

Document downloaded from:

<http://hdl.handle.net/10251/71941>

This paper must be cited as:

Bernal García, Á.; Miró Herrero, R.; Verdú Martín, GJ. (2014). Desarrollo de un código neutrónico de difusión 2D y 3D estacionario por el Método de Volúmenes Finitos. Sociedad Nuclear Española. <http://hdl.handle.net/10251/71941>.



The final publication is available at

Copyright Sociedad Nuclear Española

Additional Information

Desarrollo de un código neutrónico de difusión 2D y 3D estacionario por el Método de Volúmenes Finitos

Álvaro Bernal¹, Rafael Miró¹ y Gumersindo Verdú¹

¹Instituto de Seguridad Industrial, Radiofísica y Medioambiental
Universitat Politècnica de València, Valencia, España.
abernal@iqn.upv.es, rmiro@iqn.upv.es, gverdu@iqn.upv.es

Resumen – El objetivo del trabajo es el desarrollo de un código neutrónico modal de difusión en 2D y 3D estacionario utilizando el Método de Volúmenes Finitos, a partir de códigos libres y que se pueda aplicar a reactores de cualquier geometría. Actualmente, los métodos numéricos que más se utilizan en los códigos de difusión proporcionan buenos resultados en malla estructurada, pero su aplicación a malla no estructurada no es fácil y puede presentar problemas de convergencia y estabilidad de la solución. Respecto a la malla no estructurada, su uso está justificado por su fácil adaptación a geometrías complejas y por el desarrollo de códigos acoplados termohidráulico-neutrónico, así como el desarrollo de códigos fluidodinámicos (CFD) que incentivan el desarrollo de un código neutrónico que tenga la misma malla que la de los códigos fluidodinámicos, que en general suele ser no estructurada. Por otra parte, el refinamiento de la malla y su adaptación a geometrías complejas es otro estímulo de cara a conocer con más detalle lo que ocurre en el núcleo del reactor. Finalmente, el código se ha validado con la simulación de un reactor homogéneo y otro heterogéneo para 2D y 3D.

1. INTRODUCCIÓN

La tesina fin de máster tiene como objetivo el desarrollo de nuevos algoritmos numéricos mediante volúmenes finitos que sean capaces de determinar la distribución neutrónica, el estado de Criticidad y la evolución temporal de los reactores nucleares. En concreto, la finalidad de dichos algoritmos es su aplicación al Análisis Determinista de Seguridad Nuclear para realizar cálculos más detallados a nivel geométrico.

Para calcular la distribución neutrónica en reactores nucleares comerciales, se utiliza la teoría de la difusión que consta de ecuaciones integro-diferenciales y que no se pueden resolver para reactores nucleares comerciales de forma analítica, y por tanto se utilizan métodos numéricos para su resolución. Existen diversos métodos, pero el Método de los Volúmenes Finitos (MVF) no es uno de los de mayor aplicación en el campo de la neutrónica. No obstante, este método es uno de los más utilizados en el campo de la Termohidráulica y de la Fluidodinámica Computacional (CFD), que es otro de los aspectos más importantes de los reactores nucleares y su cálculo está ligado al cálculo neutrónico y viceversa. Además, el Método de Volúmenes Finitos es adecuado para resolver mallas no estructuradas, lo cual que permite modelar cualquier tipo de geometría.

Se ha desarrollado un análisis modal (cálculo de varios autovalores y autovectores) ya que permite calcular la distribución neutrónica de diversos modos armónicos, de utilidad en el análisis de inestabilidades en reactores BWR, que está comenzando a ser un aspecto a tener en cuenta para la normativa estadounidense para la legislación de reactores nucleares, que es la que tiene como norma España. Además, también es de utilidad para

el desarrollo de una cinética modal que permite realizar cálculos transitorios con menos actualizaciones del flujo neutrónico espacial respecto a otras cinéticas, que se traduce en la posibilidad de reducir el tiempo de computación del cálculo transitorio. No obstante, esta tesina sólo ha podido abarcar el cálculo estacionario.

En este trabajo, se ha desarrollado un código neutrónico de difusión en 2D y en 3D a partir de códigos libres que es capaz de resolver el estado estacionario de reactores de cualquier geometría y el cálculo de más de un autovalor, discretizando las ecuaciones con el MVF.

