

Documentación del código AZTHECA usando la herramienta DOXYGEN

**Zaira I. Jiménez-Balbuena, José A. Meza-Andrade, Miguel I. Sánchez-Romero,
Alejandría D. Pérez-Valseca, y Gilberto Espinosa-Paredes**
*Área de Ingeniería en Recursos Energéticos, Universidad Autónoma Metropolitana - Iztapalapa
San Rafael Atlixco No. 186, Col. Vicentina, Iztapalapa, 09340, México.*
*zai_ijb579@hotmail.com; antoniomezaandrade@outlook.es; mgl.snchz.rmr@gmail.com;
alejandria.peval@gmail.com; gepe@xanum.uam.mx;*

Armando M. Gómez-Torres
*Instituto Nacional de Investigaciones Nucleares
Carretera México-Toluca La Marquesa s/n, Ocoyoacac, Estado de México C.P. 52750, México.;*
armando.gomez@inin.gob.mx

Resumen

El principal objetivo es explorar la herramienta DOXYGEN para auto-documentación del código AZTHECA. El código termohidráulico AZTHECA permite predecir el comportamiento de un reactor tipo agua en ebullición (BWR) en condiciones de estado estacionario y transitorio, y su desarrollo forma parte de las actividades del grupo de trabajo de termohidráulica (GT-2) del proyecto “AZTLAN Platform”. El código está formado por los modelos matemáticos de la cinética neutrónica, generación de potencia, transferencia de calor, termohidráulica del núcleo, sistemas de recirculación, modelos dinámicos de presión, nivel y sistema de control, todo esto programado en un módulo principal y 32 sub-módulos. Como complemento a las actividades del GT-2 sobre el código AZTHECA se encuentra la documentación de este, permitiendo entender el código y su arquitectura de una forma breve y sencilla, siendo de gran utilidad para los desarrolladores de la plataforma AZTLAN. Adicionalmente se desarrolla un manual de usuario e instalación de DOXYGEN, el cual puede ser utilizado para documentar los códigos de la Plataforma AZTLAN. Para realizar la documentación se ha seleccionado el software DOXYGEN, el cual es una herramienta estándar para generar documentación en línea (HTML) además de manuales de referencia en formatos como LATEX, RTF, PostScript, PDF, HTML comprimido y páginas de manual de Unix.

1. INTRODUCCIÓN

El proyecto AZTLAN platform es una iniciativa nacional liderada por el Instituto Nacional de Investigaciones Nucleares (ININ), que busca situar a México en un nivel internacional competitivo en temas de software para análisis de reactores nucleares, reuniendo a las principales casas de estudios superiores públicas de México como son el Universidad Autónoma

Metropolitana (UAM), Instituto Politécnico Nacional (IPN) y la Universidad Nacional Autónoma de México (UNAM). En este proyecto se modernizan, mejoran e integran los códigos neutrónicos, termohidráulicos y termomecánicos, desarrollados en las instituciones mexicanas, en una plataforma integrada, desarrollada y mantenida por expertos mexicanos para beneficio de las instituciones mexicanas [1]. El proyecto se encuentra dividido técnicamente en 4 Grupos de Trabajo: grupo de neutrónica (GT-1), grupo de termohidráulica (GT-2), grupo de acoplamiento y análisis (GT-3) y el grupo de usuarios (GT-4) [2].

Dentro de las actividades del grupo de termohidráulica se encuentra el desarrollo del código AZTHECA. Este código permite predecir el comportamiento de un reactor tipo agua en ebullición (BWR) en condiciones de estado estacionario y transitorio [3], está formado por los modelos matemáticos de la cinética neutrónica, generación de potencia, transferencia de calor, termohidráulica del núcleo, sistemas de recirculación, modelos dinámicos de presión, nivel y sistema de control. Este código ha sido validado en estado estacionario mediante datos de planta y en estado transitorio con los resultados obtenidos de RELAP 5 [2].

