

## ARTICLE

## 3D Neutron Transport and HPC: A PWR Full Core Calculation Using PENTRAN SN Code and IBM BLUEGENE/P Computers

Tanguy COURAU<sup>1,\*</sup> and Glenn SJODEN<sup>2</sup><sup>1</sup>EDF R&D, 1 Avenue du Général de Gaulle, F92141 Clamart CEDEX, France<sup>2</sup>Nuclear and Radiological Engineering Program, Georgia Institute of Technology, Atlanta, Georgia, 30332

When dealing with nuclear reactor calculation schemes, the need for 3D transport-based reference solutions is essential for validation and optimization purposes. As SN transport methods may be considered promising with respect to comprehensive parallel computations, a 3D full PWR core benchmark was proposed to challenge the capabilities of the PENTRAN parallel SN code utilizing an IBM-BG/P computer. After a brief description of the benchmark, a parallel performance analysis is carried out, and shows that the parallelizable (Amdahl) fraction of PENTRAN is comprised between  $0.994 \leq f \leq 0.996$  for a number of BG/P nodes ranging from 17 to 1,156. The associated speedup reaches a value greater than 200 with 1,156 nodes. Using a best estimate model, PENTRAN results are then compared to Monte Carlo results rendered using the MCNP5 code. Good consistency is observed between the two methods (SN and Monte Carlo), with discrepancies less than 65 pcm for the  $k_{eff}$ , and less than 2.5% for the flux at the pincell level.

**KEYWORDS:** PWR core, SN method, Monte-Carlo method, PENTRAN, MCNP5, IBM-BG/P

### I. Introduction

EDF/R&D is currently renewing its nuclear reactor calculation scheme. In this context, the need for three dimensional (3D) transport reference solutions is essential for validation and optimization purposes. As SN neutron transport methods may be considered promising with respect to comprehensive parallel computations, a collaboration between: EDF R&D, previously University of Florida and currently pursued with the Georgia Institute of Technology, and HSW Technologies LLC, was initiated with strong support from IBM. A 3D full pressurized water reactor (PWR) core benchmark was proposed to challenge the capabilities of the PENTRAN<sup>1,2)</sup> parallel SN code utilizing an IBM-BG/P computer.

The first section deals with the PWR core PENTRAN model. It briefly describes the benchmark which corresponds to a simplified PWR first core loaded with 3 different types of fuel assemblies characterized by a specific <sup>235</sup>U enrichment.

Then, a parallel performance analysis is carried out in order to assess the parallel efficiency of PENTRAN when running on an IBM-BG/P computer.

Finally, the last section is devoted to a best estimate calculation performed with PENTRAN. The results obtained are then compared to Monte Carlo results rendered using the MCNP5 code.<sup>3)</sup>

### II. 3D PWR Core PENTRAN Model

This section presents the PWR Core model set up for

PENTRAN. After presenting PENTRAN parallel features, it deals with the core geometry, the calculational parameters, and the convergence criteria.

#### 1. PENTRAN Parallel Features

PENTRAN is a 3-D discrete ordinates code that can automatically distribute the problem phase space among the angular, energy, and spatial variables using parallel memory and task allocation across a virtual processor array (VPA). The VPA is a parallelepiped of processing cores allocated to angular, energy, and spatial decomposition, respectively, where the total number of allocated cores must be the product of AxE<sub>g</sub>S, where A=angular, E=energy group, S=spatial allocated process cores, respectively, devoted to each domain decomposition. Spatial domain decomposition is restricted to coarse mesh units containing some density of fine mesh cells; because a block adaptive design enables varying densities of fine meshes to be contained within each coarse mesh, this can lead to load imbalances depending upon specific process core work assignment and the number of cores allocated. Load imbalancing can be mitigated by enabling the “automatic load balance” feature of PENTRAN. In addition, only the locally computed phase space is stored on each core other than minimal key datalinks, enabling larger problems to be readily solved by distributing the problem across increasing numbers of processor cores.

