



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



*im*<sup>2</sup>  
Instituto Universitario  
de Matemática Multidisciplinar

Manual de usuario

---

# FEMFFUSION

## Un código de elementos finitos para el modelado de sistemas nucleares

---

A. Vidal-Ferràndiz, A. Carreño, Y. Fontenla, D. Ginestar y G. Verdú

2 de mayo de 2023

Versión 0.1

# Índice

Introducción al código	3
1 Introducción al código	3
Instalación del programa	3
2 Instalación del programa	3
Comandos para la ejecución del programa	4
3 Ejecución del programa	4
Cálculo del problema estacionario	4
4 Cálculo del problema estacionario	4
Representación de la geometría	5
5 Representación de la geometría	5
Archivos de entrada	5
6 Archivos de entrada	5
Archivo principal o archivo <b>' .prm '</b> .....	6
6.1 Archivo principal o archivo <b>' .prm '</b> .....	6
Archivos de salida	9
7 Archivos de salida	9
Ejemplos de problema	10
8 Ejemplos	10
9 Proyectos Relacionados	14
Bibliografía	15

## 1. Introducción al código

FEMFFUSION es un código neutrónico escrito en lenguaje de programación C++ y resuelve la ecuación multigrupo del transporte neutrónico usando la aproximación de la difusión y la aproximación de esféricos armónicos simplificados, también conocida como las ecuaciones  $SP_N$  [1]. El código utiliza para la discretización espacial de las ecuaciones elementos finitos continuos de Galerkin para ser capaz de resolver todo tipo de geometrías y dimensiones del problema (1D, 2D y 3D) [2].

El código está construido sobre la librería de código abierto DEAL.II que le suministra, además del soporte, avances en el método de elementos finitos [3]. Esta librería incluye muchas técnicas de desarrollo propio para la resolución de ecuaciones en derivadas parciales, problemas de álgebra lineal y estrategias de ciencia computacional, pero además es compatible y posee interfaz con otras librerías de código abierto punteras en estos campos. En nuestro caso, además del DEAL.II, nos basamos de las estructuras de PETSC que es una herramienta con técnicas muy sofisticadas para la resolución en paralelo de problemas de álgebra lineal [4]. Por otra parte, utilizamos técnicas desarrolladas por la librería SLEPC [5] junto con otras metodologías propias para la resolución de problemas de autovalores. Para la integración de problemas que dependen del tiempo, incluimos la librería SUNDIALS [6].

Por último, destacar que el código permite el uso de técnicas *matrix-free*, que no necesitan montar ni guardar las matrices involucradas en el sistema, para mantener demandas de la memoria computacional en valores razonables y aumentar la velocidad de cálculo [7].

## 2. Instalación del programa

FEMFFUSION funciona con sistemas GNU/Linux, como la mayoría de los códigos de aplicación en álgebra numérica. Para distribuciones basadas en Debian se provee de un script de instalación. Con otro tipo de distribuciones, la instalación puede realizarse utilizando los comandos análogos.

En primer lugar, se deben instalar algunas herramientas y librerías numéricas. Esto lo podemos hacer escribiendo en la terminal:

```
sudo apt-get install make g++ git cmake
sudo apt-get install petsc-dev slepc-dev libdeal.ii-dev
sudo apt-get install gmsh paraview
```

Tras estos pasos, crea o accede al lugar donde quieres tener el código de FEMFFUSION. Una vez aquí, clonamos el repositorio del código y se descargan, compilan y prueban las librerías auxiliares. Estos últimos pasos, que requieren conexión a Internet, pueden realizarse introduciendo en la terminal:

```
git clone https://Zonni@bitbucket.org/Zonni/femffusion.git
cd femffusion
cmake .
make release
make -j4
```

La última instrucción tardará bastante tiempo, ya que debe compilar las librerías PETSC, SLEPC y DEAL.II (en torno a 30 minutos).