Actualmente, como parte de las actividades del GT-2, se realiza la documentación del código AZTHECA. La documentación de un código permite entender el código y su estructura de una forma breve y sencilla, lo que es de utilidad para el o los desarrolladores de la plataforma AZTLAN. En la documentación es importante definir qué es lo que se hace en cada rutina, es decir, añadir explicaciones de lo que no es evidente, además de cuáles son los algoritmos utilizados, variables, constantes y la fuente de donde se obtuvo. Existen diversas formas de documentar un código de acuerdo con el lenguaje de programación utilizado y si la documentación es interna o externa.

Para realizar la documentación del código AZTHECA se ha seleccionado el software DOXYGEN, el cual es una herramienta estándar para generar documentación en línea (HTML) además de manuales de referencia en formatos como LATEX, RTF, PostScript, PDF, HTML comprimido y páginas de manual de Unix. Por otra parte, puede realizar funciones como [4]:

- Extracción de estructura de códigos no documentados,
- Visualizar relaciones de elementos gráficamente con diagramas de herencia y de colaboración que se generan automáticamente,
- Crear documentación normal a partir de fuentes como: C, C ++, *objective-C*, *C#*, *PHP*, *Java*, *Python*, *ID*, *Fortran*, *VDHL*, *Tcl* y *D*.

El uso de DOXYGEN ha permitido la documentación en investigaciones de diferentes áreas del conocimiento como por ejemplo las del código CLUMPY para señales de rayos- γ de estructuras de materia oscura [5]; código abierto ERGO para cálculos de estructura electrónica de escala lineal [6]; y del proyecto CosmoBolognaLib para calculo cosmológicos [7].

2. METODOLOGÍA DE DOCUMENTACIÓN

Como se mencionó anteriormente, dentro de las actividades del GT-2 se realiza la documentación del código AZTHECA, las etapas planteadas para esta actividad son principalmente tres:

- (1) Elaboración de un manual de instalación y uso de DOXYGEN para códigos desarrollados en Fortran, el cual describirá de manera concisa los pasos para utilizar la herramienta,
- (2) Análisis de la estructura jerárquica del código AZTHECA, definiendo el orden de documentación de las rutinas y obteniendo información faltante para la documentación (proporcionada por el desarrollador del código), y
- (3) Documentación de AZTHECA, ya definido el orden y teniendo la información necesaria, se realiza la documentación.

2.1. Manual de Instalación y Uso de DOXYGEN

El manual consta hasta el momento de las siguientes cinco secciones:

- **INTRODUCCIÓN A DOXYGEN.** En la cual se presenta una breve explicación de lo que es la herramienta DOXYGEN y los complementos necesarios para poder hacer la compilación desde la fuente *UNIX*.
- **INSTALACIÓN.** Sección en la que se describe detalladamente cada uno de los pasos necesarios para instalar la herramienta DOXYGEN.
- **USO DE DOXYGEN.** Permite la utilización de DOXYGEN desde la GUI, señalando los puntos a considerar para la documentación de fortran en DOXYGEN, además se describen la estructura de las funciones de una subrutina, lo que hace posible prevenir la posible no documentación de una variable y dar a conocer la solución a ésta misma.
- **GLOSARIO.** Se presenta una breve descripción de los complementos necesarios para la instalación y utilización de la herramienta DOXYGEN.
- **BIBLIOGRAFÍA.** En esta sección se incluyen las fuentes de información utilizadas [4].

Este manual es probado por varios usuarios, con la finalidad de obtener retroalimentación de su claridad y funcionalidad.

2.2. Documentación con DOXYGEN

A continuación, se describen de forma breve los pasos para la documentación.