2. DESARROLLO DEL CÓDIGO

El código se ha desarrollado a partir de los siguientes códigos: *Gmsh* [1], *Arb* [2] y *SLEPc* [3]. Además, se han utilizado también los siguientes códigos: *ParaView*, *VALKIN* [4] y *PARCS* [5].

El desarrollo del código se ha dividido en 4 secciones: Pre-Procesado, Discretización de las ecuaciones, Solucionador y Post-procesado. El Pre-procesado está basado en el código *Gmsh*, la discretización de las ecuaciones en el código *Arb*, el solucionador en la librería *SLEPc* y el post-procesado en la aplicación *ParaView*.

2.1. Descripción de los códigos utilizados

Gmsh es un generador de malla tridimensional de elementos finitos de cualquier geometría definida en formato paramétrico, es decir, con un programa CAD; la definición de la geometría se puede hacer en *Gmsh* o en otro CAD externo.

En cuanto a *Arb*, éste discretiza ecuaciones diferenciales en derivadas parciales definidas por el usuario en mallas no estructuradas usando el Método de Volúmenes Finitos y las resuelve utilizando el método de Newton-Raphson. La malla no estructurada debe estar en formato “.msh”, que puede obtenerse usando el código *Gmsh*. Las ecuaciones se introducen en un archivo denominado “equations.in” y que contendrá la definición de variables, incógnitas y ecuaciones en formato del tipo matemático.

SLEPc es un paquete de programas para resolver problemas de autovalores para matrices grandes y dispersas, tanto para el problema estándar como el problema generalizado.

Respecto a *ParaView*, es una aplicación de post-procesado que se utiliza para la visualización y análisis de datos científicos. Destaca por admitir una gran cantidad de formatos de archivo y su ejecución en paralelo lo que permite una rápida visualización de ficheros de gran tamaño, que sería imposible visualizar y manipular con un único núcleo.

En cuanto a *VALKIN*, se trata de un código de difusión neutrónica 3D que resuelve tanto el estado estacionario como el transitorio. Utiliza el Método de Colocación Nodal para discretizar las ecuaciones y el Método Modal para el análisis del transitorio. El Método Modal se basa en el cálculo de los autovalores dominantes y sus respectivos autovectores en el estado estacionario, para su aplicación en el cálculo transitorio que se realiza a

continuación. En particular, se ha escogido este código para realizar la validación del código desarrollado en este trabajo.

Finalmente, *PARCS* es un código de difusión neutrónica 3D que resuelve tanto el estado estacionario como el transitorio, pero sólo permite el cálculo de 1 autovalor. *PARCS* se ha utilizado para validación de diversos códigos neutrónicos y así se ha hecho también en este trabajo.

2.2. Pre-procesado

El Pre-procesado se ha realizado utilizando el código *Gmsh* sin ninguna modificación. Este código permite la definición de la geometría tanto por comandos como por interfaz gráfica, pues incorpora una interfaz tipo CAD. Posee diversas funciones para la definición de la geometría, pero tiene como inconveniente que se ejecuta con un solo núcleo, que supone una ejecución lenta para archivos de gran tamaño. En este trabajo, se han desarrollado manuales para un correcto modelado de la geometría en *Gmsh* para obtener tanto mallas estructuradas como no estructuradas en 2D y 3D.

2.3. Discretización de las ecuaciones

En referencia a la Discretización de las ecuaciones, primero se plantean las ecuaciones a discretizar, luego se formulan las ecuaciones en el lenguaje del código *Arb*, después se modifica el código para que cumpla todos los requisitos. Estos requisitos son: que sirva para un número genérico de materiales, que las condiciones de contorno sean genéricas y que se cumpla la condición de corriente en la superficie de separación de celdas, sin tener que recompilar el código cada vez que cambie el problema.

Las ecuaciones que se han utilizado son las de la difusión neutrónica en estado estacionario para 2 grupos de energía, y por tanto, hay que formular un problema de autovalores. *Arb* no permite el modelado de problemas de autovalores, por lo que hubo que reformular las ecuaciones para obtener las matrices del problema de autovalores.