Después de una compilación satisfactoria del código, se recomienda ejecutar los ejemplos de aplicación de FEMFFUSION para comprobar que el código funciona correctamente. Estos ejemplos están descritos en la Sección 8. Podemos ejecutarlos con la instrucción:

```
./run_examples.sh
```

### 3. Ejecución del programa

Una vez instalado FEMFFUSION ya podemos ejecutar los distintos problemas. Para ello, solo necesitamos conocer el nombre del archivo principal (o `archivo.prm`) que contiene todos los parámetros necesarios para resolver el problema e introducir en la terminal la línea de comandos:

```
./femffusion.exe -f carpeta/archivo.prm
```

donde `carpeta` indica la ruta en la cual está ubicado el archivo `.prm`, los dos directorios posibles con estos archivos son `test` o `Examples`. Posibles ejemplos de ruta se muestran en el archivo `run_examples.sh`. Tras el nombre del archivo a ejecutar, es posible introducir otros comandos para la ejecución del código, propias de FEMFFUSION (Tabla 1) o procedentes de las librerías. En este caso, el comando `-f` (o `--file`) indica que a continuación está el nombre del fichero. Otras opciones muy útiles en la línea de comandos son:

- `-v` o `--verbose`. Muestra más detalles de la ejecución del código en la salida por pantalla.
- `-s` o `--silent`. No muestra nada por pantalla.
- `-t` o `--test`. Ejecuta una serie de casos tests.
- `-p` o `--performance`. Evalúa el funcionamiento de los productos en *matrix-free* y los resolvers de autovalores en algunos casos test.

### 4. Cálculo del problema estacionario

FEMFFUSION puede ser visto como una especialización de DEAL.II para la aproximación de la ecuación del transporte neutrónico representada por la ecuación multigrupo de difusión neutrónica o las ecuaciones  $SP_N$ . Estas ecuaciones en estado estacionario tienen la forma

$$\mathcal{L}\phi = \frac{1}{k_{\text{eff}}}\mathcal{M}\phi,$$

donde  $\mathcal{L}$  es el operador diferencial del transporte,  $\mathcal{M}$  es el operador diferencial de la fuente de neutrones, el vector  $\phi$  denota el flujo de neutrones en estado estacionario y la constante  $k_{\text{eff}}$  es denominada factor de multiplicación efectivo e indica el estado de criticidad del sistema.

Tras aplicar elementos finitos de Galerkin sobre las ecuaciones diferenciales en una malla espacial concreta, se obtiene un problema algebraico de autovalores (ver más detalles en [8]). Las matrices halladas tienen una estructura a bloques, son dispersas y se construyen en formato de PETSC para poder trabajar con los algoritmos ya implementados por librería y por SLEPC. Además, estas estructuras permitirían la paralelización del código mediante PETSC. Otra posibilidad es la utilización de la ya mencionada metodología *matrix-free*, en la que solo los bloques diagonales se guardan en la memoria y el resto se usan mediante el producto matriz-vector.

Para la solución del problema de autovalores FEMFFUSION tiene implementados numerosos tipos de resolvedores además de los proporcionados por la librería SLEPC. En particular, el usuario puede utilizar el método de la potencia [1], el método de Krylov-Schur [9] (de SLEPC), el método *Generalized Davidson* [10] (de SLEPC), el método *Block Inverse-Free Preconditioned Arnoldi* (BIFPAM) [11], el método *Modified Generalized Block Newton* (MGBNM) [11] y un método híbrido basado en los dos anteriores (hybrid) [11]. Cada uno de estos métodos a su vez posee distintos tipos de inicialización y preconditionadores que optimizan los recursos computacionales.

Una vez resuelto el problema, FEMFFUSION realiza una segunda capa que enlaza la solución del problema de autovalores con las herramientas de postproceso de los resultados.

## 5. Representación de la geometría

El código está preparado para geometrías de cuatro tipos: geometrías rectangulares, geometrías hexagonales, geometrías a nivel de pin. Para cada uno de estos tipos se hace una serie de recomendaciones en cuanto a la representación de la geometría del reactor nuclear.

- **Reactores rectangulares.** El código está preparado para que, a través de una serie de parámetros, la librería DEAL.II construya la malla discreta del problema. Estos valores son: el número máximo de celdas en cada dimensión, el tamaño de cada celda en todas las dimensiones, y los valores del paralelepípedo rectangular (formado por el conjunto de celdas máximo en cada dimensión) que forma parten o no de la geometría del reactor.
- **Reactores hexagonales.** La librería DEAL.II solo permite utilizar como elementos finitos paralelepípedos, por tanto, cada elemento hexagonal se divide en tres elementos de este tipo para su representación. Para la creación de estas mallas los autores hacen uso de la librería GMSH [12, 13].
- **Reactores a nivel de pin.** La construcción de la geometría se realiza a través de la librería DEAL.II, donde, por una parte, se indica la geometría a nivel de pin y por otra, la geometría de cada uno de los pins que contienen el fuel.

