

CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
 SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
 Integrated Infrastructure Initiative

JRA3: Turbulence code GENE prepared for usage in DEISA;
 final report on new ORB5 code for extreme computing

Deliverable ID: DEISA-D-JRA3-4

Due date : April 30, 2006
Actual delivery date: May 17, 2006
Lead contractor for this deliverable: RZG, Germany

Project start date : May 1st, 2004
Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Table of Contents.....	1
INTRODUCTION	2
1.1 Executive Summary.....	2
1.2 References and Applicable Documents	2
1.3 List of Acronyms and Abbreviations	2
2 EXTREME COMPUTING BY EUROPEAN PLASMA PHYSICISTS.....	4
3 REPORT ON GENE CODE ENABLING.....	4
3.1 Introduction.....	4
3.2 Code evaluation.....	4
3.3 Scalability analysis	5
3.4 Enabling for hyperscalability.....	6
3.5 Porting to SGI Altix	8
4 FINAL REPORT ON ORB5 TURBULENCE CODE	8
4.1 Introduction.....	8
4.2 Single processor optimization and portability	8
4.3 ORB5 enabling: optimisation of the scaling property.....	10
4.3.1 The domain cloning concept.....	10
4.3.2 The initial concept for a scalable solution of the field equations.....	11
4.3.3 The enhanced concept and its implementation	12
4.3.4 Evaluation of enhanced ORB5	13
5 FURTHER RELEVANT APPLICATION CANDIDATES FROM PLASMA PHYSICS.....	14

Introduction

1.1 Executive Summary

This document “JRA3: Turbulence code GENE prepared for usage in DEISA; final report on new ORB5 code for extreme computing” is the 24-month deliverable DEISA-D-JRA3-4 for the Joint Research Activities in Plasma Physics. Besides providing an application plugin, GENE has now been prepared for extreme computing. For the ORB5 code – an ITER relevant code – all the work done so far is described.

1.2 References and Applicable Documents

- [1] Allfrey, S.J. and Hatzky R.: A revised delta-f algorithm for nonlinear PIC simulation. *Comp. Phys. Commun*, **154**: 98, 2003
- [2] Hatzky, R., Tran, T.M., Könies, A., Kleiber, R., and Allfrey, S.J.: Energy Conservation in a Nonlinear Gyrokinetic Particle-in-cell Code for Ion-Temperature-Gradient-driven (ITG) Modes in theta-Pinch Geometry. *Phys. of Plasmas*, **9**: 898, 2002
- [3] Villard, L., Allfrey, S.J., Bottino, A., Brunetti, M., Falchetto, G.L., Grandgirard, V., Hatzky, R., Nührenberg, J., Sauter, O., Sorge, S., and Vaclavik, J.: Full Radius Linear and Nonlinear Gyrokinetic Simulations for Tokamaks and Stellarators: Zonal Flows, Applied ExB Flows, Trapped Electrons and Finite Beta. *Nucl. Fusion*, **44**: 172, 2004
- [4] <http://sourceforge.net/projects/unicore/>
- [5] Kim, C.C. and Parker S.E.: Massively Parallel Three-Dimensional Toroidal Gyrokinetic Flux-Tube Turbulence Simulation. *J. Comp. Phys.*, **161**: 589, 2000
- [6] R. Hatzky, Domain Cloning for a Particle-in-Cell (PIC) Code on a Cluster of Symmetric-Multiprocessor (SMP) Computers, *Parallel Computing*, in press

1.3 List of Acronyms and Abbreviations

BSS	Batch Sub-System on the target machine
CEA	Comité de l'énergie atomique, see http://www-cad.cea.fr
CIEMAT	Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, http://www.ciemat.es
CINECA	Consorzio Interuniversitario, Bologna, http://www.cineca.it
CRPP	Centre de Recherches en Physiques des Plasmas, Lausanne see http://crppwww.epfl.ch
DEISA site	partner site of the DEISA consortium
ECMWF	European Centre for Medium-Range Weather Forecast, see http://www.ecmwf.int
EPCC	Edinburgh Parallel Computing Centre, see http://www.epcc.ed.ac.uk
Fortran	FORmula TRANslation, scientific programming language