Otra limitación de *Arb* consiste en que las ecuaciones no se pueden cambiar sin recompilar, lo que supone que no se pueden definir las ecuaciones para un número arbitrario de materiales, ni se pueden definir condiciones de contorno genéricas. Para solucionarlo, se cambió el código *Arb*, para que leyera tanto el número de materiales como las condiciones de contorno de un fichero y formulara las ecuaciones de forma genérica.

Además, en *Arb* el número de ecuaciones y el número de incógnitas tienen que ser iguales, lo cual dificulta el cumplimiento de la condición de corriente en las intercarras. Las incógnitas del problema son el flujo rápido y térmico medio en cada celda y en las fronteras y se tienen 2 ecuaciones de la difusión por celda y condiciones de contorno tanto para el flujo térmico como para el rápido. La condición de corriente en las intercarras supone un aumento del número de ecuaciones y su cálculo debe realizarse a partir del gradiente del flujo en cada celda. Sin embargo, *Arb* realiza el cálculo del gradiente en la intercarras de 2 celdas, a partir del valor medio de las celdas que rodean a dicha cara, y por tanto es independiente de la celda, ya que sólo depende de la intercarras y de todas las celdas que rodean a ésta. Así, el gradiente en la intercarras de la celda 1 será igual al gradiente de la

intercara de la celda 2, y si se tienen materiales con distintos coeficientes de difusión no se cumplirá la condición de corriente. Para solucionarlo, se ha forzado en la programación del código que la corriente en ambas celdas sea la misma utilizando el mismo coeficiente de difusión, que se ha llamado “coeficiente de difusión global”, y se han desarrollado 4 modelos de dicho coeficiente.

2.4. Solucionador

En cuanto al solucionador, se necesita un algoritmo que resuelva el problema de autovalores. Dado que las matrices resultantes son de gran tamaño y de naturaleza dispersa, se ha utilizado *SLEPc*.

SLEPc es una librería de propósito general que se puede aplicar a cualquier problema físico siempre que se le pasen las matrices del problema así como los parámetros que definen el problema.

En este caso, se trata de un problema generalizado de autovalores donde se buscan los mayores valores de la constante de multiplicación (k). Además, contiene diversos métodos para la resolución del problema de autovalores, pero en este caso se ha utilizado el método de Krylov-Schur. No obstante, las matrices que se obtienen de la discretización con *Arb* no pueden resolverse con *SLEPc* sin cálculos adicionales. En particular, se tuvo que utilizar una transformación espectral, el preconditionamiento de las matrices (preconditionamiento LU) y la utilización de un solucionador directo (se utilizó MUMPS como solucionador directo).

2.5. Post-procesado

Respecto al Post-procesado, se calcula la potencia a partir de los flujos calculados en el problema de autovalores y se normalizan estas magnitudes para que la potencia media sea igual a la unidad.

Además, se ha utilizado un formato del archivo de salida diferente al que realizaba *Arb*, ya que el formato de éste se visualiza con *Gmsh* que no se puede ejecutar en paralelo, y por tanto la visualización de los resultados de mallas de muchas celdas es imposible. En su lugar, se utiliza el formato VTK que se puede visualizar con el código *ParaView* que es ampliamente utilizado en el análisis de datos científicos y en particular en el post-procesado de códigos CFD, y que permite la ejecución en paralelo.

3. VALIDACIÓN DEL CÓDIGO

Para validar el código se han utilizado 4 reactores: uno homogéneo y otro heterogéneo para cada caso 2D y 3D. Los homogéneos se utilizan para validar la discretización de las ecuaciones sin tener en cuenta la condición de corriente en las intercara, ya que al tener el mismo coeficiente de difusión en todas las celdas, se cumplirá dicha condición. Por otra parte, los heterogéneos se utilizan para validar las aproximaciones del coeficiente de difusión global desarrolladas para forzar el cumplimiento de la condición de la corriente. En todos los casos, se realiza un cálculo de 5 autovalores.

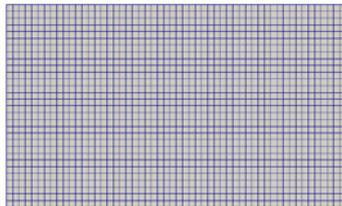
Además, los homogéneos se simulan con condiciones de contorno de flujo nulas y los heterogéneos con condiciones de contorno nulas en ciertas fronteras y condiciones de flujo reflectivo en las otras para validar todas las prestaciones del código.

