

MONOGRAFÍA FINAL DE LA MATERIA
INTRODUCCIÓN AL MÉTODO DE ELEMENTOS FINITOS

**Difusión de neutrones en mallas no estructuradas:
comparación entre volúmenes y elementos finitos**

Mg. Ing. Germán Theler



Facultad de Ingeniería
Universidad Nacional
de Buenos Aires



Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica

Buenos Aires, Argentina
Diciembre de 2013

Resumen

Difusión de neutrones en mallas no estructuradas: comparación entre volúmenes y elementos finitos

Mg. Ing. Germán Theler

El objetivo principal de esta monografía es resolver numéricamente una ecuación matemática que eventualmente puede llegar a tener aplicaciones de interés—tanto académicas como industriales—para analizar, estudiar y optimizar reactores nucleares. Para ello hemos desarrollado un código computacional llamado milonga y distribuido libremente bajo los términos de la Licencia Pública General GNU. En el presente trabajo, primero introducimos dos esquemas numéricos para resolver la ecuación de difusión de neutrones multigrupo sobre una malla no estructurada, uno basado en el método de volúmenes finitos y otro basado en el método de elementos finitos. Luego resolvemos algunos problemas sencillos—y otros no tanto—utilizando ambos esquemas y comparamos los resultados obtenidos. Introducimos primero los conceptos matemáticos básicos relacionados con la ecuación de difusión de neutrones multigrupo y luego desarrollamos las ideas de ambos esquemas numéricos, que son los que hemos programado en la versión actual del código milonga y que utilizamos para resolver problemas introducidos en orden de complejidad creciente, haciendo especial énfasis en la comparación entre volúmenes y elementos finitos.



Difusión de neutrones en mallas no estructuradas: comparación entre volúmenes y elementos finitos by jeremy theler is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Abstract

Neutron diffusion on unstructured grids: comparison between finite volumes and finite elements

Mg. Ing. Germán Theler

The main objective of this monograph is to numerically solve a mathematical equation that eventually can have applications of interest—both in the academia and in the industry—to analyze, study and optimize nuclear reactors. In order to do that, we have developed a computational code named milonga which is freely distributed under the terms of the GNU General Public License. IN this work, we first introduce two numerical schemes to solve the multigroup neutron diffusion equation over an unstructured mesh—one based on the finite volumes method and the other one based on the finite element method. Then we solve some simple—and others no to simple—using both schemes and we compare the obtained results. We introduce first the basic mathematical concepts related to the multigroup neutron diffusion equation and the we develop the ideas of both numerical schemes, which are the ones that are programmed in the current version of the milonga, used to solve the problems hereby discussed. The cases are presented in increasing order of complexity, paying special attention to the comparison between finite volumes and finite elements.



Difusión de neutrones en mallas no estructuradas: comparación entre volúmenes y elementos finitos by jeremy theler is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Contenidos

1. Introducción	1
2. La ecuación de difusión de neutrones	7
2.1. Marco físico-matemático	7
2.2. Difusión de neutrones	10
2.2.1. Grupos de energía discreta	13
2.2.2. Condiciones de contorno	14
2.2.3. El problema de autovalores	16
2.3. La ecuación de difusión discreta	18
3. Volúmenes finitos	21
3.1. Esquema basado en volúmenes	21
3.1.1. Reacciones totales	23
3.1.2. Fisiones	24
3.1.3. Scattering	24
3.1.4. Fugas	25
3.2. Formulación matricial	31
4. Elementos finitos	34
4.1. Residuos ponderados	35
4.1.1. Ecuación escalar en una dimensión	35
4.1.2. Ecuación escalar en varias dimensiones	42
4.1.3. Ecuación vectorial en varias dimensiones	46
4.2. El método de elementos finitos	51
4.2.1. Ensamble de matrices	57
4.2.2. Funciones de forma	59
4.3. Formulación débil de la ecuación de difusión de neutrones	74
4.4. Esquema basado en elementos	81
4.4.1. Fisiones	83
4.4.2. Remociones y scattering	83
4.4.3. Fugas	84
4.4.4. Condiciones de contorno	86
4.5. Equivalencia con el método de Galerkin	92
4.6. Formulación matricial	95
5. Resultados	100
5.1. Slabs unidimensionales	103
5.1.1. Slabs desnudos con propiedades homogéneas	103
5.1.2. Slab a dos zonas	127
5.1.3. Slabs reflejados a dos grupos	144