The difficult task of projecting a large SN problem over an arbitrary VPA first requires some user preferences about the processor aspect ratio, or the relative number of processors used along each axis of the VPA. In PENTRAN, this is accomplished with a user defined decomposition weight vector (*decmpv*). This decomposition vector is an ordered

\*Corresponding author, E-mail: tanguy.courau@edf.fr

triple that contains arbitrary weighting factors for angular, energy group, and/or spatial decomposition, respectively. This vector allows a user to “prioritize” the decomposition strategy used in a parallel execution of the problem, without specifically assigning an exact number of processors; a user can also block decomposition in a particular variable with a zero weight, or lock-in any specific number of processors using a negative weight, if desired. Therefore, *decmpv* defines the user desired aspect ratio of the 3-D processor array for the problem. A consideration in assigning decomposition vector weights is that PENTRAN always breaks the angles up into sweep octants (on the unit sphere), with a subsequent number of directions per octant. This will affect the way angular decomposition is applied for a given number of processors, since there are two levels of decomposition that follow processor assignments in angular decomposition: there are always 8 octants, and for a given SN quadrature order  $(SN \times (SN+2)/8)$  directions per octant. Therefore, octants should be decomposed using a processor assignment that evenly divides into either 8 (octants), or into the number of angles within each octant.

After the input deck is read and the VPA is established, a complex but efficient initialization sequence begins. If selected, load balancing assignments based on spatial mesh density and group scattering ratio are made, followed by Block-Jacobi (default) or Red-Black ordering for spatial decomposition. It should be noted that while load balancing indeed flattens core workload, it can induce an iterative convergence penalty, since it significantly alters the order of spatial sweeping. Block-Jacobi assignments without load balancing are made by evenly dividing up coarse meshes in the order they are numbered in an  $x$ - $y$ - $z$  sequential numbering scheme. Angular sweeps are also ordered and decomposed using alternating direction sweeping to provide optimum convergence. Then, process mapping arrays that track local-global and global-local variable references for parallel memory and parallel execution are established for each processor.

A communication network is also constructed using MPI communicators in PENTRAN to permit sub-groups of processors (angle-group, angle-space, etc) to communicate as efficiently as possible, again depending on the decomposition selected. To retain scalability, PENTRAN automatically minimizes the number of unique communicators “built” during each problem execution, re-using them if they contact the same set of processors because these do consume parallel buffer resources; any excess communicators not usable by the processor are assigned a null value by MPI (set to `MPI_COMM_NULL`). The entire process of assigning all communicators in PENTRAN typically takes a small fraction of a second to complete. Communications overhead is further minimized in PENTRAN by array packing routines; all array data transferred among processors is packed by the sending processor, then unpacked by the receiving processor after communications are finished, and includes a data “integrity” tag to certify that the data source agrees with the mapping assignment.

Following these steps, PENTRAN completes problem

distribution across the VPA, and proceeds with the parallel iterative solution using one of two energy group inner iteration algorithms *multigroup* or *Hiroto-Wienke (H-W)*. The multigroup scheme converges each group one group at a time, while the H-W scheme executes a single iteration in each group with continual group sweeping. When there is neutron upscattering, the H-W scheme is typically more efficient, since there is immediate feedback to all energy groups involved. The outer loop eigenvalue computation is accomplished with the standard power iteration method, with an option of an Aitken extrapolation of the  $k_{eff}$ .