GPFS	Global Parallel File System
HPCx consortium	HPCx consortium, see http://www.hpcx.ac.uk
IPP	Max-Planck-Institut für Plasmaphysik, Garching, see http://www.ipp.mpg.de
ITER	International Thermo-Nuclear Experimental Reactor
JRA	Joint Research Activity
RZG	Rechenzentrum Garching, see http://www.rzg.mpg.de
SMP	Symmetric MultiProcessor
UNICORE	UNiform Interface to COmputing Resources, see http://www.unicore.org

2 Extreme Computing by European Plasma Physicists

According to leading plasma physicists, a challenging scientific goal is the “numerical tokamak or stellarator” in which the simulation of microscopic turbulence covers, in a single run, both the whole spatial plasma region and a time interval corresponding to global profile adjustment (the “energy confinement time”). This time interval is a few seconds in a real reactor, simulated with sub-microsecond time steps. At the same time the predictions of these and other models should satisfy stringent accuracy requirements to make them useful for engineering applications such as the design of a fusion power plant.

The two codes GENE and ORB5 have been proposed for DEISA enabling and for support and preparation for a DEISA Extreme Computing Initiative (DECI) project.

The advantage of using both codes, ORB5 and GENE, is the complementary numerical approaches they take to solve the same gyrokinetic equations. On one hand, the PIC code ORB5 is based on an intuitively clear picture of particles following the characteristics of the underlying partial differential equations. On the other hand, the numerical method of the continuum code GENE is more abstract but without the problem of statistical noise. As the simulation of turbulence in plasmas is at the forefront of nonlinear dynamics studies it is very important to have the opportunity to compare the results of both codes with each other to get more insight into the behaviour of the different numerical methods.

3 Report on GENE code enabling

3.1 Introduction

GENE is a so-called continuum (or Vlasov) code. Here, the nonlinear gyrokinetic equations are solved on a fixed grid in five-dimensional phase space. All differential operators in phase space are discretised by fourth-order (compact) finite differences. For maximum efficiency, GENE uses a coordinate system which is aligned to the equilibrium magnetic field and a reduced (flux-tube) simulation domain. This reduces the computational effort by 2-3 orders of magnitude. Moreover, it can deal with arbitrary toroidal geometry (tokamaks or stellarators) and retains full ion/electron dynamics as well as magnetic field fluctuations. At present, GENE is the only plasma turbulence code in Europe with such capabilities.

3.2 Code evaluation

The GENE version provided by the authors was analyzed. A mixed parallelization model using both OpenMP (for usage within an SMP) and MPI (for usage between SMP nodes) is supported, fully exploiting the architectural characteristics of a large SMP-based system such as the IBM Regatta.

However, a hard limit of 64 MPI tasks was detected. This limit is in principle not too restrictive on IBM Regatta-like architectures with large SMPs, since theoretically up to $32 \times 64 = 2048$ processors of a Regatta-based system could be used.

Scalability tests with scientifically relevant problem cases revealed scalability bottlenecks already starting by 256 processors, as illustrated in Fig. 1.

