

## **Paralelización, en Grupos de Energía y Direcciones Angulares en el Código AZTRAN usando MPI**

**Julian A. Duran Gonzalez, Edmundo del Valle Gallegos**  
*Instituto Politécnico Nacional, Escuela Superior de Física y Matemáticas*  
Av. Luis Enrique Erro s/n, Unidad Profesional Adolfo López Mateos, Zacatenco, Delegación  
Gustavo A. Madero, C.P. 07738, Ciudad de México; México.  
redfield1290@gmail.com, edmundo.delvalle@gmail.com

**Armando M. Gómez Torres, Alejandro Campos Muñoz, Juan Carmona Lemus**  
*Instituto Nacional de Investigaciones Nucleares*  
Carretera México Toluca-La Marquesa s/n, Ocoyoacac, Estado de México. C.P. 52750  
armando.gomez@inin.gob.mx, acamposm1204@gmail.com, juan.carmona@inin.gob.mx

### **Resumen**

En el presente trabajo se describe un primer ejercicio con miras a la implementación de una versión en paralelo del código de transporte AZTRAN (AZtlan TRANsport), el cual es una herramienta para realizar análisis neutrónicos, que pertenece a la Plataforma AZTLAN. El código es capaz de resolver numéricamente la ecuación de transporte de neutrones dependiente del tiempo en geometría cartesiana y está escrito en el lenguaje Fortran 90, por lo tanto, para implementar una versión en paralelo se puede optar por una interfaz de memoria compartida (OpenMp) o una interfaz de paso de mensajes (MPI), inclusive un híbrido con ambos. En este caso se decidió utilizar MPI ya que por la forma en que AZTRAN fue programado permite obtener mayor rendimiento, además en la mayoría de los códigos de transporte actuales es el camino que se está tomando para su implementación en paralelo. En el desarrollo del trabajo se implementaron tres formas para paralelizar los procesos en el código: por grupo de energía, grupo de energía-dirección angular en un cuadrante y finalmente por grupo de energía-dirección angular en todos los cuadrantes. En esta versión de AZTRAN se simuló algunos problemas Benchmark en estado estacionario, mostrando en general aceleraciones plausibles que motivan a seguir haciendo mayores esfuerzos para mejorar el rendimiento del código.

### **1. INTRODUCCIÓN**

En la actualidad las computadoras cuentan con un poder de cómputo que hace años no se tenía, incluso sin tener acceso a un clúster, las computadoras portátiles cuentan con procesadores con varios núcleos los cuales pueden ser explotados para realizar cálculos en paralelo.

Es debido a esto que se decidió realizar una implementación en paralelo en el código AZTRAN [1][2][3] sobre los grupos de energía y direcciones angulares los cuales no requieren una cantidad de procesos considerable, una metodología diferente se implementó y probó con éxito en el código AZKIND [4].

AZTRAN resuelve la ecuación de transporte dependiente del tiempo usando el método de ordenadas discretas en geometría XYZ, el cual usa la técnica de multigrupos para discretizar la energía, el método nodal RTN-0 para la parte espacial y el método  $\theta$  para la parte dependiente del tiempo. Además, utiliza el método de iteración de fuente para resolver las iteraciones internas y el método de las potencias para las iteraciones externas.

En este trabajo la solución numérica de la ecuación de transporte en estado estacionario es paralelizada usando una interfaz de paso de mensajes (MPI) [5] en una versión bidimensional de AZTRAN, con lo cual se obtiene una reducción de tiempo plausible, el inconveniente de esta metodología es que se limita el número de procesos en el orden de las ordenadas discretas y los grupos de energía que presenta el problema a resolver.

