



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



*im*²
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, D. Ginestar y G. Verdú

5 de abril de 2022

Versión 0.1

Índice

1	Introducción	3
2	Instalación	3
3	Uso	4
4	Cálculo del problema estacionario	4
5	Representación de la geometría	5
6	Archivos de entrada	5
7	Archivos de salida	9
8	Ejemplos	10

1. Introducción

FEMFFUSION es un código neutrónico escrito en C++ que 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

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 se puede realizar 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 autoconf automake gcc g++ git cmake
sudo apt-get install valgrind libgsl-dev petsc-dev slepc-dev
sudo apt-get install gmsh gnuplot 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, se pueden realizar introduciendo en la terminal:

```
git clone https://Zonni@bitbucket.org/Zonni/femffusion.git
cd femffusion
./install.sh
```

La última instrucción tardará bastante tiempo, ya que debe compilar las librerías PETSC, SLEPC y DEAL.II (alrededor de 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. Uso

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 archivo.prm
```

A continuación del nombre del archivo, se pueden introducir otros comandos para la ejecución del código, propias de FEMFFUSION (Tabla 1) o provenientes de las librerías. En este caso, el comando -f indica que a continuación está el nombre del fichero. Otras opciones muy útiles en la línea de comandos son:

- -v. Muestra más detalles de la ejecución del código en la salida por pantalla.
- -s. No muestra nada por pantalla.
- -t. Ejecuta una serie de casos tests.
- -p. Evalúa el funcionamiento de los productos en matrix-free y los resolvedores 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 mediante la ecuación multigrupo de difusión neutrónica o las ecuaciones SP_N . Estas ecuaciones en estado estacionario tienen la forma

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

donde \mathcal{L} es el operador diferencial del transporte, \mathcal{M} es el operador diferencial de la fuente de neutrones, el vector ϕ denota el flujo de neutrones en estado estacionario y la constante k_{eff} es el valor de la k efectiva que 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 que se obtienen 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 sólo 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 Block Generalized 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.

- **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.
