Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew
Gatlinburg, Tennessee, April 6-11, 2003, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2003)

TESTING AN EXPERT SYSTEM FOR SELECTION OF MESH AND DOMAIN DECOMPOSITION OF PARALLEL S_N METHOD

Apisit Patchimpattapong and Alireza Haghighat
Department of Nuclear and Radiological Engineering
University of Florida
202 Nuclear Sciences Building, Gainesville, FL 32611, USA
axp227@ufl.edu; haghighat@ufl.edu

ABSTRACT

An expert system for generating an effective mesh distribution for the S_N particle transport method has been developed. This expert system consists of two main parts: 1) an algorithm for generating an effective mesh distribution in a serial environment, and 2) an algorithm for inference of an effective domain decomposition strategy for parallel computing. For the first part, the algorithm consists of four steps: creation of a 3-D geometric model and coarse meshes, calculation of uncollided fluxes, selection of differencing schemes, and generation of a fine mesh distribution. A 3-D geometric model was created using AutoCAD. A parallel code PENFC (Parallel Environment Neutral-Particle First Collision) has been developed to calculate the uncollided flux in 3-D Cartesian geometry. Differencing schemes were selected based on the uncollided flux distribution using a least squares methodology. A serial code PENXMSH has been developed to generate a fine mesh distribution that preserves problem geometry and physics. For the second part, the algorithm accounts for four factors that affect parallel performance: number of processors and memory available per processor, load balance, granularity, and degree-of-coupling among processors. These factors are used to derive a parallel-performance-index which provides expected performance of an algorithm depending on computing environment and resources. A large index indicates a high granularity algorithm with relatively low coupling among processors. This expert system has been successfully tested within the PENTRAN (Parallel Environment Neutral-Particle Transport) code system for simulating real-life shielding problems.

Key Words: discrete ordinates method, mesh generation, uncollided flux calculation, parallel performance model, domain decomposition strategy

1. INTRODUCTION

The discrete ordinates (S_N) method has been widely used to obtain numerical solutions of the transport equation. The method calls for discretization of independent variables: angle, energy, and space. To generate an “effective” spatial mesh distribution, one has to consider various factors including particle mean-free-path, material and source discontinuities, and problem objectives. This process becomes more complicated if we consider the effects of numerics, parallel processing and computing resources. To overcome these difficulties, we have developed an expert system for generating an effective mesh distribution for the S_N method. This expert system comprises two major components: 1) an algorithm for generating an effective mesh distribution in a serial environment, common for both serial and parallel calculations; and 2) an algorithm for selecting an effective domain decomposition strategy for parallel computing. The first component has been previously reported in Ref. 1. In this paper, first briefly we review the algorithm discussed in Ref. 1, then elaborate on the algorithm developed for selection of domain decomposition strategy, and finally examine the capability of our expert system using the PENTRAN (Parallel Environment Neutral-Particle Transport) code system [2], and the VENUS-3 benchmark experimental facility [3].

2. ALGORITHM FOR GENERATING AN EFFECTIVE MESH DISTRIBUTION IN A SERIAL ENVIRONMENT

This algorithm can be divided into four steps: 1) creation of a 3-D geometric model and coarse meshes, 2) calculation of uncollided fluxes, 3) selection of differencing schemes, and 4) generation of a fine mesh distribution. Following subsections describe methodologies and formulations used in each step.

2.1. Creation of a 3-D Geometric Model and Coarse Meshes

A 3-D physical model can be partitioned into x-y-z coarse meshes based on material and source boundaries and problem objectives. These coarse meshes are continuous, i.e. coarse mesh boundaries along each of the coordinate axes are applied throughout the entire model. To obtain a 3-D geometric model, we first partition a physical system into z-levels along the z-axis. For each z-level, we utilize AutoCAD [4] to create a 2-D geometric model and coarse mesh layout. To obtain a 2-D model, we:

1. Create layers of simple geometric shapes such as circles, triangles and rectangles, each of which is colored to represent its material content
2. Create the x-y coarse mesh layout as a separate layer
3. Project all layers onto a plane

We export the complete contents of an AutoCAD drawing as a *Drawing Interchange File* (DXF).