## 2. IMPLEMENTACIÓN DE LA VERSIÓN PARALELA EN AZTRAN

Para la implementación en paralelo se utilizaron tres caminos; en el primero se paralelizó por grupo de energía, el segundo por grupo de energía y dirección angular en un cuadrante y finalmente para cada grupo de energía y dirección angular en todos los cuadrantes, todos estos caminos parten de la ecuación de transporte de neutrones bidimensional en estado estacionario para un medio multiplicativo sin fuente externa discretizada en energía y ángulo.

$$\begin{aligned} \mu_n \frac{\partial}{\partial x} \psi_{gn}(\vec{r}) + \eta_n \frac{\partial}{\partial y} \psi_{gn}(\vec{r}) + \Sigma_{tg}(\vec{r}) \psi_{gn}(\vec{r}) = \\ \sum_{g'=1}^G \Sigma_{s_{g' \rightarrow g}}(\vec{r}) \sum_{n'=1}^N \omega_{n'} \psi_{g'n'}(\vec{r}) + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^G \nu \Sigma_{fg'}(\vec{r}) \sum_{n'=1}^N \omega_{n'} \psi_{g'n'}(\vec{r}) \end{aligned} \quad (1)$$

$$g = 1, \dots, G$$

$$n = 1, \dots, N$$

donde

$g$  : Número del grupo de energía.

$G$  : Número total de grupos de energías.

$n$  : Número de dirección angular.

$N$  : Número total de direcciones angulares.

$\mu_n, \eta_n$  : Dirección discreta correspondiente a la  $n$ -ésima dirección angular.

$\psi_{gn}(\vec{r}) = \psi_g(\vec{r}, \mu_n, \eta_n)$  : Flujo angular correspondiente a la  $n$ -ésima dirección angular y  $g$ -ésimo grupo de energía.

$\Sigma_{tg}(\vec{r})$  : Sección eficaz macroscópica total correspondiente al grupo de energía  $g$ .

$\Sigma_{s_{g' \rightarrow g}}(\vec{r})$  : Sección eficaz macroscópica de dispersión del grupo  $g'$  al  $g$ .

$\Sigma_{fg}(\vec{r})$  : Sección eficaz macroscópica de fisión correspondiente al grupo de energía  $g$ .

$\nu$  : Número promedio de neutrones liberados por fisión.

$\chi_g$  : Probabilidad de que los neutrones surgen de las fisiones en el grupo de energía  $g$ .

$k_{eff}$  : Factor de multiplicación efectiva.

La forma en que AZTRAN resuelve la ecuación de transporte en el dominio es de tal manera que resuelve una matriz local para cada celda en un grupo de energía y una dirección angular así que el número total de ecuaciones que resuelve está dado por:

$$\text{Número de ecuaciones} = (NGR) \times (NSN) \times (3) \times (N_x) \times (N_y) \quad (2)$$

donde

$NGR$  : Número total de grupos de energía.

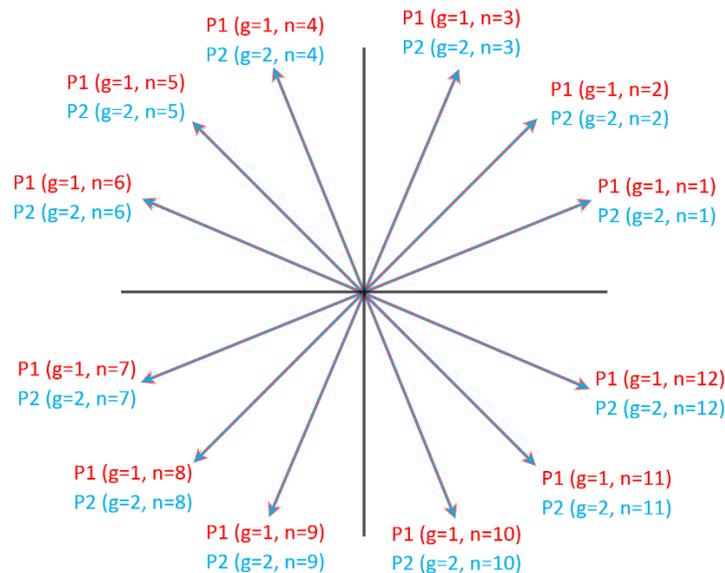
$NSN$  : Número total de ordenadas discretas.

$N_x$  : Número de intervalos en la dirección  $x$ .

$N_y$  : Número de intervalos en la dirección  $y$ .