## 6. Archivos de entrada

Para la ejecución de un problema FEMFFUSION necesita siempre el archivo principal ‘.prm’ y, al menos, un otro tipo de archivo que tenga las secciones eficaces (XS) de los materia-

les y su distribución en la geometría. Sin embargo, es posible que según el tipo de problema, se requieran otros archivos de entrada. Una lista de los archivos de entrada usados por FEMFFUSION son descritos a continuación:

- **archivo.prm.** Es el archivo principal y contendrá todos los parámetros que podemos introducir al código como variables de entrada para la resolución del problema y su postproceso. Estas variables se detallarán en la Sección 6.
- **archivo.xsec** o **archivo.xlm.** El primer archivo es para problemas con dos grupos de energía y el segundo para más de dos grupos energéticos. En estos archivos se encuentra toda la información relativa a los materiales del problema (distribución y secciones eficaces).
- **archivo.msh.** El código también puede leer mallas que provienen de la librería GMSH [12, 13]. Este tipo de archivo se utiliza principalmente para mallas hexagonales.
- **archivo.bar.** Si el problema estacionario es el estado inicial de un transitorio donde se realiza un movimiento de barras, podemos introducir la posición de las barras con este archivo.
- **archivo.tri** y **archivo.tri.vec.** Es un archivo generado por FEMFFUSION (la primera vez que se ejecuta el ejemplo) para la creación de la malla en el caso de la discretización espacial de los pines.
- **archivo.xlm.** Contiene datos referentes a la geometría del reactor. Este archivo se utiliza para cálculos a nivel de pin.

El nombre de los archivos se introduce en el archivo principal ‘.prm’ como variable de entrada.

Los archivos de entrada ‘.xsec’ y ‘.xlm’ tienen un formato completamente distintos entre sí. Un archivo de entrada ‘.xsec’ proporciona información geométrica plano a plano del reactor nuclear por medio de valores que representan cada uno de los materiales (i.e. 1, 2, 3, 4, 5, 6), proporciona las características del material representadas por sus secciones eficaces para un máximo de 2 grupos de energías, las características  $\beta$  y  $\lambda$  de los precursores, y, las velocidades de los neutrones. Un archivo de entrada ‘.xlm’ recoge los valores vectoriales y matriciales que representan las características del reactor con ayuda de las secciones eficaces de los materiales del mismo para cada uno de los grupos de energía. El benchmark C5G7 trabaja con archivos de entrada ‘.xlm’ y usa 7 grupos de energía. Más información sobre ello se halla en el informe de validación FEMFFUSION: <https://femffusion.webs.upv.es/>

### 6.1. Archivo principal o archivo ‘.prm’

El archivo ‘.prm’ contiene los parámetros de entrada del código. Para añadir valores introducimos en el fichero

```
set Nombre = Opcion
```

donde nombre es una de las variables que se listan en la Tabla 1 y opción una de las opciones que contiene esa variable. En el caso de los vectores, éstos se introducen como números consecutivos separados por espacios.