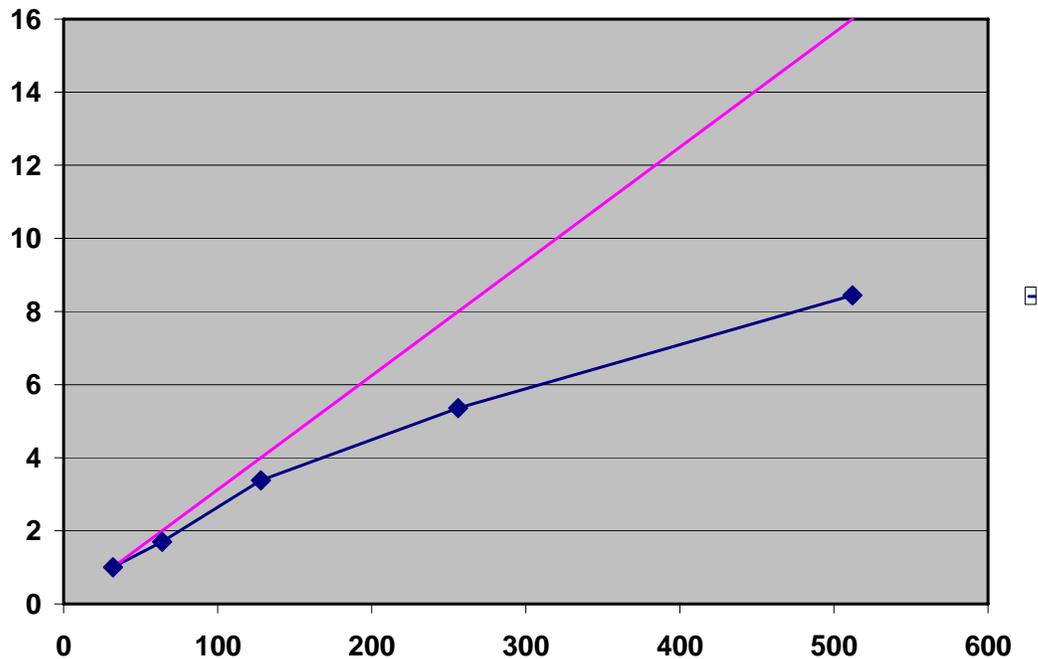


Fig. 1 Scalability curve of GENE9 for a typical problem size (blue line) with measurement points for 32, 64, 128, 256 and 512 processors. Ideal linear scaling is indicated by the pink line. The base point is the measurement on 32 processors, which was assigned a speedup of 1.

As the reference point in Fig. 1 is 32 processors, ideal scaling on 512 processors would result in a speedup of 16. With GENE9 we are far from that with an actual speedup of 8.4. As the relative speedup from 256 to 512 is only 1.5, there is no potential for efficient usage of larger processor configurations. Furthermore, SMP mode (with 32-way nodes on IBM Regatta) was used. For small-node systems with only two processors per node, the technical upper limit would be only 128 processors (with 64 MPI tasks and 2 OpenMP threads per task).

3.3 Scalability analysis

To understand the measured scalability problem, the structure and the parallelization scheme of the code was further analyzed:

- The currently used dimensions for the spatial coordinates x and y in production runs are typically 128 for x and 64 for y . The scientists expect that these values could be increased for really big runs in the future up to 4096. The x and y directions currently have no domain decomposition.

- The spatial coordinate s typically has a value of 16 and domain decomposition so far supports up to 4 MPI tasks.
- For the parallel velocity z a value of 16 is currently used. No domain decomposition exists so far, but the OpenMP parallelization uses the loops over z .
- For the perpendicular velocity m a typical dimension is 8. A domain decomposition allows up to 8 MPI tasks.
- The number of particle species can be either 1, 2 or 3, and is normally set to 2. Each species is a separate group of MPI tasks.

Therefore the maximum number of MPI tasks supported is currently 64. The scalability, however, already reaches its efficiency limit with 128 processors (see Fig. 1). In addition, to efficiently run the code on a machine with much smaller SMP nodes, a much higher number of supported MPI tasks is required.

We conclude that in order to enable the GENE code for extreme computing, the most important task is a significant extension of the scalability.

3.4 Enabling for hyperscalability

The GENE code has a total of 6 dimensions with the potential for parallelization. The spatial coordinates x and y , so far only treated serially, contain the highest potential for domain decomposition. A large number of 2-dimensional FFTs are done on the xy planes. If the xy plane is distributed, it must be transposed in order to perform the FFT in the x and y directions. However, the transposition requires an all-to-all communication which results in a high communication overhead on the IBM Regatta and many other systems.

Another option would be domain decomposition of the parallel velocity z . It would have less communication overhead, but its maximum number of MPI tasks would probably be limited to only 8.

Since the coordinates x and y contain the highest potential for domain decomposition we focused on these axes. If we could leave the y -coordinate in k -space, the FFTs would have to be performed on the x -coordinate only, which is contiguous in memory. A transpose of the xy -plane would not be necessary. There are some sections in the code where transformations to the configuration space are necessary. For these transformations, transpositions of the xy -plane have to be performed. However, the number of these transformations is much smaller than the number of FFTs.

Since this implies a major change to the overall data structure, it has consequences for many parts of the code. As a prerequisite the code was adapted by the authors according to the new role of the y -coordinate. Our main task was now the design and the realization of the domain decomposition of the y -coordinate.

The number of points available in the y direction has to be dynamically mapped on the number of processors selected for treating the y direction. This scheme was consequently applied on all loops in the y direction (a number in the range of 20 to 40).

