

Pre Procesamiento de archivos de entrada para el código AZTRAN.

Samuel Vargas E.

Instituto Nacional de investigaciones Nucleares
Carretera México-Toluca s/n, La Marquesa, Ocoyoacac, México C.P. 52750
samuel.vargas@inin.gob.mx

Guillermo Ibarra

Instituto Politécnico Nacional
Av. IPN s/n, C.P. 07738, México
guillermoibarra@gmail.com

Resumen

El código AZTRAN comenzó a desarrollarse en el Departamento de Ingeniería Nuclear de la Escuela Superior de Física y Matemáticas del Instituto Politécnico Nacional, con el propósito de resolver numéricamente diversos modelos que surgen de la física y de la ingeniería de reactores nucleares. El código aún se encuentra en desarrollo y es parte de la AZTLAN Platform: Desarrollo de una plataforma mexicana para el análisis y diseño de reactores nucleares. Debido a la complejidad para generar un archivo de entrada para el código, se desarrolla un script basado en lenguaje D, con la finalidad de hacer más fácil su elaboración, basándose en un nuevo formato de archivo de entrada el cual incluye tarjetas específicas, las cuales se han dividido en dos bloques, las tarjetas obligatorias y las tarjetas opcionales, incluyendo un pre procesamiento del archivo de entrada para poder identificar posibles errores dentro del mismo, así como un generador de imágenes para el problema específico basado en el interprete python.

1. INTRODUCCIÓN

El código AZTRAN[1] (AZtlan TRAnspOrt Neutronic), el cual comenzó a desarrollarse en el Departamento de Ingeniería Nuclear de la Escuela Superior de Física y Matemáticas, actualmente forma parte de AZTLAN Platform[2]. El código resuelve la ecuación de transporte de neutrones para varios grupos de energía en estado estacionario, en geometría xyz , usando el método de ordenadas discretas S_n para la discretización de la variable angular, el método nodal RTN-0 para la discretización espacial y proporciona una solución al problema de valor propio y determina la distribución de neutrones en un medio específico para el número de grupos de energía que se requieran.

Para resolver el sistema de ecuaciones se define un dominio cartesiano en tres dimensiones para las variables espaciales y se hace pasar sobre él una malla regular para dividirlo en nodos o celdas. Si en el dominio hay diferentes tipos de materiales, debe buscarse que las líneas de discretización espacial de la malla coincidan con cada una de las interfaces que separan a los materiales.

La elaboración de un archivo de entrada para el código AZTRAN puede resultar difícil e incluso tedioso ya que se necesita especificar el número de materiales, grupos de energía, el número de

líneas de corte o puntos en cada una de las direcciones espaciales x , y y z , las condiciones de frontera y sobre todo el orden de la aproximación angular S_n .

2. ARCHIVO DE ENTRADA PARA EL CÓDIGO AZTRAN

El archivo de entrada para el código AZTRAN[1], es un documento de texto plano ASCII forzosamente con el nombre *input* y sin extensión alguna, esta dividido en siete bloques de información.

1. Información general del problema.
2. Mallado y condiciones de frontera.
3. Discretización del dominio.
4. Opciones de Rebalance.
5. Distribución de elementos barra en el núcleo y asignación axial de materiales.
6. Criterios de convergencia.
7. Parámetros neutronicos para cada material.

Cada uno de los bloques tiene asociado diferentes datos de entrada y deben ser leídos en un orden específico por el código AZTRAN. Dando como resultado un archivo de entrada con un formato rígido, ya que cualquier alteración en el orden de introducir la información llevaría a resultados equivocados o errores durante la lectura del archivo de entrada. En las Tabla I a VII, se muestran los datos de entrada que forman el archivo de entrada para cada bloque de información.

Tabla I: **Información general del problema.**

ID	Título del Problema.
NUM_MAT	Número total de materiales
GR_ENER	Número de grupos de energía.
NUM_BAR	Número de elementos barra.
SN	Índice de la aproximación de la ordenada discreta.
ANISO	Opción de anisotropía.
REB	Opción de Rebalance.
POT	Potencia del sistema.
PRT	Opción de impresion.

2.1 Ejemplo de archivo de entrada

Consideremos un ejemplo benchmark[3] para un arreglo BWR 4×4 , con dos barras de Gadolina. La descripción geométrica es descrita en la Figura 1. Las barras 1-5 estan constituidas por UO_2 con 3wt % de enriquecimiento, mientras que las barras con Gadolina contienen 3wt. % UO_2 y 3wt. % Gd_2O_3 . En la Figura 2 se muestra el archivo de entrada para el código AZTRAN, considerando que cada barra es homogénea.

Tabla II: Mallado y condiciones de frontera.