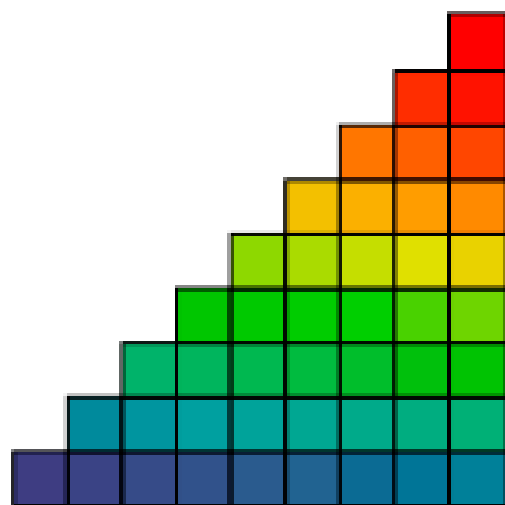
## 2. Geometry and Materials

We assembled a 3D model of a PWR for testing and evaluation purposes. Three fuel assembly types corresponding to Low, Medium, and Highly Enriched Uranium (LEU, MEU, and HEU) are used to simulate a simplified first core loaded with fresh fuel characterized by a  $^{235}\text{U}$  enrichment ranging from 1.5% to 3.25%. For simplicity’s sake, no inserted control device was considered in this core model.

For each  $17 \times 17$  fuel assembly type, the associated pin-specific two-group cross sections were generated using an assembly calculation carried out with the DRAGON cell code.<sup>4,5)</sup> Taking into account the 8-fold symmetry, **Fig. 1** presents the 45 material assembly geometry associated with DRAGON output. The pin-cell lattice pitch was set to 1.26 cm, no inter-assembly water gap was modeled.

The DRAGON code was used to generate a 135 material and 2-energy group cross-section library associated with the fuel assemblies and the reflector properties. The cross-section library was produced at a temperature of 300 K, corresponding to Cold Zero Power (CZP) conditions, without any xenon saturation and with an average boron concentration lower than the beginning of cycle concentration.

**Figure 2** shows the core model used with PENTRAN where the core is represented in a  $17 \times 17$  array with 157 fuel



**Fig. 1** PWR  $17 \times 17$  Assembly: DRAGON Pin-by-Pin Description

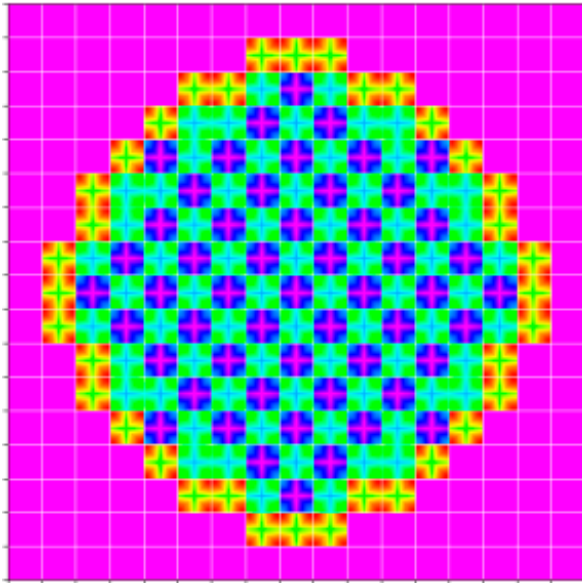


Fig. 2 PWR Core  $xy$ -plane – PENTRAN code

assemblies (LEU, MEU, HEU) and 132 reflector assemblies surrounding the active core. Along the  $z$ -axis, each 360 cm height assembly is reflected by 30 cm of water.

The Cartesian features of the PENTRAN code fully benefits from such a core description. It is also consistent with the pin-by-pin description of nuclear reactor cores when modeled with an industrial core code using diffusion theory, or an SPN approach.

PENTRAN SN code then intends to provide a 3D transport solution ( $\phi$ ,  $k_{eff}$ ) that may be used as a reference for an industrial core code validation and optimization.

### 3. Calculational Parameters

PENTRAN uses a coarse/fine mesh approach where each coarse mesh is subdivided into fine meshes; as noted, coarse meshes can be allocated for spatial parallel decomposition. As noted, PENTRAN also enables one to select decomposition options among discrete angles and energy groups.