Furthermore major adaptations of the FFTs had to be done:

- For the transformation to Fourier space only the x direction has to be transformed since the y direction is already in Fourier space. The same applies for the back transformation.
- For the transformation to configuration space only the y direction has to be transformed since the x direction is already in configuration space.
- In order to get the y direction contiguous in memory, the xy planes have to be transposed. For the realization of this operation, an all-to-all communication scheme was used.

The new code was tested with real problems sizes (see Fig. 2) and delivered correct results.

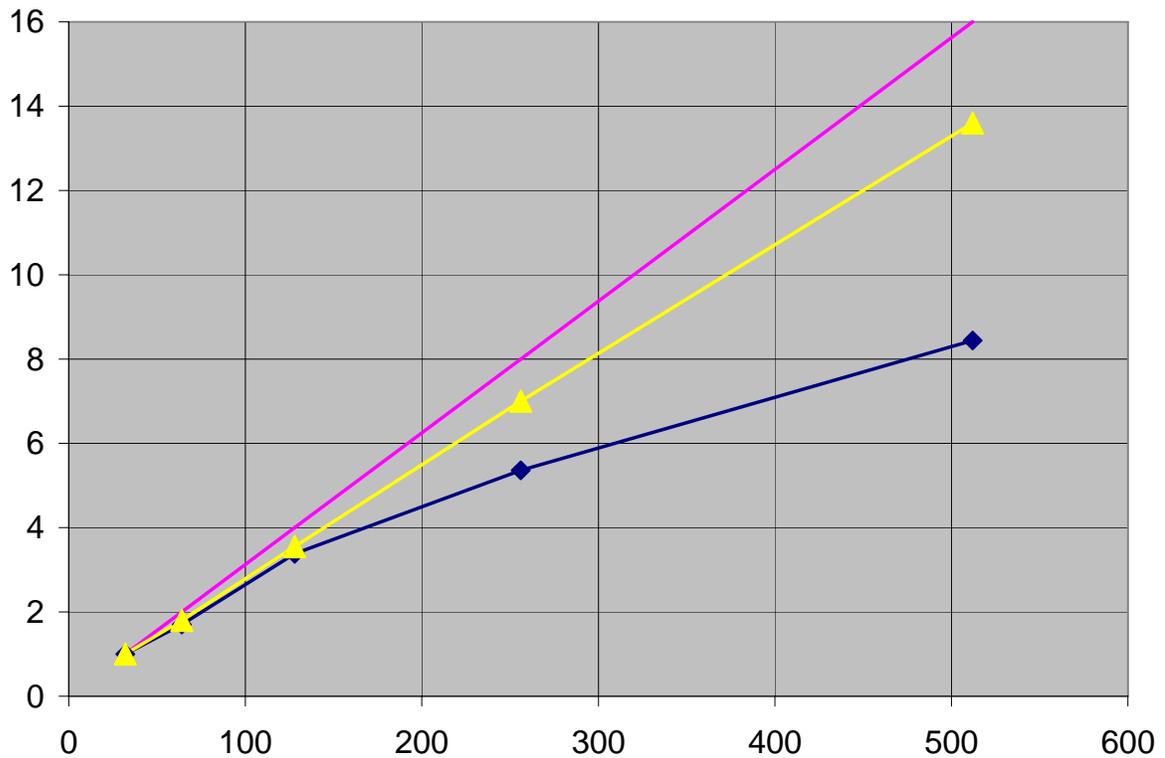


Fig. 2 Scalability curve of GENE10 for a typical problem size (triangles and yellow line), with measurement points for 32, 64, 128, 256 and 512 processors. Ideal linear scaling is indicated by the pink line. Diamonds and the blue line refer to GENE9 scaling.

As shown in Fig. 2, with the new GENE10 code the original speedup of 1.5 from 256 to 512 IBM Regatta processors was increased significantly to 1.94, pushing the scalability of the code towards efficient usability on several thousand processors.

3.5 Porting to SGI Altix

Since LRZ's new SGI Altix platform is expected to soon become a very strong machine in the DEISA infrastructure, GENE10 should also be ported to that architecture.

The porting work carried out so far on a small machine required adaptations at different levels. A new platform-specific makefile had to be created, and since the available FORTRAN compiler on ALTIX is the Intel FORTRAN compiler the compiler switches (in the makefile) had to be adapted to this compiler.

The GENE10 code used Bessel functions from the NAG library. Since the NAG library was not available on the ALTIX platform, three new subroutines had to be written with an implementation of these Bessel functions required for GENE10.