NX	Número de líneas de corte o de discretización en la dirección x.
NY	Número de líneas de corte o de discretización en la dirección y.
NZ	Número de líneas de corte o de discretización en la dirección z.
L_BC	Condición de frontera en la cara izquierda.
R_BC	Condición de frontera en la cara derecha.
N_BC	Condición de frontera en la cara más cercana.
F_BC	Condición de frontera en la cara más alejada.
B_BC	Condición de frontera en la cara inferior.
T_BC	Condición de frontera en la cara superior.

Tabla III: Discretización del dominio.

NNX	Número de divisiones por subintervalo de malla gruesa en el eje x.
POS_X	Cortes de la malla gruesa en dirección x.
NNY	Número de divisiones por subintervalo de malla gruesa en el eje y.
POS_Y	Cortes de la malla gruesa en dirección y
NNZ	Número de divisiones por subintervalo de malla gruesa en el eje z.
POS_Z	Cortes de la malla gruesa en dirección z.

Tabla IV: Opciones de Rebalance.

REB_OP	Opción para mallado de rebalance.
--------	-----------------------------------

Tabla V: Distribución de elementos barra en el núcleo y asignación axial de materiales.

MAP	Matriz que define el arreglo geométrico en malla gruesa.
BAR_MAT	Asignación axial de materiales a los elementos barra.
0	Fin de asignación de materiales.

Tabla VI: Criterios de convergencia.

IT_INT	Número de iteraciones internas.
IT_EXT	Número de iteraciones externas.
DIF_INT	Tolerancia para la convergencia de las iteraciones internas.
DIF_EXT	Tolerancia para la convergencia de las iteraciones externas.

3. PRE PROCESAMIENTO

Como se puede ver en la Figura 2, la escritura de un archivo de entrada debe ser en un orden secuencial y cuenta con valores que están relacionados tales como: **NX** que tiene relación con: **NNX** y **POS_X**. Para evitar al usuario escribir información que puede ser omitida, se generó un script en lenguaje D[4] con la finalidad de que el usuario del código AZTRAN pueda generar con mayor facilidad un archivo de entrada. El nuevo formato del archivo de entrada, consta de diferentes tarjetas, que hacen alusión a los diferentes parámetros que necesita el código, dividiendo las tarjetas en dos bloques, **tarjetas obligatorias** y **tarjetas opcionales**.

Tabla VII: **Parámetros neutronicos para cada material.**

Valor	Especificación
$\nu * X_{S_F}$	Producto del número de neutrones producidos por fisión (ν) sigma de fisión y la sección macroscópica de fisión (Σ_f).
KAPPA_F	Producto de la sección macroscópica de fisión (Σ_f) por el valor de producción de energía por fisión (κ).
CHI	Fracción de neutrones que surgen de la fisión en cada grupo de energía.
MAT_DISP	Matriz de dispersión.
F_EXT	Fuente de neutrones independiente.

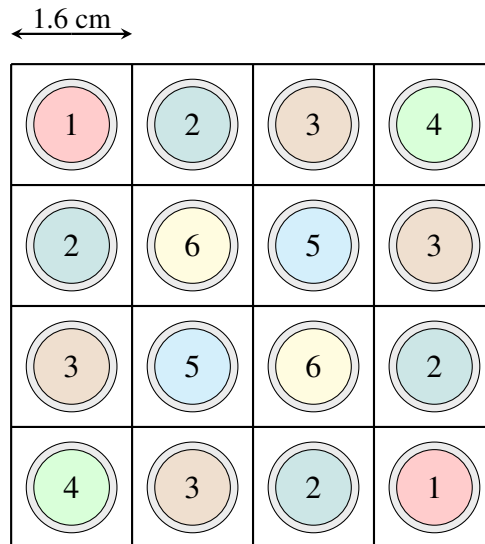


Figura 1: Descripción del arreglo BWR 4×4. Barra 1-5: UO₂, Barra 6: UO₂ + Gd₂O₃

3.1 Tarjetas obligatorias y opcionales

Las tarjetas obligatorias, son las mínimas requeridas para generar un archivo de entrada para el código AZTRAN. En la Tabla VIII se muestran estas tarjetas, así como una descripción de las mismas. Por otra parte las tarjetas opcionales, cuentan con un valor por defecto, por lo que se pueden omitir en el archivo de entrada. En la Tabla IX se muestran las tarjetas opcionales, así como una descripción de las mismas.

3.3 Nuevo archivo de entrada

Con base a las nuevas tarjetas que se incluyen en el pre procesamiento, se puede generar el archivo de entrada para el problema benchmark BWR 4×4, utilizando solamente las tarjetas obligatorias. En la Figura 3 se muestra el nuevo formato del archivo de entrada para el código AZTRAN utilizando el script basado en D. Adicionalmente al pre procesamiento se le ha agregado un script generador de imagenes[5] basado en el interprete Python3 del problema en especifico. La figura 4 muestra una imagen generada para el problema benchmark.

