Progress in NUCLEAR SCIENCE and TECHNOLOGY, Vol. 2, pp.700-705 (2011)

ARTICLE

HPC Challenges for Deterministic Neutronics Simulations Using APOLLO3® Code

Christophe CALVIN *

CEA/DEN/DANS/DM2S/SERMA, Commissariat à l'Energie Atomique, CEA/Saclay, 91191 Gif-sur-Yvette Cedex, France

The aim of this paper is to present some major HPC challenges for deterministic neutronics simulations and how these challenges are addressed in the APOLLO3® code. Different levels of HPC are illustrated on different kind of applications and parallel paradigms techniques in the frame of the APOLLO3® code. Results obtained for fuel load management using genetic algorithm, domain decomposition for transport solvers, GPU acceleration for the Boltzmann equation solution are given using from few cores to massively parallel computing using more than 10,000 cores.

KEYWORDS: APOLLO3®, neutronics, transport and simplified transport solvers, domain decomposition methods, genetic algorithm, hybrid programming, HPC, MPI, GPGPU

I. Introduction

The aim of this paper is to present some major challenges using High Performance Computing (HPC) for deterministic neutronics simulations and how these challenges are addressed in the APOLLO3® code.

APOLLO3® is a common project of CEA, AREVA and EDF for the development of new generation system for core physics analysis.

We can consider different targets of use for high performance computing in Reactor Physics. Depending on the target, different level and techniques can be used. Nevertheless, the different techniques will allow us to fall back on to higher level of simulation, like:

- Parameterized calculations: this is the basic technique for optimization. HPC is a great opportunity to take into account more parameters and to reduce "time to market". This allows the use of optimization techniques, like neural networks, in order to find automatically and optimize set of parameters.
- High resolution physics: greater memory capacity and greater CPU power are required for more refined models in each discipline. For instance deterministic transport instead of few groups' diffusion approach.
- A more realistic physics by using systematically real physical model instead of simplified model or pre-tabulated values. This implies, not only greater CPU power, but robust and easy use coupled system
- Real time simulation: this already exists, but we can imagine improving modeling in order to obtain more realistic simulators and decreases the number of assumptions.

All these improvements are needed in order to:

- Increase Margin by reducing uncertainties;
- Optimize designs;
- Improve safety;
- Optimize operating conditions;
- Increase physics knowledge.

In Section II, we will present briefly the APOLLO3® code. Major challenges that can be addressed in Reactor Physics thanks to HPC and the description of how parallelism and HPC is taken into account in APOLLO3® code will be described in Section III. Finally, we will present some illustration in Section IV.

II. APOLLO3® Code

During the last decade, there has been a growing interest in the nuclear industry for improved nuclear code systems. Based on an assessment of the design of different reactor concept (PWR, BWR, GFR, SFR, SCWR…) and of operating margins to cover uncertainties, there has been a growing interest for the development of a new deterministic multi-purpose code including Lattice, Core and Lattice-Core calculations.

In addition, the evolution of the computer mainframe industry has proceeded along different lines with respect to hardware development. Thus, it is possible to benefit from the constant increase in high performance computing resources and particularly in the ability to perform calculations on parallel computers. It will give new guidelines for the development of future codes and for the extension of the methods to solve large-scale deterministic problems. Within the neutronic modeling framework, important evolutions are required with regard to energy mesh structure, extension of numerical methods, new applications... It appears that the boundary condition (external and internal) should become an essential aspect for the implementation of multi-solver methods and parallelized calculations. A complete and coherent nuclear system must be considered from nuclear data treat-

