



CONTRACT NUMBER 508830

**DEISA**  
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR  
SUPERCOMPUTING APPLICATIONS**

**European Community Sixth Framework Programme**  
**RESEARCH INFRASTRUCTURES**  
Integrated Infrastructure Initiative

JRA3: Final report on simulation code TORB, progress report  
on new ORB5 code for extreme computing within DEISA.

Deliverable ID: DEISA-D-JRA3-3

**Due date : Oct, 31, 2005**  
**Actual delivery date: November 25, 2005**  
**Lead contractor for this deliverable: RZG, Germany**

**Project start date : May 1<sup>st</sup>, 2004**  
**Duration: 4 years**

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



## Table of Contents

Table of Contents .....	1
<b>1 INTRODUCTION .....</b>	<b>2</b>
1.1 Executive Summary.....	2
1.2 References and Applicable Documents .....	2
1.3 List of Acronyms and Abbreviations .....	2
<b>2 FINAL REPORT ON TURBULENCE CODE TORB .....</b>	<b>4</b>
2.1 Provision of a Portal for the ORB code suite .....	4
2.2 TORB code preparation for multi-site usage .....	8
2.3 Scaling of TORB on AIX systems.....	12
2.4 Scaling of TORB on a Linux system.....	15
<b>3 ORB5 CODE .....</b>	<b>17</b>
3.1 ORB5 simulation model.....	17
3.2 Study for ORB5 enabling.....	17
<b>4 EUROPEAN PLASMA PHYSICS COMMUNITY .....</b>	<b>18</b>

## Introduction

### 1.1 Executive Summary

This document “JRA3: Final report on simulation code TORB, progress report on new ORB5 code for extreme computing within DEISA” is the 18-month deliverable DEISA-D-JRA3-3 for Joint Research Activities in Plasma Physics. It describes all actions taken with the TORB simulation code which was selected as a DEISA proof-of-concept application for the plasma physics scientific community. Furthermore, first actions taken with ORB5 – an ITER relevant code – are described.

### 1.2 References and Applicable Documents

- [1] <http://www.UNICORE.org/documents/UNICOREPlus-Final-Report.pdf>
- [2] Allfrey, S.J. and Hatzky R.: A revised delta-f algorithm for nonlinear PIC simulation. *Comp. Phys. Commun*, **154**: 98, 2003
- [3] 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
- [4] 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
- [5] <http://sourceforge.net/projects/unicore/>
- [6] Kim, C.C. and Parker S.E.: Massively Parallel Three-Dimensional Toroidal Gyrokinetic Flux-Tube Turbulence Simulation. *J. Comp. Phys.*, **161**: 589, 2000

### 1.3 List of Acronyms and Abbreviations

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

GPFS	Global Parallel File System
HPCx consortium	HPCx consortium, see <a href="http://www.hpcx.ac.uk">http://www.hpcx.ac.uk</a>
IPP	Max-Planck-Institut für Plasmaphysik, Garching, see <a href="http://www.ipp.mpg.de">http://www.ipp.mpg.de</a>
ITER	International Thermo-Nuclear Experimental Reactor
JRA	Joint Research Activity
RZG	Rechenzentrum Garching, see <a href="http://www.rzg.mpg.de">http://www.rzg.mpg.de</a>
SMP	Symmetric MultiProcessor
TSI	Target System Interface: UNICORE server component which represents the interface to the batch sub-scheduler
UNICORE	UNiform Interface to COmputing Resources, see <a href="http://www.unicore.org">http://www.unicore.org</a>

## 2 Final Report on Turbulence code TORB

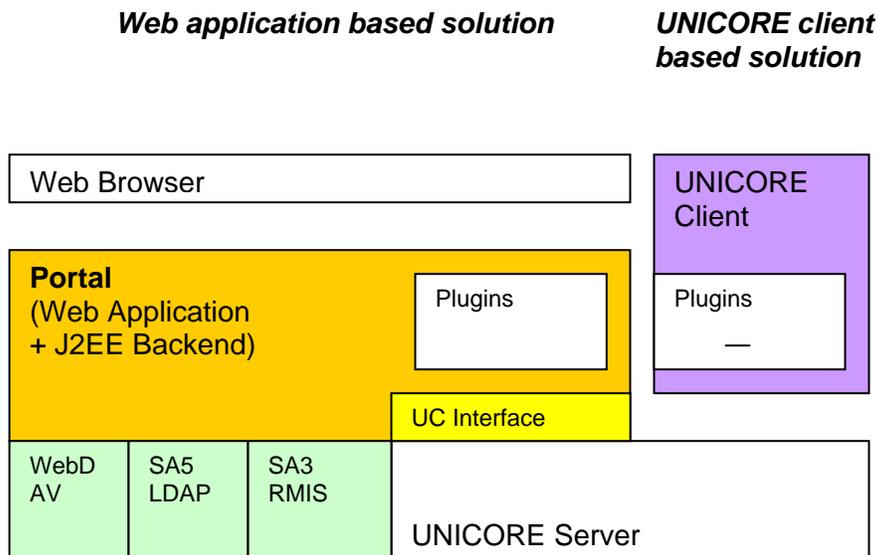
The ORB code suite is a set of related codes solving identical gyrokinetic equations for different geometrical configurations. For the JRA3 activities with ORB as a prototype, the theta-Pinch geometry was selected for DEISA, as already explained in the JRA3 6-month deliverable DEISA\_D-JRA3-1. This version is referred to as TORB, since new versions are under development.

Parts of this chapter and figures have already been presented in the 12-month deliverable, but are repeated here for the final report.

### 2.1 Provision of a Portal for the ORB code suite

New users and scientists not familiar with the instrumentation of the ORB code suite should benefit from a portal which provides a uniform access to the DEISA infrastructure. Such a portal has to facilitate the application handling, i.e. job preparation, job submission and output handling. This is a way to hide details of the infrastructure from the user.

