Improvements of the Particle-In-Cell Code EUTERPE for Petascaling Machines

Xavier Sáez[a,*], Alejandro Soba[a], Edilberto Sánchez[b], Ralf Kleiber[c], Francisco Castejón[b], José M. Cela[a]

[a]*Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain*
[b]*Laboratorio Nacional de Fusión, Madrid, Spain*
[c]*Max-Planck-Institut für Plasmaphysik, EURATOM-Association, Greifswald, Germany*

Abstract

In the present work we report some performance measures and computational improvements recently carried out using the gyrokinetic code EUTERPE [1, 2], which is based on the general particle-in-cell (PIC) method. The scalability of the code has been studied for up to sixty thousand processing elements and some steps towards a complete hybridization of the code were made. As a numerical example, non-linear simulations of Ion Temperature Gradient (ITG) instabilities have been carried out in screw-pinch geometry and the results are compared with earlier works. A parametric study of the influence of variables (step size of the time integrator, number of markers, grid size) on the quality of the simulation is presented.

Keywords: PIC code; MPI-OpenMP; hybrid codes; ITG instabilities.

1. Introduction

EUTERPE is a particle-in-cell (PIC) gyrokinetic code for the simulation of fusion plasmas. The aim of this code is to address global linear and nonlinear simulations of fusion plasmas in three-dimensional geometries, in particular in stellarators. It was developed as a parallel code, using Message Passing Interface (MPI), and has been adapted to different computing platforms. A kinetic treatment of electrons and a third species, as well as electromagnetic effects, have been included recently and work is in progress to include collisions. The Barcelona Supercomputing Center (BSC) collaborates with the Fusion Theory Unit of CIEMAT and IPP-Greifswald for the development and exploitation of this code.

EUTERPE is at the forefront of plasma simulations and requires a huge amount of computational resources. The code provided good results both in linear and nonlinear simulations of Ion Temperature Gradient (ITG) instabilities [1, 3, 4, 5, 6, 7]. In those simulations, especially for the nonlinear ones, it became clear that the amount of computational resources that a global three-dimensional PIC code requires for typical simulations is huge, and it is of crucial importance that it can run efficiently on multi-processor architectures. The code had been optimized and its scalability had been studied for up to 60000 processors in the framework of the project PRACE [8].

The code had already been parallelized using MPI and a domain cloning technique was used in order to use many more processors without increasing the inter-processor communications to prohibitive levels [9].

In this work, we present a computational study of the capabilities of the code on several architectures taking into account the future PetaScaling machines which are being developed in the next years. Besides, we developed a new solver for the quasineutrality equation completely adapted to EUTERPE possibilities based on MPI+OpenMP parallel directives, which can be used instead of the solvers from the external package PETSc [13].

A critical point is related to the numerical noise and its dependence on the amount of markers and on the time step used in the integration of the equations of motion. We analyze in this work the close relation between these parameters.

This paper is organised as follows. In section 2 the code and the basics of its numerical model are introduced. Some results related to the parallelization strategies implemented in the code and some performance measurements are summarized in section 3. Some simulations done with the code, referred to as parameter optimizations, are presented in section 4. Finally some conclusion are drawn in section 5.

2. The EUTERPE code

The EUTERPE code solves the gyroaveraged Vlasov equation for the distribution function of each kinetically treated species (ions, electrons, or a third species)

$$\frac{\partial f_s}{\partial t} + \frac{\mathrm{d}\mathrm{v}_\parallel}{\mathrm{d}t}\frac{\partial f_s}{\partial v_\parallel} + \frac{\mathrm{d}\vec{\mathrm{R}}}{\mathrm{d}t}\frac{\partial f_s}{\partial \vec{R}} = 0, \qquad (1)$$

where f_s is the distribution function of the species s. The code is based on the particle-in-cell (PIC) scheme, i.e. the distribution function f_s is discretized using markers. The markers follow the equations of motion, giving the evolution with time of $\vec{R}$ and $\mathrm{v}_\parallel$ as functions of the electric and magnetic fields.