*Corresponding author, E-mail: christophe.calvin@cea.fr

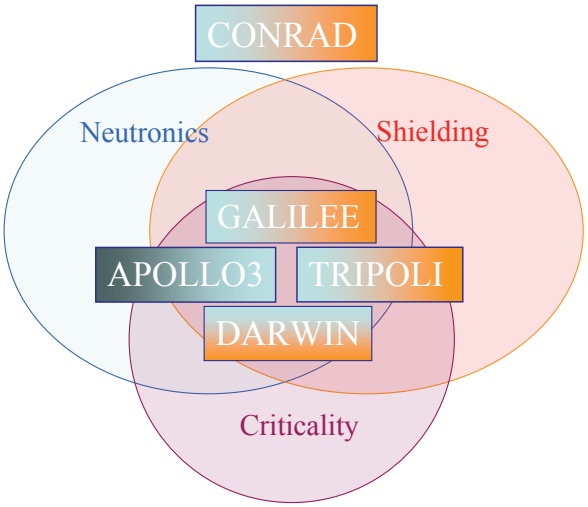

Fig. 1 CEA global development program dedicated to fine neutronic modeling of nuclear systems

ment tools to calculation codes (deterministic and Monte-Carlo codes). According to the codes and the applications, the uncertainties data treatment plays an important role.

Up to now, CEA, with EDF and AREVA support, is renewing its main application codes (deterministic codes "APOLLO2, CRONOS2, and ERANOS2",[1-6] Monte Carlo code TRIPOLI-4™,[7] depletion and fuel cycle MENDEL, successor of DARWIN code,[9] its nuclear data processing system GALILEE,[8] in order to have a complete and advanced nuclear modeling system (see **Fig. 1**) and CONRAD,[10] a nuclear reaction analysis tool in order to create evaluations.

In order to reach a new stage in the modeling of the nuclear systems a new generation of codes will be developed on an approach where core/lattice, deterministic/Monte-Carlo, reference/industrial calculation routes are compatible.

APOLLO3®,[11] is a common project of CEA, AREVA and EDF for the development of new generation system for core physics analysis. Therefore, requirements both for R&D and industrial applications were taken into account for the design of the new system architecture. The new system APOLLO3® is the continuation of the "APOLLO2, CRONOS2, and ERANOS2" code family.[1-6] The experience on APOLLO2, CRONOS2, ERANOS2 codes and their applications provides an initial and complete set of calculation routes for the neutronic evaluation; this experience draws the ways of improving the models (flux solvers and self-shielding methods with new acceleration or effective parallelization methods…).

In order to meet these objectives, the main APOLLO3® requirements are the following:

- Flexibility: from best-estimate calculations to industrial design;
- Coupling with codes from other disciplines (thermal-mechanics, thermal-hydraulics) with SALOME platform;[12]
- Easy coupling with Monte-Carlo codes: in particular with TRIPOLI[7] through a dedicated shared data interface;
- Extended application domain: criticality, shielding of all types of reactors (PWR, BWR, GFR, SFR, SCWR…);
- Uncertainties assessment with perturbation methods and non-intrusive methods (i.e. URANIE);[13]
- Ability to perform calculations on parallel computers;
- User friendly: user interface, databases…
- Portability

III. High Performance Computing within APOLLO3® Code

1. The Different Level of Parallelism

In deterministic simulation, we can consider three main levels of parallelism:

- The first level concerns multiparameterized calculations. This level is typically implanted using a distributed computing approach, since each calculation is an independent one from the other. Classical example is a multiparameter assembly's calculation, where one has to compute different kinds of reactor assemblies for core code calculation.
- The second is what we can call multi-domain calculations. It is implemented using a coarse grain parallelism approach. For example, it concerns all the treatment dealing cross sections, depletion process, thermal-hydraulic feedback, etc. In this level, usually we can consider that the spatial dependency of the data is very tight and a massively parallel approach is well suited. Concerning the case where the spatial dependency of the data is strong, domain decomposition techniques are used. It concerns, for example, the transport equation resolution.
- The third level is based on a fine grain parallelism model and mainly concerns the fine grain parallelization of solvers by exploiting intrinsic parallelism of the involved numerical methods.

In the following we will illustrate these different levels by describing some typical examples.