In addition to completely hybrid decomposition options, PENTRAN implements an adaptive differencing algorithm to optimally select linear or exponential SN differencing schemes based on performance metrics aliased to solution physics in each coarse mesh region, as well as higher order Taylor coupling of angular fluxes when localized spatial grids are dissimilar due to zone refinement. The code also has other features not tested here.

In our model, each assembly corresponds to a coarse mesh level, each of which is further subdivided with one cell per pin in the  $xy$ -plane, and 84 cells in the  $z$ -axis which correspond to 5 cm height cells. The associated calculational parameters are presented in **Table 1**.

Here,  $\xi_k$  and  $\xi_\phi$  are associated with the convergence criteria for the  $k_{eff}$  and the flux  $\phi$  that are discussed in Section II.3.

We note here that the spatial discretization and angular quadrature presented in Table 1 are not refined enough to

Table 1 PENTRAN Core Model

Core Height	Assembly Height	Lattice Pitch	Type of assemblies Fuel & Reflector
420cm=	360cm	1.26cm	157 (F) + 132 (R)
360cm + 2×30cm			
Coarse Mesh	Fine/Coarse Mesh	#groups	#regions
17×17=289	17×17×84=	2	7,015,764
	24,276		
3D Angular quadrature	# unknowns	$\xi_k$	$\xi_\phi$
S6=48 directions	673,513,344	$1.0 \times 10^{-5}$	$5.0 \times 10^{-3}$ $1.0 \times 10^{-2}$

guarantee that the results obtained may be interpreted as a reference for code to code validation. However, the choice of these parameters results from a trade-off between computation time and accuracy in order to facilitate the parallel performance analysis presented in Section III.

### 4. Convergence Criteria

The SN transport solution is based on the power method. It involves a two-level iterative algorithm to compute the neutron flux. The convergence criteria that must be satisfied between iterations  $l$  and  $l-1$  are:  $\varepsilon_k \leq \xi_k$  and  $\varepsilon_\phi \leq \xi_\phi$ .

Equation (1) defines  $\varepsilon_k$  and  $\varepsilon_\phi$  with  $\phi_i^g(l)$  the flux in the energy group  $g$  and the region  $i$  and  $k_{eff}(l)$  the eigenvalue of the problem. The flux convergence criteria must be satisfied at both the inner and outer iteration level for 2 successive iterations before exiting the iteration loop.

$$\varepsilon_k = \frac{|k_{eff}(l) - k_{eff}(l-1)|}{k_{eff}(l)} \text{ and } \varepsilon_\phi = \max_{i,g} \left\{ \frac{|\phi_i^g(l) - \phi_i^g(l-1)|}{|\phi_i^g(l)|} \right\} \quad (1)$$

The flux convergence parameter was fixed to  $5.0 \cdot 10^{-3}$  in coarse meshes associated with the fuel and the surrounding reflector, and it was relaxed to  $1.0 \cdot 10^{-2}$  in the coarse meshes associated with the reflector not in contact with the fuel (see Fig. 2).

## III. Parallel Performance Analysis

Based on this model, a parallel performance analysis was performed by varying the number of BG/P nodes from 17 to 1,156. The 5 different parallel decompositions considered were either related to single or hybrid decompositions of space (S), angle (A), or energy groups (E).

For each configuration studied, the convergence was reached within 59 to 66 outer iterations, which corresponds to less than 2,400 inner iterations.

This section briefly reviews Amdahl's law properties, and then focuses on the results obtained with PENTRAN and the BG/P computer.