The evolution of the electric and magnetic fields is given by the quasi-neutrality equation and Ampère's law (in the electromagnetic version) where the charge and current density is calculated using a charge assignment procedure. The δf approximation is used: the distribution function is decomposed into an

[*]Corresponding author.
E-mail address: xavier.saez@bsc.es

equilibrium part (Maxwellian) and a time-dependent perturbation. Only the evolution of the perturbation is followed, which allows to reduce the noise and the needed resources, as compared to the alternative of simulating the evolution of the full distribution function.

The electromagnetic potential is represented on a spatial grid, the electric charge being carried by the markers. Two coordinate systems are used in the code: a system of magnetic coordinates (PEST) (s, θ, ϕ) is used for the potential and cylindrical coordinates (r, z, ϕ) are used for pushing the particles, where $s = \Psi/\Psi_0$ is the normalized toroidal flux. The change between coordinate systems is facilitated by the existence of the common coordinate (ϕ). The equations for the fields are discretized using finite elements (B-splines) and the PETSc library is used for solving the resulting matrix equations. The integration of the equations of motion of the markers is done using a fourth order Runge-Kutta scheme.

An equilibrium state calculated with the code VMEC [11] is used as a starting point. The equilibrium quantities computed by VMEC are transformet to PEST coordinates and mapped onto the spatial grid using an intermediate program.

EUTERPE features several techniques for noise control: filtering of Fourier modes (square and diagonal filters can be used) and optimized loading [12]. More details about the code can be found in the Refs [1, 3, 4, 5, 6, 7]

3. Parallelization strategies in EUTERPE

The parallelization of PIC codes is an important and necessary task for increasing the capabilities of simulations. However, it is a hard mission, since PIC codes have data accesses with multiple levels of indirection and complicated message-passing patterns.

The first technique used to parallelize EUTERPE is the *domain decomposition*. The distribution of work between processors is based on the division of the physical domain into portions in the ϕ direction. The particles are distributed according to their physical coordinates in the domains in order to split the computation efficiently among processors. At the end of each time step, the particles which have moved from its original domain to a new one are transferred to their new processor, also the electromagnetic potential values from boundary nodes of the grid are sent to the neighbour processors.

A main advantage of this technique is the intrinsic scalability of the physical-space resolution when the number of processors is increased, although the parallelization is limited by the grid divisions. On the other hand, due to particle migration between domain partitions on different processors, the load balancing varies during the simulation and imbalances may appear between processors.

The other technique used in EUTERPE is the *domain cloning*. It is a combination between domain decomposition and particle decomposition. The processors are distributed into a number of groups, each one of which is assigned to one of the domain clones. These clones are copies of the same domain and the particles are distributed between them. A domain decomposition is applied also to the clones and as a result a larger amount of processors can be used.

The suitability of these techniques in executions on a large number of processors can be observed in the following computational study performed on Huygens (SARA) and Jugene (FZJ) supercomputers in the framework of the PRACE project.

Huygens is an IBM pSeries 575 system. It consists of 104 nodes, 16 dual core processors (IBM Power6, 4.7 GHz) per node and either 128 GBytes or 256 GBytes of memory per node. The total peak performance is 60 Teraflops. Jugene is a Blue Gene/P system. It consists of 73728 nodes, 4 core processors (32-bit PowerPC 450 850 MHz) per node and 2 GByte of memory per node. The total peak performance is about 1 Petaflops.

The chosen data set to study the parallelization of EUTERPE is a typical scenario of ITGs, specifically a cylindrical geometry. The initial equilibrium corresponds to a θ-pinch with radius $a = 0.55$ m, length $l = 5.55$ m and a fixed homogeneous magnetic field along the axis of the cylinder. The resolution of the spatial grid used in the simulations was $n_s \times n_\theta \times n_\phi = 32 \times 512 \times 512$, so the grid can be distributed on up to 512 processors. The use of clones allows us to perform runs on up to several tens of thousands of processors. The number of markers used in the simulation is 10^9.

The scalability of the code has been studied as follows: the size of the problem has been maintained fixed while the number of processors used in the simulation has been increased (hard scaling). On Huygens, the simulations ran on 128 up to 2560 processors, while on Jugene the number of processors ranged from 512 up to 61440 processors.