- **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 de los materiales y su distribución en la geometría. Sin embargo, es posible que según el tipo de problema, se requieran otros archivos de entrada. Un resumen de los archivos de entrada de FEMFFUSION es:

- **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 de energía. En éstos 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]. 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.

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.

Nombre	Descripción	Comando	Opciones
Dimension	<i>Dimensión del problema</i>		1, 2, 3*
Transport_Appr	<i>Tipo de aproximación del transporte</i>		Diffusion* SPN
N_SPN	<i>Número 'N' de las ecuaciones SP_N</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

Nombre	Descripción	Comando	Opciones
Energy_Groups	Número de grupos de energía		1,2*,...,7
Geometry_Type	Tipo de geometría del reactor		Rectangular* Hexagonal Unstructured Composed
N_Refinements	Número de refinamientos globales	-n_refinements	0*,...,10
N_Refs_Radial	Número de refinamientos radiales		0*,...,10
N_Refs_Axial	Número de refinamientos axiales		0*,...,10
Mesh_Filename	Archivo donde se declara la malla *.msh		
Geometry_Filename	Archivo donde se declara la malla *.xml		
Mesh_Size	Número de celdas por dimensión		Vector
Cell_Pitch_x	Tamaño de las celdas en la dimensión x		Vector
Cell_Pitch_y	Tamaño de las celdas en la dimensión y		Vector
Cell_Pitch_z	Tamaño de las celdas en la dimensión z		Vector
Geometry_Points	Primera y última celda de cada fila en la planta del reactor		Vector
Geometry_Matrix	Matriz con la geometría en la planta reactor		0: no material 1: fuel 2: reflector
Boundary_Conditions	Condiciones de contorno: x1 x2 y1 y2 z1 z2		0: Flujo cero 1: Simetría 2: Albedo 3: Vacío
Albedo_Factors	Factores de Albedo para cada grupo		Vector
Triang_Filename	Archivo donde la triangulación se crea y se guarda		
Refinement_Model	Modelo de refinamiento para el caso de geometría compuesta		Local Uniform

Nombre	Descripción	Comando	Opciones
Output_Filename	Archivo donde se guarda el output	-out_file	
Output_Flag	Crear o no un archivo *.vtk con el output	-out_flag	True, False*
Print_Grid_Flag	Dibujar la malla en un archivo *.eps	-print_grid	True, False*
Out_Refinements	Número de refinamientos por celda para el output *.vtk	-n_out_ref	[0,1*,...,10]
Solver_Type	Método para el problema de autovalores	-solver_type	power_it bifpam slepc_2g ks gd newton hybrid
N_Eigenvalues	Número de autovalores a calcular	-n_eigenvalues	1*,...,50