### 1. Amdahl's Law

Defining the speedup as:  $S_p = T_s/T_p$ , where  $T_s$  is the wall clock time for a single task, and  $T_p$  is the wall clock time for the tasks associated with  $p$  processors. It is then possible to

write Amdahl's law as:

$$S_p = \left[ (1-f) + \frac{f}{p} \right]^{-1} \tag{2}$$

where  $f$  is the parallelizable fraction of the code. This parallelizable fraction  $f$  could be computed considering a speedup ratio equation  $S_p/S_k$  derived from Eq. (2) where the subscripts  $p$  and  $k$  refer to 2 calculations performed with  $p$ , respectively  $k$ , processors or MPI processes.

$$\frac{S_p}{S_k} = \frac{(1-f) + \frac{f}{k}}{(1-f) + \frac{f}{p}} \Rightarrow f = \frac{S_k - S_p}{S_k(1-1/k) - S_p(1-1/p)} \tag{3}$$

**2. Parallel Efficiency Results**

**Table 2** presents the various simulations performed. One will note that:

- The number of 4-core BG/P nodes used varies from 17 to 1,156. PENTRAN has a pure MPI programming model.
- Two modes were considered with the BG/P nodes utilized. SMP: 1 MPI process per node with full node memory available for this process. DUAL: 2 MPI processes per node with half the memory per process.
- The decomposition strategy involved angular, spatial, and energetic aspects.

Using Eq. (3) it was possible to evaluate the parallelizable fraction  $f$  of the code when running on an IBM-BG/P computer.

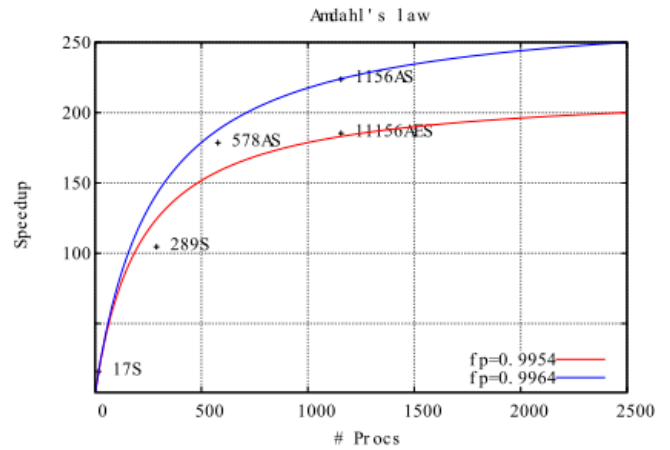
Based on the several different simulations performed, the parallel fraction was estimated to be fairly constant with:  $0.994 \leq f \leq 0.996$ . A speedup greater than 200 was achieved with 1,156 BG/P nodes. **Figure 3** plots the associated results and shows that Amdahl's law is reasonably well adapted to depict the parallel efficiency of our system.

Due to the nature of the transport SN equation, the angular decomposition induces almost no extra computation cost compared to spatial or energy decomposition. In addition, energy decomposition is also driven by convergence in the slowest converging group, since the spectral radius of the source iteration is proportional to the scattering ratio, which is higher in lower energy neutron groups; this results in ad-

**Table 2** Parallel Speedup

(\*Case 17S: The speedup was extrapolated assuming  $f=0.9955$ )

Case	17S	289S	578AS	1156AES	1156AS
# MPI processes	17	289	578	1156	1156
Mode	SMP	SMP	SMP	SMP	DUAL
BG/P nodes	17	289	578	1156	578
Decomp:	1,1,17	1,1,289	2,1,289	2,2,289	4,1,289
A,E,S					
Runtime (s)	125926	18649	11289	10772	9044
Relative speedup	1	6.75	11.5	11.68	13.92
$f$	-	0.9939	0.9961	0.9955	0.9964
Speedup	15.8*	104.5	178.5	185.3	223.4



**Fig. 3** PENTRAN Simulations with BG/P – Amdahl's Law (each point is characterized by a # MPI processes and the associated decomposition with respect to Angle, Energy, and Space.)

ditional iterations in all groups with a parallel-synchronized energy decomposition. This explains the results presented on Fig. 3, where the performance obtained with 1,156 processors is better when relying on spatial and angular decomposition only (1,156 AS) rather than spatial, angular, and energy<sup>a</sup> decomposition (1156 AES).