2.2. Calculation of Uncollided Flux

To select an appropriate differencing scheme, one needs to know the behavior of the particle flux. We seek a methodology that provides an approximate flux shape in a relatively short time. An uncollided flux distribution is a good candidate because it is a part of the actual flux distribution and it can be calculated semi-analytically. For this purpose, we have developed a parallel code, PENFC (Parallel Environment Neutral-Particle First Collision) [1]. PENFC is capable of calculating uncollided and first collision fluxes in a 3-D Cartesian geometry in a parallel environment. PENFC solves for uncollided flux using Eq. 1.

$$\phi_g^u(\underline{r}) = \frac{S_g}{4\pi\sigma_{t,g,0}} \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \sin\theta \left[e^{-\sigma_{t,g,1} l_a} (1 - e^{-\sigma_{t,g,0}(l_b - l_a)}) \right] \tag{1}$$

where for an energy group g, $\phi^u{}_g(\underline{r})$ = uncollided flux at a position $\underline{r}$; S_g = source strength; $\sigma_{t,g,o}$ = total cross section inside the source; $\sigma_{t,g,l}$ = total cross section outside the source; $l_b - l_a$ = path length inside the source; and l_a = path length outside the source

We utilize the Trapezoidal cubature formula to approximate the double integrals in Eq. 1. It is worth noting that the uncollided flux at one location can be calculated independently and PENFC includes an MPI-based algorithm for parallel processing. As a result, the amount of computation time is reduced significantly with parallel processing.

2.3. Selection of Differencing Scheme

In the S_N method, we utilize a differencing scheme to solve a particle balance equation in each spatial mesh. A differencing scheme is a fitting formulation that represents the behavior of angular flux within a spatial mesh. We fit a function, representing the behavior of differencing scheme, to an uncollided flux distribution obtained from PENFC. Then, we apply the least squares method to determine which differencing scheme best suits the flux distribution.

2.4. Generation of Fine Mesh Distribution

We have developed a serial code PENXMSH, which generates a fine mesh distribution based on preserving material boundaries, conserving material masses (volumes), and considering particle interactions (mean-free-path). PENXMSH utilizes a geometric model and coarse mesh layout from the AutoCAD DXF file provided from Step 1 (Section 2.1). A coarse mesh is partitioned into fine meshes based on user-specified mean-free-path. To assign material to a fine mesh, PENXMSH checks the center of the fine mesh against the layers of geometric shapes described in Section 2.1. To achieve certain accuracy in preserving material volumes, an iterative procedure is used to examine different mesh sizes.

3. ALGORITHM FOR SELECTING AN EFFECTIVE DOMAIN DECOMPOSITION STRATEGY FOR PARALLEL COMPUTING

To perform a parallel S_N calculation, one can decompose the three independent domains (variables): angle, energy and space. Depending on how sub-domains are distributed among processors, each decomposition strategy may require different amount of memory, and results in different parallel performance. There are four main factors that affect parallel performance: 1) number of processors and memory available per processor, 2) load balance, 3) granularity, and 4) degree-of-coupling. We now discuss the nature of each factor, our approach to estimate them, and their use for selection of an effective domain decomposition strategy.

3.1. Factors Affecting Parallel Computing

3.1.1. Number of processors and memory available per processor

This factor indicates which domain decomposition strategies are feasible within available computing resources. The majority of memory in PENTRAN is used for storing angular fluxes and flux moments. These arrays are allocated based on the number of local coarse meshes, local energy groups, and local angular sweep octants on each processor. Therefore, different decomposition strategies may require different amounts of memory. For this, we utilize a mapping algorithm used in PENTRAN to estimate the memory requirement.

3.1.2. Load balance

This factor indicates how workload is distributed among processors. For any decomposition strategy, the number of octants, groups and coarse meshes is always divided equally among processors. However, each coarse mesh may contain different numbers of fine meshes and may use different numerics (differencing scheme). These differences result in a load imbalance among processors.

3.1.3. Granularity

Granularity is defined as the number of operations performed per number of communications. Each decomposition strategy has its own unique computation and communication structure, which results in different granularity. We measure granularity in terms of computation-to-communication time ratio (CPCM) in a transport sweep. This ratio gives not only a measure of granularity of the parallel algorithm but also a degree of load balance (indicated by a waiting time for message passing). In the S_N calculation, the bulk of computing time is spent in the transport sweep. Moreover, the sweep algorithm is affected the most by parallel domain decomposition. To simulate a transport sweep without performing the actual transport calculation, we need the following information:

1. Number of fine meshes per coarse mesh
2. Differencing scheme of each coarse mesh
3. Computation time of a transport sweep for each differencing scheme
4. Computation time of a scattering source
5. Communication time of a transport sweep

Fine mesh distribution and differencing schemes can be obtained from Steps 4 and 3 (Sections 2.4 and 2.3) of the serial algorithm, respectively. To estimate the computation time of the transport sweep, we perform serial calculations using a simple model (e.g. 1 coarse mesh, 1,000 fine meshes, S6, P3, 1 group) with differencing schemes of interest. For the computation time of the scattering source, we utilize a scattering source calculation algorithm available in PENTRAN to estimate its computation time. The computation times of each differencing scheme and the scattering source are estimated only once for each computing platform to account for processor characteristics. To estimate the communication time of the transport sweep, we need to obtain a parallel message passing time and a waiting time caused by load imbalance as described in Section 3.1.2. We utilize a communication structure as it is used in PENTRAN to: 1) determine sending and receiving processors; and 2) simulate parallel message passing. The first part is carried out in a serial environment and the second part is performed in the parallel environment of interest to measure the time spent for message passing. By using the preceding information, we can simulate the transport sweep and the scattering source calculation to estimate the waiting time. This waiting time is added to the parallel message passing time to obtain the communication time of the transport sweep. Consequently, the CPCM can be obtained from a ratio of the total computation time to the communication time of the transport sweep.