! Bloque Información general.	! Barras.
bwr_4x4	1
2, 2, 2, 4, 0, 1, 1.0E+00, 0	1
! Mallado y condiciones de frontera.	2
5,5,2,1,1,1,1,1,0	2
! Discretización del dominio.	0 ! fin de asignación de materiales
1,1,1,1	! Criterios de convergencia
0	1000, 50, 1.0e-04, 1.0e-04
1.6	! Parámetros neutronicos
3.2	! Mat 1:
4.8	5.234990E-01, 9.958700E-01
6.4	5.784860E-03, 1.013300E-01
1,1,1,1	4.576370E-01, 8.401140E+00
0	1.000000E+00, 0.000000E+00
1.6	4.960860E-01, 1.869730E-02,
3.2	6.415180E-05, 9.346000E-01
4.8	0.000000E+00, 0.000000E+00
6.4	! Mat 2:
1	5.305960E-01, 1.287740E+00
0	5.466620E-03, 2.876210E-02
1.0	4.334190E-01, 2.384610E+00
! Opción de rebalance.	1.000000E+00, 0.000000E+00
0	5.012810E-01, 1.911050E-02,
! Núcleo y materiales	8.562880E-05, 1.014820E+00
! MAP	0.000000E+00, 0.000000E+00
1 1 1 1	
1 2 1 1	
1 1 2 1	
1 1 1 1	

Figura 2: Archivo de entrada para el problema benchmark BWR 4×4.

Tabla VIII: Tarjetas obligatorias

Tarjeta	Valor
TTL	'NOMBRE' Título de identificación del problema.
GR	'GR_ENER' Número de grupos de energía.
BC	'LBC' 'RBC' 'NBC' 'FBC' 'BBC' 'TBC' LBC: Condición de frontera en la cara izquierda. RBC: Condición de frontera en la cara derecha. NBC: Condición de frontera en la cara más cercana. FBC: Condición de frontera en la cara más alejada. BBC: Condición de frontera en la cara inferior. TBC: Condición de frontera en la cara superior.
PIN	'ID' 'MAT' ID: Número de identificación del PIN. MAT: Número(s) de material definido en la tarjeta MATXS
LATTICE	'ID TP X0 Y0 NX NY P' ID: Número de identificación del lattice. TL: Tipo de lattice. X0: Coordenada en el eje x. Y0: Coordenada en el eje Y. NX: Número de elementos lattice en la dirección X. NY: Número de elementos lattice en la dirección Y. PL: Pitch del lattice.
MATXS	'ID' 'TOTAL' 'NU*FIS' 'KAPPA_FIS' 'CHI' 'MAT_DISP' 'F_EXT' ID: Número de material. TOT: sección macroscópica total (Σ_t). NFIS: Producto de número de neutrones producidos por fisión (ν) y la sección macroscópica de fisión (Σ_f). KFIS: Producto del valor de producción de energía por fisión (κ) y la sección macroscópica de fisión (Σ_f). CHI: Valores de las fracciones de neutrones que surgen de la fisión con energías pertenecientes a cada grupo. MAT_DISP: Elementos de a matriz de dispersión. F_EXT: Fuente de neutrones (n/cm^3s)

Tabla IX: Tarjetas opcionales

Tarjeta	Valor
SN	'IN_SN' Índice de aproximación de ordenadas discretas. Los valores que puede tomar: 2, 4, 6, 8, 10, 12, 16. (4 por defecto)
POW	'POWER' Potencia en Watts del sistema. (1.0 por defecto)
ANISO	'DISP' Opción para las matrices de dispersión en la definición de materiales. 0, para las matrices P0. (por defecto) 1, para las matrices P0 y P1. 2, para las matrices P0, P1 y P2. 3, para las matrices P0, P1, P2 y P3. 4, para las matrices P0, P1, P2, P3 y P4.
REB	'REB_OPT' Opción de rebalance 0, rebalance desactivado. (por defecto) 1, rebalance activado.
PRT	'PRT_OPT' Opción de impresión en el archivo de salida. 0: Imprime potencia axial y radial. (por defecto) 1: Imprime mapa de materiales y potencias por planos.
CONV	'IT_INT IT_EXT DIF_INT DIF_EXT' Criterios de convergencia IT_INT: Número de iteraciones internas. (50, por defecto) IT_EXT: Número de iteraciones externas. (1000, por defecto) DIF_INT: Tolerancia para la convergencia de las iteraciones internas. (1E-04, por defecto) DIF_EXT: Tolerancia para la convergencia de las iteraciones externas. (1E-04, por defecto)
SUBMESH	'SUBX SUBY SUBZ' SUBX: Número de divisiones de cada intervalo en el eje X. (1 por defecto) SUBY: Número de divisiones de cada intervalo en el eje Y. (1 por defecto) SUBZ: Número de divisiones de cada intervalo en el eje Z. (1 por defecto)

```
! Titulo del Problema
TTL bwr_4x4

! Número de grupos
GR 2

! Condiciones de frontera
BC 1 1 1 1 1 0

! Definicion de pins
PIN 1 1
PIN 2 2

! Lattice 2D
LATTICE 1 0 0.0 0.0 4 4 1.6
1 1 1 1
1 2 1 1
1 1 2 1
1 1 1 1

! Definición de materiales
MATXS 1
5.23499E-01 9.95870E-01
5.78486E-03 1.01330E-01
4.57637E-01 8.40114E+00
1.00000E+00 0.00000E+00
4.96086E-01 1.86973E-02
6.41518E-05 9.34600E-01
0.00000E+00 0.00000E+00

MATXS 2
5.30596E-01 1.28774E+00
5.46662E-03 2.87621E-02
4.33419E-01 2.38461E+00
1.00000E+00 0.00000E+00
5.01281E-01 1.91105E-02
8.56288E-05 1.01482E+00
0.00000E+00 0.00000E+00
```