2. First Level: Multiparameterized Calculations

The main interest of this level is to use the brute force of HPC to solve problems with huge amount of independent calculation in a "human" time. (Some examples have been presented by Prof. Turinsky in his talk during the M&C-SNA conference in 2007 in Monterey). Different uses of HPC for reducing uncertainties in simulator predictions of limiting nuclear plant attributes or how to gain margins using optimization techniques are discussed below.

One way to reduce uncertainties is of course to use deterministic (forward and adjoint) approaches. Another way is to use a less-intrusive method and is based on a stochastic (sampling) approach. Of course this approach is challenged in regard to computational resources required. This sampling approach is also very interesting in case of problem where deterministic approach is too complex. One can cite coupled problems (thermo-hydraulic–neutronic) or depletion ones.

Concerning optimization problems, one very good example is fuel optimization. Grand challenge problem attributes include:

- Multiobjective
- Multicycle
- Lattice optimization
- Bundle optimization
- Loading pattern optimization
- Excess reactivity control optimization (e.g., BWR: CRP and core flow)

3. Second Level: Multi-Domain Calculations

This level is the most classical one. In most of parallel scientific applications this one is used through domain decomposition techniques. To be more precise, one could say that this domain is mainly based on spatial decomposition. Applied to neutronics applications, all the calculations which are spatially independent are included. For instance, in a typical two-stage calculation, at the core level, all the steps concerning cross-sections loading and management, thermal feedback, isotopic depletion, etc. are local to the cell of the geometrical domain, and thus could be done in parallel. To summarize in a standard deterministic 3D core calculation, all the steps are spatially independent and thus could be done naturally in parallel, except one, the flux calculation. Even if all the precedent steps could be done in parallel, the main problem is still the data flow management and the data distribution between the processes. One has to think about it in the architecture code design in order to have optimum data structures to mitigate this problem.

Concerning the flux calculation itself, classical domain decomposition techniques can be used.[14] Concerning the Boltzmann transport equation solver,[15] other parallelism degree can be found, since the other dimensions of the equations can be used, for instance the angular or energetic ones. Many solvers have been implemented in parallel, exploiting either angular or energetic parallelism,[16-17] spatial one[18-22] or both.[23]

Another degree of parallelism can be used when exploiting multilevel techniques. For instance, fine transport solution on one assembly coupled with full 3D coarse solution. Typical examples of such techniques can be found in the COBAYA code[24] or applied to the MINOS diffusion solver within an original approach based on a component mode synthesis.[18]

4. Third Level: Fine Grain Parallelism Model

This level is usually used on shared memory architecture and exploits intrinsic parallelism of the algorithms. These techniques had a great infatuation in early 2000's with HPF language and after that OpenMP.[21-22,25] It becomes more and more interesting in today's computing environment with the many-core architectures which have to be combined with the second level of parallelism in order to improve the overall performances of the algorithm.

A variation of this model concerns the use of accelerators. With the performance exponential increase of these devices, like GPGPU, it becomes more and more interesting to try to utilize this large amount of computational power, despite of specific programming language and paradigm.

Finally, all these levels can be combined to exploit as much as possible all the parallelism degree in the application and the whole performance of the targeted architectures.

In the next section, we will present some practical examples of these parallelism levels.

IV. Example of HPC Application Using APOLLO3®

1. Fuel Loading Optimization

As an example of multiparameterized calculations, one can cite an exercise achieved for fuel loading pattern optimization with genetic algorithm.[26] A tool based on URANIE/VIZIR and APOLLO3® code has been designed and has been successfully applied to the optimization of fuel loading pattern in the case of highly heterogeneous LWR cores. This tool allows the evaluation of more than ten million different configurations in less than 24 hours using more than 4,000 processors. An illustration of different kinds of solutions is provided in **Fig. 2**.