3.1.4. Degree-of-coupling

PENTRAN allows hybrid domain decompositions including any combination of angular, energy and spatial decompositions. Each sub-domain (octant, group and coarse mesh), is processed by one processor only. Degree-of-coupling (DCP) is defined as the contribution to the total source in a sub-domain from other processors. This quantity indicates how one processor depends on data from others. Eq. 2 gives a formulation of the DCP.

$$DCP^{P}_{V,g,k} = \frac{\left(\sum_{surface=1}^{6} \left(J^{In}_{g,k} A\right)_{surface} + \sum_{g'=1}^{g} \sigma_{g' \to g} \phi_{g'} V \right)_{P}}{\sum_{i=1}^{Nproc} \left(\sum_{surface=1}^{6} \left(J^{In}_{g,k} A\right)_{surface} + \sum_{g'=1}^{g} \sigma_{g' \to g} \phi_{g'} V \right)_{i} + S^{Fixed}_{g} V} \tag{2}$$

where $DCP^{P}{}_{V,g,k}$ = degree-of-coupling of a processor P to the total source of volume V, group g and octant k; $J^{In}{}_{g,k}$ = incoming current; A = surface area; $\sigma_{g' \to g}$ = scattering cross section from group $g' \to g$; $\phi_{g'}$= scalar flux of the energy group g'; $S^{Fixed}{}_{g}$= fixed source strength

To estimate DCP using Eq. 2, we obtain an angular flux distribution, partial currents, and a source distribution from a "small model" serial calculation. This small model must represent the problem physics yet should require a relatively short computation time on a single processor. We create this model by coarsening meshes and reducing a quadrature order. We find the maximum DCP of each energy group. To account for different group convergence behaviors, we weight the DCP of each group by its corresponding c-ratio ($C_g = (\sigma_s/\sigma_t)_g$, which is related to the spectral radius, i.e. a measure of the rate of convergence). We define the DCP of the whole problem by:

$$DCP = \sum_{g=1}^{G} C_g DCP^{Max}_{g} \tag{3}$$

where $DCP^{Max}{}_{g}$ = maximum degree-of-coupling of the energy group g; C_g = c-ratio of the energy group g; and G = number of energy groups

3.2 Estimation of the Parallel-Performance-Index

The CPCM gives the relation of computation and communication, while the DCP represents the convergence (numerical) behavior of the parallel algorithm. To obtain an index that accounts for both factors, we define a parallel-performance-index (PPI) as:

$$PPI = \frac{CPCM}{DCP} \tag{4}$$

A large value of the PPI indicates that the parallel algorithm can perform a large number of computations relative to the number of communications, while it does not diminish the rate of convergence.

4. TEST PROBLEM DESCRIPTION

The VENUS-3 facility is the LWR-PVS benchmark experimental facility with partial length shielded assemblies located at SCK•CEN, Mol (Belgium). Its overall dimensions are 65.573x65.573 cm^2 and 70 cm high. In the following sections, we examine different aspects of our expert system by simulating the VENUS-3 benchmark facility. All calculations were performed on the PCPEN cluster[1].

5. EVALUATION OF THE ALGORITHM FOR GENERATING AN EFFECTIVE MESH DISTRIBUTION IN A SERIAL ENVIRONMENT

5.1. Evaluation of the Algorithm for Creating a 3-D Geometric Model and Coarse Meshes

We partition a 3-D geometric model of the VENUS-3 facility into four z-levels. Fig. 1 illustrates the process of creating a 2-D model for z-level 2. For this z-level, there are ten layers of shapes and a layer of coarse mesh layout. We utilize AutoCAD to create these layers and to project them onto a plane to create a 2-D model. Repeat the same process for other z-levels to complete the 3-D model.