3.1. Reactor homogéneo 2D

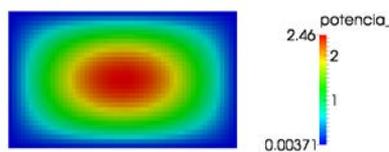
El reactor homogéneo 2D consiste en un rectángulo compuesto de un único material. Este reactor tiene solución analítica y es la que se utiliza como referencia. Además, se han simulado distintos tamaños y tipos de malla. En concreto se han simulado 7 mallas estructuradas de tamaños en cm: 10, 5, 2, 1, 0.5, 0.2 y 1; y también se han simulado 6 mallas no estructuradas de tamaños en cm: 10, 5, 2, 1, 0.5 y 0.2.

Se comprueba en los resultados una disminución del error de los autovalores y de las potencias para mallas más finas, así como el aumento del tiempo de computación. En resumen, se tienen errores y tiempos de computación aceptables para la malla de tamaño 2 cm. En las siguientes figuras se muestra la potencia para los primeros 5 autovalores, y las mallas estructurada y no estructurada de 2 cm.

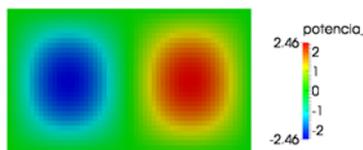
Malla estructurada



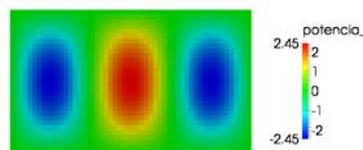
Malla



Modo 1

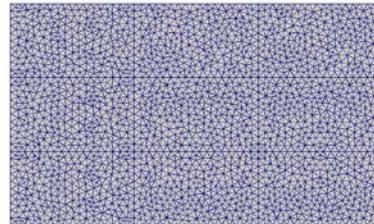


Modo 2

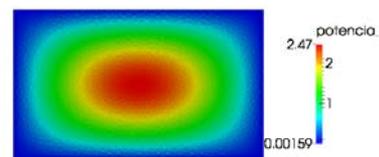


Modo 3

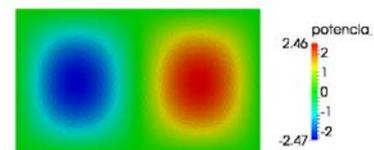
Malla no estructurada



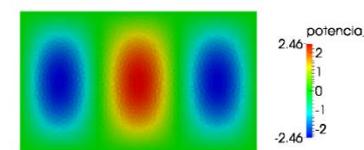
Malla



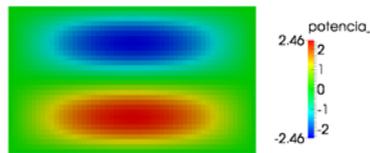
Modo 1



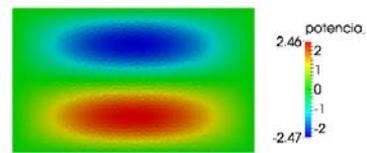
Modo 2



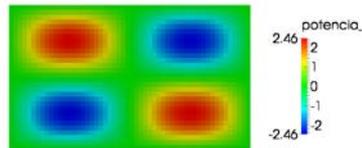
Modo 3



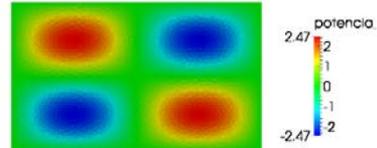
Modo 4



Modo 4



Modo 5



Modo 5

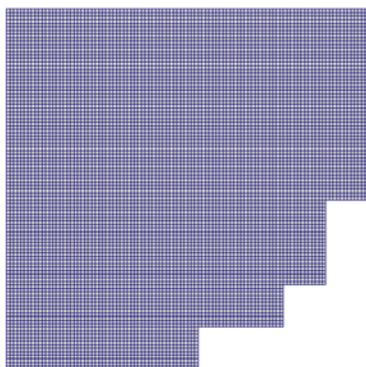
3.2. Reactor Biblis

El reactor heterogéneo 2D considerado para la validación del código es el reactor Biblis, que se ha utilizado en diversos Benchmark. Se han utilizado varios tamaños de malla tanto para la malla estructurada como no estructurada. En particular se han utilizado los siguientes tamaños en cm: 10, 5, 2 y 1.