**IV. Comparing PENTRAN with MCNP**

The parallel performance analysis discussed in Section III showed that 3D full core transport calculations were achievable with PENTRAN when relying on a massively parallel computer such as BG/P.

This section focuses on the validation of the results obtained. It presents a PENTRAN refined solution and discusses convergence issues. Then, it compares PENTRAN results to a Monte Carlo model based on the same multi-group cross-sections and rendered with the code MCNP5.

**1. PENTRAN Best Estimate Calculation**

In order to produce a best-estimate calculation with PENTRAN, we refined the discretization parameters described in Table 1.

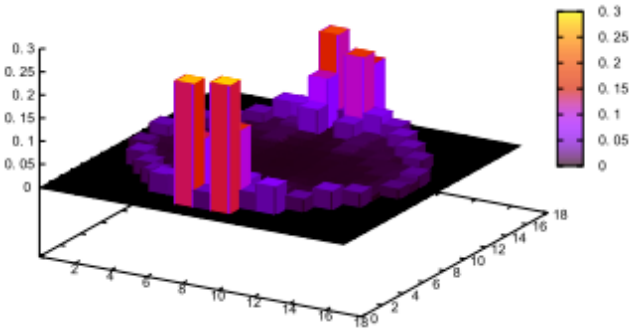
Each fuel pin was subdivided in 2x2 cells and the height of each cell was reduced to 3 cm. The total number of regions was then increased to 46,771,760 resulting in a total number of unknowns equal to 4,490,088,960.

**(1) Convergence Analysis**

Using 578 MPI tasks in SMP mode (578 BG/P nodes involved) the full core calculation was performed with a spatial (289 coarse meshes) and angular (2 angles) decomposition. The maximum number of outer iterations was set to 200, and was reached before convergence of the algorithm.

**Figure 4** presents the maximum relative error for the fine meshes within a given coarse mesh. Results are given for

<sup>a</sup> Here, the ‘‘Multigroup’’ energy decomposition was used. ‘‘H-W scheme’’, which was not tested, may have improved the performance with respect to energy decomposition.



**Fig. 4** PENTRAN Flux Relative Error (%) in Thermal Group ( $g=2$ )

each coarse mesh. It clearly shows that convergence was reached in almost every coarse mesh. Figure 4 focuses on the thermal group ( $g=2$ ) only, since the behavior is similar in the fast group. It shows that the sweep algorithm used to solve the transport equation favors a direction which induces a dissymmetry in the error shape.

When comparing Section III results where a coarser mesh division was used, it appears that the number of iterations needed to reach convergence increases as the mesh is refined.

This difficulty has to be investigated further, and may be partly overcome by tuning the acceleration algorithms and performing double precision floating point operations.

(2) Results Obtained

The  $k_{eff}$  was found equal to 1.17231 with an associated error of 0.4 pcm. This  $k_{eff}$  value is significantly higher than an expected value of 1.0 for a critical reactor.

This is directly related to the fact that the library used was produced at a temperature of 300 K (CZP conditions) without any inserted control device, no xenon and a boron concentration lower than the beginning of cycle concentration.