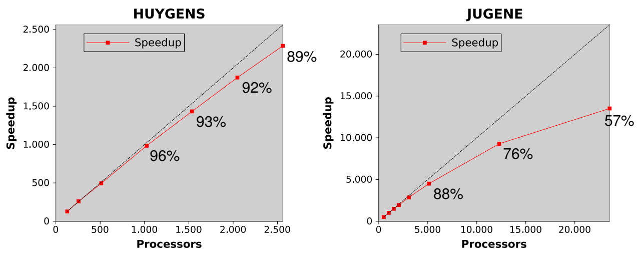

Figure 1: Scalability of EUTERPE using 10^9 markers. The percentage with respect to the ideal speedup is specified for each measurement.

The figure 1 shows a very good scalability on Huygens, but not so good on Jugene, where a degradation of the speedup appears for more than 12288 processors. The reason for this fact is that the amount of work per processor is not enough to hide the increasing time for the inter-processors communications for a large number of processors. When the number of markers is increased up to $5 \cdot 10^9$ (this number is limited by the small available memory per processor on Jugene) the scalability improves considerably, as it is shown in figure 2.

The implementation of the mentioned techniques in EUTERPE uses only the parallelism at task level which is provided by MPI. For that reason our idea was to develop an hybrid code that would take advantage of all the levels of parallelism that a multicore architecture offers and also of all the memory of a node.

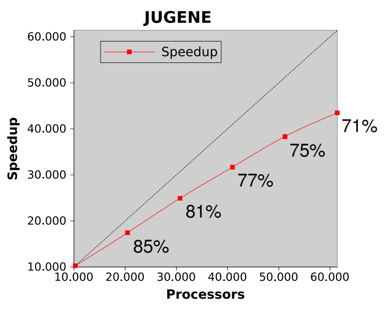

Figure 2: Scalability of EUTERPE using $5 \cdot 10^9$ markers. The percentage with respect to the ideal speedup is specified for each measurement.

Firstly, we analyzed the distribution of the execution time and we identified the most time-consuming routines: the routine that moves the particles, the one that determines the charge density and the one that solves the field equation.

To solve the field equation, EUTERPE uses the PETSc library. It is a well known package for solving Partial Differential Equations on parallel machines. As PETSc routines are not thread safe, we decided to develop a hybrid version of the solver. The new solver (PCG) is a Jacobi preconditioned conjugate gradient and it has been completely parallelized. The dependencies between iterations are due to the dot products and they have been solved with reductions. The results obtained show that the speedup of the hybrid version with respect to a sequential execution is near linear (Figure 3). Moreover, the most time consuming part in the solver is the sparse matrix-vector multiplication, where the memory access has a higher cost respect to the computation. This is due to a low data reuse and a lack of float operations to hide the elapsed time in memory operations. So, we are in front of a memory bound problem, in other words, the limiting factor is the memory access speed. This explains the performance obtained by PETSc and our new solver.

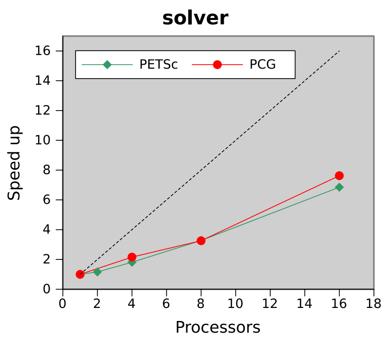

Figure 3: Scalability of the hybrid version of our solver (PCG) compared to the scalability of PETSc. To run PCG solver, it has been assigned one MPI task per node and one OpenMP thread per MPI task.

In the rest of the identified routines, OpenMP was introduced looking for loops with coarse granularity, for example, loops that move the particles inside a domain. Since some of the particles can write data into the same memory positions, critical regions were needed to avoid conflicts between threads. However, the critical sections were avoided whenever possible by creating private copies of the conflicting data structures (one per thread).

Figure 4 shows graphically the behavior of the hybrid version of EUTERPE (MPI+OpenMP) using the PARAVER tool [14]. A limited case is shown in order to perform a detailed analysis using the tracing tools. Similar behavior is expected with bigger cases. The dark blue colour means that the thread is running and the light blue colour means that the thread is idle. Therefore we can observe that the work is well balanced and the threads are running almost all the time.