El algoritmo está diseñado de forma que calcula las fuentes de neutrones por dispersión y fisión primero y a partir de ellas se resuelve los flujos angulares, esto permite que haya una independencia en los cálculos, por ejemplo, se obtendría el mismo resultado en un problema de dos grupos de energía si se resuelve primero el grupo rápido y después el térmico o viceversa.

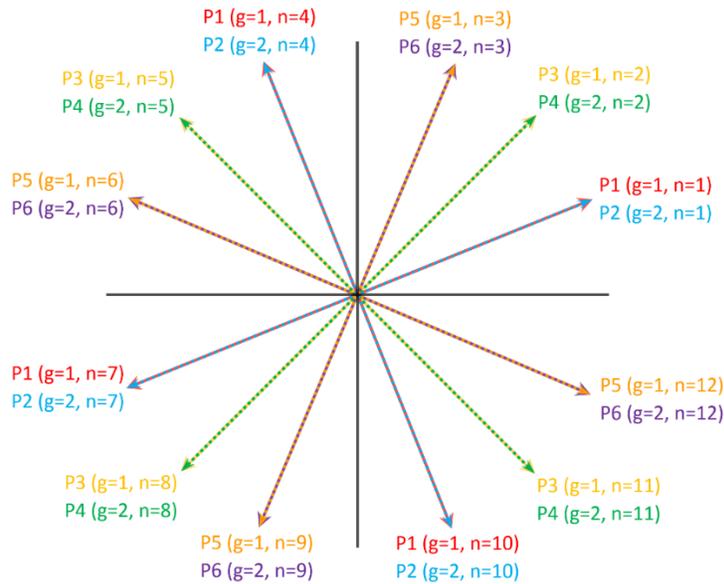
El primer camino en el que fue llevado a cabo la paralelización es por grupo de energía, lo cual solo permite usar un proceso o usar  $NGR$  procesos. En la Figura 1 se muestra la distribución de los procesadores para un problema de dos grupos de energía y una aproximación  $S_4$ , en el cual solo se puede usar dos procesadores, los cuales resuelven la ecuación para cada grupo.



**Figura 1. Designación de cada procesador por grupo de energía**

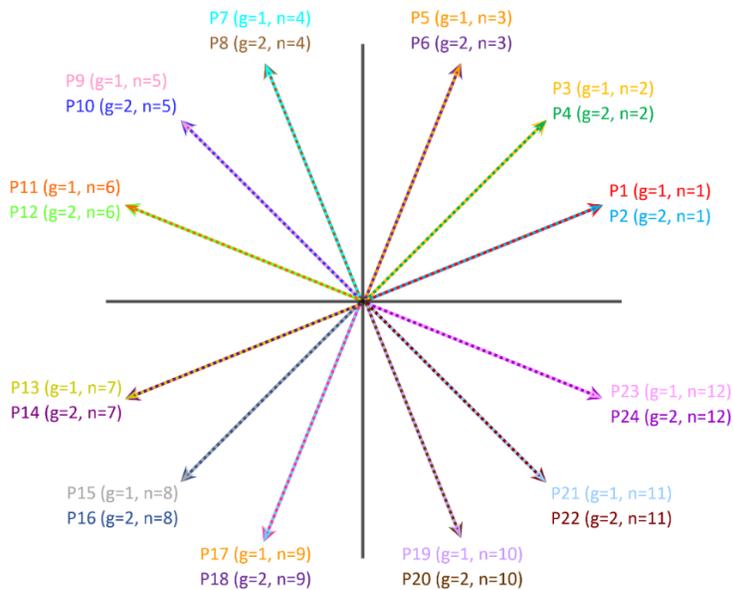
La siguiente forma de paralelizar es por grupo de energía y dirección angular en un cuadrante, la cual solo permite usar un proceso o  $NGR \times NSN/4$  procesos. En la Figura 2 se muestra la distribución de los procesadores para el caso análogo de la Figura 1, solo que en este caso se puede usar seis

procesadores, los cuales resuelven independientemente un grupo y una dirección angular en un cuadrante.