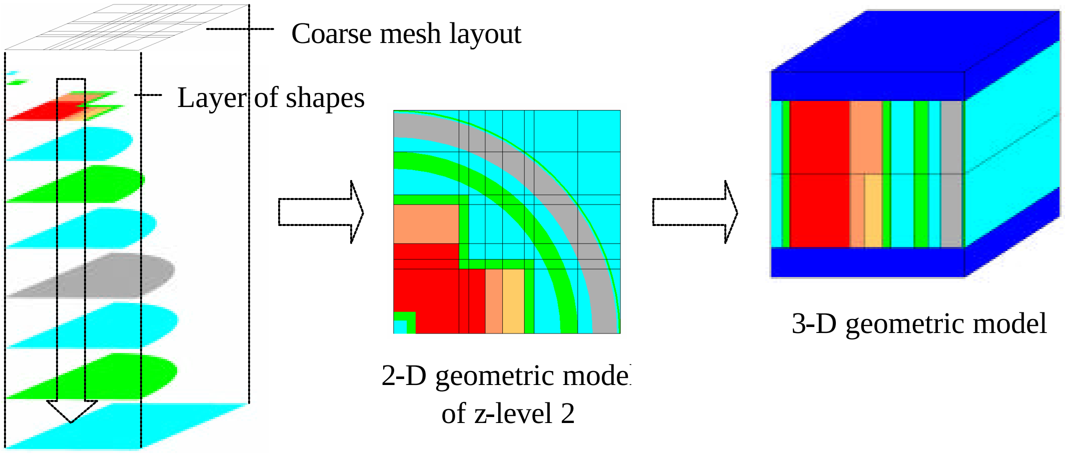


Figure 1. Illustration of the process to create a 2-D geometric model for z-level 2 of the VENUS-3 facility.

5.2. Evaluation of the Algorithm for Calculating Uncollided Fluxes

The VENUS-3 problem model is partitioned into 6x5x4 x-y-z coarse meshes. For simplicity, we homogenize materials in each coarse mesh, and use the first energy group (15.76 MeV) of the

[1] PCPEN cluster is owned by the Department of Nuclear and Radiological Engineering, University of Florida. This cluster has 8 nodes: 7 nodes of 1 GHz Pentium III processors and the head node of 1.7 GHz AMD Athlon processor. Each node has 2 GB RAM.

BUGLE-96 library [5]. For the reference PENTRAN model, we utilize a uniform fine mesh distribution of 5x5x5 per coarse mesh, and S8 level-symmetric quadrature set. For PENFC, we calculate the flux at seven locations per coarse mesh, i.e., three locations along each axis through the center of the mesh. We use 10x10 angles (polar x azimuthal angles) for the numerical integration in Eq. 1. Fig. 2 shows a comparison of the normalized uncollided fluxes for the mid-point energy of 15.76 MeV along x-axis at y = 3.15 cm and z = 22.5 cm.

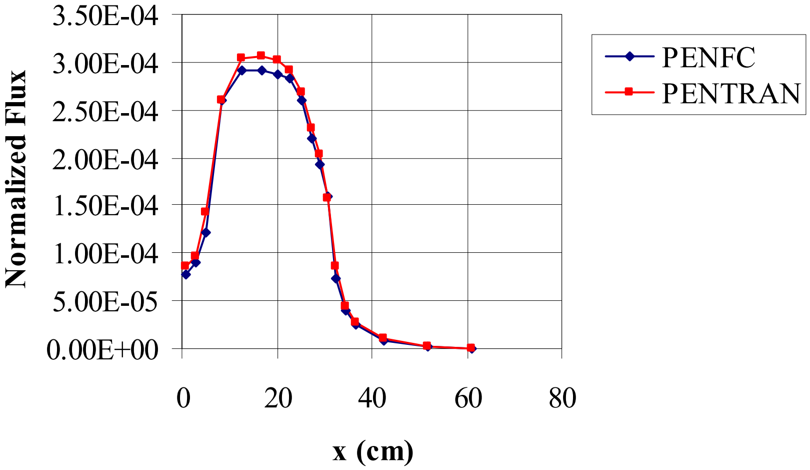


Figure 2. Comparison of the normalized uncollided fluxes along x-axis at y = 3.15 cm and z = 22.5 cm, at 15.76 MeV.

We observe a good agreement between PENFC and PENTRAN flux distributions. The computation times of PENFC are 147.99 sec for a serial calculation and 38.51 sec for a 4-processor parallel calculation while PENTRAN requires 33.3 sec on one processor. We observe a high parallel efficiency of 96%. This high efficiency is expected because PENFC formulation is highly parallelizable, i.e. each position can be processed independently. For this test problem, PENTRAN requires less computation time than PENFC. However, PENFC solves for uncollided fluxes semi-analytically; therefore, it is not suffered by the ray-effect, i.e. the unphysical oscillation of scalar fluxes due to insufficient number of directions in problems with very low scattering and localized sources.

5.3. Evaluation of the Algorithm for Selecting Differencing Schemes

We examine the capability of our algorithm (described in Section 2.3) in predicting an appropriate differencing scheme based on the uncollided flux shape obtained from PENFC in the previous step. We select a differencing scheme within the PENTRAN adaptive differencing strategy [6]. For this test problem, our predictions agree with the PENTRAN predictions for 84% of the coarse meshes. This demonstrates that the uncollided flux distribution is adequate for determination of an appropriate differencing scheme. The computation time of this step is a few seconds.