4. Parametric analysis

The optimization task could require the change of the simulation parameters in order to better adapt the simulation to the computer architecture. In this section, the effect of changing parameters on the quality of the simulation is studied.

In a previous work [7], the code provided good results both in linear and nonlinear simulations of Ion Temperature Gradient (ITG) instabilities. Comparison with results obtained with the TORB code in screw-pinch geometry [15] showed that the time step can be increased from 1 to 20 (in units of the ion cyclotron frequency) obtaining similar results. The saturation of energies, heat flux and also the structures that appeared in the potential in the non-linear phase were quite similar in both cases. This is very important because it means that similar results can be obtained with less CPU time.

Here we make a deeper study of the dependence of the simulation quality on the time step used in the integration of the equations of motion. A Runge-Kutta integrator of fourth order is used in our simulations. Here we address the question how the time step used for time integration depends on the size of the problem and on the grid used to discretize the domain.

Runs were done with several time steps ($\Delta t/\Omega_i$ = 5, 10, 20, 40, 80, 120, 160, 200, 300, where $\Omega_i = 1.2 \cdot 10^8 s^{-1}$ is the ion cyclotron frequency) and particle numbers in order to study its influence on results. Simulations have been run with N=1, 4, 16, 32, 64 and 256 million particles. Firstly, we performed non-linear electrostatic simulations of ITG in a screw-pinch with $\iota/2\pi = 0.8$ and a grid of 100x64x64 nodes. Afterwards the simulations have been extended to grids with 100x128x128 and 100x256x256 nodes.

The signal to noise ratio (S/N) is used as a measure of the quality of the simulation. It is defined as the ratio of the spectral power kept inside the filter to the spectral power filtered out. In EUTERPE a filter on the density in Fourier space is always used to reduce particle noise. For the simulations we used a diagonal filter along the $n/m = \iota/2\pi$ line, with a width of $\Delta m = 4$. The squared filter limit is always set to 2/3 of the Nyquist frequency.

In figure 5 one can see the S/N ratio for several test cases with different number of markers and time steps. In all the cases the grid was 100x64x64. The S/N ratio does not show any dependency on the time step and the numerical noise decreases when the number of markers increases for a fixed grid. Furthermore, the numerical noise increases with the size of the grid in agreement with results obtained by other authors [16, 17].

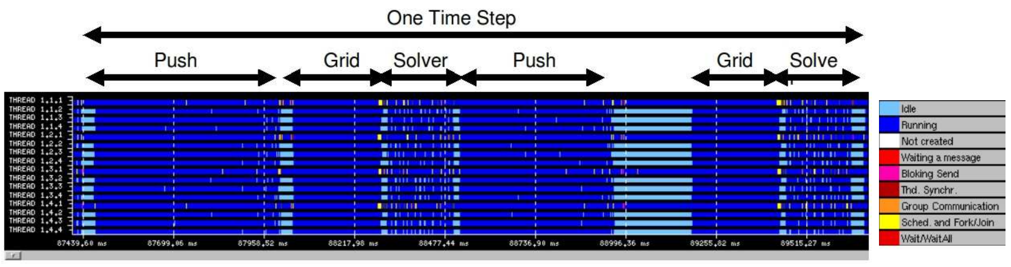

Figure 4: Trace of one time iteration of the hybrid version of EUTERPE. The name of the routines is specified on the top of the graphic: push (computes the motion of the particles), solver (solves the field equation) and grid (determines the charge density). On the right, there is the legend to interpret the state of threads.

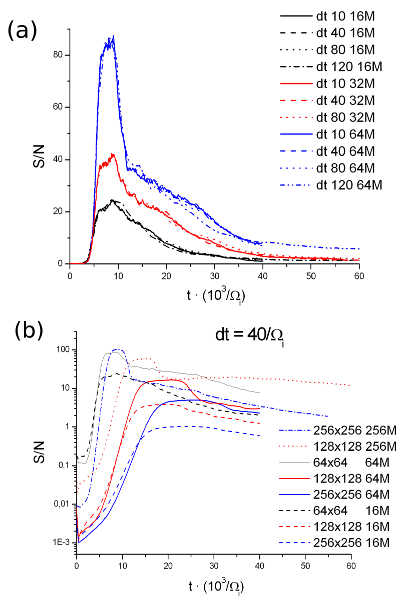

Figure 5: S/N signal for different simulations. (a) Comparison between time steps and particle numbers in a grid of 100x64x64 nodes. (b) Comparison for three grids sizes for time step dt=40.