GENE10 so far could either use FFT-routines from the FFTW package or from the IBM proprietary ESSL-library. On ALTIX systems the optimized scientific library MKL from INTEL is available. It contains the necessary FFT-routines, but with a different interface. Consequently, the calling interface to the FFT-routines had to be adapted to MKL.

Tests of the ported version will start soon.

4 Final Report on ORB5 Turbulence Code

4.1 Introduction

The ORB code solves the nonlinear gyrokinetic equations pertinent to the study of transport-related instabilities and turbulence. It uses a particle-in-cell (PIC) time evolution approach, and takes advantage of all the recent techniques of noise reduction and control in PIC simulations. In particular, it uses a statistical optimisation technique that increases the accuracy by orders of magnitude. In the past the TORB code has been successfully enabled for DEISA scaling up to 4096 processors. Its physical simulation domain is restricted to cylindrical symmetry, the so-called theta-pinch configuration. This can be interpreted as a limiting case of the more complex tokamak symmetry which has in addition varying equilibrium quantities in the poloidal plane. The ORB5 code can deal with the more general tokamak case. Thus, ORB5 is able to simulate plasmas of higher complexity which can be used to simulate effects such as long-living zonal flow structures analogous to those seen in the Jovian atmosphere, or so-called Geodesic Acoustic Modes (GAM). Of course, a higher complexity of the simulation model has its impact on the complexity of the programming model. Additional effort therefore has to be given to the ORB5 code to enable it to run with high scalability on the DEISA hardware.

4.2 Single processor optimization and portability

The enabling of the ORB5 code for DEISA started with a scan for potential improvements of the single processor performance. For identification of the most CPU time consuming routines the code has been instrumented by the simple but efficient perf library. The perf library has been programmed by the RZG scientist Reinhard Tisma and

gives information about the time spent and the Mflop rate achieved in a detected region. The regions have to be defined by hand and can be marked by arbitrary labels. Table 1 Performance output of the ORB5 code for a 256 processor run, shows the perf output for a short 256 processor run (IBM POWER4, 1.3 GHz) of the ORB5 code lasting just one time step,. Hence, the initialization phase (init) is disproportionately more expensive than the main loop (mainloop) compared to the many time steps that would be taken in a real simulation.

Subroutine	#calls	Time(s)	Inclusive		Time(s)	Exclusive	
			%	Mflop/s		%	Mflop/s
init	1	282.892	204.4	203.955	14.282	10.3	108.032
initfiel	1	133.756	96.7	286.017	127.999	92.5	269.768
wsmpl_ini	1	5.757	4.2	647.316	5.757	4.2	647.316
initiation	1	9.350	6.8	-0.022	9.350	6.8	-0.022
inital	1	93.223	67.4	136.265	93.223	67.4	136.265
setrho	5	69.261	50.1	365.235	0.575	0.4	110.054
grid	5	65.584	47.4	382.804	0.210	0.2	19.717
cub	5	63.139	45.6	397.546	63.139	45.6	397.546
guard	10	2.286	1.7	0.438	2.286	1.7	0.438
fourcol	20	0.209	0.2	811.927	0.209	0.2	811.927
pptransp	20	4.195	3.0	-0.006	4.195	3.0	-0.006
poisson	5	6.410	4.6	98.027	2.431	1.8	25.733
scal_sol	5	1.915	1.4	99.639	1.915	1.4	99.639
wsmpl_sol	5	0.712	0.5	467.276	0.712	0.5	467.276
mainloop	1	138.372	100.0	477.808	0.036	0.0	0.005
onestep	1	136.329	98.5	481.024	8.593	6.2	20.329
push	4	74.072	53.5	600.169	62.977	45.5	702.814
isort	1	11.095	8.0	17.514	11.095	8.0	17.514
parmove	4	10.273	7.4	21.144	10.273	7.4	21.144
dump_dia	1	2.006	1.4	267.946	2.006	1.4	267.946

Table 1 Performance output of the ORB5 code for a 256 processor run

Two bottlenecks have been identified and improved. One was the implementation of the FFT (fast Fourier transform) used in the code (fourcol) and the other the cache sort of the Monte Carlo particles (isort).

A module has been written for the FFT which contains different interfaces to specialised FFT libraries. The original code included the source code of an implementation of the FFT and had a very poor Mflop rate of less than 100 MFlop/s.