5.4. Evaluation of the Algorithm for Generating a Fine Mesh Distribution

We investigate the effectiveness of PENXMSH for generating a fine mesh distribution. For demonstration, we compare flux values of the following three fine mesh distributions (generated by PENXMSH).

Reference-mesh: Max. material loss of 2 %, 104,920 fine meshes (0.1-0.2 mean-free-paths)
Variable-mesh: Max. material loss of 13 %, 20,560 fine meshes (0.3-0.35 mean-free-paths)
Uniform-mesh: Max. material loss of 8 %, 57,928 fine meshes (1-cm uniform mesh)

Fig. 3 shows the fine mesh distributions for z-level 2 of the VENUS-3 model. Numbers of coarse meshes along x, y and z axes are 8x7x4, respectively. We use the first energy group of the BUGLE-96 library, S8 level-symmetric quadrature set and P3 Legendre scattering order. The computation time of PENXMSH is less than 10 sec for each mesh distribution.

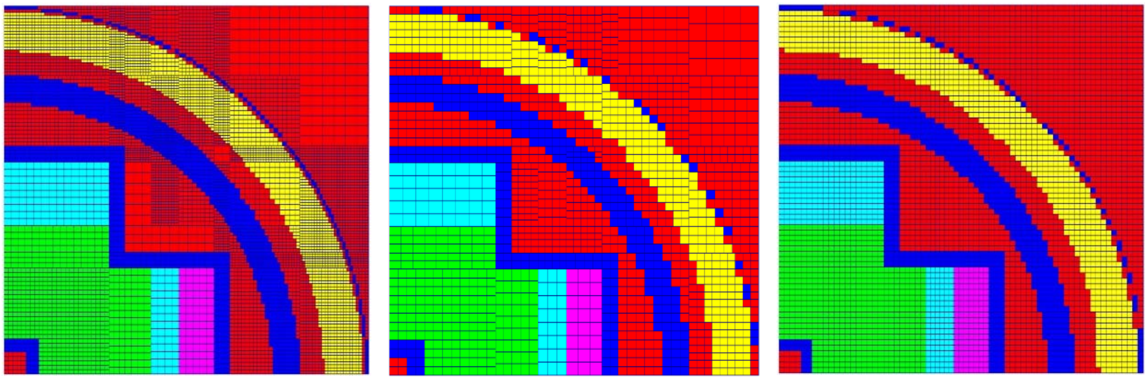

Reference-mesh Variable-mesh Uniform-mesh

Figure 3. Fine mesh distributions for z-level 2 of the simplified VENUS-3 model.

The flux distributions obtained from all three mesh distributions are in good agreement. We observe the maximum differences of ~10% and ~16% for the variable-mesh and the uniform-mesh compared to the reference-mesh, respectively. The computation times of PENTRAN are: 1,233.9 sec for the reference-mesh (8-processor parallel calculation), 284.2 sec for the variable-mesh (serial calculation), and 875.5 sec for the uniform-mesh (serial calculation). The variable-mesh yields more accurate results than the uniform-mesh, while requiring significantly less computation time. This is caused by the fact that the variable-mesh is generated based on expert's knowledge of problem physics, i.e. preserving material boundaries and considering particle interactions (mean-free-path). As a result the mesh density varies depending on problem geometry and material properties. In contrast, the uniform-mesh leads to over-meshing and large computing time. Note that the reference-mesh calculation was performed on 8 processors, because it requires more memory than what is available per processor.

6. EVALUATION OF THE ALGORITHM FOR SELECTING AN EFFECTIVE DOMAIN DECOMPOSITION STRATEGY FOR PARALLEL COMPUTING

Thus far, we have examined the effectiveness of our expert system in a serial environment. In this section, we evaluate our algorithm for selecting an effective domain decomposition strategy for parallel processing.

For reference calculations, we utilize the variable-mesh distribution as described in Section 5.4 with S8 level-symmetric quadrature set, P3 Legendre scattering order, and the first 26 energy groups of the BUGLE-96 library. We perform 4-processor parallel calculations for three different domain decomposition strategies: spatial, angular-spatial, and angular, and examine the PPI of each strategy.

6.1. Evaluation of Algorithms for Estimating the Four Factors Affecting Parallel Performance

6.1.1. Evaluation of the algorithm for estimating memory required per processor

The amounts of memory required per processor for spatial, angular-spatial, and angular domain decomposition strategies are 290.3 MB, 414.4 MB and 665.8 MB, respectively. The spatial decomposition strategy results in the lowest amount of memory per processor, followed by the angular-spatial and the angular decomposition strategies. This is because the angular decomposition requires all of the spatial arrays for angular fluxes and moments, while the spatial decomposition partitions them among processors.

6.1.2. Evaluation of the algorithm for estimating load balance

As mentioned earlier, the difference in total number of fine meshes per processor causes load imbalance. For this problem, there are four z-levels. Z-levels 1 and 4 have significantly fewer numbers of fine meshes as compared to z-levels 2 and 3; therefore, there is a significant load imbalance for the spatial decomposition strategy.