The difference between the electrostatic and the kinetic energy of the markers (sum of quantities with different signs) constitutes a measure of the energy conservation, and consequently, another measure of the quality of the simulation.

An influence of the time step on the energy conservation was readily observed. There is a linear phase during which both energies grow exponentially and, after this phase, the non-linear interactions between modes become important. In order to study how the energy conservation depends on the time step we define an average measure of energy conservation as

$$q =< (|E_f + E_k|)/|E_f| > \qquad (2)$$

where E_f and E_k are the electrostatic and kinetic energies and <> means ensemble average. This average is computed omitting the initial phase of the simulation, before the linear growth.

The average quality measure is computed in a time window of $10^4/\Omega_i$ for all the cases. As in simulations with different number of markers the initial perturbation is different, at a given simulation time the energy E_f (and E_k) can reach different levels in different simulations. The levels can be very different

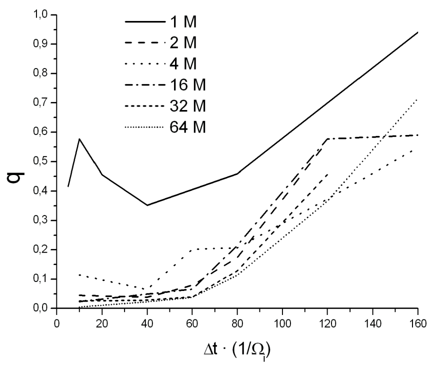

Figure 6: Average energy conservation ($< (|E_f + E_k|)/|E_f| >$) as a function of the time step for a fixed grid size of 100x64x64 cells.

for times corresponding to the exponential growth phase. To prevent distortions due to this effect, the beginning of the averaging window is not located at a fixed time, but when the E_f reach a minimum level (10^{-4}) in normalized units instead. As can be seen in figure 6 the time step influences the energy conservation. For small time steps the energy conservation is almost the same for different number of markers (N=16, 32 and 64 M). As the time step increases, the energy conservation gets worse. Looking at the simulations we noted that the field energy changes only slightly when the time step is increased. However, the kinetic energy of the markers increases significantly. The increase in the kinetic energy is larger for larger time steps.

Figure 7, where the difference of the energy with respect to the initial value is plotted, shows this increment for selected time steps using 64M markers. This figure also shows that the kinetic energy is responsible for the degradation in the conservation of energy. For a time step of 80 the kinetic energy is 40% larger (in absolute value) than the potential.

5. Conclusions

We have studied the scalability of EUTERPE up to 60000 processors with a very good performance up to several thousands. For larger numbers, the scalability has been worse due to the increase of inter processor communication and the reduction of the work per processor. We observe that after increasing the number of markers, the speedup was better.

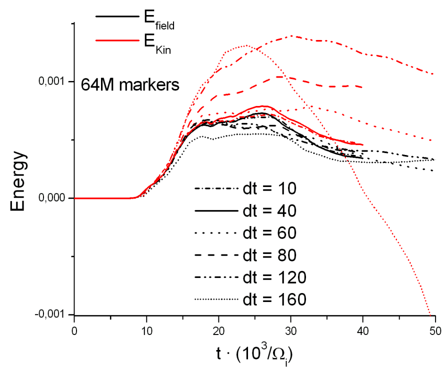

Figure 7: Potential and kinetic energies for different time steps. The grid size was 100x64x64 for all the cases.

Even though the scalability of the code has proven to be good, more work is needed to maintain a very good performance in coming computer architectures with multi-core nodes and millions of cores in total. A progress has been made in this direction, introducing a new hybrid version of the solver with a similar behaviour to the original PETSc solver included in the previous version of the code.

A parameter analysis made for time step, grid size and number of markers allows to clarify their close relation and helps in planning simulation needs.