**Tabla 1:** Parámetros de entrada para el archivo '.prm'. El símbolo \* indica el valor por defecto.

	<b>Nombre</b>	<b>Descripción</b>	<b>Comando</b>	<b>Opciones</b>
<b>Parámetros generales</b>	Dimension	<i>Dimensión del problema</i>		1, 2, 3*
	Transport_Appr	<i>Tipo de aproximación del transporte</i>	-transport	Diffusion* SPN
	N_SPN	<i>Número 'N' de las ecuaciones SP<sub>N</sub></i>		1,3*,5
	FE_Degree	<i>Grado de los polinomios en el método de elementos finitos</i>	-fe_degree	1,2,3*,4,5
	Energy_Groups	<i>Número de grupos energéticos</i>		1,2*,...,7
<b>Parámetros geométricos</b>	Geometry_Type	<i>Geometrías del reactor</i>		Rectangular* Hexagonal Unstructured Composed
	N_Refinements	<i>Número de refinamientos globales</i>	-n_refinements	0*,...,10
	N_Refs_Radial	<i>Número de refinamientos radiales</i>		0*,...,10
	N_Refs_Axial	<i>Número de refinamientos axiales</i>		0*,...,10
	Mesh_Filename	<i>Archivo donde se declara la malla '.msh'</i>		
	Geometry_Filename	<i>Archivo donde se declara la malla '.xml'</i>		
	Mesh_Size	<i>Número de celdas por dimensión</i>		Vector
	Cell_Pitch_x	<i>Tamaño de las celdas en la dimensión x</i>		Vector
	Cell_Pitch_y	<i>Tamaño de las celdas en la dimensión y</i>		Vector
	Cell_Pitch_z	<i>Tamaño de las celdas en la dimensión z</i>		Vector
	Geometry_Points	<i>Primera y última celda de cada fila en la planta del reactor</i>		Vector
	Geometry_Matrix	<i>Matriz con la geometría en la planta reactor</i> <i>0: Sin material</i> <i>1: Fuel</i> <i>2: Reflector</i>		0 1 2

	<b>Nombre</b>	<b>Descripción</b>	<b>Comando</b>	<b>Opciones</b>
<b>Parámetros geométricos</b>	Boundary_Conditions	<i>Condiciones de contorno: x1, x2; y1, y2; z1, z2 0: Flujo cero 1: Simetría 2: Albedo 3: Vacío</i>		0 1 2 3
	Albedo_Factors	<i>Factores de Albedo para cada grupo</i>		Vector
	Triang_Filename	<i>Archivo donde la triangulación se crea y se guarda</i>		
	Refinement_Model	<i>Modelo de refinamiento para el caso de geometría compuesta</i>		Local Uniform
<b>Archivos de salida</b>	Output_Filename	<i>Archivo donde se guarda el output</i>	-out_file	
	Output_Flag	<i>Crear o no un archivo '.vtk' con el output</i>	-out_flag	true, false*
	Print_Grid_Flag	<i>Dibujar la malla en un archivo '.eps'</i>	-print_grid	true, false*
	Out_Refinements	<i>Número de refinamientos por celda para el output '.vtk'</i>	-n_out_ref	[0,1*,...,10]
	Show_Eps_Convergence	<i>Imprime o no los valores de convergencia por pantalla</i>	-show_eps_convergence	true*, false
<b>Métodos de Cálculo</b>	Solver_Type	<i>Método para el problema de autovalores</i>	-solver_type	power_it bifpam slepc_2g ks gd newton hybrid
	N_Eigenvalues	<i>Número de autovalores a calcular</i>	-n_eigenvalues	1*,...,50
	EPS_Tolerance	<i>Tolerancia relativa para el problema de autovalores</i>	-tol_eps	1e-7 <sup>a</sup>
	KSP_Tolerance	<i>Tolerancia relativa para resolver sistemas lineales</i>	-tol_ksp	1e-12*
	Static_KSP_Tolerance	<i>Activar la tolerancia estática para los sistemas lineales</i>	-static_ksp_tol	true, false*
	Adjoint	<i>Resolver el problema adjunto</i>	-adjoint	true, false*

	<b>Nombre</b>	<b>Descripción</b>	<b>Comando</b>	<b>Opciones</b>
	Matrix_Free_Type	<i>Metodologías de uso matrix-free</i>	-matrixfree_type	full_allocated non_diagonal full_matrixfree
	P_Init	<i>Activar la inicialización multinivel</i>	-p_init	true*, false
	Init_Type	<i>Método de cálculo</i>	-init_type	multinivel* random krylov
Métodos de Cálculo	XSEC_Type	<i>Secciones eficaces de entrada</i> <i>XS2G: 2 grupos de energía por XS</i> <i>XSEC: 1 o 2 grupos de energía por XS*</i> <i>XLM: Más de 2 grupos de energía por XS (nivel de pin)</i>		XS2G XSEC* XLM
	XSECS_Filename	<i>Archivo con los valores de secciones eficaces</i>		
	Bar_Filename	<i>Archivo donde se define el movimiento de barras</i>		
	T_Modes	<i>Cálculo del valor propio usando un determinado modo</i>	-modes	lambda alpha gamma
	RodCuspingMethod	<i>Método de cálculo Rod Cusping</i> <i>1: Volumen de homogeneización*</i> <i>2: Mover todo el plano</i>		1* 2
Par. temporal	Time_Delta	<i>Paso temporal entre iteraciones</i>		0.01*
	Time_End	<i>Tiempo final de las iteraciones</i>		1*
	Transient	<i>Activar la transición o dinámica del sistema</i>	-transient	true, false*