6.1.3. Evaluation of the algorithm for estimating the CPCM

Fig. 4 shows a comparison of the actual and the predicted CPCMs. We observe that our prediction follows a similar trend as the actual CPCM.

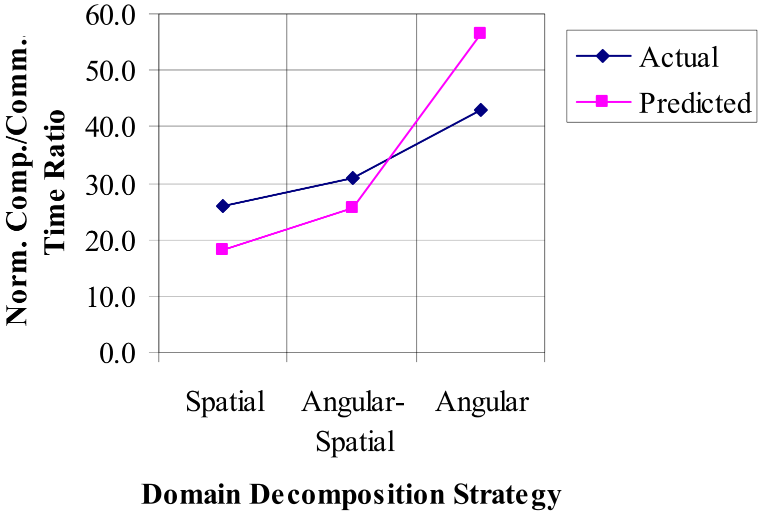


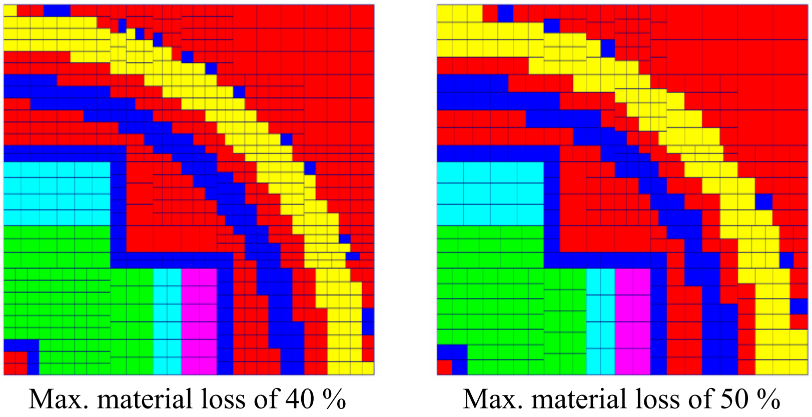
Figure 4. Comparison of the actual and the predicted CPCMs.

6.1.4. Evaluation of the algorithm for estimating the DCP

We utilize three mesh distributions, created using PENXMSH, to examine our DCP estimation algorithm.

Max. material loss of 13 % (reference), S8 level-symmetric quadrature set
Max. material loss of 40 %, S6 level-symmetric quadrature set
Max. material loss of 50 %, S6 level-symmetric quadrature set

Fig. 5 shows the last two fine mesh distributions for z-level 2 of the simplified VENUS-3 model.



Max. material loss of 40 %

Max. material loss of 50 %

Figure 5. Fine mesh distributions for z-level 2 of the simplified VENUS-3 model.

We perform PENTRAN serial calculations using P0 and 26 energy groups, with the inner flux iteration tolerance of 10.0%. Note that for the reference calculation, the inner flux tolerance is 0.1%. Fig. 6 shows a comparison of the inverses of the DCPs of the three mesh distributions.

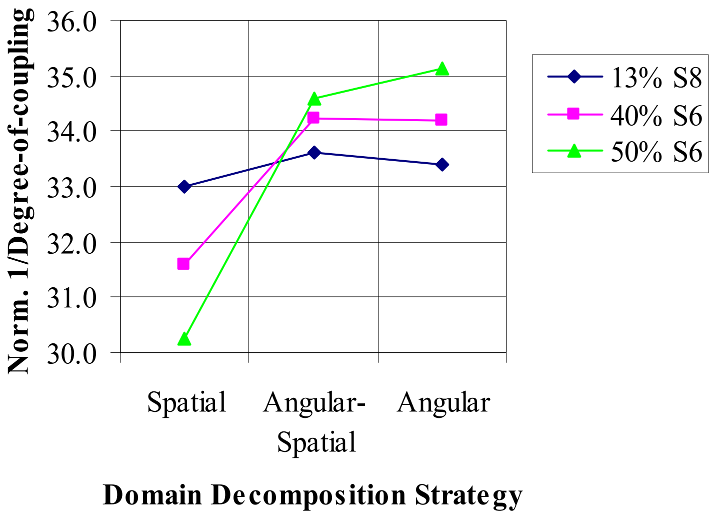


Figure 6. Comparison of the inverses of the DCPs of different mesh distributions.

