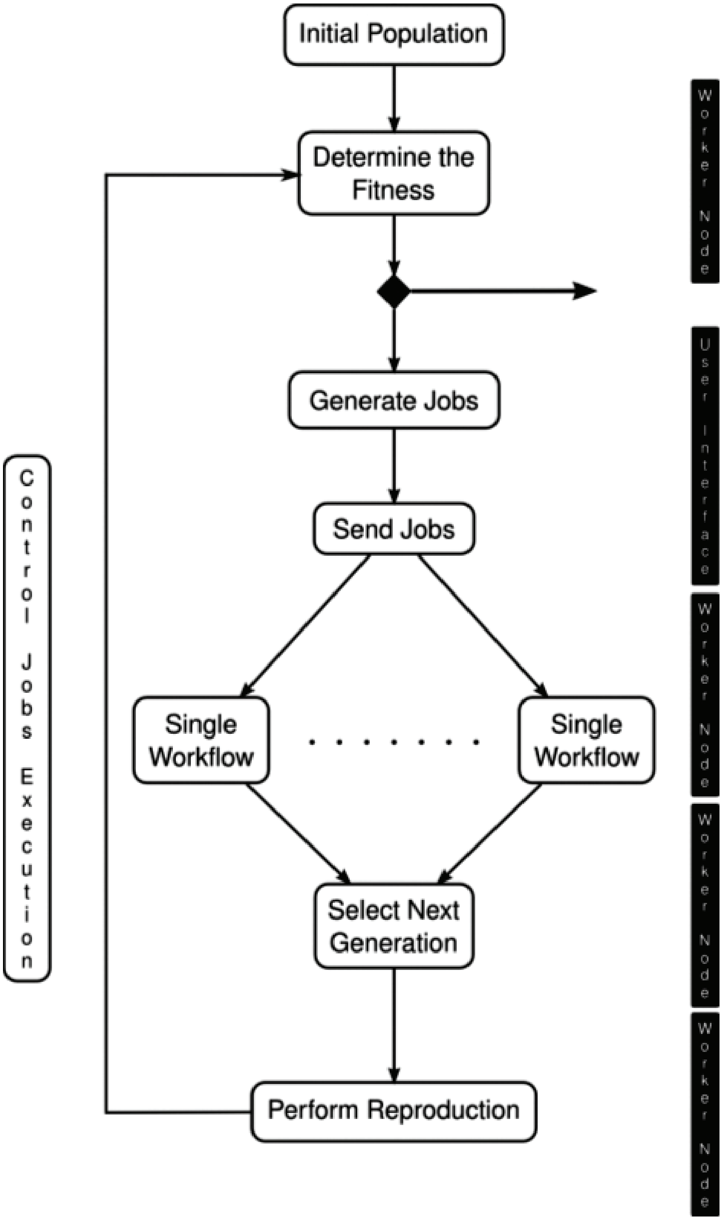


Fig. 4. Overview of a GA on the grid.

Once the equilibrium for the configurations has been obtained, the algorithm selects the different individuals of the population by pairs, using a tournament selection method, and performs a crossover replacing the chromosomes of the worst element, i.e., the one with higher value for the fitness function, and inserting the new element instead of the selected one.

The second technique is the Mutation-based GA, which uses the sample standard deviation of each chromosome in the entire population to perform the mutation. This function assures some level of convergence in the values of chromosomes, even though this is only noticed after a long number of generations, as well as a large diversity of the population. Each selected chromosome is added or subtracted with a value between 0 and the standard deviation for that value not including these extreme values. The mutation using the standard deviation value could be used too, but with that approach the dispersion of the population would become higher. Figure 4 shows a GA running in the grid.
A third algorithm has been tested: the Scatter Search (SS) one. It is a metaheuristic process that has been applied to many different optimisation areas with optimal results. SS works over a set of solutions, combining them to get new ones that improve the original set. But, as main difference with other evolutionary methods, such as GA, SS is not based on large random populations but in strategic selections among small populations: while GAs usually work with populations of hundreds or thousands of individuals, SS uses a small set of different solutions. An overview of this algorithm is shown in Figure 5. Like for GAs, in this case all the control over the execution of the fitness functions, in fact, all the logic needed to carry out this algorithm, is executed in the User Interface, while the fitness function, which is the time demanding process, is executed in the different Worker Nodes of the grid infrastructure.