It is clear that, for a fixed time step, the increment in markers improves the quality of the simulations. The same happens for a fixed number of markers if we decrease the time step. The S/N ratio seems to be independent of the time step used in the integrator, even for very high time steps (>= 200) when the energy conservation is very bad. The S/N ratio decreases very fast with the grid size and grows with the number of markers.

A large loss of energy conservation is observed when the time step is increased. The potential energy is less affected by a change in time step than the kinetic energy which grows very quickly with increasing time step. This could be an indication of numerical noise introduced by the temporal integrator (a fourth order Runge Kutta), particularly for large time steps. Besides, the use of a new time integrator (for example a symplectic integrator) for the equations of motion of the markers should be considered. This could allow for a better conservation of energy or the use of a larger integration time step with still acceptable results. A deeper analysis needs to be done to determine how one can find an optimal time step for a fixed number of markers and grid size.

6. Acknowledgements

The authors want to thank I. Calvo for stimulating discussions. This work has been partially funded by the Spanish Ministry of Science and Innovation under project ENE2008/06082/FTN. The scalability studies were possible thanks to the PRACE project resources.

References

[1] G. Jost. Simulations Particulaires d'ondes de drive dans des configurations magntiques 3D. Technical report, CRPP-REPORT-2000-015, 2000.

[2] G. Jost, T. M. Tran, K. Appert, et al. Global linear gyrokinetic PIC simulations in 3D magnetic configurations. in Theory of Fusion Plasmas, International Workshop, Varenna, September 1998. Editrice Compositori, Società Italiana di Fisica, Bologna, 1999, p. 419.

[3] G. Jost, T. M. Tran, W. Cooper, et al. Global linear gyrokinetic simulations in quasi-symmetric configurations. Physics of Plasmas 8, 3321, 2001.

[4] V. Kornilov, R. Kleiber, R. Hatzky, et al. Gyrokinetic global three-dimensional simulations of linear ion-temperature-gradient modes in Wendelstein 7-X. Physics of Plasmas 11, 3196, 2004.

[5] V. Kornilov, R. Kleiber and R. Hatzky. Gyrokinetic global electrostatic ion-temperature-gradient modes in finite β equilibria of Wendelstein 7-X. Nuclear Fusion 45, 238, 2005.

[6] R. Kleiber. Global linear gyrokinetic simulations for stellarator and axisymmetric equilibria. Joint Varenna-Lausanne International Workshop. AIP Conference Proceedings, 871, p. 136, 2006.

[7] E. Sánchez, R. Kleiber, R. Hatzky, et al. Linear and non-linear simulations using theEUTERPE gyrokinetic code. To appear in IEEE Transaction on Plasma Science (Aug. 2010).

[8] http://www.prace-project.eu/

[9] R. Hatzky, Parallel Computing **32**, 325 (2006).

[10] T. S. Hahm. Nonlinear gyrokinetic equations for tokamak microturbulence. Physics of Fluids 31, 2670, 1988.

[11] S. P. Hirshman, and J. C. Whitson. Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria. Physics of Fluids 26, 3553, 1983.

[12] R. Hatzky, T. M. Tran, A. Könies, et al. Energy conservation in a nonlinear gyrokinetic particle-in-cell code for ion-temperature-gradient-driven modes in theta-pinch geometry. Physics of Plasmas 9, 898, 2002.

[13] S. Balay and K. Buschelman and W. D. Gropp et al. PETSc Web page. http://www.mcs.anl.gov/petsc

[14] V. Pillet, J. Labarta, T. Cortés and S. Girona. Paraver: A tool to visualize and analyze parallel code. Transputer and occam Developments, pages 17-32, April 1995. http://www.bsc.es/plantillaA.php?cat_id=485.

[15] C. Nührenberg, R. Hatzky, S. Sorge, et al. Global ITG Turbulence in Screw-Pinch Geometry. IAEA TM on Innovative Concepts and Theory of Stellarators, Madrid 2005.

[16] A. Bottino and A. G. Peeters, R. Hatzky, et al. Nonlinear low noise particle-in-cell simulations of electron temperature gradient driven turbulence. Physics of plasmas, 14, 2007.

[17] W. Nevins, G. Hammett, A. Dimits, et al. Discrete particle noise in particle-in-cell simulations of plasma microturbulence. Physics of plasmas, 12, 2005.