<sup>a</sup> Leído como  $1 \cdot 10^{-7}$

## 7. Archivos de salida

El objetivo del código es conocer el factor de multiplicación efectivo,  $k_{\text{eff}}$  (o los siguientes modos) y la distribución del flujo neutrónico o de la potencia neutrónica. Para la utilización de los resultados y su posterior visualización con otras herramientas, FEMFFUSION crea los siguientes archivos de salida:

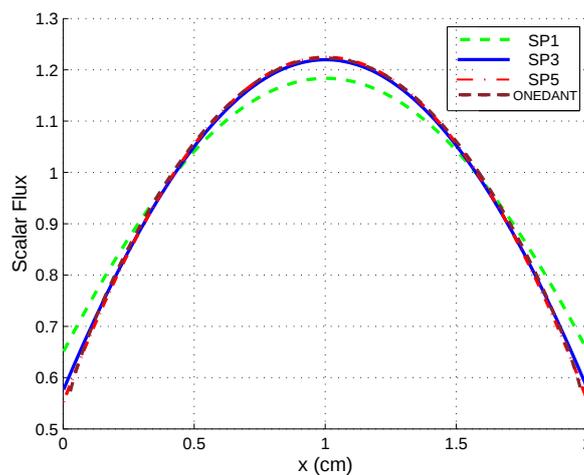
- **archivo.out.** Contiene los principales datos del problema (número de celdas, grados de libertad, ...), los autovalores calculados, el valor por celda de los flujos y la potencia, y en el caso de reactores tridimensionales el perfil axial de potencia.

- **archivo.vtk.** Contiene los valores de los flujos y la potencia en los nodos del problema para su visualización en programas como Paraview o Visit.
- **archivo.eps.** Dibujo de la malla que hemos utilizado en la resolución del problema. Este archivo se recomienda obtener solo en casos bidimensionales.
- **archivo.log.** Contiene información respectiva a la resolución del problema de autovalores: tipo de resolvidor, preconditionador, número de iteraciones, número de productos matrix-vector, tiempo de resolución, etc, ...

## 8. Ejemplos

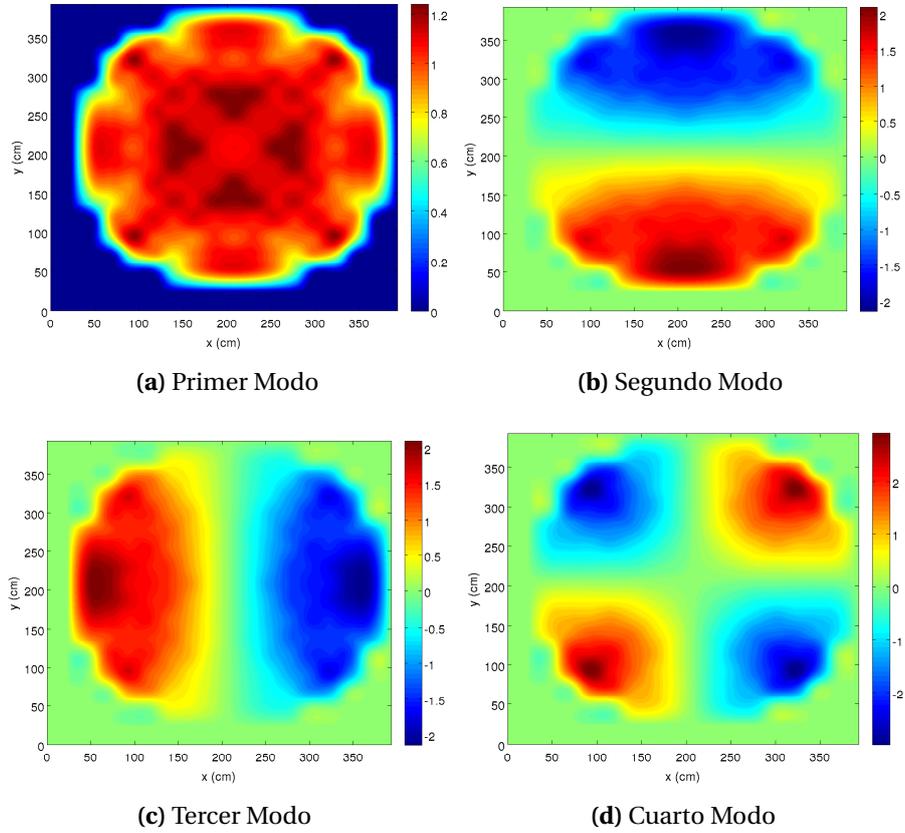
Consisten en una serie de reactores nucleares *benchmark* interesantes para estudiar tanto el flujo como la potencia de neutrones. A continuación se muestran los resultados para cada uno de los casos interesante a estudiar tras la ejecución de los problemas con el programa FEMFFUSION.