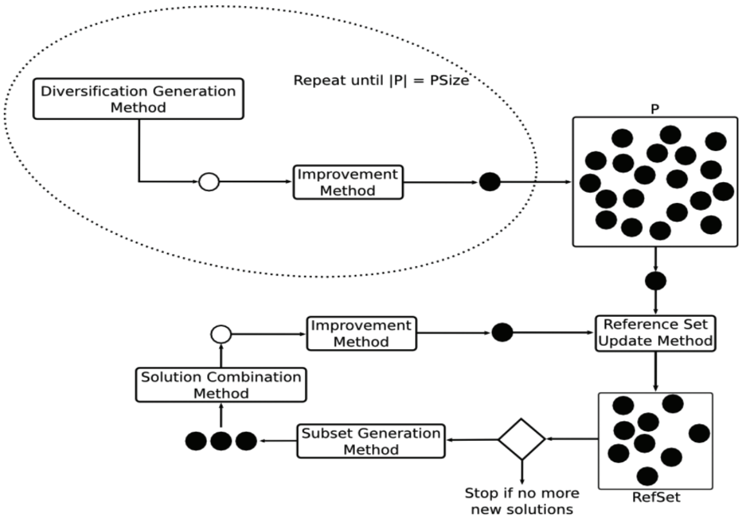


Fig. 5. Overview of Scattter Search algorithm.

The next step will be to include more target functions that can take into account different optimization criteria and not only one, as has been performed in this work. The main advantage of porting this application to the grid is the capability of exploring a wide extension of the parameter space, which happens to be huge for this problem. The usual optimization applications has the disadvantage of exploring a limited zone of the parameter space, so a local optimum is reached, while the parameter exploration performed here allows one to explore a wider parameter range and, probably, to find different locally optimal configurations.

5. Porting of a PIC code for plasma-wall interaction: BIT1

BIT1 is a particle in cell application. PIC simulations are used practically in all branches of laboratory and astrophysical plasma physics. These applications are highly time consuming because of the large number of simulation particles (10^5-10^{10}) and the phase space grid cells (10^2-10^7), which could be needed in the simulation. BIT1 consist of two different parts:

- Traditional PIC module: solver of the Maxwell equations.
- Monte Carlo code simulating collisional plasmas.

The PIC simulation is based on a simple idea: it simulates the motion of each plasma particle and calculates all macro-quantities from the position and velocity of these particles. During a PIC simulation the trajectory of all particles is followed, which requires the solution of the equations of motion for each of them. But the plasma-surface interaction processes cannot be attributed to a classical PIC method, so here is where we need a Monte Carlo code.

This code is important to simulate the interaction of the plasma with some critical points of fusion devices, specially the divertor at the bottom of a vacuum vessel of a fusion reactor: its function is to protect the walls from the strong plasma fluxes and to exhaust the escaping power. Almost all the nonlinear Coulomb collision operators used in PIC codes are based on the binary collision model, where each particle inside a cell is collided with one particle from the same cell.

We have successfully ported to the grid the serial version of this application. Some problems were found during this process:

- The use of X11 libraries.
- The makefile was not working in grid environments.
- Some bugs appeared during our tests.

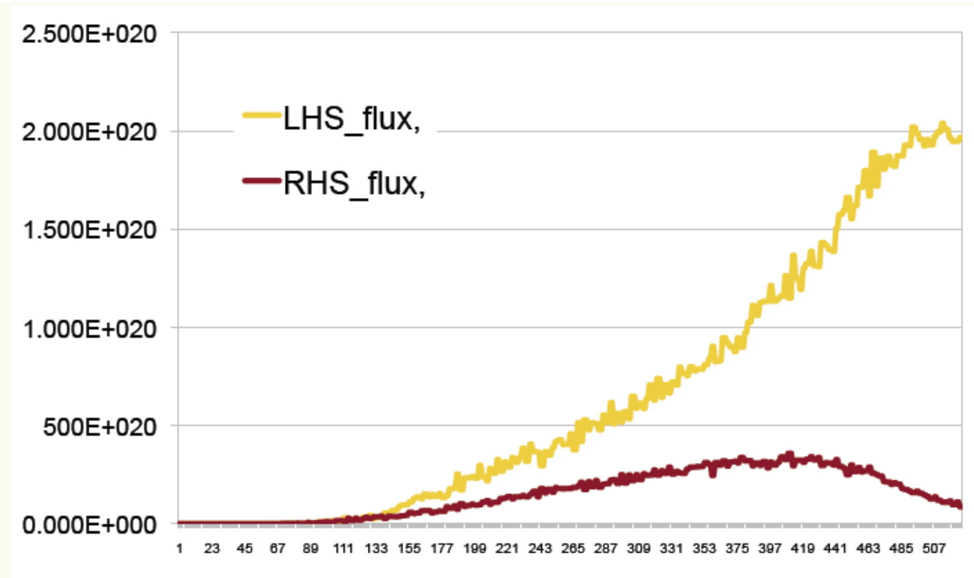


Fig. 6. Time evolution of inwards (LHS) and outwards (RHS) particle fluxes (in arbitrary units) as functions of time, estimated on the grid.