EPS_Tolerance	Tolerancia relativa para el problema de autovalores	-tol_eps	1e-7*
KSP_Tolerance	Tolerancia relativa para resolver sistemas lineales	-tol_ksp	1e-12*
Static_KSP_Tolerance	Usar una tolerancia estática para los sistemas lineales	-static_ksp_tol	True, False*
Adjoint	Resolver el problema adjunto	-adjoint	True, False*
Matrix_Free_Type	Tipo de metodología matrix-free	-matrixfree_type	full_allocated non_diagonal full_matrixfree
P_Init	Usar inicialización multinivel	-p_init	True*, False
XSEC_Type	Tipo de secciones eficaces a leer		Femffusion XSEC Forest Valkin
XSECS_Filename	Archivo de las secciones eficaces		
Bar_Filename	Archivo donde se define el movimiento de barras		

Nombre	Descripción	Comando	Opciones
RodCuspingMethod	Método para el rod cusping		1: Homogeneización* 2: Mover todo el plano

7. Archivos de salida

El objetivo del código es conocer la constante multiplicativa, k_{eff} (o los siguiente 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 siguiente 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:

1. **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 SP₁, SP₃ SP₅ se puede observar en la Figura 1 junto a una referencia.

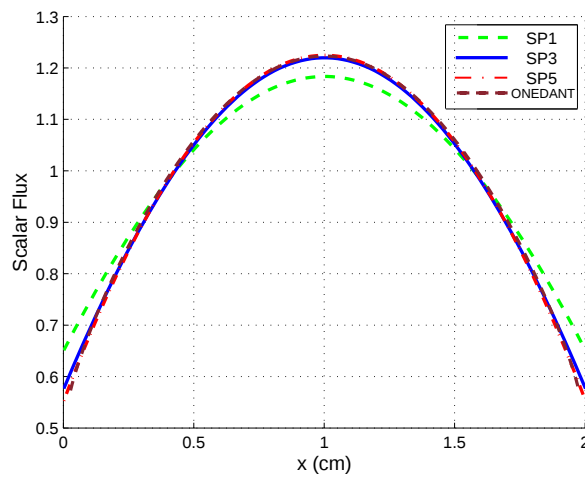


Figura 1

2. **2D_BIBLIS:** Se trata de un benchmark clásico de difusión neutrónica con características de tablero de ajedrez [13]. Resuelto con aproximaciones SP₁, SP₃ y SP₅. La solución para los primeros modos se puede ver en la Figura 2.

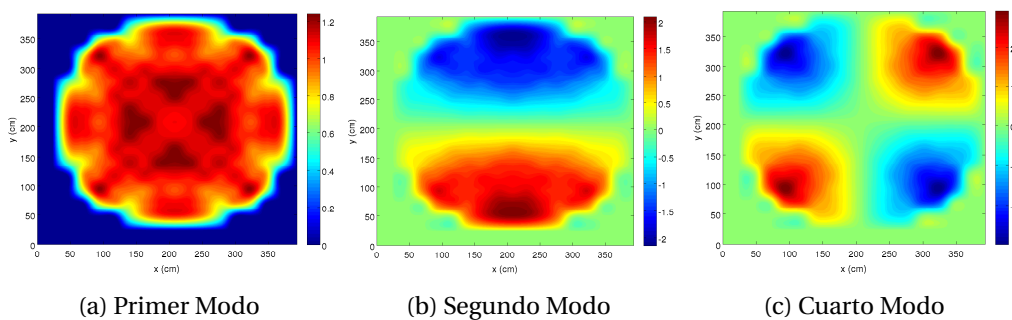


Figura 2: Potencia neutrónica de los primeros modos en el reactor BIBLIS 2D.

3. **2D_C5G7**: El benchmark numérico 2D-C5G5 de la NEA para cálculos determinísticos de transporte a nivel de pin sin homogenización espacial. La definición completa se puede encontrar en [14].

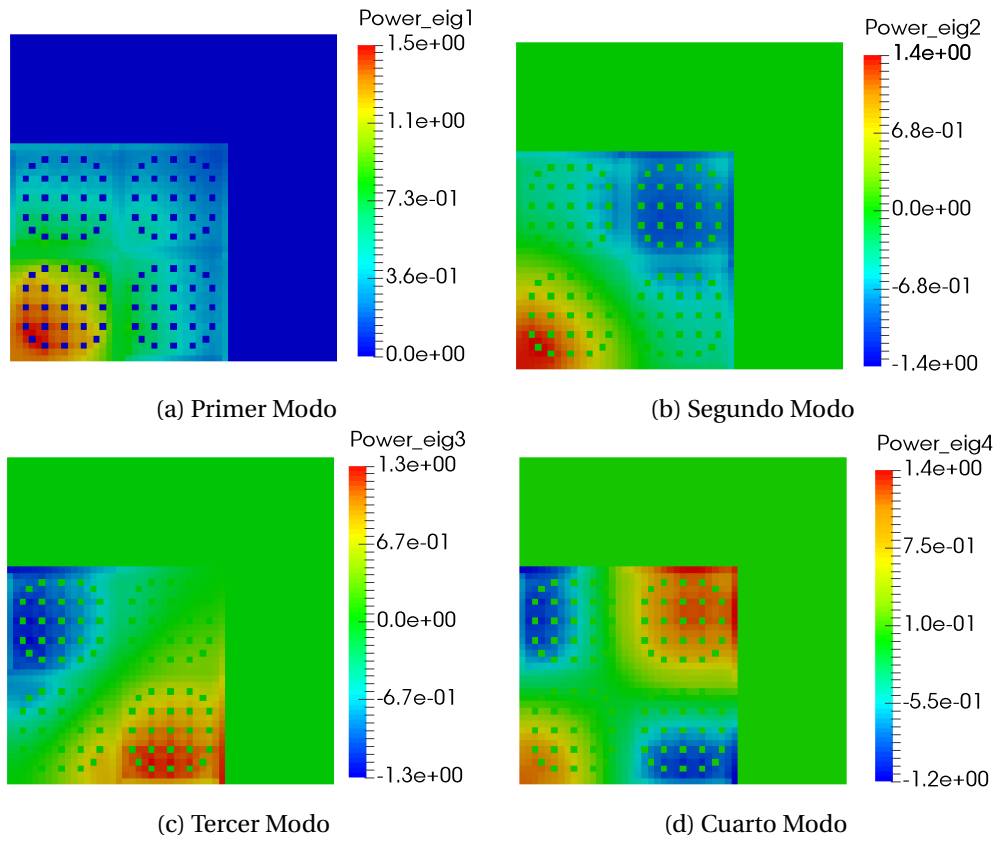


Figura 3: Potencia neutrónica calculada con las SP_3 para los 4 primeros modos en el reactor 2D-C5G7

4. **3D_IAEA:** Un benchmark 3D clásico con geometría rectangular. La definición completa se pueda encontrar en [15].

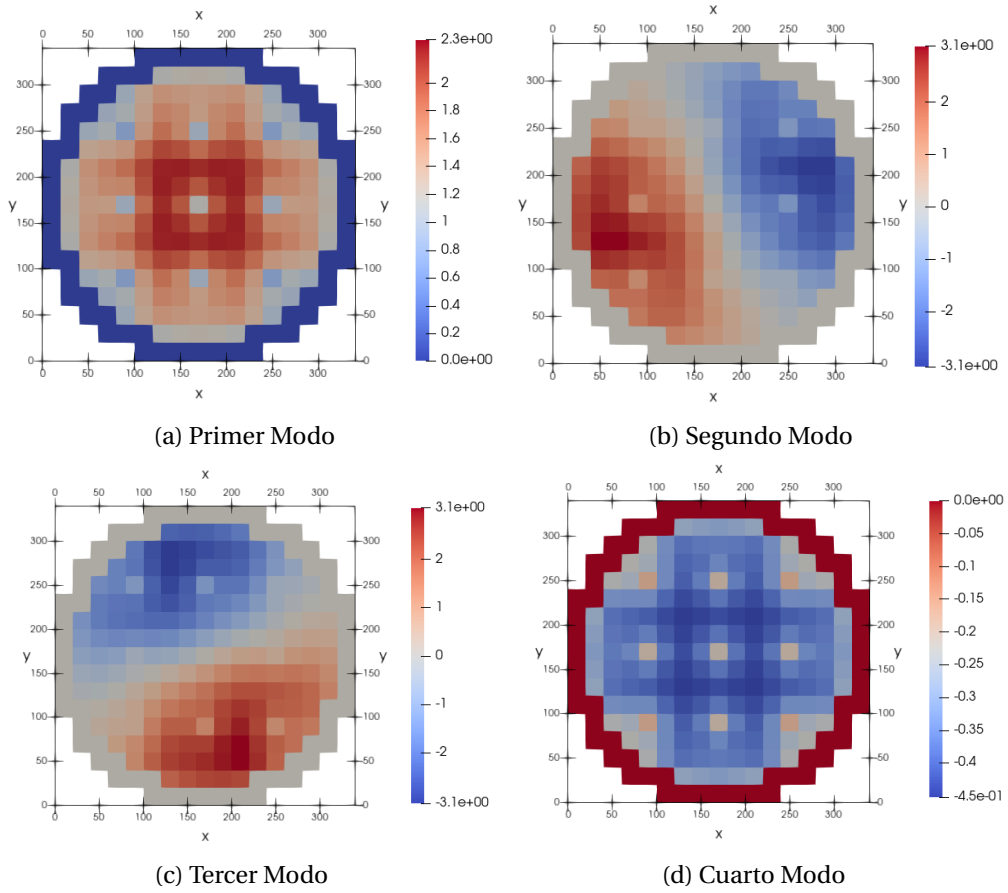


Figura 4: Potencia neutrónica para los primeros modos en el reactor 3D-IAEA

5. **3D_VVER440**: Un reactor hexagonal 3D. La definición completa se pueda encontrar en [16].

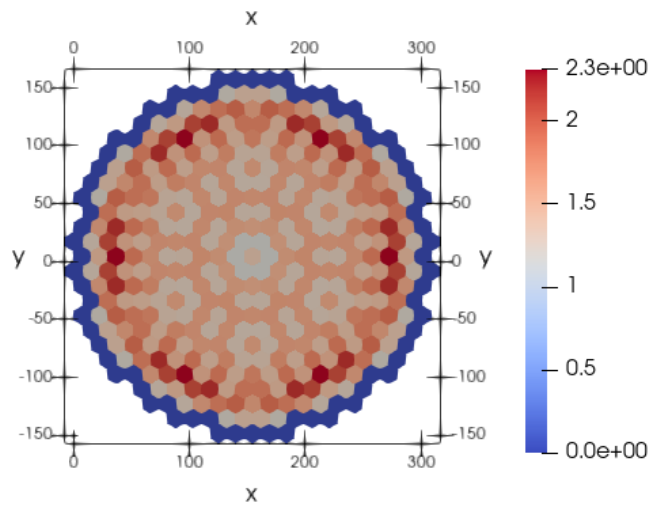


Figura 5: Potencia neutrónica para el primer modo en el reactor 3D-VVER440

Referencias

- [1] W. Stacey, *Nuclear reactor physics*. John Wiley & Sons, 2018.