**Figura 2. Designación de cada procesador por grupo de energía y dirección angular en un cuadrante**

Finalmente, la última forma seleccionada fue resolver por grupo de energía y dirección angular en todos los cuadrantes el cual permite usar hasta  $NGR \times NSN$  procesos en paralelo, el inconveniente de esta forma es que está limitada a problemas con condiciones de frontera de vacío en las cuatro caras, si suponemos el mismo problema de los casos anteriores se puede usar hasta veinticuatro procesadores, los cuales resuelven independientemente para cada grupo de energía y dirección angular, en la Figura 3 se muestra esta distribución.

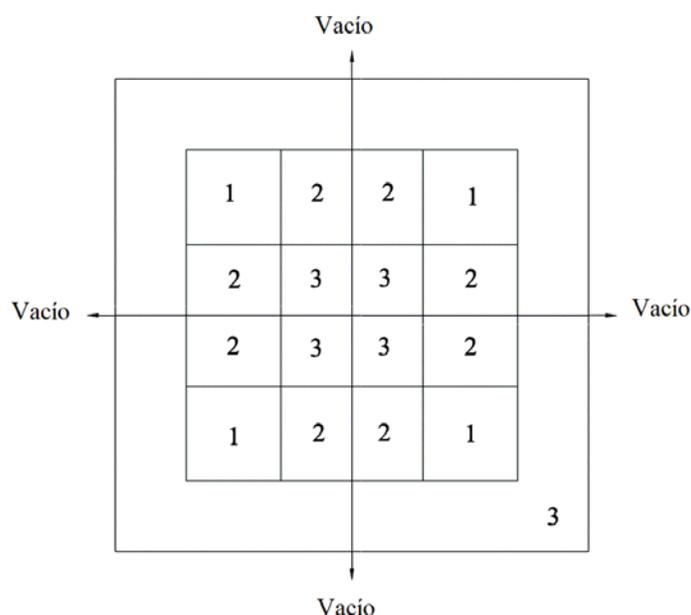


**Figura 3. Designación de cada procesador por grupo de energía y dirección angular en todos los cuadrantes**

### 3. RESULTADOS

Para la obtención de resultados se utilizaron dos problemas de referencia muy conocidos y además probados en el área, el equipo utilizado para realizar los cálculos cuenta con una familia de procesadores Intel® Xeon® CPU E5-2699 v3 @2.30GHz x 56 con el cual se pudieron realizar los cálculos y una comparación de las metodologías.

El primer Benchmark corresponde al TWIGL-2D [6], el cual es un modelo de un reactor de agua ligera para dos grupos de energía el cual está basado en la teoría de difusión, este modelo tiene una dimensión de 160cm x 160 cm y consta de tres materiales, el material fisible primario ocupa las zonas 1 y 2, mientras la zona 3 contiene al cobertor, que también contiene material fisible. En la Figura 4 se muestra un modelo del problema, así como las condiciones de frontera.



**Figura 4. Modelo TWIGL-2D**

En la referencia [6] está descrito el Benchmark, así como los resultados obtenidos por el código MPACT el cual es un código que resuelve la ecuación de transporte de neutrones utilizando el método de las características (MOC), el cual permite tener una mejor referencia si se compara con los resultados obtenidos por la teoría de difusión. En la Tabla I se compara el valor del factor de multiplicación efectiva obtenido por AZTRAN con un mallado de la celda de 0.5cm x 0.5 cm utilizando una aproximación  $S_6$  mientras MPACT utilizó un espaciado de rayos de 0.03 cm, y en la parte angular 16 azimutales y 3 polares con una cuadratura Chebyshev-Yamamoto.

**Tabla I. Resultados del factor de multiplicación efectiva TWIGL-2D**

AZTRAN	MPACT
0.91615	0.91601
Diferencia pcm	-15 pcm

$$pcm = \frac{k_{ref} - k_{AZTRAN}}{k_{ref}} \times 10^5$$

Se puede observar que se obtiene resultados adecuados al comparar contra MPACT, tomando en cuenta que ambos códigos usan diferentes metodologías para resolver la ecuación de transporte.

El siguiente paso fue resolver este problema con AZTRAN con diferentes tamaños de la malla y variando las ordenadas discretas, para ello utilizando diferentes procesos para observar la reducción de tiempo obtenido por la paralelización.