**1D:** Ejemplo de un reactor unidimensional de 2 cm resuelto con condiciones de contorno de vacío y un solo grupo de energía. La solución para las aproximaciones SP<sub>1</sub>, SP<sub>3</sub> y SP<sub>5</sub> pueden observarse en la Figura 1 junto a una referencia.



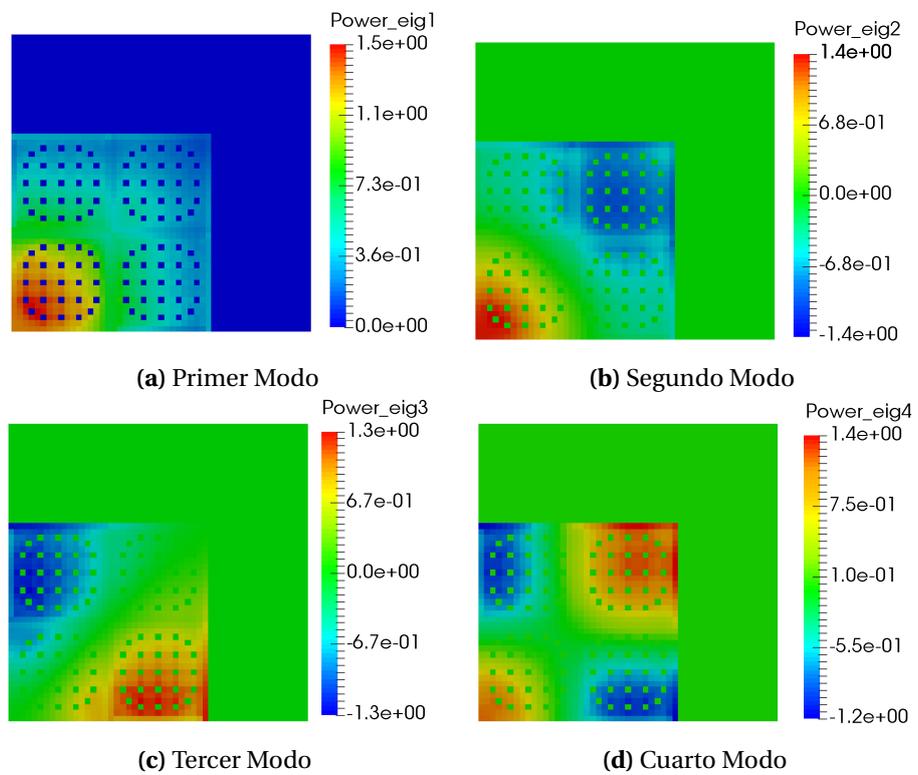
**Figura 1:** Flujo escalar para la losa homogénea 1D.

**2D\_BIBLIS:** Se trata de un *benchmark* clásico de difusión neutrónica con características de tablero de ajedrez [14]. Resuelto con aproximaciones SP1, SP3 y SP5. La solución para los primeros modos puede observarse en la Figura 2.