With the new FFT module both the FFTW v3.1 (Fastest Fourier Transform in the West, <http://www.fftw.org>) and the IBM ESSL (Engineering and Scientific Software Library, <http://www-03.ibm.com/systems/p/software/essl.html>) are available. The FFT from the commercial ESSL library gives slightly better results than the publicly available FFT of the FFTW library. A maximum Mflop/s rate of approximately 800 MFlop/s has been achieved per processor. To improve the portability of the code to Linux platforms, an additional interface for the FFT of the Intel MKL (math kernel library, <http://www.intel.com/cd/software/products/asmo-na/eng/perflib/mkl/index.htm>) has been included in the FFT module. It has been tested as part of the ORB5 code under Linux on ia32 and em64t hardware. All the FFT library routines presented here, FFTW, ESSL and MKL, have the advantage that they are not restricted to vector lengths of powers of two as was the case for the original implementation of the FFT algorithm. This results in a much higher degree of flexibility when choosing the grid resolution of the electrostatic potential.

The cache sort algorithm sorts the Monte Carlo particles relative to their position in the grid cells of the electrostatic potential. Hence, a very high cache reuse of the electrostatic field data is achieved which significantly improves the Mflop rates of the routines `setrho` and `push`. The overhead caused by the sort routine itself (`isort`) had been minimized. A switch has been implemented to give the choice to enlarge a work array for the sorting process which can speed up the `isort` routine by a factor of three. Usually the resident memory size of the simulation is small enough to take advantage of this new feature.

4.3 ORB5 enabling: optimisation of the scaling property

4.3.1 *The domain cloning concept*

In analogy to the theta-pinch version, TORB, the domain cloning concept has been implemented in the ORB5 code to optimise scaling and decouple the selectable grid resolution from the number of processors used for the simulation. It is a relatively new parallelisation concept which builds on the following two decomposition techniques.

For the domain decomposition, different portions of the physical domain and of the corresponding electrostatic field grid are assigned to different processors together with the Monte Carlo particles that reside on them. As particles move from one region to another, they are communicated to the processor which is associated with the new region.

The other concept called particle decomposition uses the following strategy: the whole spatial grid is assigned to every processor, but each processor takes care of only a subset of the particle population. Partial contributions to the ion density, which is required to update the electrostatic field, are communicated among processors and summed together (using a global sum).

The domain cloning concept is a combination of particle decomposition and domain decomposition including both of them as limiting cases. With this strategy, which was introduced in Ref. [5], the simulation domain is cloned, i.e. multiple copies of the same domain are made. The processors are divided into a number of groups equal to the number of clones. Each group of processors is assigned to one of the domain clones. The concept of domain cloning is illustrated in Fig. 3 as an example of two clones of a periodic cylindrical simulation domain. The overall number of 16 processors is divided into two groups of processors and each group is assigned to one of the domain clones. Each domain clone is loaded with its own set of particles and a one-dimensional decomposition is performed such that each processor has a corresponding subdomain clone (for details see R. Hatzky, *Parallel Computing*, in press).

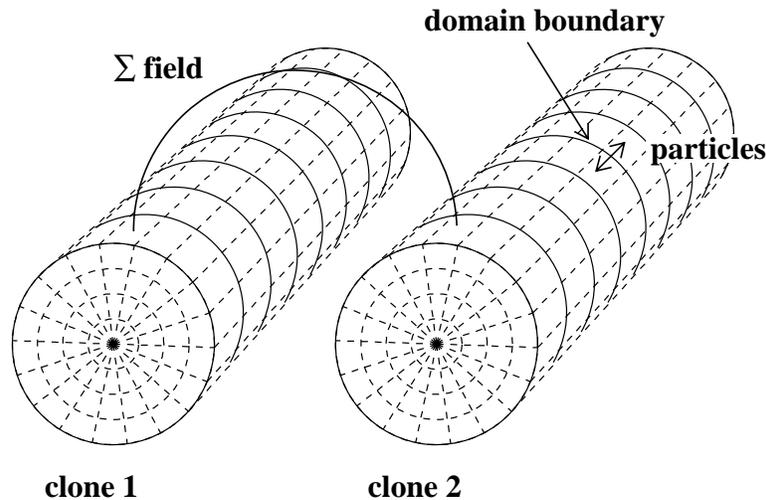


Fig. 3 A two-fold domain clone of the simulation domain. Particle communication occurs inside the clones; field data are summed over subdomain clones