- [2] O. Zienkiewicz, R. Taylor, P. Nithiarasu, and J. Zhu, *The finite element method*, vol. 3. McGraw-hill London, 1977.
- [3] W. Bangerth, R. Hartmann, and G. Kanschat, “deal.ii – a general-purpose object-oriented finite element library,” *ACM Trans. Math. Softw.*, vol. 33, no. 4, 2007.
- [4] S. Abhyankar, J. Brown, E. M. Constantinescu, D. Ghosh, B. Smith, and H. Zhang, “Petsc/ts: A modern scalable ode/dae solver library,” *arXiv preprint arXiv:1806.01437*, 2018.
- [5] V. Hernandez, J. Roman, and V. Vidal, “SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 351–362, 2005.
- [6] A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker, and C. Woodward, “Sundials: Suite of nonlinear and differential/algebraic equation solvers,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.
- [7] P. Bastian, E. Müller, S. Müthing, and M. Piatkowski, “Matrix-free multigrid block-preconditioners for higher order discontinuous galerkin discretisations,” *Journal of Computational Physics*, 2019.

- [8] A. Vidal Ferràndiz, *Development of a finite element method for neutron transport equation approximations*. PhD thesis, Universitat Politècnica de València, 2018.
- [9] G. Stewart, “A Krylov–Schur algorithm for large eigenproblems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 601–614, 2002.
- [10] R. Morgan and D. Scott, “Generalizations of davidson’s method for computing eigenvalues of sparse symmetric matrices,” *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 817–825, 1986.
- [11] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú, “Block hybrid multilevel method to compute the dominant λ -modes of the neutron diffusion equation,” *Annals of Nuclear Energy*, vol. 121, pp. 513–524, 2018.
- [12] C. Geuzaine and J. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities,” *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [13] E. Müller and Z. Weiss, “Benchmarking with the multigroup diffusion high-order response matrix method,” *Annals of Nuclear Energy*, vol. 18, no. 9, pp. 535–544, 1991.
- [14] M. A. Smith, E. E. Lewis, and B. C. Na, “Benchmark on deterministic transport calculations without spatial homogenisation – A 2-D/3-D MOX Fuel Assembly Benchmark (C5G7 MOX Benchmark),” Tech. Rep. NEA/NSC/DOC(2003)16, OECD/NEA, 2003.
- [15] American Nuclear Society, “Argonne code center: benchmark problem book, anl-7416(suppl.2),” tech. rep., ANS, 1977.
- [16] Y. Chao and Y. Shatilla, “Conformal mapping and hexagonal nodal methods—ii: implementation in the anc-h code,” *Nuclear Science and Engineering*, vol. 121, no. 2, pp. 210–225, 1995.