Figura 3: Nuevo archivo de entrada para el problema benchmark BWR 4×4 .

Nivel 1

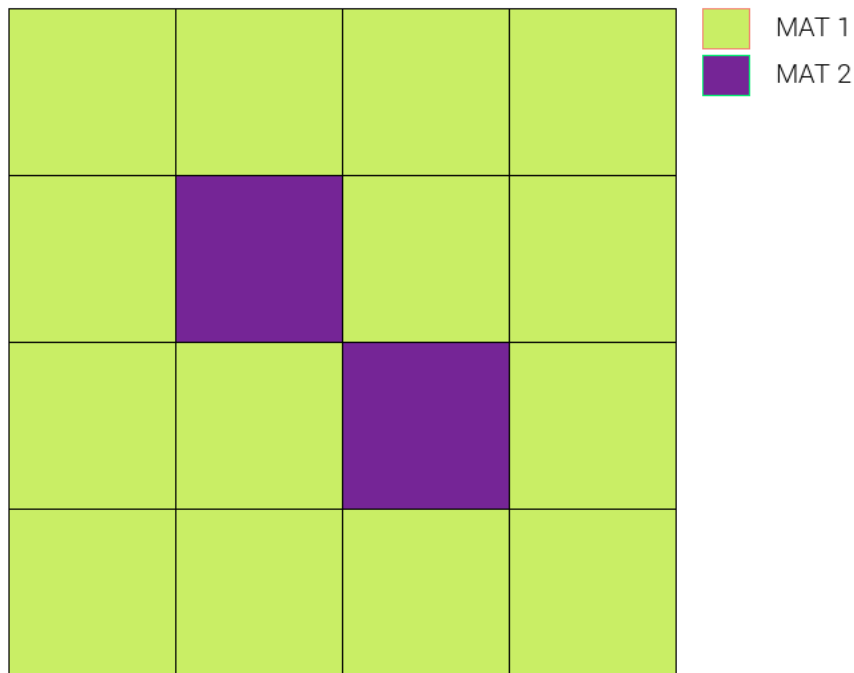


Figura 4: Imagen generada con el script Python para el problema Benchmark BWR 4×4

CONCLUSIONES

El pre procesamiento de archivos de entrada para el código AZTRAN ha comenzado a tener interés por el grupo de usuarios de dicho código, dicho pre procesamiento se encuentra en constante desarrollo ya que los códigos AZTRAN, AZNHEX y AZKIND también lo están. Actualmente se cuenta con una primera versión para el código AZTRAN, y se esta trabajando en las versiones para los códigos restantes.

AGRADECIMIENTOS

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

REFERENCIAS

1. Samuel Vargas y col. *AZTRAN: AZtlan TRAsport Neutronic code, Manual de usuario*. 2016, págs. 21-41.
2. A. M. Gómez-Torres, F. Puente-Espel y E. Del-Valle-Gallegos. "Mexican platform for analysis and design of nuclear reactors". En: *Proceedings of the International Congress on Advances in Nuclear Power Plants ICAPP (2015)*.
3. Xue Yang y Nader Satvat. "MOCUM: A two-dimensional method of characteristics code based on constructive geometry and unstructured meshing for general geometries". En: *Annals of Nuclear Energy (2012)*.
4. D Programming Language. <http://dlang.org/>.
5. Hirepan Palomares, Guillermo Ibarra y Samuel Vargas. "Pre Procesamineto de archivos de entrada con el intérprete Python para los códigos AZTRAN, AZNHEX y AZKIND". En: *Congreso Técnico Científico ININ-SUTIN (2016)*.