Para validar los resultados, se han utilizado los resultados del código nodal analítico PANIC, que se obtuvieron de un artículo [6]. En dicho artículo sólo están los resultados correspondientes al primer autovalor. Por otra parte, se han utilizado las 4 aproximaciones del coeficiente de difusión global para forzar el cumplimiento de la condición de corriente en las intercaras.

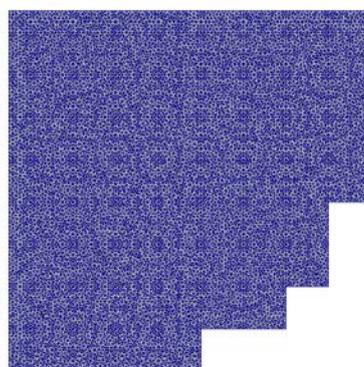
Con todas estas simulaciones se ha desarrollado un análisis de sensibilidad que muestra qué aproximación del coeficiente de difusión global tiene menores errores y que tamaño de malla presenta la mejor relación error-tiempo de computación. A continuación se muestra la potencia para los primeros 5 autovalores, y las mallas estructurada y no estructurada de 2 cm para el reactor Biblis.

Malla estructurada

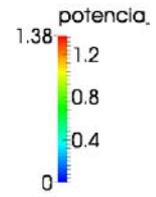
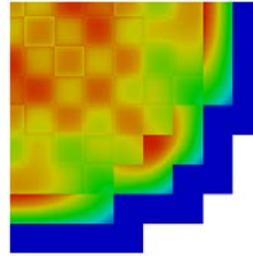
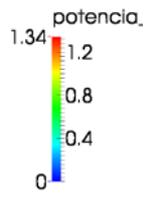
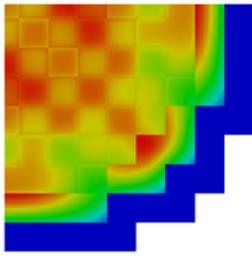


Malla

Malla no estructurada

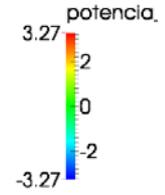
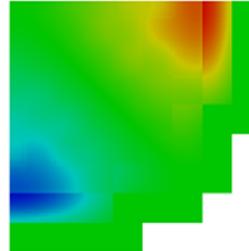
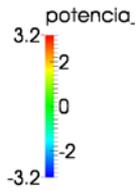
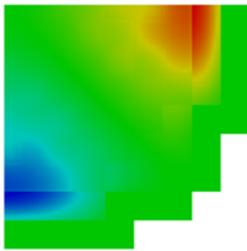


Malla



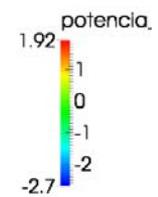
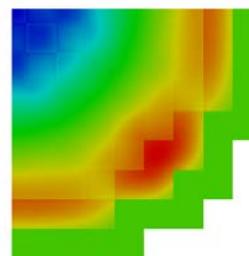
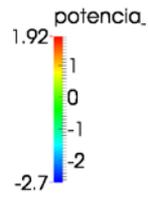
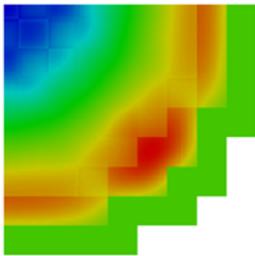
Modo 1

Modo 1



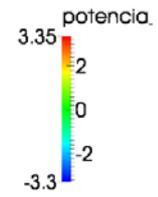
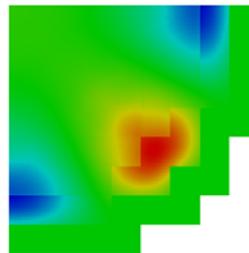
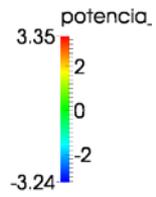
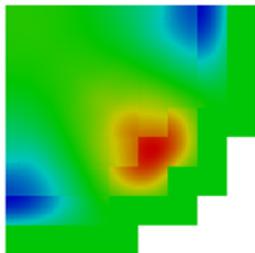
Modo 2