1. Desde la terminal se utiliza el comando *Doxywizard*

2. Al ejecutar el comando anterior se obtiene la salida de una interfaz que se divide en tres secciones diferentes: *Wizard*, *Expert* y *Run* las cuales a su vez se clasifican cada una en *topics*.
3. La sección *Wizard* contiene los siguientes *topics*: *Project*, *Mode*, *Output* y *Diagrams*. En el *topic Project* se selecciona el directorio de trabajo desde el que se ejecutará DOXYGEN, se da un nombre al proyecto y una sinopsis del mismo, se añade un logo si se desea, además se especifica la dirección para escanear el código fuente y finalmente se especifica la dirección de destino de la documentación. Para el paso anterior es recomendable que tanto la dirección del código fuente como la dirección de destino de documentación sean la misma para fines de organización.
4. Ahora bien, en el *topic Mode* perteneciente a la sección *Wizard* se selecciona el modo de extracción deseado y el lenguaje de programación para optimizar los resultados.
5. En el *topic* siguiente *Output* se elige el tipo de formato que se necesita para la documentación, ya sea LaTeX o HTML.
6. En el último *topic* de la sección *Wizard* correspondiente a *diagrams* se elige el tipo de estructura esquemática que se desea en la salida de la documentación.
7. Ahora bien, en la sección *Expert*, se cuenta con los *topics*: *Project*, *Build*, *Messages*, *Input*, *Source Browser*, *Index*, además de un *topic* específico para cada una de las salidas de la documentación ya sea HTML, LaTeX, RTF, etc. Para cada *topic* de los mencionados anteriormente se utiliza la configuración ya establecida por el programa más alguna específica que deberá seleccionarse. En el caso del *topic: Project* además de la configuración establecida debe seleccionarse la opción *BUILTIN_STL_SUPPORT*.
8. Para el *topic Build* deben seleccionarse las opciones: *EXTRACT_ALL* y *EXTRACT_PRIVATE*, sin modificar la configuración ya establecida. Cabe mencionar que para el *topic Messages* la configuración se conserva tal cual se encuentra especificada.
9. Las opciones de entrada, o mejor dicho el *topic: Input* además de la configuración ya establecida debe contar con la selección de la opción *RECURSIVE*.
10. El *topic: Source Browser* debe especificar las opciones *SOURCE_BROWSER* y *INLINE_SOURCES* adicionalmente a la configuración establecida, mientras que para el *topic Index* no se tiene configuración diferente a la ya especificada.
11. Para concluir con la configuración de la sección *Expert* en los *topic HTML* y *LaTeX* que son en este caso los de interés se seleccionan las opciones *GENERATE_TREEVIEW* y *COMPACT_LATEX* respectivamente, como extra a la configuración ya establecida, mientras que los *topic* restantes no reciben ninguna modificación.
12. Finalmente, en la sección *Run* es donde se ejecuta DOXYGEN para generar la documentación. Cabe mencionar que el código de Fortran a documentar es extraído desde la dirección dada inicialmente para escanear el código fuente.

Los pasos descritos anteriormente hacen posible la documentación en DOXYGEN de fuentes tales como: *C*, *C++*, *objective-C*, *C#*, *PHP*, *Java*, *Python*, *ID*, *Fortran*, *VDHL*, *Tcl* y *D* [4].

3. DESCRIPCIÓN DEL CÓDIGO AZTHECA

La segunda etapa de la documentación está enfocada al análisis de la estructura del código AZTHECA. Como se mencionó anteriormente, el código está formado por diversos modelos que permiten predecir el comportamiento de un reactor tipo BWR, estos modelos están programados en diferentes rutinas. Todo el código está escrito en lenguaje Fortran, teniendo dos versiones: (1) la versión compatible con Windows, escrita en Digital Visual FORTRAN, (2) la versión compatible con Linux fue compilada con GFORTAN [2][3]. Para la documentación descrita en este trabajo se utiliza la versión compatible con Linux.

3.1 Estructura y descripción del código

En esta sección se describe la arquitectura del código y las principales rutinas [2], con la finalidad de conocer el mismo y determinar el orden de documentación. El código está estructurado por un módulo (programa) principal y 31 sub-módulos (subrutinas). El programa principal tiene por nombre AZTHECA su función es leer los datos de condiciones iniciales, condiciones de frontera, datos de entrada del modelo para la configuración del núcleo y para ejecutar las rutinas del modelo. Adicionalmente el programa permite crear archivos de escritura, almacenar información en archivos en cualquier condición de simulación (FOTO) y modificar las condiciones de entrada como función del tiempo. Las principales rutinas del código se muestran en la Figura 1 y se describen posteriormente.