This radial configuration associated to a square array core composed of under-moderated assemblies leads to a harder neutron spectrum inside the core and thus allows us to reach a high conversion rate. However, the presence of fertile assemblies produces a radial power peak or radial shape factor (Fxy) of 1.7 on some fissile assemblies. Moreover the draining coefficient (DC) defined as the reactivity discrepancy at the beginning of the cycle, between nominal core configuration and the one where accidentally it has almost completely lost its water (moderator density equal to 0.007) is worth 290 pcm. In term of safety constraints the Fxy must be lower than 1.5, that corresponds to a feasible linear power density of 450 W/cm and the DC must be strictly negative.

The main advantage of this kind of approach is to allow engineers to test many different kinds of configurations and relax some constraints which are not possible without genetic algorithms and HPC. An example of different solutions found

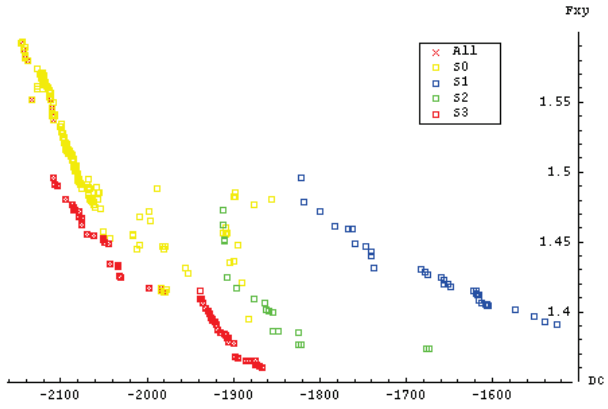

Fig. 2 Example of set of solution found by the genetic algorithm depending on different strategies (Pareto front): optimization of two parameters (Fxy on vertical axis and DC on horizontal axis) – "All", "S0" to "S3" represent different strategies depending on the number of samples and sample sizes.

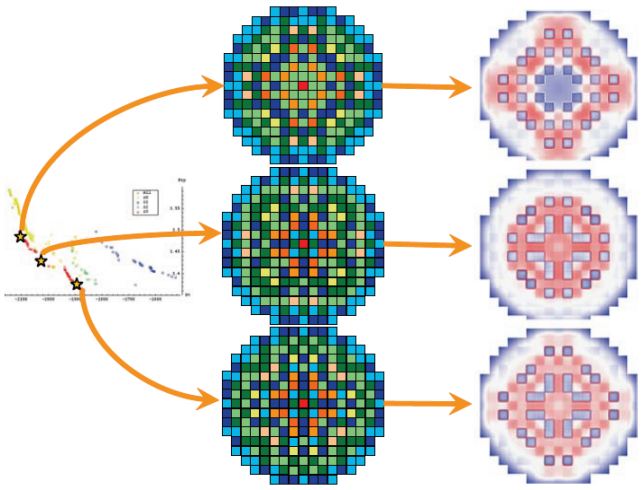

Fig. 3 Illustration of different solutions: Pareto front on the left – loading pattern on the middle – corresponding power map on the right

is given in **Fig. 3**.

2. 3D Core Heterogeneous Calculation Using Domain Decomposition

Highly heterogeneous core calculations as cell by cell are currently too expensive for industrial applications, even if a simplified transport (SP_N) approximation is used. A way to decrease the computation time and the local memory requirement is to use a domain decomposition method. It is particularly well fitted for parallel computers; calculations are distributed on several subdomains, and as many processors as subdomains can be used. We propose here an iterative method using non-overlapping subdomains and Robin interface conditions.

This method could be applied to SP_N approximation, but is presently limited to the diffusion model and to Cartesian grids. It has been implemented in the framework of the existing MINOS solver,[27] which uses a mixed dual finite element method for the resolution of diffusion equation in 3D Cartesian homogenized geometries.