Modo 2



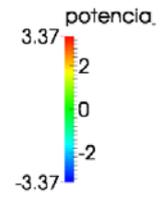
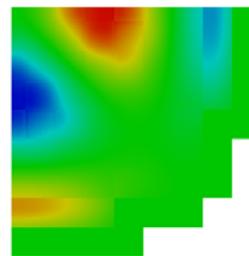
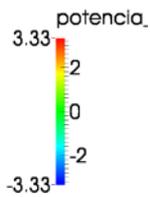
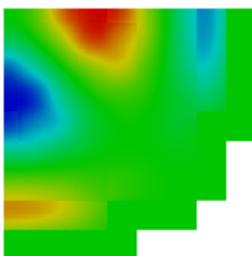
Modo 3

Modo 3



Modo 4

Modo 4



Modo 5

Modo 5

3.3. Reactor homogéneo 3D

En cuanto al reactor homogéneo 3D, se utiliza un reactor con forma de paralelepípedo compuesto del mismo material que el caso 2D. Este reactor también tiene solución analítica y será la solución de referencia. Sin embargo, sólo se han simulado casos de mallas estructuradas porque contienen menor número de celdas que las mallas no estructuradas, ya que las mallas no estructuradas no tenían tiempos de computación viables. En concreto, se han utilizado 3 tamaños de malla en cm: 10, 5 y 2. Los tiempos de computación de estos tamaños son del orden de segundos, minutos y horas respectivamente. Por consiguiente, los tiempos de computación para el caso de 2 cm no son prácticos. No obstante, se tienen tiempos de computación y errores aceptables con la malla de 5 cm.

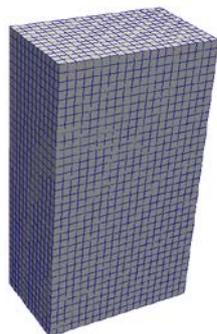
3.4. Reactor Langenbuch

Finalmente, el reactor heterogéneo 3D utilizado es el reactor de Langenbuch. Del mismo modo que el reactor homogéneo 3D, sólo se han utilizado los tamaños de malla en cm: 10, 5 y 2; para malla estructurada por cuestión de tiempos de computación.

Para validar los resultados, se ha simulado el reactor de Langenbuch con los códigos PARCS y VALKIN. Además, también se han simulado las 4 aproximaciones del coeficiente de difusión global para verificar cuál funcionaba mejor. Los tiempos de computación son del mismo orden que el caso homogéneo y los errores un poco más altos, pero se tienen errores aceptables con la malla de 5 cm.

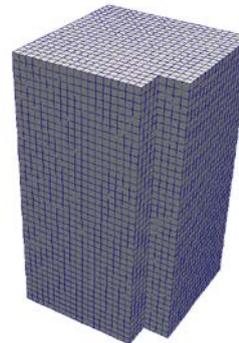
A continuación se expone los resultados de la potencia de los primeros 5 autovalores para los reactores 3D, para la malla estructurada de 5 cm.