- AZTHECA: Programa principal del código AZTHECA. Mediante este programa se realiza la interacción entre el usuario y el código, permite obtener fotografías, archivos de graficación, resultados y modificar variables.
- REACTOR_GEOMETRY: Contiene los datos de entrada sobre las dimensiones de la geometría del reactor.
- ALLOC_ALL_TH: Se le asignan las dimensiones a las matrices que dependen del número de nodos.
- READ_INPUT: Lee los datos del archivo input.
- POINT_KINETICS_INI: Contiene los datos iniciales para resolver el modelo de la cinética neutrónica.
- FOPROP: Calcula las propiedades termodinámicas.
- MMDUMY: Simula el flujo de vapor y el flujo de agua de alimentación.
- KINETI: Contiene el modelo puntual de la cinética neutrónica.
- PRDOMO: Calcula la presión en el domo y el nivel de la vasija.
- VALV: Calcula la derivada del cambio de la apertura de las válvulas con respecto al tiempo.

- FODRIV: Calcula la longitud de ebullición y se hace el llamado por nodo de las rutinas que contienen la transferencia de calor y la termohidráulica.
- RRS: Simula el sistema de recirculación del reactor, incluye las bombas de recirculación del reactor, las bombas jet, válvulas de corte, succión, descarga y control.
- FOEULD: Contiene el método numérico de integración de Euler.
- PRINT_PARTIAL_RESULTS: Crea e imprime los resultados de la simulación.
- CONDUCCION: Calcula el perfil radial de temperatura.
- MAHIFL, MAPCHF, FOTPFM, SEPARA: Calculan la variación de la fracción vacíos respecto al tiempo, la variación de la entalpía con respecto a tiempo, densidad de la mezcla en nodos bifásicos para diferentes condiciones de flujo.
- FOTRAF: Contiene el tráfico de correlaciones para el cálculo de los coeficientes de transferencia de calor.

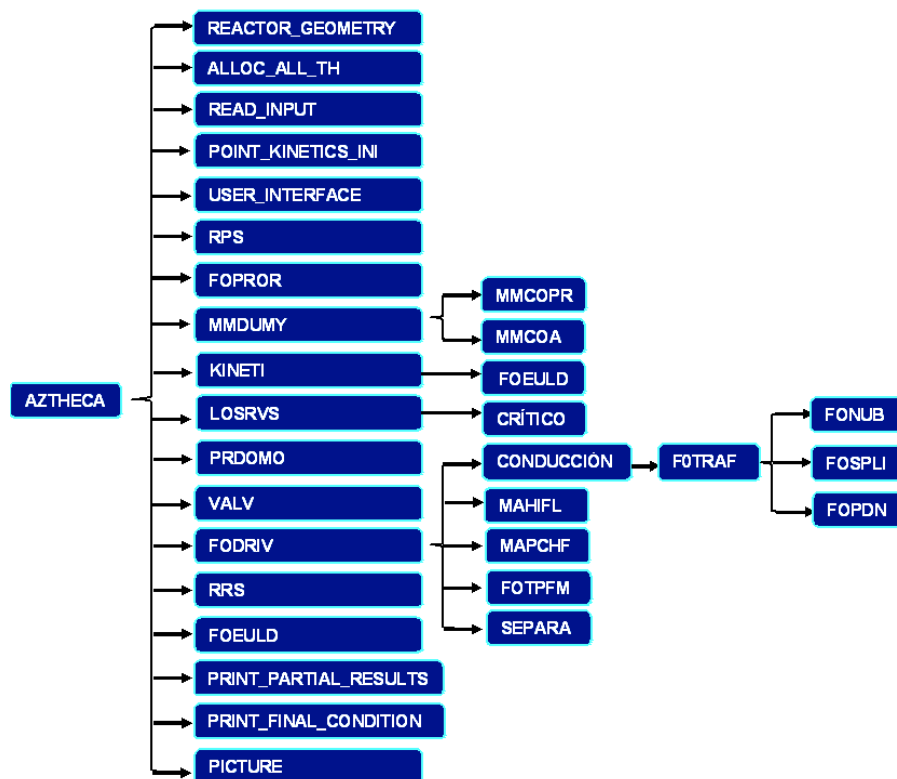


Figura 1. Diagrama de la arquitectura del código AZTHECA por llamadas.