The domain decomposition method is applied to the mixed dual formulation of the diffusion equation, which is discretized using Raviart-Thomas finite elements, and which is implemented into the MINOS solver.[27] The method is based on non-overlapping subdomains. The idea is to iterate the resolution of local problems on each subdomain, using Robin interface condition. At a given iteration, this condition consists in to impose the corresponding boundary value of the solution obtained on the adjacent subdomain at the previous iteration.[18-20]

The flow chart of the parallel algorithm is presented in **Fig. 4.**

The test is a 3D PWR 900 MWe core with two energy groups. The mesh size is 289×289×40. The speed up obtained is presented in **Fig. 5**. As one can see, we obtained very good speedup up to 32 processors. Beyond this number, the mesh size is too small compared to the number of processors. However, the CPU time still decreases, and it tooks less than 10 seconds onto 128 processors to compute a full heterogeneous 3D power map of a PWR core.

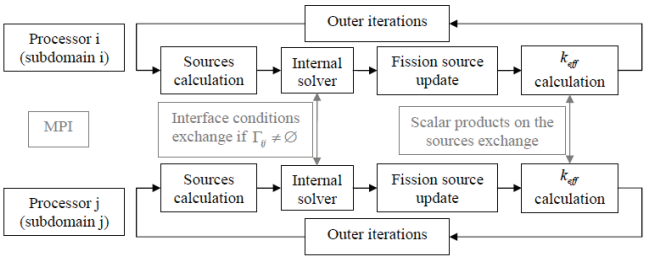

Fig. 4 Flow chart of the parallel MINOS algorithm

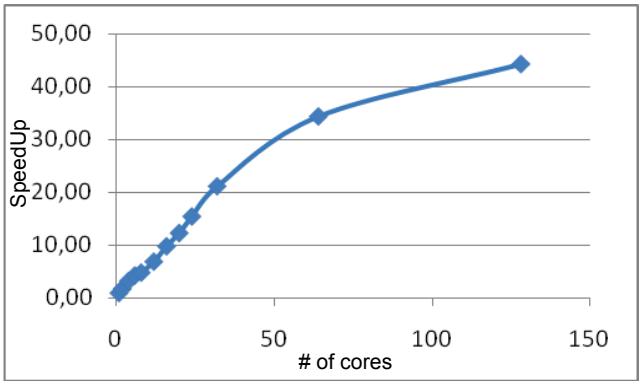

Fig. 5 Speedup obtained on BULL cluster (Intel Nehalem processor and Infiniband network)

3. 3D Massively Parallel Sn Transport Using Hybrid MPI/OpenMP Parallelization

Minaret[28] is a 2D/3D transport solver developed in the frame of APOLLO3® code. The transport equation is solved using a Sn approximation on unstructured mesh – triangular in 2D and prismatic mesh in 3D and is based on discontinuous Galerkin FE method.

A natural way to parallelize this kind of method is to benefit of angular direction idependence. Thus in 3D, the maximum parallelism degree is more than 4,000 for S_{64} approximation. Thus, one can affect a set of angular directions by MPI process, up to one angular direction per MPI process. The parallel flow chart of the algorithm is presented in **Fig. 6**.

Moreover, we can parallelize the computation along each angular direction for the assembly and the inversion of each 4x4 system for each triangle (see **Fig. 7**). This fine grain parallelization is achieved using OpenMP multithreading.

Using this two level parallelism, we have planned, by the end of year 2010, to compute a full 3D GENIV reactor core between 30 and 300 energy groups using more than 10,000 cores in few hours on the last new French petaflop supercomputer TERA100.