Two solutions have been developed and implemented. Both rely on UNICORE, a key middleware component of DEISA. This is illustrated in Fig. 1.



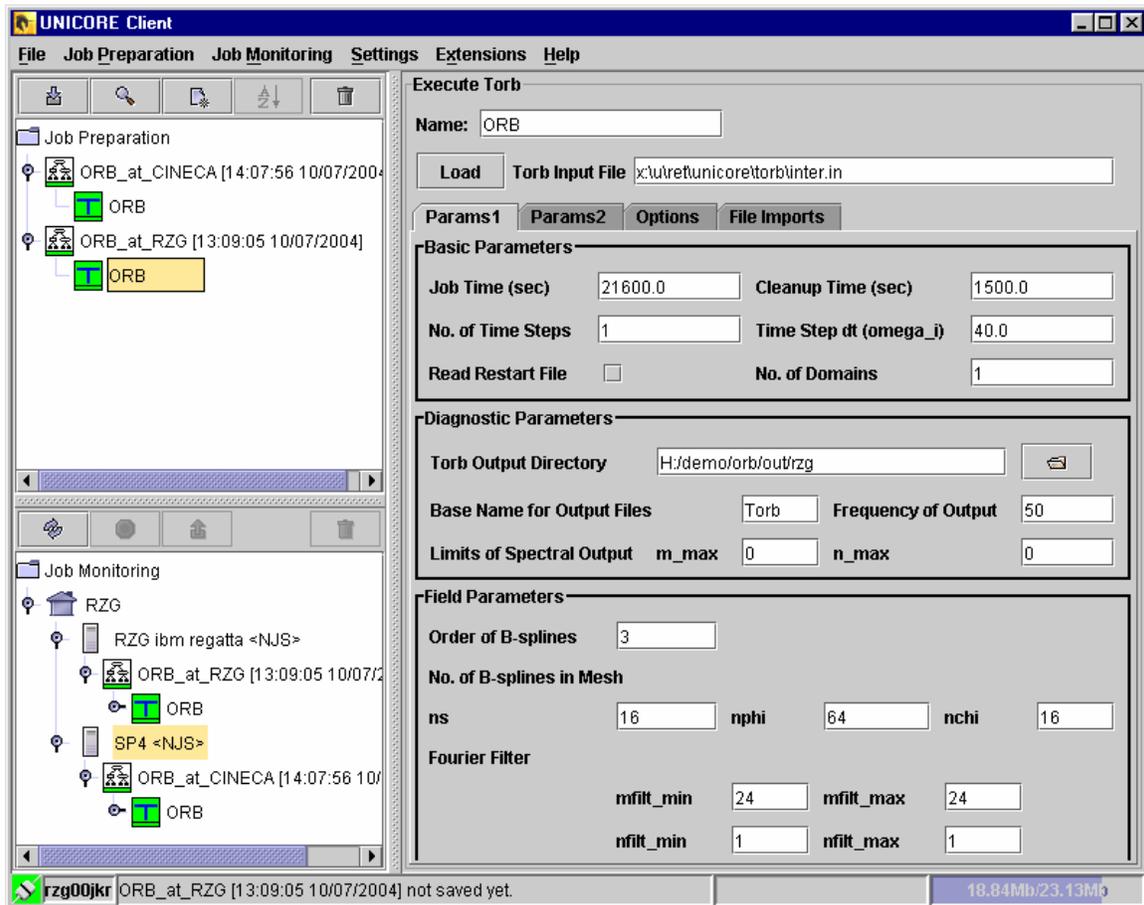
**Fig. 1**

DEISA portal options for JRA3 (and JRA1)

Right side: Unicore client plus plug-in (UNICORE client compatible solution)

Left side: More general approach with web application interface to UNICORE server, including plug-in(s)

The desired functionality can principally be provided by special features of a part of the UNICORE software package, the so-called plug-ins for the UNICORE client. An advanced plug-in for TORB has been developed for the UNICORE client, focusing on ease of use. A TORB-specific form sheet has also been developed. It is designed so that it already contains reasonable default values, generic to most users, which can be individually modified and stored. The form can now also be backfilled with the content of pre-existing input files, since a parser for the input file has been written.



**Fig. 2** TORB plug-in for the UNICORE client

This setup is assembled into a TORB-compliant parameter file that is transferred to the relevant Target System Interface (TSI) before the TORB executable is submitted to the Batch Sub-System (BSS) on the target machine.