4. RESULTADOS

Se aplicó la documentación en DOXYGEN para la rutina principal y una subrutina del Código AZTHECA siguiendo los pasos mencionados en la Sección 2.2. Para el caso de la rutina principal se obtiene un documento HTML en el que se encuentra una descripción detallada del programa y se indica la función o subprograma utilizado como se muestra en la Figura 2(a) y 2(b).



Figura 2(a). Documentación en HTML de la rutina principal del código AZTHECA.



Figura 2(b). Descripción detallada de la documentación de la rutina principal Código AZTHECA.

Se obtiene también mediante la documentación una sección en la que se muestra el código fuente utilizado, además de una lista desplegable con las siguientes secciones: Código AZTHECA, que en éste caso es la rutina principal, una lista de archivos en la que aparece definición del archivo

“aztheca.for”, el programa utilizado llamado torio y finalmente dentro de la sección miembros de los ficheros pueden encontrarse las funciones y subprogramas del código documentado como se muestra en la Figura 3.



Figura 3. Menú desplegable de la documentación de la rutina principal del Código AZTHECA.

En la Figura 4 se muestra la documentación en HTML de una subrutina, en este caso es de la Subrutina “Crítico”, en la cual se obtiene una referencia del archivo en donde se muestran los parámetros utilizados, la descripción de las variables y comentarios dentro del código fuente. Como se observa en la Figura 4, con la herramienta es posible definir el tipo de comandos que integran cada sección, diferenciando de programas, subrutinas y funciones.

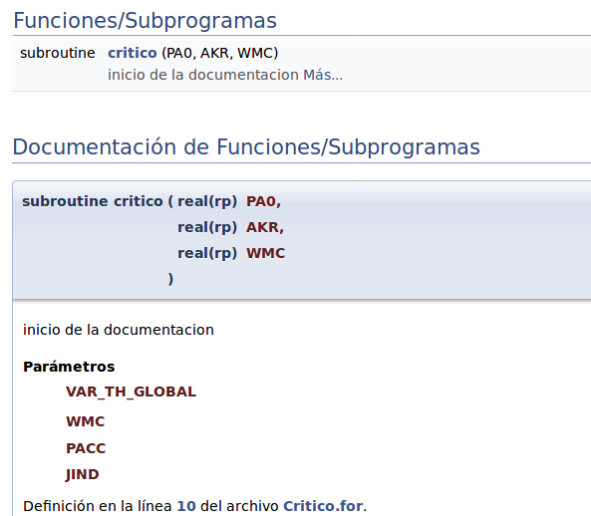


Figura 4. Descripción detallada de una de una de las subrutinas del código AZTHECA.

Dentro de la documentación en HTML es posible visualizar, sin modificar el código fuente, en la Figura 5 se presenta el ejemplo de código de la subrutina “crítico”.

```
Critico.for  
  
Ir a la documentación de este archivo.  
  
1  
3  
4  
5  
6  
7  
8  
12  
13  
14  
15  
16  
17  
18  
21  
22  
23  
24  
25  
26  
27  
28  
31  
32  
33  
34  
35  
36  
37  
  
SUBROUTINE critico (PA0, AKR, WMC)  
  
!! variable use  
USE var_th_global  
  
IMPLICIT NONE  
INTEGER :: jind, indice  
real(rp) :: alfa_l, bterm1, bterm2, pac, rogc  
real(rp) :: wmc, vc, pa0, akr, sqp, dsqdp, dnsatv  
real(rp) :: pacc, drogdpc, rogcc  
  
alfa_l = 1.0d0  
bterm1=44851601.82d0+alfa_l*(-173870767.61d0+  
alfa_l*(212037335.94d0-alfa_l*83185742.62d0))  
bterm2=-5.682256d0+alfa_l*(24.606936d0+  
alfa_l*(-29.542277d0+alfa_l*11.241647d0))  
pac = bterm1+bterm2*pa0  
  
pacc = pac  
jind= indice(pacc,sqp,dsqdp)  
rogcc = dnsatv(jind,pacc,drogdpc)  
vc = 1.0d0/drogdpc**0.5d0  
wmc = rogcc*akr*vc  
RETURN  
END SUBROUTINE critico
```