En las siguientes tablas se muestran los resultados obtenidos.

En la Tabla II se muestran los tiempos de cómputo requeridos para resolver el problema usando una aproximación  $S_2$ , el cual equivale a cuatro direcciones angulares en todos los cuadrantes, así que se resolvió usando por el método por grupo de energía y también por grupo de energía-dirección angular en todos los cuadrantes.

**Tabla II. Tiempos de cómputo obtenidos para el TWIGL-2D usando un  $S_2$**

Procesadores	4cm x 4cm	2cm x 2cm	1cm x 1cm	0.5cm x 0.5cm
1	2.2 s	8.5 s	43.7 s	176.4 s
2 (NGR)	1.5 s	5.7 s	34.7 s	160.1 s
8 (NGR x NSN)	1.2 s	4.2 s	20.9 s	114.2 s

En la Tabla III se resuelve ahora con  $S_4$ , el cual es equiparable a tres direcciones angulares por cuadrante, el método usado fue por grupo de energía y por grupo de energía-dirección angular en un cuadrante.

**Tabla III. Tiempos de cómputo obtenidos para el TWIGL-2D usando un  $S_4$**

Procesadores	4cm x 4cm	2cm x 2cm	1cm x 1cm	0.5cm x 0.5cm
1	6.3 s	23.6 s	107.5 s	526.9 s
2 (NGR)	4.5 s	16 s	94.2 s	448.5 s
6 (NGR x NSN/4)	3.2 s	11.5 s	71.4 s	309.2 s

Finalmente, en la Tabla IV el problema usa una aproximación  $S_6$ , el cual es análogo a seis direcciones angulares por cuadrante, los métodos utilizados fueron por grupo de energía y grupo de energía-dirección angular en un cuadrante.

**Tabla IV. Tiempos de cómputo obtenidos para el TWIGL-2D usando un  $S_6$**

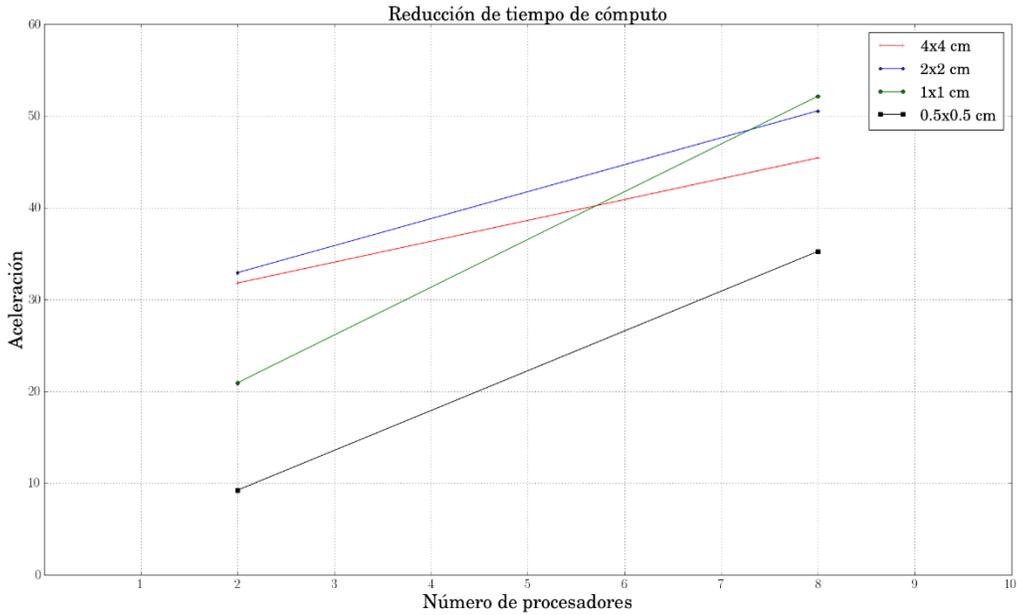
Procesadores	4cm x 4cm	2cm x 2cm	1cm x 1cm	0.5cm x 0.5cm
1	11.7 s	47 s	219.7 s	1062.9 s
2 (NGR)	7.9 s	34.1 s	197.3 s	912.2 s
12 (NGR x NSN/4)	5.9 s	31.8 s	160.5 s	667.5 s

La razón de que no se utilizó por grupo de energía-dirección angular en todos los cuadrantes fue que se encontró que a partir de un número de procesadores el proceso en paralelo es más lento que con un procesador ya que la comunicación entre procesadores no está optimizada.