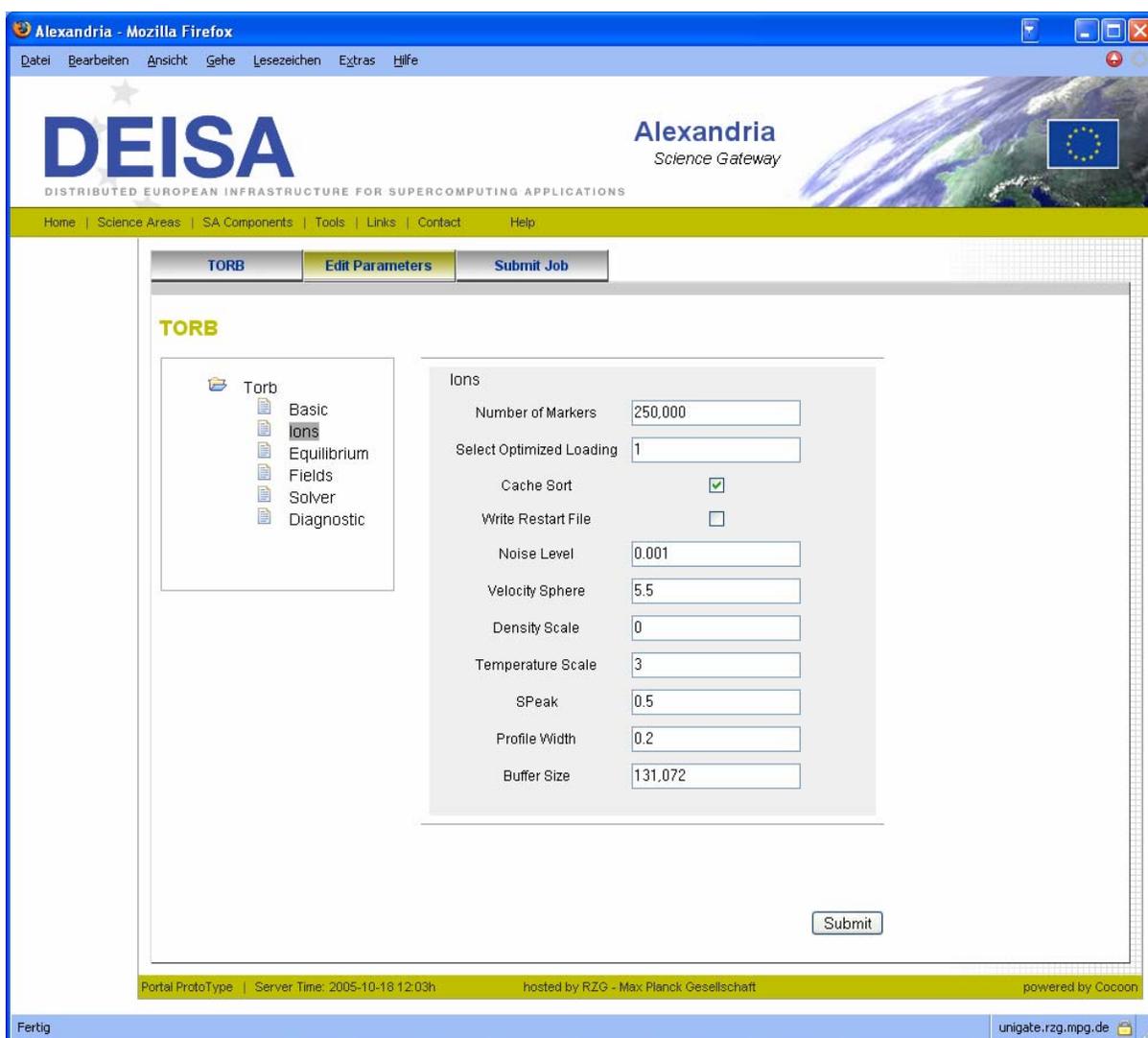
Authentication by certificates, job preparation, submission, execution and result retrieval have been tested. The functional components are verified by a number of test cases.

The initial prototype has been converted to a production version that can readily be used on DEISA core sites, together with the UNICORE client and the associated ORB plug-in.

The more general approach to portal functionality is via web applications which have also been selected for applications in JRA1 for materials science. This approach, and the reasons for it, are described in detail in the 12-month deliverable of JRA1 (DEISA\_D-JRA1-2.doc).

Essentially, the DEISA infrastructure access has been transferred from the UNICORE client to a web application at the server side, directly interfaced to the UNICORE server. This portal can now be directly accessed with a standard web browser. No further client software installation is needed at the user side, as illustrated in Fig. 1.

The new general portal approach has been implemented for TORB. The TORB plug-in has been adapted and TORB can now, in principle, be started in DEISA core sites using a simple standard web browser.



**Fig. 3**

TORB plug-in integrated in the web application/portal facilitates TORB application start via web browser Mozilla.

Though not promised for this deliverable, the portal concept for TORB has further been tested by developing another plug-in for a second important plasma physics code, GENE. Plug-ins have been built both for the UNICORE client and the web application

The plug-in for the plasma simulation code GENE implements 3 panels for submission of a job.

- **Forms panel defining the problem size**  
It provides a form to enter the sizes of the dimensions of the arrays and the size of the processor grid.
- **Forms panel for physical parameters**  
This form provides input for the physical parameters.
- **Editor panel for config.h**  
The file config.h enables or disables different parts of the code and sets the compile time parameters. It should be changed only by experts of the code.