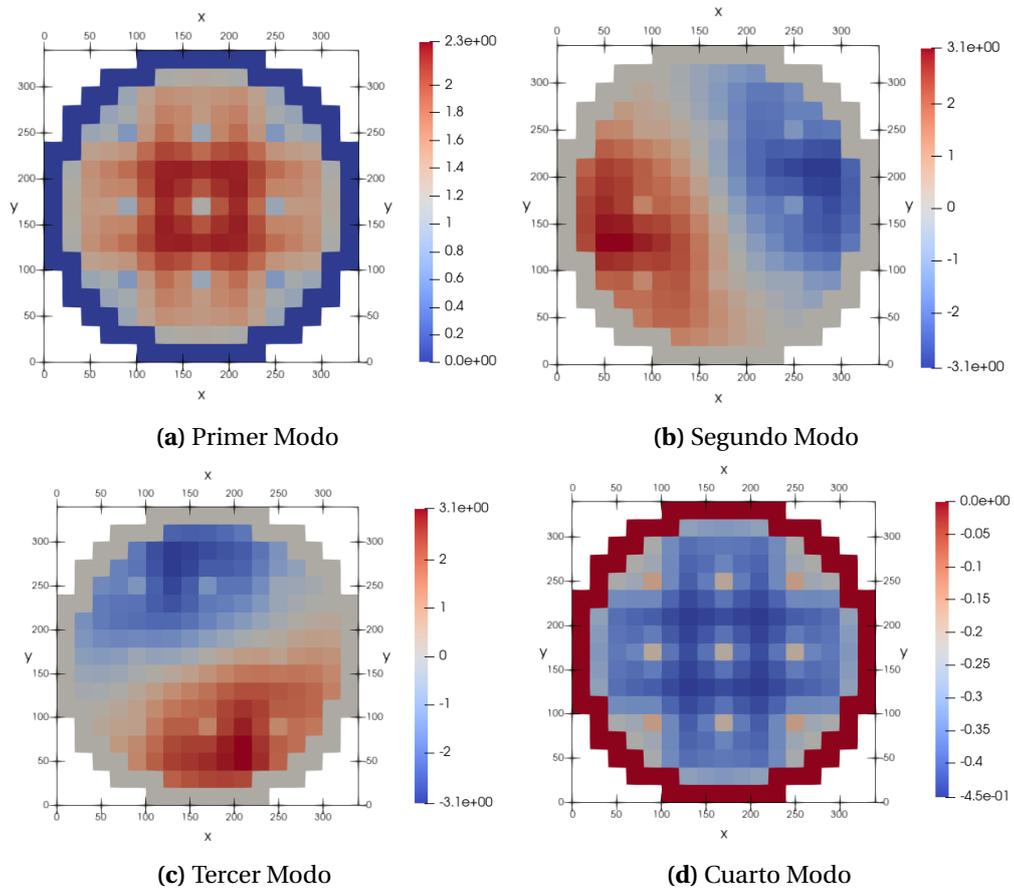
**Figura 2:** Potencia neutrónica para los cuatro modos en el reactor BIBLIS 2D.

**2D\_C5G7:** El *benchmark* numérico 2D C5G7 de la NEA para cálculos determinísticos de transporte a nivel de pin sin homogenización espacial (Figura 3). Para más información detallada sobre este *benchmark*, puede consultarse la referencia [15].



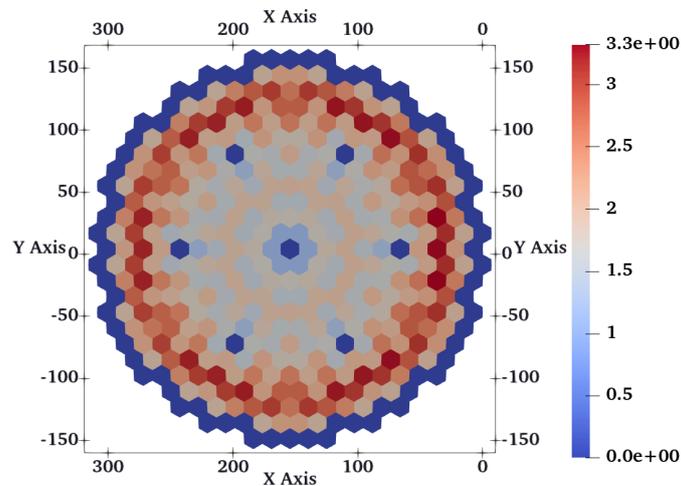
**Figura 3:** Potencia neutrónica calculada con las  $SP_3$  para los 4 primeros modos en el reactor bidimensional C5G7.