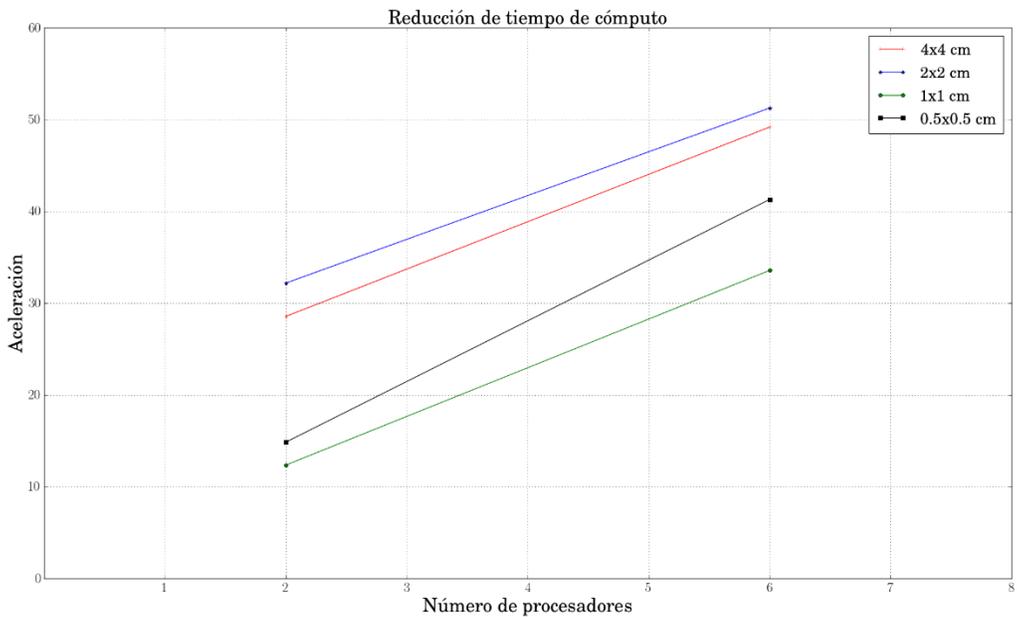
En la siguiente ecuación se define el porcentaje de aceleración debido a la paralelización.

$$\text{Aceleración} = \left(1 - \frac{\text{Tiempo cómputo paralelo}}{\text{Tiempo cómputo serie}}\right) \times 100 \quad (3)$$

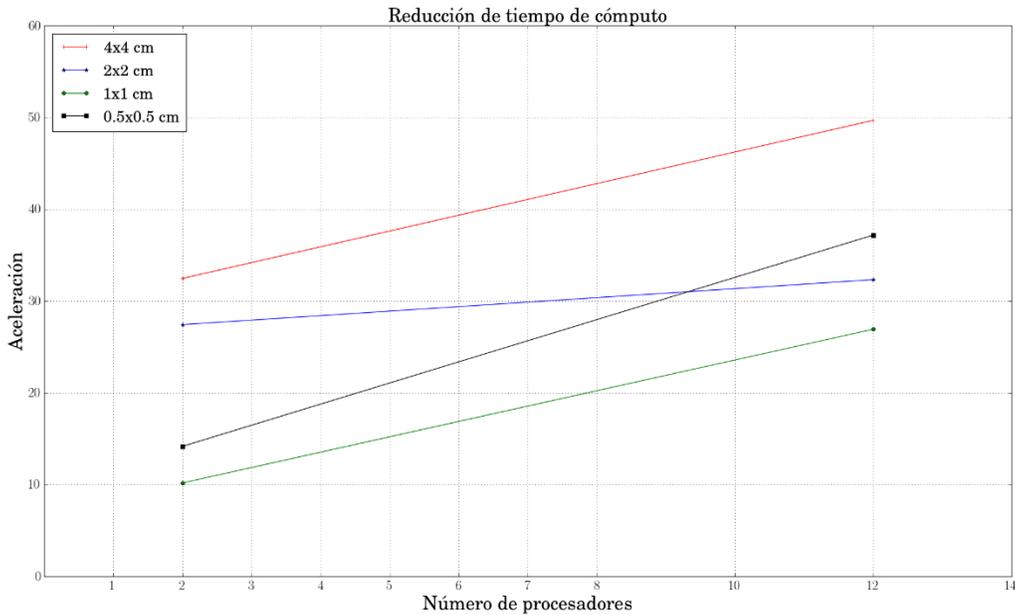
En la Figuras 5-7 se muestran el aceleramiento producido por los procesos en paralelo.