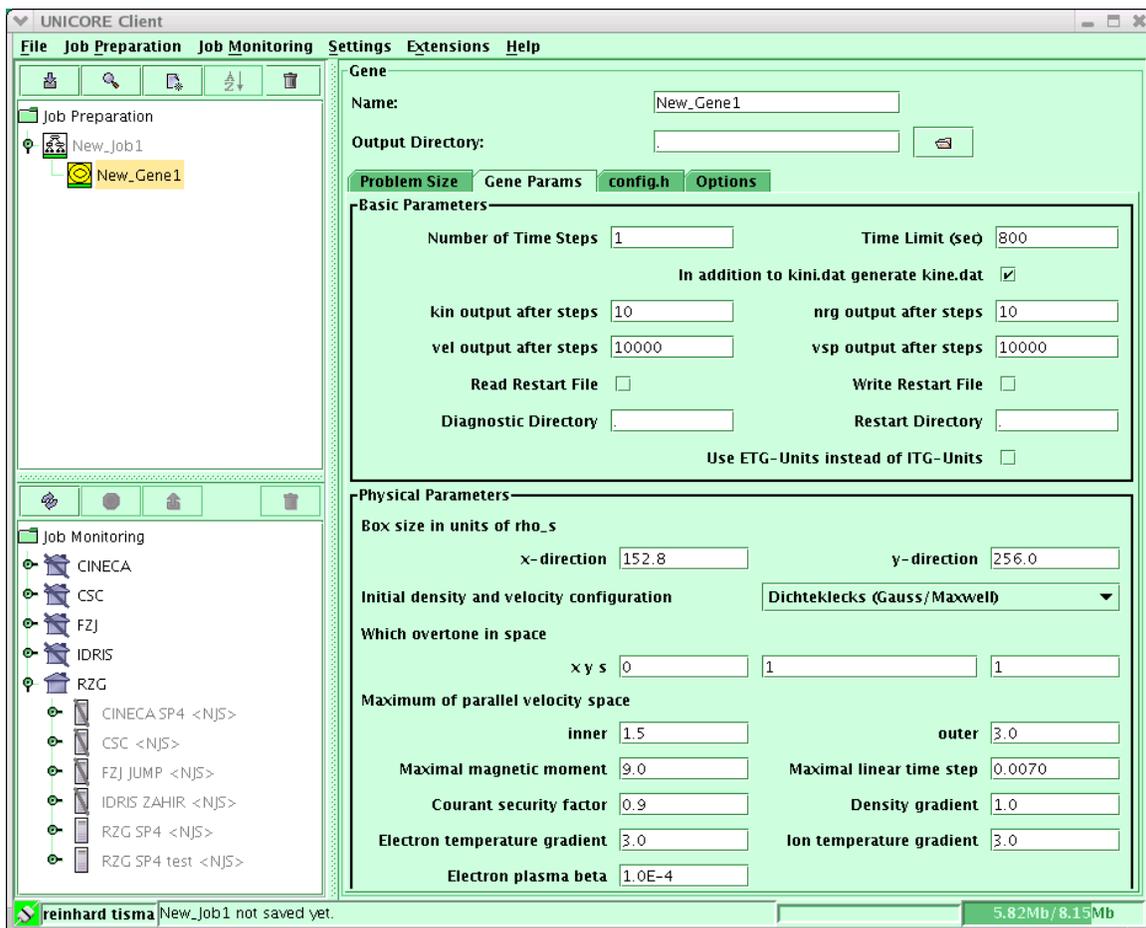


Fig. 4 GENE plug-in for the UNICORE client

This plug-in functionality has also been ported to the web application. When addressed via a web browser, the plug-in appears as displayed in Fig. 5.

**Fig. 5**

GENE plug-in integrated in the web application/portal facilitates GENE application start via web browser (here: Mozilla)

## 2.2 TORB code preparation for multi-site usage

The term multi-site usage could be used in different contexts:

- synchronous multi-site usage of coupled applications;
- a single application being spread over more than one site;
- asynchronous or sequential multi-site usage in a workflow of different functions;
- a continuous series of jobs and optional post-processing and/or visualization of results.

Since synchronous single application distribution over different sites (“metacomputing”) has not become a major operational model of DEISA, but job re-routing across sites will be a focus, we have addressed and tested sequential multi-site usage. The aspect of being able to use large resources for a single application (one original motivation for

single application metacomputing in the late nineties) is addressed in chapters 2.3 and 2.4. Scalability expansion has become a new perspective with the availability of more than a thousand processors in a single site. Access to such large-scale computing resources via DEISA has already been realized with ECMWF and BSC, and with LRZ as of next year.

Here we have focused on starting TORB execution at one site, writing a restart file, doing a continuation run at the same site, continuing the simulation at a second site, and starting a post-processing application for visualization at a third site.

Such a usage type has been prepared and conducted involving compute systems at three of the DEISA core sites, IDRIS, FZJ and RZG. CINECA systems will be included in the next test series.

TORB multi-site usage could already benefit from the new global file system multi-cluster GPFS among the DEISA core sites, and has also been used as the first real application from a JRA to test the setup.

Due to the ease of use of GPFS it was not necessary to adapt the TORB code in a special way. The files are read and written to the GPFS as it would be a local file system. However the I/O of the TORB code can be optimized to support the soft- and hardware layers of the GPFS.

The number of the restart files of the TORB code scales with the number of processors used. Hence, at the end of a 512 processor run, all the processors start simultaneously to write their own output file into the GPFS. In this situation it is the responsibility of the GPFS to achieve optimal performance by managing the communication to and writing on the hard disks. Instead of writing all output files simultaneously, the output routines have been enhanced to give the possibility to write only a certain number of files simultaneously. Therefore an overloading of the file system with data of hundreds of files written at once is in principle excluded.

Thus the parameter which fixes the number of simultaneous output files can be used to optimize the parallel output process to the GPFS.

**1. Start of TORB simulation run****2. Continuation of TORB sim. run****3. Start of TORB post-processing run****Figure 6**

Multi-site usage of TORB involving the GPFS new global file system among DEISA core sites: start of a TORB simulation at IDRIS; continuation of simulation at FZJ; post-processing application at RZG.

**Multi-site tests with TORB**

TORB has been tested on the dedicated systems where early versions of the multi-cluster GPFS were available. Typically, eight processors have been used for a problem size of 12 GB, a simulation with 20 million, particles, and reading/writing restart files of about 2 GB.

The first TORB simulation was started at IDRIS (Orsay, France) writing/reading data to/from GPFS disks located at IDRIS (three continuation runs were done).

The following lines were taken from the log file, referring to restart file I/O:

```
Particle out wall clock time = 8.3466280E+00 => 224.6MB/s
```

```
Particle in wall clock time = 9.6171534E+00 => 195.0MB/s
```

```
Particle out wall clock time = 8.4093769E+00 => 223.0MB/s
```

```
Particle in wall clock time = 9.9078562E+00 => 189.2MB/s
```

```
Particle out wall clock time = 8.5982108E+00 => 218.1MB/s
```

The TORB simulation was then continued at FZJ (Juelich, Germany), writing/reading data to/from GPFS disks located at IDRIS (again three continuation runs were done).