Working together with the code author we have successfully fixed all these bugs, which represents a complete success in the achievements of our work. After fixing these problems, a python script interacting with the grid environment was developed. Thanks to this, users do not have to interact with the grid, take care of proxy problems or job management, since this script performs all these tasks without human supervision.
In order to obtain relevant physical results we decided to carry out a parameter scan of the input file. Several different parameters can be explored like scrape-off layer width or the impurity species that are under consideration. In our case, we have different scrape-off layers and impurity concentrations. 64 jobs were executed with the following computational results:

Average CPU Time per Job	156:23:00
Cumulative CPU Time	10010:32:53

The code provides the time evolution of several quantities, estimated in the inner and outer walls. For example, in Figure 6, we show particle fluxes in a grid simulation. The use of the grid for this application allows to explore a huge number of input parameters, which will provide an extensive study of the plasma-wall interaction depending of those parameters.

6. Complex workflows in fusion

Fusion modelling is characterised by the wide range of applications that work on different fields of physics, which makes that one can have serial, parallel and shared memory applications. This applications work on different plasma regions and use different physical models, so it is not easy at all to include all of them in a single code. Moreover, a large range of time and space scales makes very difficult to include all of them in a single application. In order to simulate complex phenomena that interact one another it is mandatory to communicate applications and to build complex workflows. These ones can be cyclic, linear or more complex and can include applications that run on different infrastructures. In fact, fusion research needs the plug of grid applications with HPC ones. Ideally, both types of large scale computing platforms must be available in order that the suitable architecture can be used for the application to run since both parallel and serial applications are needed in fusion, as has been discussed above. Moreover, the future numerical fusion reactor will need the data interchange between both type of codes. So it is necessary to develop experiences in this direction. Here we show several examples of workflows that have been built up to date.

6.1 FAFNER-ISDEP

This is an example of basic binary workflow. As it has been described above, FAFNER estimates the birth point of fast ions in the 5D phase space (3D in real space plus 2D in velocity space). These birth points can be taken as starting positions for running ISDEP and this exactly what we have done. In this way it is possible to study the confinement properties of fast ions in arbitrary geometries. Specifically, the hit points of these fast ions have been studied, which allows one to estaimate the heat losses on the stellaator vacuum vessel and hence determine if there exist hot points on the vessel. Figure 3 shows the distribution of hit points on the vacuum vessel of the TJ-II stellarator. FAFNER can run both on an HPC with its version that uses MPI or on the grid, while ISDEP is a grid code.

6.2 ASTRA-MaRaTra

A very fruitful and flexible way to build workflows is to take the transport equations of the plasma. For instance, the heat transport equation is given by:

$$\frac{3}{2}n\frac{\partial T}{\partial t}+\nabla\cdot\left(n\chi\nabla T\right)=P_{in}-P_{loss}$$

Here n and T are the plasma density and temperature, P_{in} and P_{loss} are the power input and losses respectively and χ is the heat diffusivity. This equation could be integrated symbolically, giving the temperature evolution, in this way:

$$\Delta T=\frac{2n}{3}\left[P_{in}-P_{loss}-\nabla\cdot\left(n\chi\nabla T\right)\right]\Delta t$$

The right hand side of this equation is can be given by function as complex as ne could imagine that need to be estimated numerically on HPC or on the grid. For instance, this evolution equation can be solved by the transport code ASTRA (Authomatic System for Transport Analysis), where a complex heat diffusivity that must be estimated on an HPC using MPI has been introduced. As a case example we estimate the input power P_{in} coming from a microwave beam and given by MaRaTra code that runs on the grid. The physical problem is that the results of MaRaTra, which are an input for ASTRA, strongly depend on the plasma parameters that are estimated by ASTRA. So this code calls MaRaTra when the plasma parameters are changed. An example of the plasma evolution estimated using this workflow can be seen in Figure 7. The results produced by this workflw are physically relevant and have published in [17].