**Figura 5. Aceleración del TWIGL-2D S<sub>2</sub>**



**Figura 6. Aceleración del TWIGL-2D S<sub>4</sub>**



**Figura 7. Aceleración del TWIGL-2D S<sub>6</sub>**

En las Figuras 5-7 se puede observar la aceleración obtenida por los procesadores donde se puede observar una reducción del 10% - 50 % dependiendo del problema además se encuentra mayor reducción en donde la malla es menos fina, y esto es debido a que solo se paralelizó la parte de energía y angular.

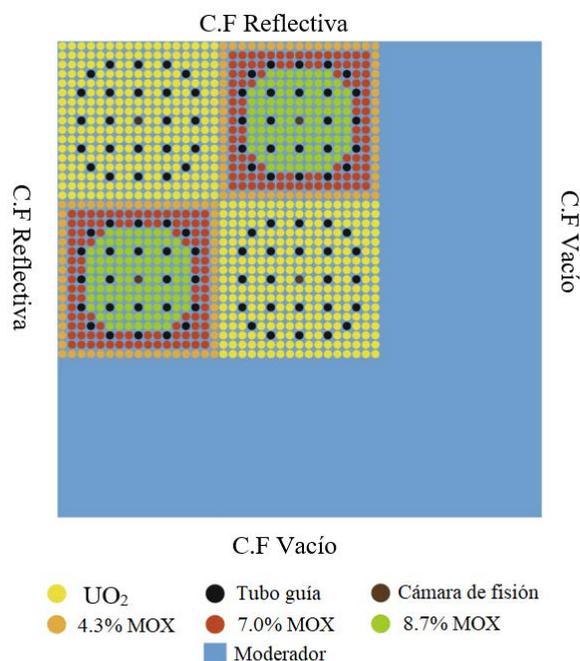
El siguiente Benchmark que se reprodujo es el C5G7-2D OECD/NEA [7] el cual es un problema muy popular para cálculos de transporte determinísticos. Las dimensiones generales de la configuración 2D son 64.26 x 64.26 cm, mientras que cada ensamble es de 21.42 x 21.42 cm. Las condiciones de frontera aplicadas son reflexivas en la parte superior e izquierda, y de vacío sobre la derecha e inferior de la geometría. Cada ensamble de combustible está compuesto por una celda de 17x17, la descripción detallada del problema puede ser encontrada en la referencia [7].

En la Figura 6 se muestra la configuración del problema.

En la Tabla V se muestran los valores del factor de multiplicación obtenidos por AZTRAN para diferentes ordenadas discretas y para dos diferentes refinamientos.

**Tabla V. Factores de multiplicación efectiva C5G7-2D**

AZTRAN S <sub>2</sub> (1x1)	AZTRAN S <sub>2</sub> (2x2)
1.1865	1.1854
AZTRAN S <sub>4</sub>	AZTRAN S <sub>4</sub>
1.1847	1.1848
AZTRAN S <sub>6</sub>	AZTRAN S <sub>6</sub>
1.1848	1.1849



**Figura 6. Modelo del ensamblaje C5G7-2D**

Al igual que el caso anterior se procedió a resolver el problema con diferentes procesadores, pero en este problema solo se utilizó la metodología en grupos de energía ya que como se mencionó anteriormente, al usar un determinado número de procesadores, los procesos en paralelo tardan más que el problema en serie.