Reactor homogéneo 3D

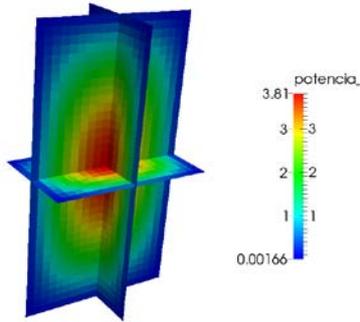


Malla

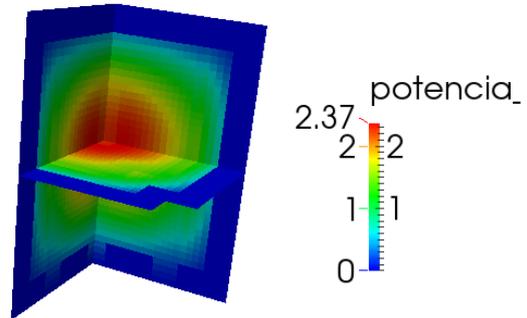
Reactor de Langenbuch



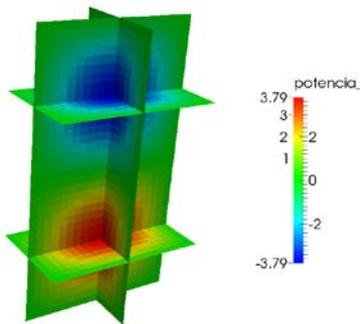
Malla



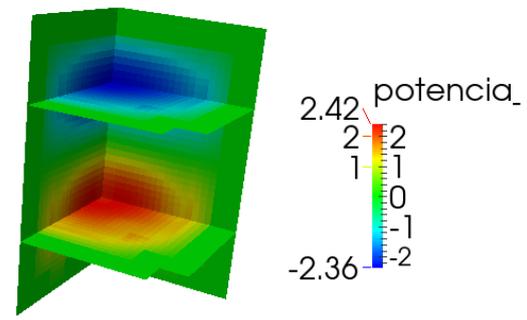
Modo 1



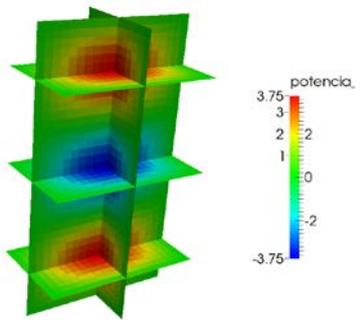
Modo 1



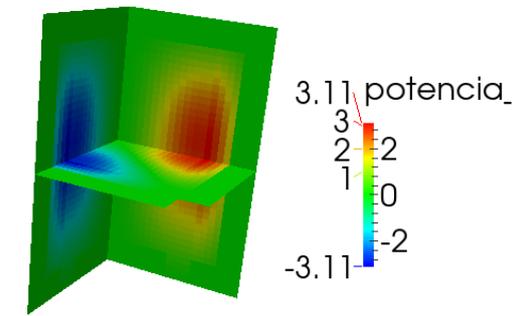
Modo 2



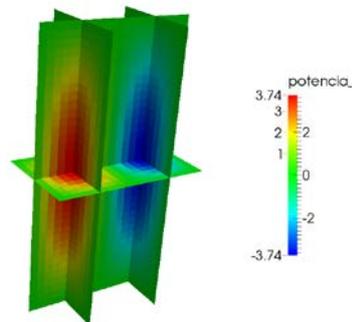
Modo 2



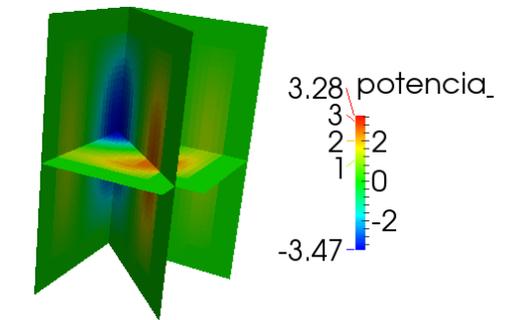
Modo 3



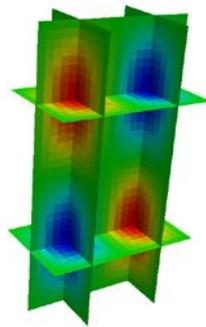
Modo 3



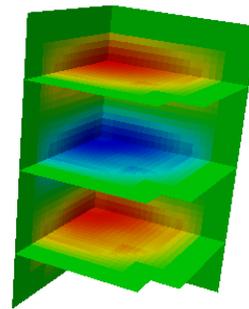
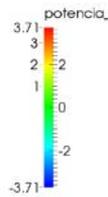
Modo 4