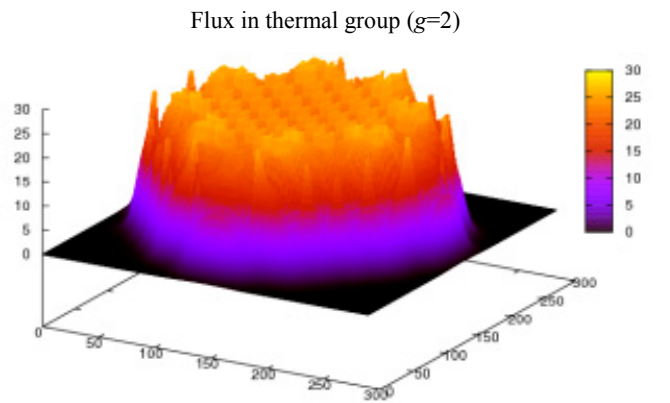
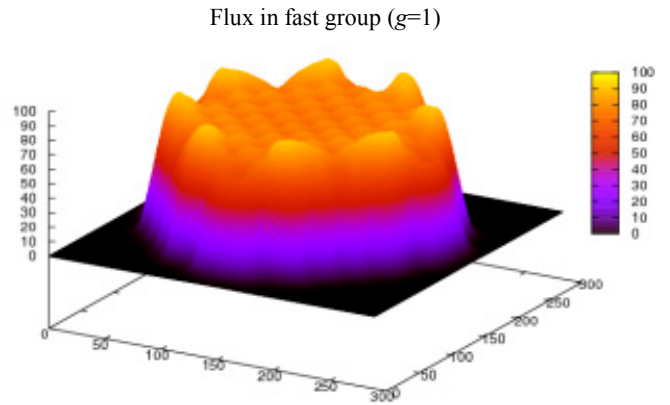
Finally, it was also possible to compute the  $f_{xy}$  ratio which compares the power (assimilated here to the production rate) of the hottest fuel pin to the average power. A value of 1.36 was found, which is consistent with industrial core configurations.

**Figure 5** presents the neutron flux averaged over each pin and integrated along the  $z$ -axis. One clearly sees that the flux distribution is flatter in the fast group ( $g=1$ ) while the thermal ( $g=2$ ) group exhibits more peaks associated with the neutron flux thermalization in the water holes and in the reflector vicinity.

**2. MCNP Results & Comparison with PENTRAN**

Monte Carlo results were generated using MCNP5 in multigroup mode using the same cross-sections as for PENTRAN calculations.

MCNP results were generated with 10 statistically independent runs to limit the impact associated with the source convergence issues. Each MCNP run consisted in a simulation of 1,000 cycles of 64,000 particles, after having discarded 200 cycles for source convergence.



**Fig. 5** PENTRAN Pin Cell flux

The  $k_{eff}$  was estimated to  $1.17302 \pm 5$  pcm. Averaging the flux over each independent simulation and accounting for the symmetry, the pinflux convergence was estimated to 2% and 1% respectively for groups  $g=1$  and  $g=2$ . The number of particles simulated was, however, not sufficient to reach a satisfying convergence level for local tallies along  $z$ -axis.

Consequently, only fully integrated data along  $z$ -axis were compared. **Table 3** and **Figure 6** compare the results obtained with PENTRAN and MCNP. It is shown that results are consistent between the two codes. However, there is need for a more converged Monte Carlo solution in order to provide 3D data with an acceptable level of convergence.

It clearly appears on Figure 6 that the order of magnitude of the plotted error is greater in thermal group ( $g=2$ ). This may be related to the fact that the fast flux is generally smoother than the thermal flux. One may also observe that the error is primarily located in the reflector vicinity where the flux gradient is greater.

**Table 3** PENTRAN – MCNP Comparison for  $k_{eff}$  and 2-group flux

	$k_{eff}$ ( $\delta k$ -pcm)	Flux Error (%)
PENTRAN	1.17231 (0.4)	[0.12, 0.3]
MCNP	1.17302 (5)	[2, 1]
COMPARISON	64 pcm	[1.54, 2.23]