En las tablas a continuación se muestran los tiempos de cómputo obtenidos para el problema.

**Tabla VI. Tiempos de cómputo obtenidos para el C5G7-2D usando una aproximación S<sub>2</sub>**

Procesadores	1x1	2x2
1	272.7 s	1436.2 s
7	196 s	870.6 s
Aceleración	28.12 %	39.38 %

**Tabla VII. Tiempos de cómputo obtenidos para el C5G7-2D usando una aproximación S<sub>4</sub>**

Procesadores	1x1	2x2
1	775.6 s	3462.6 s
7	490.3 s	2087 s
Aceleración	36.78 %	39.72 %

**Tabla VIII. Tiempos de cómputo obtenidos para el C5G7-2D usando una aproximación S<sub>6</sub>**

Procesadores	1x1	2x2
1	1602.7 s	6715.9 s
7	968.1 s	4053.1 s
Aceleración	39.59 %	39.64 %

En las Tablas VI-VIII se puede observar una aceleración en promedio del 39 % el cual no es un tiempo óptimo para el número de procesadores que se utilizan, por lo que esto motiva a que se siga trabajando para optimizar la comunicación entre procesadores y esto aumentaría la reducción de tiempo de cómputo.

#### 4. CONCLUSIONES

En el presente trabajo se presentó una forma de implementar la paralelización en el código AZTRAN por medio de los grupos de energía y direcciones angulares los cuales permiten usar hasta  $(NGR) \times (NSN)$  procesos, la limitante de esta forma de implementación es que solo se puede usar un solo proceso o el máximo determinado por el problema además se encontró que a partir de un número de procesos la comunicación de información entre procesos se satura de tal forma que el proceso en paralelo es más tardado que en serie, los tiempos de cómputo obtenidos pueden ser mejorados si se optimiza el intercambio de comunicación entre procesadores, esto impulsa a trabajar en las metodologías para optimizarlas y así se reduciría más los tiempos de cómputo y además se podría usar más procesos en paralelo. Finalmente, cuando estas formas de paralelización estén optimizadas se procederá a paralelizar la parte espacial, la cual permitirá obtener una mayor reducción en los tiempos de cómputo.

#### AGRADECIMIENTOS

Los autores agradecen el apoyo financiero recibido del proyecto estratégico No. 212602 (AZTLAN Platform) del Fondo Sectorial de Sustentabilidad Energética CONACYT-SENER.

El primer autor agradece al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca que está siendo otorgada para la realización de sus estudios de doctorado.

#### REFERENCIAS

1. “AZTLAN Platform”, [www.aztlanplatform.mx](http://www.aztlanplatform.mx) (2018).
2. Gómez Torres, A. M., Puente Espel, F., del Valle Gallegos, E., François, J. L., Martín-del-Campo, C. and Espinosa-Paredes, G. (2015). AZTLAN: Mexican Platform for Analysis and Design of Nuclear Reactors. In *Proceedings of ICAPP 2015*, Nice, France. May 03-06. Paper 15493.
3. Duran Gonzalez, J. A. *Implementación de la Cinética en el Código de Transporte Tridimensional AZTRAN*, tesis maestría, Instituto Politécnico Nacional, ESFM, México (2017).
4. Hernández Rodríguez, A., Gómez Torres, A. M., del Valle Gallegos, E. “HPC implementation in the time-dependent neutron diffusion code AZKIND.” *Annals of Nuclear Energy*, **99**, pp. 174-182 (2017).
5. Gropp, W., E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, Cambridge, Massachusetts: MIT Press, (1994).
6. Ang Zhu, Yunlin Xu, Aaron Graham, Mitchell Young, Thomas Downar and Liangzhi Cao, “Transient Methods for pin-resolved whole core transport using the 2D-1D Methodology in MPACT”, *Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method*, Nashville, April 19-23, ANS (2015).
7. “Benchmark on Deterministic Transport Calculations Without Spatial Homogenization”, [www.oecd-nea.org/science/docs/2003/nsc-doc2003-16.pdf](http://www.oecd-nea.org/science/docs/2003/nsc-doc2003-16.pdf) (2003).