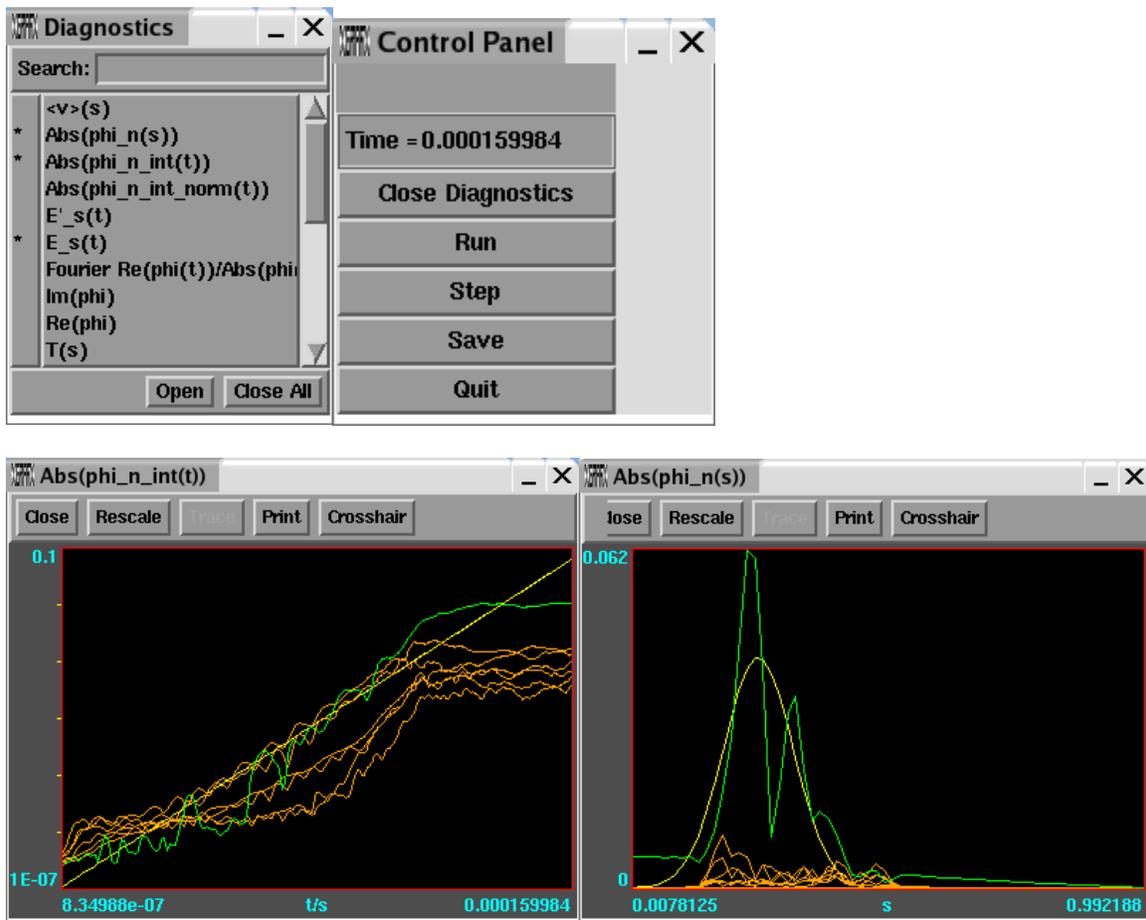
The following lines were taken from the log file, referring to restart file I/O:

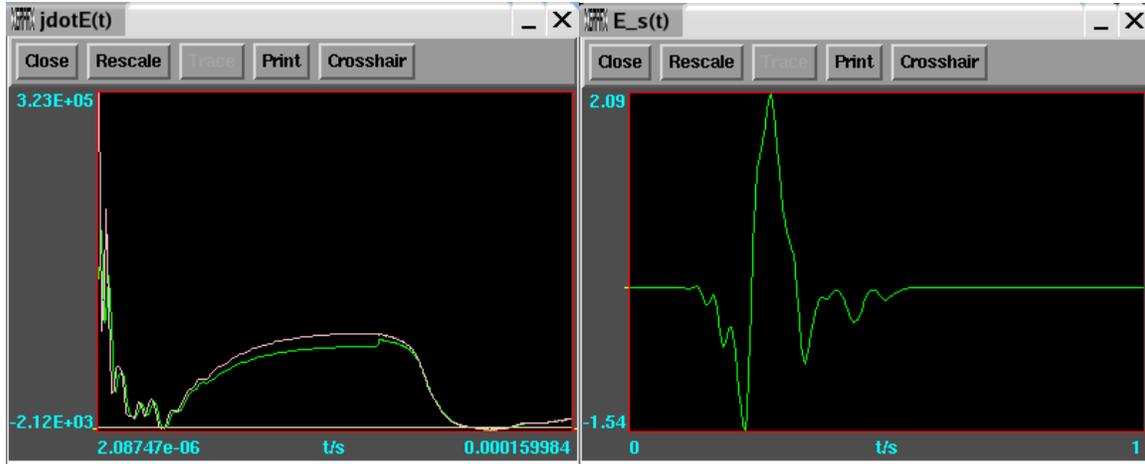
```
Particle in wall clock time = 1.9522469E+01 => 96.0MB/s
Particle out wall clock time = 3.6178846E+01 => 51.8MB/s
```

```
Particle in wall clock time = 1.7238927E+01 => 108.8MB/s
Particle out wall clock time = 4.2755378E+01 => 43.9MB/s
```

```
Particle in wall clock time = 1.6924593E+01 => 110.9MB/s
Particle out wall clock time = 3.5461993E+01 => 52.9MB/s
```

In a third step, the post-processing tool ANGY for TORB output data evaluation was started at RZG, accessing GPFS data located at IDRIS. The six screenshots in Fig. 7 document this post-processing step.





**Fig. 7**

Visualization of various TORB simulation output parameters with analysis tool ANGY. Screenshots of 6 different windows are displayed.

For its graphical user interface, the data analysis tool ANGY uses the graphical software package xgrafx written by the Plasma Theory and Simulation Group (PTSG) at UC Berkeley (URL: <http://langmuir.nuc.berkeley.edu/pub/codes/xgrafx/>). The flow chart of ANGY was adapted to upgrade the xgrafx package from version 1.94 to the recent version 2.60.