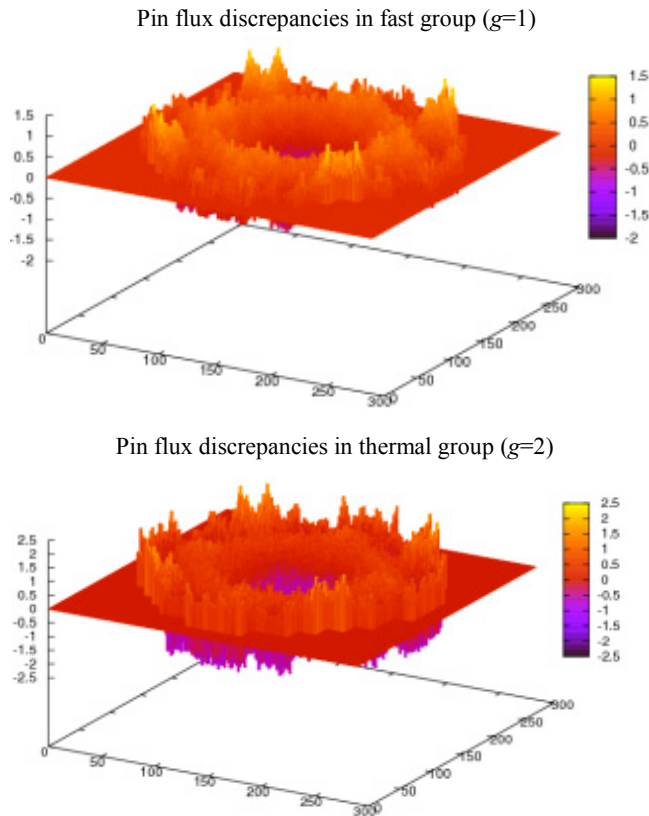


Fig. 6 PENTRAN – MCNP Pin Flux Discrepancies (%)

## V. Conclusion

This paper shows that PENTRAN code and IBM-BG/P computers perform very well together, with very good speedups achieved when using up to 1,156 BG/P nodes.

In the deterministic neutron transport context, the main advantage of using a massively parallel computer is the ability to deal with very large problems requiring a lot of memory to be solved. This is feasible when relying on a code using a parallel distributed memory architecture.

Consequently, the use of SN parallel method as a refer-

ence tool for transport calculations is considered as a realistic approach provided one relies on a comprehensively parallel computer such as IBM-BG/P.

It is also shown that convergence issues may be exhibited when dealing with very large 3D full core calculations. The authors believe that there is a need for improved numerics adapted to this type of full core calculation.

Having demonstrated the feasibility of full PWR core 3D transport calculations, an increased number of energy groups will have to be considered in order to improve the physical quality of the solution. This type of calculation will then provide reference solutions that could be used for industrial core solvers validation and optimization.

## Acknowledgment

The authors wish to thank Cindy Mestad and her Blue Gene team at IBM for their support with the Rochester IBM-BG/P computer. This work would not have been possible without their much appreciated help.

## References

- 1) G. E. Sjoden, A. Haghghat, "PENTRAN, a Parallel Environment Neutral Particle TRANsport Code in 3-D Cartesian Geometry", *Mathematics Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, NY, 1997 (1997).
- 2) G. E. Sjoden, A. Haghghat, "PENTRAN, a Parallel Environment Neutral Particle TRANsport Code in 3-D Cartesian Geometry," *Code Users's Guide/Manual, Version 9.4.X.5*, HSW Technologies LLC (2008).
- 3) X-5 Monte Carlo Team, *MCNP - A General Monte Carlo N-Particle Transport Code. Version 5, Volume I: Overview and Theory*, LA-UR-03-1987 (2003), (Revised March 2005).
- 4) G. Marleau, A. Hébert, R. Roy, *A User Guide for DRAGON 3.05*, IGE-174 Rev. 6, Institut de Génie Nucléaire, École Polytechnique de Montréal, Québec, Canada (2006).
- 5) T. Courau, G. E. Sjoden, G. Marleau, "PWR Assembly Transport Calculation: A Validation Benchmark Using DRAGON, PENTRAN, and MCNP," *Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, Saratoga Springs, NY 2009 (2009).