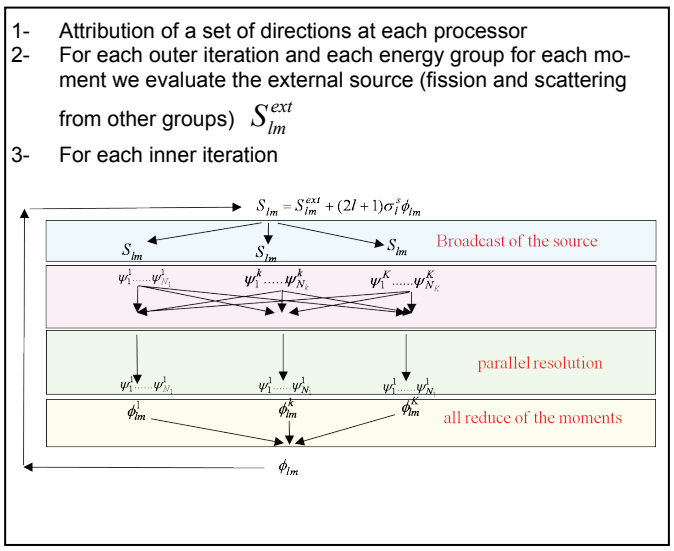

Fig. 6 Flow chart for the exchange of the angular flux and moments

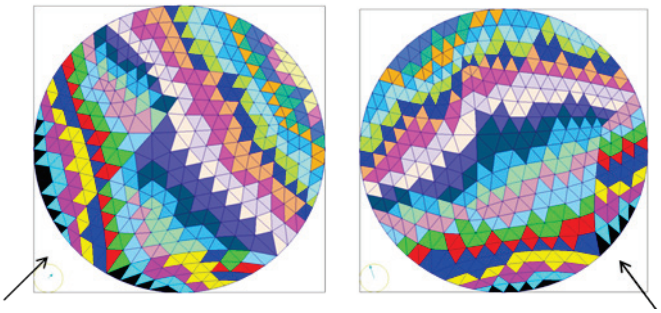

Fig. 7 Fine grain parallelization along one angular direction. At each step all the triangles of the same color can be computed in parallel.

4. GPGPU Programming to Solve the Boltzman Neutron Transport Equation

In this application, we are interested in how accelerate the resolution of the Boltzman equation using GPGPU accelerators. We study the very fine grain parallelization of GPGPU applied to the MINOS solver.[29] In the MINOS solver,[27] the eigenvalue algorithm is based on the power iterations method. At each step, one has to invert the mass matrices using a Cholesky algorithm. All the numerical kernels have been implemented using basic BLAS1 type operations. Thus, we use GPU acceleration without changing the global MINOS algorithm, using CUBLAS and specific implementation of basic computing kernels in CUDA. First experimentations show an acceleration up to 20 times the CPU time. The final implementation in the APOLLO3® code leads to decrease the final acceleration due to lack of generality and specific optimization in the original GPGPU implementation mockup. But we obtained up to factor 5 acceleration on real industrial applications using MINOS-APOLLO3® solver (see **Fig. 8**).

Finally we have mixed the domain decomposition method described earlier implemented using MPI with GPGPU acceleration. Each subdomain is held by a MPI process and we applied GPGPU acceleration on each subdomaine

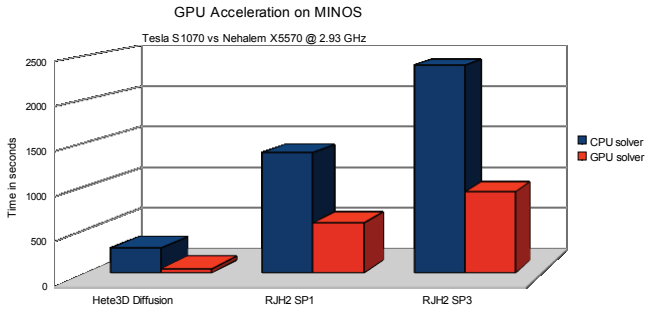

Fig. 8 GPUGPU acceleration of the MINOS-APOLLO3® solver on 3 different tests cases: from left to right: 3D PWR core, 2D and 3D JHR experimental core

calculation. We observe that the overall overhead of the application incrases and thus the speedup decreases in a much more significant way due to mainly two factors:

1. The subdomain computation is accelerated using GPGPU and thus the communication time is proportionally greater than using CPU only;
2. The overhead increases due to GPU – CPU memory transfer; since the MPI communication has to be achived by the CPU, each data exchange between subdomain leads to data tranfer from GPU to CPU, and between CPU using MPI and finally between CPU and GPU.