5.1.4. Slab con distribución continua de propiedades	159
5.2. Problemas en dos dimensiones	172
5.2.1. Círculo desnudo homogéneo	172
5.2.2. Corte transversal de un reactor con barras inclinadas	199
5.2.3. El problema de los pescaditos	225
5.2.4. Two-dimensional PWR IAEA Benchmark	235
6. Conclusiones	262
6.1. Trabajos futuros	263
A. Algunos resultados de la geometría del plano	265
A.1. La normal a un segmento	265
A.2. El área del triángulo	266
A.3. El área del cuadrángulo	266
A.4. Pertenencia de un punto a un triángulo	267
A.5. Pertenencia de un punto a un cuadrángulo	267
B. Identidades de Green	269
C. Cuadratura de Gauss	271
C.1. Una dimensión	271
C.2. Dos dimensiones	273

Introducción

- [director] ¿Sabés por qué trabajo de lo que trabajo?
- [tesista] Por la diversión (en el sentido de "for the fun").
- [director] Exacto.

*Diálogo entre el autor de este trabajo
y su director de tesis de maestría, 2008*

En diciembre de 1942, geográficamente bajo las tribunas del viejo estadio abandonado de fútbol americano de la Universidad de Chicago y orgánicamente bajo el ala del proyecto Manhattan, Enrico Fermi mostró experimentalmente que era posible lograr una reacción de fisión nuclear autosostenida. La facilidad conocida como Chicago Pile-1 consistía en una verdadera pila compuesta por ladrillos de diferentes materiales—dióxido de uranio como combustible y grafito como moderador—simplemente apoyados uno contra otros dejando algunos lugares libres por donde podían deslizar barras recubiertas de cadmio que actuaban como reguladoras. Fermi pudo, resolviendo la ecuación de difusión de neutrones con lápiz y papel, predecir cuántos ladrillos hacían falta para lograr la masa crítica y estimar a qué altura se debían extraer las barras de control para lograr la primera cadena de fisiones autosostenidas inducidas por el hombre.

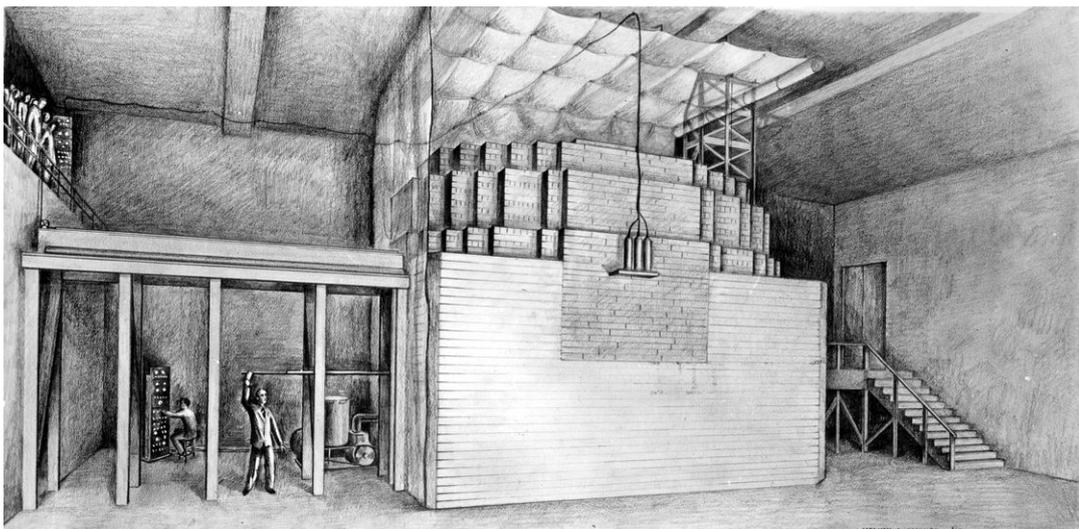


Figura 1.1 – Representación artística de la Chicago Pile-1 donde Enrico Fermi logró la primera reacción nuclear de fisión autosostenida de la historia en diciembre 1942. University of Chicago Photographic Archive, [apf2-00503], Special Collections Research Center, University of Chicago Library.