4.3.2 The initial concept for a scalable solution of the field equations

Compared to the theta-pinch version, TORB, the calculation of the electrostatic potential in the Tokamak version, ORB5, is much more complicated. It requires the construction and solution of two different matrices: one is dense and has band format and is quite large; the other has sparse format and is more than an order of magnitude smaller. In general, both matrices can be very large as they follow from the three dimensional geometry of the turbulence. However, these matrices are split in the ORB5 code into a decoupled system of two-dimensional matrices by imposing an FFT. The positive effects for the parallelisation strategy are obvious. Instead of parallelising the solution of two large 3D matrices it is possible to do a trivial parallelisation over the Fourier modes. Perfect scaling is guaranteed as long as each processor solves for the same number of Fourier modes by using the back solve of the 2D matrix. The disadvantages are that each processor would have to store the full 2D matrix and that the domain cloning concept seems not to fit naturally into this parallelisation strategy. One of the advantages of the domain cloning concept is the possibility to run simulations with much larger numbers of processors than Fourier modes. Thus, the work load of the trivial parallelisation could not be shared equally among the processors.

As a first step to overcome these obstacles the code developers had the idea of using the publicly available ScaLAPACK band matrix solver (Scalable LAPACK, http://www.netlib.org/scalapack/scalapack_home.html) to parallelise both matrix solves along the direction of the corresponding subdomain clones. For example, four clones each solving of a Fourier mode was accomplished by four processors and accordingly each copy of the 2D matrix was shared over the four domains. The problem is that storing the matrices in band matrix format is only necessary for the dense 2D matrix whereas for the sparse matrix all the filling zeros are stored unnecessarily.

The resulting memory consumption for storing the band matrices can become quite large so that a large number of domains would be necessary to distribute the matrix over many processors. Otherwise, storing would be impossible or at least would lead to a severe memory balance problem especially on machines which do not have a large

shared memory. More importantly the number of domains should not be constrained by memory management purposes.

4.3.3 The enhanced concept and its implementation

In close collaboration with the code developers, especially Alberto Bottino, the final concept was developed and implemented to completely decouple the solving of the dense and sparse matrices.

The dense matrix is still solved by the ScaLAPACK band matrix solver routine PDPBTRS. However, an additional MPI (Message Passing Interface) communicator has been defined which is not restricted by the domain cloning concept. The matrix can now be parallelised over a subset of processors of all the processors dedicated to the job. The divide and conquer algorithm used by the ScaLAPACK band matrix solver restricts the maximum number of processors involved in the parallelisation to the size of the band width. The maximum number for a matrix solve typically does not exceed 32 processors for ORB5 simulations. This is much smaller than the typical number of processors used for the whole simulation which is in the range of many hundreds of processors. Although all processors cannot be used for the solution of the dense matrix it is at least possible to distribute the matrix over the computer so that the memory consumption is balanced. For example, on a cluster of a symmetric-multiprocessor (SMP) computers of 16 nodes the matrix would be distributed over all nodes by using just two processors on each node. So in principle the memory dedicated to all processors of the simulation can be used to store the matrix.

With the knowledge that the solution of the dense matrix with ScaLAPACK does not scale to high numbers of processors we tried to speed up at least this part of the code as much as possible. For that reason we used the IBM PESSL (Parallel ESSL) implementation of ScaLAPACK on our IBM hardware. Unfortunately this implementation of ScaLAPACK does not support the BLACS standard (Basic Linear Algebra Communication Subprograms, <http://www.netlib.org/blacs>) as used in the publicly available ScaLAPACK v1.7. After communicating with IBM support it became clear that the BLACS context had to be modified throughout the code to get the IBM PESSL implementation. Finally, for a particular case, the performance of the ScaLAPACK band matrix solver routine did not speed up with the IBM PESSL implementation. Thus, it would be advantageous to find an algorithm scaling to much higher numbers of processors than the algorithm used in ScaLAPACK. To our knowledge ScaLAPACK is state of the art concerning the solution of equations of dense matrices with publicly available libraries. A better performing library could perhaps be found but this in itself would be a research project for the future..