Nevertheless, these first experimentations show that it is possible to use GPGPU in an industrial code in order to obtain interesting acceleration while maintaining generic programming and original algorithms. Now, if we want to obtain greater speedup, especially by mixing MPI and GPGPU, one has to modify both algorithms and implementation.

V. Conclusion

We have presented in this paper some major HPC challenges for deterministic neutronics simulations and how these challenges are addressed in the APOLLO3® code. We have illustrated with real applications and practical examples how HPC could help achieving high fidelity simulations or dealing with very large problems which could not be addressed otherwise. We use in APOLLO3®, different levels of parallelism, from distributed calculations, to very fine grain parallelism, depending on the problem to deal with and the target architecture. Moreover, we have shown that APOLLO3® code is ready to address both hybrid architecture and programming model, in order to solve the present end coming reactor physics challenges using the last petaflops and post petaflops machines.

References

1) S. Loubiere, R. Sanchez, M. Coste, A. Hebert, Z. Stankovski, C. Van Der Gucht, I. Zmijarevic, "APOLLO2 Twelve Years Later," *Proc. Int. Conf. on Math. and Computations, Reactor Physics and Environmental Analysis in Nucl. Applications, M&C 1999*, Madrid, Spain, Sept. 1999, p.1298 (1999).
2) R. Sanchez, J. Mondot, Z. Stankovski, A. Cossic, I. Zmijarev-