A similar scenario has also been successfully demonstrated during the June 2005 review meeting in Paris.

A 256 processor job was running at RZG (no use of RZG data) for which the data were read from IDRIS (France), where they had been produced by a former job.

The application wrote the restart data to the Jülich GPFS file system and the result data to RZG. The resulting data was then interpreted and displayed in CINECA reading the data from RZG (Germany).

## 2.3 Scaling of TORB on AIX systems

The nonlinear particle-in-cell code TORB is one of the potential candidates for the DEISA Extreme Computing Initiative. It uses a Monte Carlo particle approach to simulate the time evolution of turbulent field structures in fusion plasmas. In such simulations, a very large number of Monte Carlo particles is needed to simulate large physical domains with a very low level of statistical noise for long simulation intervals. The large amount of memory used for storing these hundreds of millions of particles is only made available by massively parallel computers with distributed memory architectures.

Although Monte Carlo codes have in general good scalability properties on parallel computers, it was not clear how TORB would scale up to thousands of processors. Due to Amdahl's law it is a nontrivial task to run codes efficiently on thousands of processors: such extreme scalability can be only reached by a small number of codes.

As a first step, the scaling properties of the TORB code measured on up to 512 processors which was limited by the largest available batch queue of the Garching computing centre RZG (see deliverable DEISA\_D-JRA3-1.doc). As a next step,

measurements on 1024 processors could be done at the supercomputer of the HPCx Consortium of which EPCC is a member. Finally, it was possible to use up to 2048 processors of the supercomputer of the European Centre for Medium-Range Weather Forecast (ECMWF). Performing such benchmarks on thousands of processors belonging to a clustered symmetric-multiprocessor (SMP) computer was only made possible since all named computer centres are members of the DEISA project.

Besides the usual changes to the batch scripts, some minor adaptations of the output format were made to handle processor numbers larger than 999 processors. More significantly, a major change became necessary to handle a number of Monte Carlo particles larger than  $2^{31}=2,147,483,648$ . For the first time the total memory of approximately 2 TByte available gave the opportunity to use 3.2 billion particles within a 2048 processor run of the TORB code which exceeded by far the  $2^{31}$  limit. The  $2^{31}$  particle limit is critical for the programming structure of the TORB code as the standard 4-Byte Integer representation of FORTRAN can not handle larger integer numbers without causing an overflow. Hence the parts of the code relying on 8-Byte Integer arithmetic had to be adapted. A general change to 8-Byte integers would have been more laborious because of the changes involved for calls to the Message Passing Interface (MPI) library interfaces which were programmed originally in TORB for 4-Byte Integers.

The time spent in sequential regions is reduced to a minimum for the TORB code. However, due to the basic structure of the particle-in-cell simulation, each Monte Carlo particles does not act completely independently from the others. Instead, the particles interact with the field resulting from a collective charge-assignment process so a parallel communication is unavoidable. An efficient implementation of this parallel communication is only possible on the deep algorithmic level.

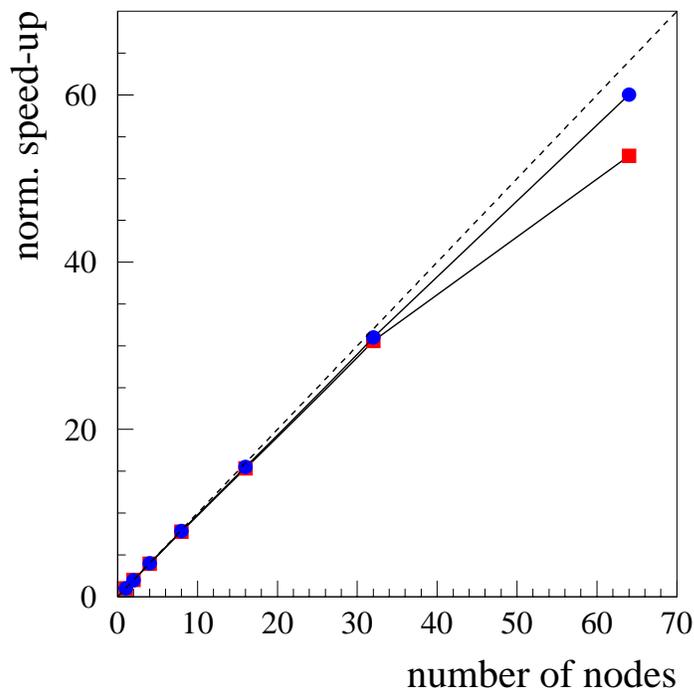
In 2000, a new approach to parallelization, domain cloning, was presented by C.C. Kim and S.E. Parker, *Journal of Computational Physics* 161 (2000) 589-604.

Domain cloning is an additional layer of parallelization, a supplement to one-dimensional domain decomposition, which gives the opportunity to optimize the scaling property of particle-in-cell codes such as the TORB code. In the past this parallelization concept was implemented into TORB at the Garching computing centre. Recently, in collaboration with the DEISA initiative, a further adaptation focused on clustered symmetric-multiprocessor computers has been developed, addressing the question of whether it would be possible to use the domain cloning concept on thousands of processors. As a result, a paper entitled "Domain Cloning for a Particle-in-Cell (PIC) Code on a Clustered Symmetric-Multiprocessor (SMP) Computer" was submitted to the journal *Parallel Computing*.

As mentioned above, the final results have been achieved on one of the two IBM Cluster 1600 supercomputer systems at the High Performance Computing Facility of ECMWF. Each separate cluster comprises 70 pSeries 690+ servers (compute nodes), each of which has 32 Power 4+ CPUs at 1.9 GHz. Up to 64 compute nodes with 32 GB memory each were available for the largest simulation. The first 1024 processors runs were conducted on the HPCx system comprising 50 IBM pSeries 690+ nodes, i.e. 1600 Power 4+ processors at 1.7 GHz, with a total of 1.6 TBytes of memory.