Figura 5. Código fuente de la subrutina “crítico” del Código AZTHECA.

5. CONCLUSIONES

El uso de la herramienta DOXYGEN permite reconocer la importancia y utilidad que tiene la documentación de las rutinas y subrutinas del código AZTHECA, ya que permite tener acceso a información específica del programa de forma organizada y detallada. Hasta el momento ha sido posible la documentación parcial del código, sin embargo, el proyecto sigue su curso para hacer posible la documentación de las subrutinas restantes, y del proyecto global de la plataforma AZTHECA.

La elaboración del manual para la herramienta DOXYGEN ha tenido grandes avances, ya que gracias al mismo ha sido posible llevar a cabo la auto-documentación. Cabe mencionar que el manual está siendo elaborado con la finalidad de que sea utilizado para documentar los códigos restantes de la plataforma AZTLAN por cualquier tipo de usuario, pues el mismo manual está siendo usado e implementado por personas externas a los desarrolladores del código.

AGRADECIMIENTOS

Los autores agradecen el apoyo económico recibido del proyecto estratégico No. 212602 del Fondo Sectorial de Sustentabilidad Energética, CONACyT-SENER.

REFERENCIAS

1. Gómez, A., Espinosa, G., François, J., Martín, C., and del Valle, E., “AZTLAN Platform: Plataforma Mexicana para el Análisis y Diseño de Reactores Nucleares”, *XXV Congreso Anual de la Sociedad Nuclear Mexicana.*, Boca del Río, Veracruz, México, del 31 de Agosto al 4 de Septiembre de 2014,
2. Quezada, S., Espinosa, G., Centeno, J., and Sánchez, H., “Código AZTHECA”, *XXVIII Congreso Anual de la Sociedad Nuclear Mexicana - 2017 LAS/ANS Symposium "New Technologies for a Nuclear Power Expansion Program"*, Ciudad de México, Junio 18 – 21, 2017,
3. Quezada, S., Centeno, J., and Espinosa, G. *AZTHECA 1.0 AZTLAN CÓDIGO TERMOHIDRÁULICO Manual de usuario.* Ciudad de México, México (2016).
4. Jiménez, Z., Meza, A., Sánchez, M., Pérez, A., and Espinosa, G. *Manual DOXYGEN.* Actividades de Servicio Social Universidad Autónoma Metropolitana, Ciudad de México, México (2018).
5. Charbonnier, A., Combet, C., and Maurin, D. *Computer Physics Communications*, vol. **183**, no 3, p. 656-668 (2012).
6. Rudberg, E., Rubensson, E., Salek, P., and Kruchinina, A. “Ergo: An open-source program for linear-scaling electronic structure calculations” *SoftwareX*, vol. **7**, p. 107-111 (2018).
7. Marulli, F., Veropalumbo, A., and Moresco, M. “CosmoBolognaLib: C++ libraries for cosmological calculations” *Astronomy and Computing*, vol. **14**, p. 35-42 (2016).
8. “Dpto. Informática e Ingeniería de Sistemas Universidad de Zaragoza”, http://webdiis.unizar.es/assignaturas/LGA/material_2004_2005/Intro_Flex_Bison.pdf. (2017).
9. “Introduction to Libiconv” <https://www.gnu.org/software/libiconv/> (2017)
10. Framework Web, http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf (2017)