ic, "APOLLO II: A user-oriented, portable, modular code for multigroup transport assembly calculations," *Nucl. Sci. Eng.*, **100**, 352-362 (1988).
3) G. Rimpault *et al*., "The ERANOS data and code system for fast reactor neutronic analyses," *Proc. of the Int. Conf. on the New Frontier of Nuclear Technology: Reactor Physics, Safety and High-Performance Computing, PHYSOR*, Seoul, South Korea, 7–10 October 2002 (2002).
4) J-M. Ruggieri, J. Tommasi, J-F. Lebrat, "ERANOS 2.1: International Code System for GEN IV Fast Reactor Analysis," *Proc. ICAPP '06*, Reno, USA (2006).
5) G. Rimpault, "Algorithmic Features of the ECCO Cell Code for Treating Heterogeneous Fast Reactor Subassemblies," *Proc. Int. Topical Meeting on Reactor Physics and Computations*, Portland, Oregon (1995).
6) R. Sanchez, C. Magnaud, J-J. Lautard, D. Caruge *et al.*, "SAPHYR : A Code System from Reactor Design to Reference Calculations," *Proc. Int. Conf. on Supercomputing in Nuclear Applications*, Paris, France, September (2003).
7) C. M. Diop *et al.*, "TRIPOLI-4: A 3D Continuous Energy Monte Carlo Transport Code," *Proc. Int. Conf. On Physics and Technology of Reactors and Applications, PHYTRA-1*, Marrakech, Morocco (2007).
8) M. Coste-Delclaux, "GALILEE: A Nuclear Data Processing System for Transport, Depletion and Shielding Codes," *Proc. Int. conf. PHYSOR'08 Advances in Reactor Physics*, Interlaken, Switzerland (2008).
9) A. Tsilanizara *et al*., "DARWIN: An Evolution Code System for a Large Range of Applications," *Proc. Int. Conf on Radiation Shielding, ICRS-9*, October 199, Tsukaba, Japan (1999).
10) C. De Saint Jean *et al*., "Status of CONRAD, a nuclear reaction analysis tool," *Proc. Int. Conf. on Nuclear Data for Science and Technology 2007, ND 2007*, Nice, France (2007).
11) H. Golfier *et al*., "APOLLO3: a common project of CEA, AREVA and EDF for the development of a new deterministic multi-purpose code for core physics analysis," *Proc. Int. Conf. on Mathematics, Computational Methods & Reactor Physics, M&C 2009*, Saratoga Springs, New York, May 3-7 (2009).
12) Open Cascade, "Salome: The open source integration platform for numerical simulation," http://www.salome-platform.org, (2005).
13) F. Gaudier, M. Marques, B. Spindler, B. Tourniare, "Uncertainty assessments in severe accident scenario using the URANIE software," *Proc. 35th ESREDA Seminar*, November 19-21, Marseille, France (2008).
14) B. F. Smith, P. Bjorstad, W. Gropp, *Domain decomposition - parallel multilevel methods for elliptic partial differential equations,* Cambridge University Press, New York (1996).
15) R. Roy, Z. Stankovski, "Parallelization of neutron transport solvers," *Recent advances in parallel virtual machine and message passing interface*, LCNS, vol.1332, 494-501 (1997).
16) M. Zeyao, F. Lianxiang, "Parallel flux sweep algorithm for neutron transport on unstructured grid," *J. Supercomput*., **30**[1], 5-17 (2004).
17) Z. Stankovski, A. Puill, L. Dullier, "Advanced plutonium assembly parallel calculations using the APOLLO2 code," *Proc. Int. Conf on Mathematics and Computations, M&C1997* (1997).
18) P. Guérin, A-M. Baudron, J-J. Lautard, "A component mode synthesis method for 3D cell by cell SPn core calculation using the mixed dual finite element solver MINOS," *Proc. Int. Conf on Mathematics and Computations, M&C2005*, Avignon, France (2005).
19) P. Guérin, A-M. Baudron, J-J. Lautard, "Component mode synthesis methods applied to 3D heterogeneous core calculations, using the mixed dual finite element solver MINOS," *Proc. Int. conf. PHYSOR'06 Advances in Reactor Physics,* Vancouver, Canada *(2006)*.
20) P. Guérin, A-M. Baudron, J-J. Lautard, "Domain decomposition methods for core calculations using the MINOS solver," *Proc. Int. Conf on Mathematics and Computations, M&C2007*, Monterey, USA (2007).
21) J. Ragusa, "Implementation of multithreaded computing in the neutronics FEM solver Minos," *Proc. Int. Conf on Mathematics and Computations, M&C2003*, Gatlinburg, Tennessee, USA, (2003).
22) J. Ragusa, "Application of multithread computing and domain decomposition to the 3-D neutronics FEM code CRONOS," *Int. Conf. on Supercomputing in Nuclear Applications, SNA2003*, Paris, France (2003).
23) G. E. Sjoden, *PENTRAN: a parallel 3D S(N) transport code with complete phase space decomposition, adaptive differencing, and iterative solution methods*, PhD thesis. The Pennsylvania State University, Source DAI-B 58/05, p.2652, 301 pp (1997).
24) J-J.Herrero, C. Ahnert, J-M Aragones, "Spatial domain decomposition for LWR cores at the pin scale," *ANS Winter meeting* (2007).
25) F. Coulomb, "Parallelization of the DSN multigroup neutron transport equation on the CRAYT3D using CRAFT," *Proc. 8th SIAM Conference on Parallel Processing for Scientific Computing,* Minneapolis, Minnesota, USA (1997).
26) J-M. Do *et al*., "Fuel loading pattern for heterogeneous EPR core configuration using a distributed evolutionary algorithm," *Proc. Int. Conf on Mathematics and Computations, M&C2009*, Saratoga Springs, New York, USA (2009).
27) A. M. Baudron, J. J. Lautard, "MINOS: A Simplified Pn Solver for Core Calculation," *Nucl. Sci. Eng.*, **155**[2], 250-263 (2007).
28) J. J. Lautard, J. Y. Moller, "Minaret: a 3D solver for unstructured calculation within the APOLLO3® system," *Proc. Int. Conf on Mathematics and Computations, M&C2011*, Rio de Janeiro, Brazil (2011).
29) C. Calvin, E. Jamelot, J. Dubois, S. Petiton, "GPGPU Programming to Solve the Boltzman Neutron Transport Equation," *14th SIAM Conference on Parallel Processing for Scientific Computing*, Seattle, USA, February (2010).