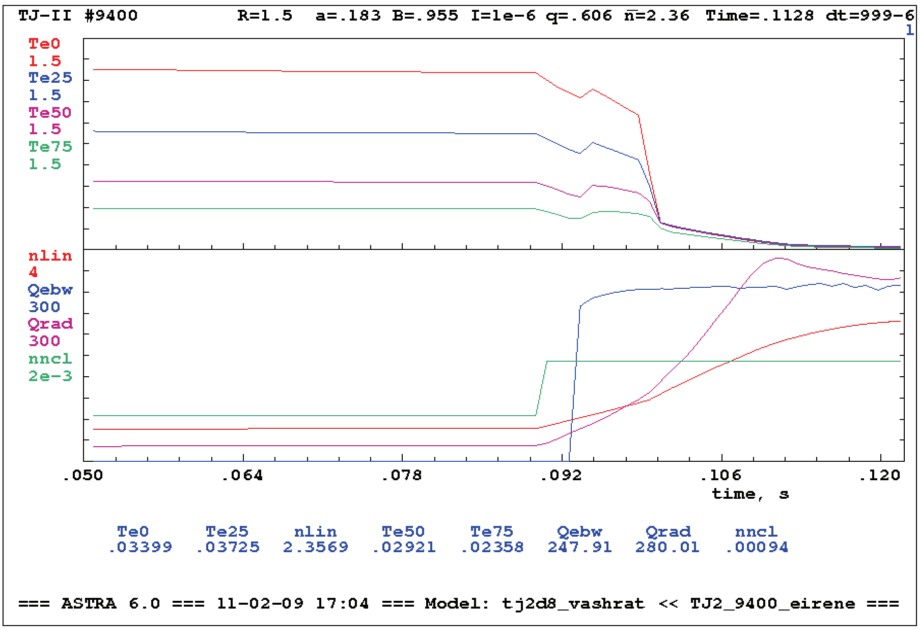


Fig. 7. Evolution of plasma parameters estimated by ASTRA, linked to MaRaTra.

7. Conclusions

An increasing number of fusion applications has been ported to the grid, showing the utility of such computing paradigm for fusion research. The range of problems solved and the techniques used have been increased from the first use of the grid for fusion. Nowadays, we are running Monte Carlo codes, parameter scan problems, ray tracing codes, genetic algorithms, etc. Besides the large variety of grid applications, it is remarkable the wide range of problems solved using the grid. The fusion research involves a large variety of physics problems that go from the Magnetohydrodynamics to the kinetic theory, including plasma heating by NBI or microwaves. Grid computing is present in almost all of them. Moreover, the present work shows the capability of grid techniques for helping to reach the full simulation of the numerical reactor, helping to the traditional HPC applications. Finally, complex workflows have been developed beyond the simple applications in order to simulate more complex phenomena in fusion devices. These workflows allow one to connect codes that run on different platforms (grid and HPCs) and that deal with different physical models and scales.

8. References

[1] K. Ikeda et al. Nuclear Fusion 47 (2007). Special Issue.

[2] http://www.ipp.mpg.de/ippcms/eng/pr/forschung/w7x/index.html

[3] F. Castejón et al. Proc. of the 2005 EGEE Conference. 2005, Pisa (Italy)

[4] EUFORIA: http://www.euforia-project.eu/

[5] F. Castejón. Summary of Fusion Session. Proc. of the 4th EGEE/OGF User Forum. 2009, Catania (Italy)

[6] F. Castejón, L. A. Fenández, J. Guasp, V. Martín-Mayor, A. Tarancón, and J. L. Velasco. Plasma Physics and Controlled Fusion 49 (2007) 753

[7] S.P. Hirshman and J. C. Winston. Physics of Fluids 26 (1983) 3553

[8] F. Castejón, Á. Cappa, M. Tereshchenko, and Á Fernández. Nuclear Fusion 48 (20089 075011

[9] J. Guasp and M.Liniers: "The Complex FAFNER/EIRENE at Ciemat: Scripts and file structure". Reports Ciemat. October 2008, Madrid, Spain.

[10] EIRENE: http://www.eirene.de/

[11] F. Castejón, A. Gómez-Iglesias, D. Tshkakaya, and A. Soba. Proc. of the 4th EGEE/OGF User Forum. 2009, Catania (Italy)

[12] F. Castejón, J. M. Reynolds, J. M. Fontdecaba, R. Balbín, J. Guasp, D. López-Bruna, I. Campos, L. A. Fernández, D. Fernández-Fraile, V. Martín-Mayor, and A. Tarancón. Fusion Science and Technology 50 (2006) 412.

[13] F. Castejón, A. López-Fraguas, A. Tarancón, and J. L. Velasco. Plasma and Fusion Research 3 (2008) S1009.

[14] J.L. Velasco, F. Castejón, L.A. Fernández,V. Martin-Mayor, A. Tarancón and T. Estrada. Nucl. Fusion 48 (2008) 065008.

[15] E. Huedo, R.S. Montero and I.M. Llorente. Journal Scalable Computing - Practice and Experience 6 (2005) 1 Editorial Nova Science. ISSN 1895-1767

[16] J. L. Vázquez-Poletti, E. Huedo, R. S. Montero and I. M. Llorente. "A Comparison Between two Grid Scheduling Philosophies: EGEE WMS and GridWay" Multiagent and Grid Systems 3 (2007) 429

[17] Á. Cappa, D. López-Bruna, F. Castejón, et al."Calculated evolution of the Electron Bernstein Wave heating deposition profile under NBI conditions in TJ-II plasmas". Contribution to Plasma Physics, in press.