**3D\_IAEA:** Un *benchmark* 3D clásico con geometría rectangular (Figura 4). Para más información detallada sobre este *benchmark*, puede consultarse la referencia [16].



**Figura 4:** Potencia neutrónica en el plano mital para los primeros modos del reactor tridimensional IAEA.

**3D\_VVER440:** Un reactor hexagonal 3D (Figura 5). Para más información detallada sobre este *benchmark*, puede consultarse la referencia [17].



**Figura 5:** Potencia neutrónica en el plano mitad del reactor tridimensional VVER440.

## 9. Proyectos Relacionados

El código FEMFFUSION está financiado parcialmente por los siguientes proyectos:

- CORTEX project from the Euratom Research and Training Programme 2014-2018 under grant agreement No 754316.
- Grant NEUCONCEPT PDC2021-121667-I00 financed by MCIN/AEI / 10.13039/501100011033 and by the European Union Next GenerationEU/PRTR
- Grand PAID-10-19 and SP20180095 by Universitat Politècnica de València.



## Bibliografía

- [1] W.M. Stacey. *Nuclear reactor physics*. John Wiley & Sons, 2018.
- [2] O.C. Zienkiewicz y col. *The finite element method*. Vol. 3. McGraw-hill London, 1977.
- [3] W. Bangerth, R. Hartmann y G. Kanschat. “deal.II – A General-purpose Object-oriented Finite Element Library”. En: *ACM Trans. Math. Softw.* 33.4 (2007). ISSN: 0098-3500.
- [4] S. Abhyankar y col. “PETSc/TS: A Modern Scalable ODE/DAE Solver Library”. En: *arXiv preprint arXiv:1806.01437* (2018).
- [5] V. Hernandez, J.E. Roman y V. Vidal. “SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems”. En: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), págs. 351-362.
- [6] A.C. Hindmarsh y col. “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers”. En: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), págs. 363-396.
- [7] P. Bastian y col. “Matrix-free multigrid block-preconditioners for higher order Discontinuous Galerkin discretisations”. En: *Journal of Computational Physics* (2019).
- [8] A. Vidal Ferràndiz. “Development of a finite element method for neutron transport equation approximations”. Tesis doct. Universitat Politècnica de València, 2018.
- [9] G.W. Stewart. “A Krylov–Schur algorithm for large eigenproblems”. En: *SIAM Journal on Matrix Analysis and Applications* 23.3 (2002), págs. 601-614.
- [10] R.B. Morgan y D.S. Scott. “Generalizations of Davidson’s method for computing eigenvalues of sparse symmetric matrices”. En: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), págs. 817-825.
- [11] A. Carreño y col. “Block hybrid multilevel method to compute the dominant  $\lambda$ -modes of the neutron diffusion equation”. En: *Annals of Nuclear Energy* 121 (2018), págs. 513-524.
- [12] Antoni Vidal-Ferràndiz y col. “Moving meshes to solve the time-dependent neutron diffusion equation in hexagonal geometry”. En: *Journal of computational and applied mathematics* 291 (2016), págs. 197-208.
- [13] C. Geuzaine y J. Remacle. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”. En: *International journal for numerical methods in engineering* 79.11 (2009), págs. 1309-1331.

- [14] EZ Müller y ZJ Weiss. “Benchmarking with the multigroup diffusion high-order response matrix method”. En: *Annals of Nuclear Energy* 18.9 (1991), págs. 535-544.
- [15] M. A. Smith, E. E. Lewis y B. C. Na. *Benchmark on deterministic transport calculations without spatial homogenisation – A 2-D/3-D MOX Fuel Assembly Benchmark (C5G7 MOX Benchmark)*. Inf. téc. NEA/NSC/DOC(2003)16. OECD/NEA, 2003.
- [16] American Nuclear Society. *Argonne Code Center: benchmark problem book, ANL-7416(Suppl.2)*. Inf. téc. ANS, 1977. DOI: [10.2172/5037820](https://doi.org/10.2172/5037820).
- [17] Y.A. Chao e Y.A. Shatilla. “Conformal mapping and hexagonal nodal methods—II: implementation in the ANC-H code”. En: *Nuclear Science and Engineering* 121.2 (1995), págs. 210-225.