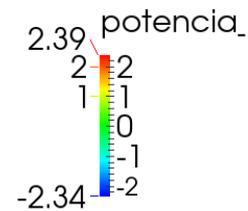
Modo 4



Modo 5



Modo 5



4. CONCLUSIONES

En conclusión, el código desarrollado es ideal para casos 2D ya que se obtienen resultados precisos y con un coste computacional pequeño, del orden de segundos. No obstante, para casos 3D el cálculo tiene un coste computacional grande y requiere elevados tiempos de computación para mallas de tamaño pequeño. Sin embargo, todavía no se ha optimizado completamente el código ni se ha implementado la ejecución en paralelo que implicaría la aceleración del cálculo. Por otra parte, se comprueba que la discretización de las ecuaciones con el Método de Volúmenes Finitos produce buenos resultados como se comprueba en los casos de los reactores estudiados. Además, las aproximaciones de la corriente neutrónica utilizadas posibilitan el cumplimiento de la condición de corriente que no se cumpliría si no se implementara en el código. De hecho, esto se ha experimentado en el desarrollo del código pero no se ha expuesto en el trabajo porque los errores relativos de los resultados estaban alrededor del 15-20 % para la potencia del primer autovalor. En resumen, se tienen errores en torno al 0.5 % para casos 2D y de 2% para 3D, aunque depende de la malla.

Cabe mencionar que los resultados de este trabajo se expusieron en el congreso *Mathematical Modelling in Engineering & Human Behaviour 2013*, siendo el título de la presentación *Resolution of the Generalized Eigenvalue Problem in the Neutron Diffusion Equation Discretized by the Finite Volume Method*. Además, también se publicaron en la revista *Abstract and Applied Analysis*, con DOI: [10.1155/2014/913043](https://doi.org/10.1155/2014/913043). Por otra parte, se han mejorado los tiempos de computación de este método cambiando la resolución del problema de autovalores y este trabajo ha sido aceptado en el congreso *PHYSOR 2014*, siendo el título de la presentación *Generalized and Standard Multigroup Neutron Diffusion Equation Eigenvalue Problem with the Finite Volume Method*.

Respecto al trabajo futuro, se debe optimizar el código, ya que se han incluido funciones y subrutinas del código *Arb* que este código que se ha desarrollado no utiliza. Además, la próxima mejora que incluirá el código será la lectura de secciones eficaces a partir de archivos *NEMTAB*, como lo hace el código *PARCS* y *VALKIN*, y tiene como objetivo el cálculo de reactores reales, ya que las secciones eficaces de estos se tienen en este formato. Por otra parte, se paralelizará el código, tanto en el análisis de la geometría como en la discretización de las ecuaciones y en la resolución del problema de autovalores. También se desarrollará una subrutina para resolver el estado transitorio, aprovechando el cálculo de varios modos, es decir, de autovalores y autovectores. En

cuanto a la aproximación de las corrientes, se están estudiando otras alternativas y ésta será otra de las mejoras con mayor prioridad. Además, se desea implementar técnicas alternativas en la discretización de las ecuaciones con el MVF que contiene *Arb*, para estimar el valor de las variables en las caras a partir de los valores de las variables en las celdas. En concreto, se pretende aplicar esquemas de alto orden. Por último, se acoplaría este código a códigos CFDs para realizar el cálculo neutrónico-termohidráulico acoplado, ya que puede utilizar el mismo mallado que estos códigos.

REFERENCIAS

- [1] C. Geuzaine and J.-F. Remacle. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009.
- [2] Dalton J. E. Harvie. *An implicit finite volume method for arbitrary transport equations*. ANZIAM J. (CTAC2010), 52:C1126–C1145, March 2012.
- [3] V. Hernandez, J. E. Roman, and V. Vidal. *SLEPc: Scalable Library for Eigenvalue Problem Computations*. Lect. Notes Comp. Sci., vol. 2565, pages 377-391. Springer, 2003.
- [4] R. Miró (2002): *Métodos Modales para el Estudio de Inestabilidades en Reactores Nucleares BWR*. Tesis Doctoral. Universitat Politècnica de València (España).
- [5] Downar, T. J., Lee, D., Xu, Y., Kozlowski, T., *PARCS v2.6. U.S. NRC Core Neutronics Simulator. Theory Manual*. RES/U.S.NRC. Rockville, Md (2004).
- [6] E.Z. Müller, Z.J. Weiss, *Benchmarking with the multigroup diffusion high-order response matrix method*, Annals of Nuclear Energy, Volume 18, Issue 9, 1991, Pages 535-544, ISSN 0306-4549.