The sparse matrix is solved by the direct parallel sparse solver IBM WSMP (Watson Sparse Matrix Package, <http://www-users.cs.umn.edu/~agupta/wsmp.html>). It has the great advantage that the storage of the matrix and the time for the back solve can be drastically reduced. This is especially important for simulations with large grids where the matrices become more and more sparse compared to the band format of ScaLAPACK. The initial idea of parallelising over the direction of the corresponding subdomain clones can be still kept. However, this type of parallelisation is already quite sophisticated with the usage of the MPI communicator already defined for the domain cloning. Hence, a close collaboration with the IBM developer of WSMP, Anshul Gupta, became necessary to amend many documentation and implementation problems. More than 60 e-mails have been exchanged in the last two months. The parallel sparse solver can also use

multiple right hand sides which means that the back solve can be used simultaneously for more than one Fourier mode, significantly improving the Mflop rate.

4.3.4 Evaluation of enhanced ORB5

If one compares the initial concept of using exclusively ScaLAPACK with the mixed concept of ScaLAPACK/IBM WSMP it can be shown that the time spent for solving the two matrices reduced by a factor of 6 for the example of a $256 \times 256 \times 256$ grid. The relative time consumption on the total main loop wall clock time becomes less than 2%. On one hand the code becomes faster, and on the other hand it is now possible to calculate much larger grids which results in the simulation of larger plasma volumes. This is of special interest as larger plasma volumes make the turbulence simulations more physically relevant.

The hard scaling property (with constant problem size) has been evaluated on the IBM POWER4 1.3 GHz super computer of RZG. For a typical simulation run over one time step a main loop wall clock time of 138.4 s with 256 processors and 72.8 s with 512 processors has been achieved. As one can see from Fig. 4 this gives a promising scaling factor or 1.9.

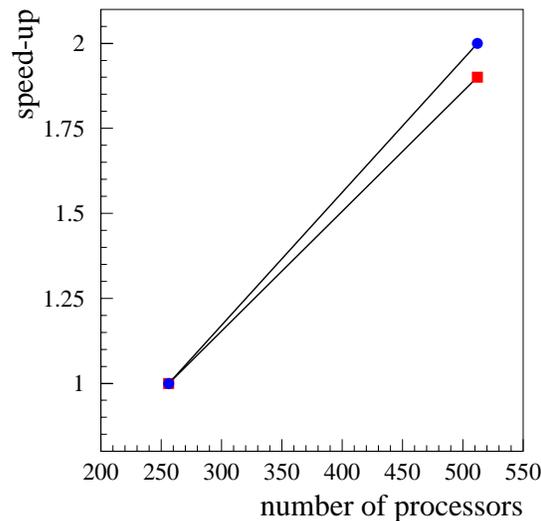


Fig. 4 Hard scaling of the ORB5 code on the IBM POWER4 1.3 GHz super computer of RZG; Squares: measured results for 256 and 512 processors, normalized to the 256 processor run Circles: perfect linear scaling for comparison

As PIC simulations suffer from an inherent statistical noise problem, larger numbers of processors would be used for simulations with larger number of particles. Hence, a weak scaling would be more realistic where the number of particles is increased in proportion to the number of available processors. Hence, it is appropriate to derive from a hard scaling factor of 1.9 an even better weak scaling property of the ORB5 code up to several thousands of processors.

5 Further relevant application candidates from plasma physics

As listed in the 24 month milestone, besides testing of further code candidates for grid enabling (done with codes GENE and ORB5), other codes should have been assessed and put in a priority list. Design of the principle structure of a portal has already been fully addressed in the month 12 and month 18 deliverables.

The priority list has been built with the support of scientists from several important European institutions, after a screening for further applications from plasma physics to be supported in DEISA has taken place. Among these institutions are CEA/Cadarache, UKAEA/Culham, CRPP/Lausanne, IPP/Garching and IPP/Greifswald.

A request for immediate action has been received from a Swiss/German collaboration (CRPP/IPP) with respect to the EUTERPE code. This code is capable of simulating micro-instabilities in global three-dimensional configurations. Hence, simulations require a massive parallel compute environment as provided by DEISA.

EUTERPE has been selected as the next code to be supported for DEISA.

A further request for JRA3 support at a later stage has been launched for the HAGIS code. HAGIS is based on a nonlinear 3D wave-particle interaction model evolving self-consistently the linear MHD perturbations in the simulated system. Hence the code has both ITER relevance (tokamak geometry) and is also able to simulate the 3D stellarator geometry.