We observe a similar behavior of DCPs among the three mesh distributions. The 40%-material-loss mesh, which has more meshes than the 50%-material-loss mesh, yields closer values to the reference mesh. Table I gives the computing resources required for the PENTRAN and the DCP calculations. The 40%-material-loss mesh provides similar information at significantly lesser cost compared to the reference mesh.

Table I. Comparison of required computing resources

Max. material loss (%)	Quadrature set	PENTRAN serial calculation		Degree-of-coupling calculation (sec)
		Memory (MB/proc)	CPU time (sec)	
13	S8	1086.8	1221.1	1033.0
40	S6	487.8	401.9	361.4
50	S6	297.8	250.9	166.5

6.2. Evaluation of the Algorithm for Estimating the PPI

The performance of any algorithm can be inversely related to its computing time. Since the PPI is estimated based on the simulation of a transport sweep, we define an actual parallel-performance-index (APPI) as the inverse of a transport sweep time. Fig. 7 shows a comparison of the APPIs and the PPIs of different mesh distributions.

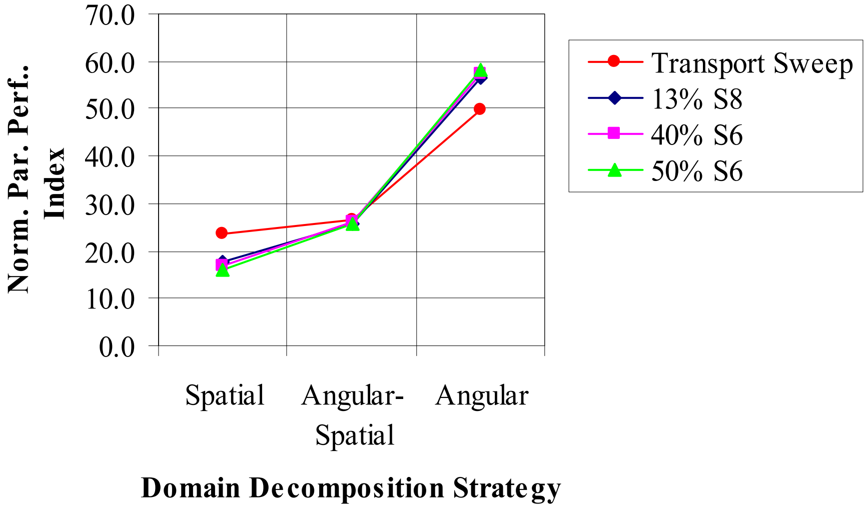


Figure 7. Comparison of the APPIs (transport sweep) and the PPIs.

Our prediction, similar to the actual transport sweep performance, indicates that the angular decomposition strategy is the most effective domain decomposition strategy. It is interesting to compare our prediction to the overall parallel performance of the calculation. Consequently, this means that the APPI is estimated based on the inverse of the problem wall-clock time. Fig. 8 shows a comparison of the APPIs (overall) and the PPIs.

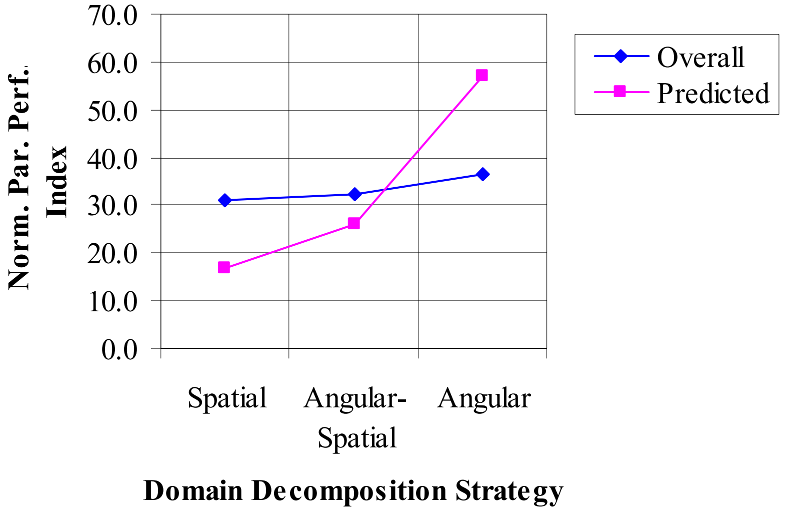


Figure 8. Comparison of the APPIs (overall) and the PPIs.

The PPIs are deviated from the APPIs. This is because there are other parts of the calculation that are affected by parallel domain decomposition and not included in our algorithm. Nonetheless, we observe a similar trend between the two indexes.