Starting with 50 million particles for a single node simulation the number of particles increases linearly with the number of nodes. The maximum number of nodes used is 64 which corresponds to a total number of 3.2 billion Monte Carlo particles in the simulation.

Figure 6 shows the speed-up with normalized simulation size as a function of the number of nodes  $n$ . The scaling proves to be excellent for up to 32 nodes. Only a further doubling of the number of nodes to  $n=64$  gives a significant decrease of the speed-up to  $53/64$ . The intensive allreduce communication (global sum) over the clones finally diminishes the very good scaling properties of the domain cloning concept. Nevertheless, a total flop rate of 1.5 Tflop/s was reached on 64 nodes (2048 processors) surpassing the Teraflop performance threshold. Two benchmark studies were undertaken at ECMWF, separated by a time period of half a year. The comparison of the results showed that the total flop rate increased by 0.2 Tflop/s due to better scaling for 2048 processors. The parallel efficiency of a 2048 processor job increased by 11 % to a value of 94 %. This significant improvement is a result of the optimizations of the system configuration performed on the ECMWF supercomputer over the past half a year.



**Figure 8**

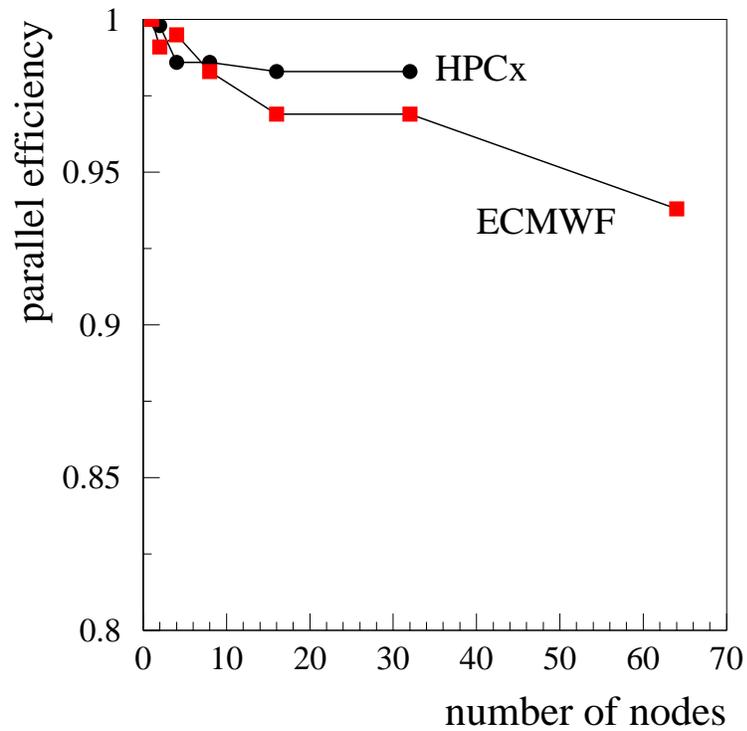
The speed-up of the TORB code with normalized simulation size vs. the number of compute nodes (with 32 processors each) involved. The problem size is scaled (increased) linearly with the number of nodes. The dashed line depicts ideal scaling.

Two benchmark campaigns have been made at the ECMWF within a time period of half a year, squares from March, 2005, cycles October, 2005. With 2048 processors (64 nodes), a parallel efficiency of 94 % was achieved in October, resulting in a performance of 1.5 TFlop/s.

After a careful study of the parallelization concept of the TORB code it has been possible to adapt the algorithmic structure in such a way that TORB can use efficiently the powerful IBM Power 4 hardware made available by many sites of the DEISA initiative.

# procs	Parallel efficiency at EPCC	Parallel efficiency at ECMWF
32	1,000	1,000
64	0,998	0,991
128	0,986	0,995
256	0,986	0,983
512	0,983	0,969
1024	0,983	0,969
2048		0,938

**Table 1**  
TORB scaling measurements at EPCC (HPCx system) and ECMWF



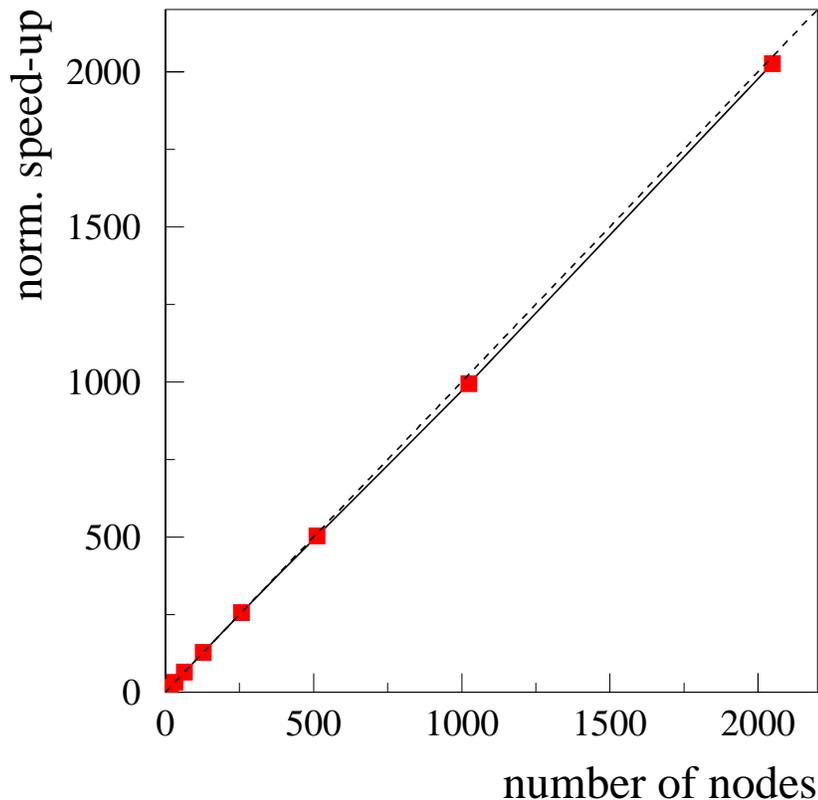
**Fig 9**  
Parallel efficiency of TORB scaling:  
Measurements at EPCC (HPCx system) and ECMWF