6.3. Computation Time of the Algorithm for Selecting an Effective Domain Decomposition Strategy

Our algorithm required a total computation time of 859.3 sec, of which 89% is spent on estimation of the DCP. The wall-clock times for spatial, angular-spatial, and angular domain decomposition strategies are 8,990.7 sec, 8,642.6 sec, and 7,703.0 sec, respectively. These results indicate that our predictive model is very effective. It accurately predicts the effective domain decomposition strategy in a reasonable amount of time (~10% of the actual computation time). However, for this problem, the differences in the DCPs (Fig. 6) do not affect the resulting PPIs (Fig. 7). This suggests that the PPI can be simply derived from the CPCM. Fig. 9 shows a comparison of the APPIs (transport sweep) and the CPCM PPIs.

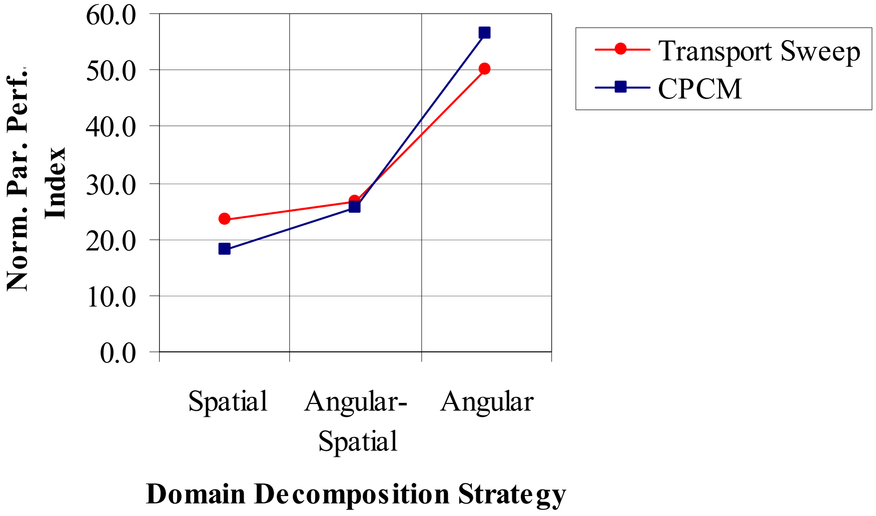


Figure 9. Comparison of the APPIs (transport sweep) and the CPCM PPIs.

We observe a good agreement between the two indexes. This demonstrates that for this problem the CPCM is sufficient for obtaining the PPI. As a result, the computation of our predictive algorithm is reduced significantly to 96 sec (~1% of the actual computation time).

7. CONCLUSIONS

An expert system for preparing an effective mesh distribution for the S_N method has been developed and verified. It consists of two main parts: 1) generation of an effective mesh distribution in a serial environment, common for both serial and parallel calculations, and 2) selection of an effective domain decomposition strategy for parallel computing. The system has been successfully tested within the PENTRAN code system for simulating the VENUS-3 experimental facility. Our analysis has demonstrated that this expert system can reduce user's time and effort for preparing an effective particle transport simulation in both serial and parallel environments.

ACKNOWLEDGEMENTS

This research is a part of an on-going project sponsored by Nuclear Engineering Education Research (NEER). The authors express their appreciations to Dr. Glenn Sjoden, USAF, and Dr. Vefa Kucukboyaci, Westinghouse Electric Company, for their valuable technical advice.

REFERENCES

1. A. Patchimpattapong and A. Haghighat, "Developing an Expert System for Preparing an Effective Mesh Distribution for the S_N Method in the Parallel Environment," *Proceedings of the 12th ANS Radiation Protection and Shielding Division Topical Meeting (RPSD 2002)*, Santa Fe, NM, April 14-18, CD (2002).
2. G. E. Sjoden and A. Haghighat, "PENTRAN – A 3-D Cartesian Parallel S_N Code with Angular, Energy, and Spatial Decomposition," *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, New York, October 5-9, Vol. 1, pp. 553-562 (1997).
3. L. Leenders, *LWR-PVS BENCHMARK EXPERIMENT VENUS-3 (with Partial Length Shielded Assemblies)*, SCK.CEN, Nuclear Research Department, Mol, Belgium (1988).
4. AutoDesk, *AutoCAD 2002 User's Guide* (2002).
5. J. E. White et al., "BUGLE-96: A Revised Multigroup Cross Section Library for LWR Applications Based on ENDF/B-VI Release 3," *Proc. Tpl. Mtg. Radiation Protection and Shielding*, No. Falmouth, Massachusetts, Vol. 2, pp. 1071 (1996).
6. A. Haghighat, G. E. Sjoden and V. N. Kucukboyaci, "Effectiveness of PENTRAN's Unique Numerics for Simulation of the Kobayashi Benchmarks," *Progress in Nuclear Energy*, **39**, n 2, pp. 191-206 (2001).