## 2.4 Scaling of TORB on a Linux system

The TORB code has been ported to the hardware of the Marenostrum super computer at the Barcelona Supercomputing Center (BSC), which is the most powerful compute system in DEISA and currently the largest machine in Europe. In a first step the makefile structure was adapted to the software environment on the IBM Power PC 970FX processors of the Marenostrum machine.

Due to the limitations of a relatively small stack size it was necessary to scan the whole code for automatic arrays which are located in FORTRAN on the stack. All large automatic arrays were transformed into allocatable arrays which are located instead on the heap. The limitation due to the relatively small stack size has been circumvented by using the much larger heap. Finally, the TORB code was tested with different compiler optimization options and different simulation test cases. As a result the TORB code is now running on the IBM Power PC hardware of Marenostrum under the Linux operating system, while previously it had only executed on DEISA sites running IBM Power 4 hardware under AIX.

Using the Myrinet interconnection network of the Marenostrum an excellent scaling up to 4096 processors was achieved. Nearly the whole machine of 4812 processors was occupied by a single TORB job, proving that the parallelization concept of the TORB code is also able to guarantee a nearly perfect scaling up to thousands of processors on Linux systems.



**Figure 10**

The speed-up of the TORB code with normalized simulation size vs. the number of compute nodes (with 2 processors each) involved. The problem size is scaled (increased) linearly with the number of nodes. The dashed line depicts ideal scaling.

With 4096 processors (2048 nodes), a parallel efficiency of 99 % was achieved, resulting in an estimated performance of 2.5 TFlop/s.

### **3 ORB5 Code**

Because of its scientific excellence and its potential for extraordinary parallel scaling properties, the ORB code became a candidate for the DEISA Extreme Computing Initiative. In the process of code development the ORB code has been enhanced to simulate more elaborated plasma configurations, for example as realized in the tokamak concept of ITER. In this phase the focus of the scientists had been especially on physical aspects and not the efficiency of the parallelization of the code. In the framework of the DEISA Extreme Computing Initiative, the focus will again be on the improvement of the scalability property of the new code version ORB5.

#### **3.1 ORB5 simulation model**

The ORB code solves the nonlinear gyrokinetic equations pertinent to the study of transport-related instabilities and turbulence. It uses a finite element, "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. The TORB code is able to perform simulations in 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. Thus, the ORB5 code 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. Although in relation to the TORB code a separate effort has to be taken for the ORB5 code to enable it to run with high scalability on the DEISA hardware. This includes the integration of the ORB5 code into the DEISA Common Production Environment (CPE).

#### **3.2 Study for ORB5 enabling**

Compared to the theta-pinch version, TORB, the communication pattern of the Tokamak version, ORB5, is much more complicated. The next study will give an overview of the parts of the programming model which have a potential for reducing the serial and communication intensive parts of the code to pave the way to high scalability. This is typically for PIC codes the so-called field solve where all the contributions of the single particles have to be summed up to calculate the self consistent fields.

The calculation of the electrostatic potential in the ORB5 code requires the construction and solution of two different matrix systems; one matrix has band format and is quite large; the other has sparse format and is more than an order of magnitude smaller. At the moment both matrices are solved in parallel with the ScaLAPACK band matrix solver routine PDPBTRS. This is convenient for the large band matrix but not for the much smaller sparse matrix. In that case it will be much more efficient to use a serial solver

with a direct method and sparse matrix format in combination with a parallelization over the different Fourier modes in the toroidal direction. Hence a communication overhead will be completely avoided. As a consequence the sparse matrix has to be stored on each processor but the memory consumption will not be much larger than using the inconvenient band matrix format for the sparse matrix and to distribute it by ScaLAPACK in parallel over the processors. The sparse solver to be selected should be of high performance on the DEISA hardware.

In addition, the domain cloning concept should be properly implemented in this field solve operation. It should be applied in the distribution of the physical domain but not necessary in solving the large band matrix. Here it seems to be advantageous to construct another MPI communicator to solve the large band matrix not only with the processors of each clone but over a larger number of processors. Thus the processors of several clones are combined to work in parallel to solve the large band matrix. In contrast the classical domain cloning concept would solve the same large band matrix multiple times on each clone.

## 4 European Plasma Physics Community

Preparation of the ORB code suite for adequate usage in DEISA helps with the expansion of the European user community. Developers are located at: IPP, Greifswald, Germany; IPP, Garching, Germany; and CRPP, Lausanne, Switzerland. With the help of principal investigators of this JRA from Germany and Switzerland, contacts with two interested groups at Cadarache in France have been initiated.

A trans-national European collaboration for developing the TORB code for stellarator configurations has now been established between a group from Spain (CIEMAT, Madrid) and the IPP, Greifswald. Development has started at IPP, Greifswald, with simulations running on the supercomputer at RZG, Garching. After the TORB code was ported to the Marenostrum supercomputer of BSC by this DEISA JRA effort (see 2.4) simulations could be continued at the BSC. This was realized by moving the restart files of the simulation to BSC where computing capacity was provided to continue the 512 processor job with a higher throughput rate than would have been the case at RZG. The results of this simulation have already been presented at the IAEA Technical Meeting on Innovative Concepts and Theory of Stellarators, Madrid, October 10-11, 2005.