Las mismas ecuaciones que utilizó Fermi para diseñar la CP-1 hace más de setenta años siguen siendo las más empleadas en el diseño y en el análisis de reactores nucleares de fisión, tanto en el ambiente académico como en la industria, ya que presenta ciertas ventajas que la hacen bastante atractiva frente a otras formulaciones matemáticas. Estas ecuaciones son básicamente un conjunto de ecuaciones diferenciales en derivadas parciales acopladas, donde los operadores diferenciales son de segundo orden y se aplican sólo a las coordenadas espaciales. La ecuación de difusión es una simplificación de la ecuación de transporte de Boltzmann—que a su vez se basa en la suposición de que los neutrones se comportan en forma determinística—y que, como todos los modelos matemáticos, es aplicable con una precisión razonable sólo en algunos casos. Además, la ecuación de difusión se utiliza principalmente en cálculos a nivel de núcleo—en contraposición de los llamados cálculos a nivel de celda. Por todos estos motivos, aunque la difusión de neutrones resulte atractiva desde el punto de vista matemático, su aplicación a sistemas físicos reales y la interpretación de los resultados obtenidos deben ser realizadas con un cuidadoso criterio ingenieril.

Si bien el espíritu de la ingeniería en general es resolver problemas prácticos de la mejor manera y con el menor costo posible, en particular este trabajo se centra solamente en la matemática involucrada detrás de la resolución de la ecuación de difusión de neutrones multigrupo. Es decir, el objetivo es resolver un sistema de ecuaciones en derivadas parciales. Sabemos que de alguna manera, los métodos y resultados obtenidos pueden ser aplicados a sistemas físicos reales. Pero el alcance de esta monografía se limita a estudiar, analizar y entender la matemática que yace detrás de la ecuación de difusión de neutrones multigrupo en estado estacionario. Es en este sentido en el cual deben ser entendidos las ideas introducidas y los resultados obtenidos.

Usualmente, en cálculos de núcleo a nivel industrial, la ecuación de difusión de neutrones se resuelve utilizando esquemas numéricos basados en diferencias sobre una malla estructurada. Sin embargo, en ciertas ocasiones es interesante resolver el mismo problema de varias maneras para analizar los errores introducidos por las diferentes aproximaciones utilizadas al transformar una ecuación diferencial continua en una cantidad finita de ecuaciones algebraicas.

Sucede que, por un lado, en la literatura de reactores nucleares donde el problema de difusión de neutrones es casi central, las mallas no estructuradas no son muy tratadas principalmente porque su aplicación en la industria es más bien escasa. Y, por otro lado, los libros de métodos numéricos que sí muestran cómo resolver ecuaciones sobre mallas no estructuradas no hablan sobre neutrónica ya que es un tema demasiado particular que no tiene una audiencia lo suficientemente grande como para ser objeto de ejemplos de elementos finitos.¹ Más aún, aunque matemáticamente la ecuación de difusión de neutrones es similar a otras ecuaciones de difusión, como por ejemplo la ecuación de conducción del calor—que sí es extensamente tratada en la literatura de elementos finitos—usualmente los coeficientes de los ejemplos resueltos se toman constantes. Los casos de interés de la ecuación de difusión de neutrones involucran varios materiales y, usualmente, una dependencia continua de los coeficientes de la ecuación con la posición espacial, aún para un mismo material. Sucede también que los libros clásicos que introducen los conceptos de elementos finitos para resolver problemas de ingeniería se enfocan más en resolver ejemplos canónicos del problema elástico lineal y de mecánica de fluidos que en explicar las características matemáticas los métodos numéricos involucrados, dejando afuera otro tipo de ecuaciones como las de Maxwell o la de Schrödinger, que bien podrían servir para introducir e ilustrar ideas generales aplicables a cualquier tipo de problemas. Sólo a modo de ejemplo, la referencia [1] indica en su título que debería tratar sobre procedimien-

¹Lo que coloquial y e incorrectamente se suele llamar “masa crítica”, configurando una paradójica situación en la que dicho término coloquial ha encontrado más aplicación en la comunidad científica que la física de reactores propiamente dicha.



tos relacionados a elementos finitos y se trata en verdad de un libro sobre la resolución de problemas de ingeniería civil utilizando métodos numéricos, arribando a algunos resultados puramente matemáticos sólo luego de resolver ejemplos prácticos que tienen interés sólo para ingenieros estructuralistas en lugar de desarrollar una teoría sólida a partir de axiomas matemáticos sencillos.

El objeto principal de estudio es la comparación de los resultados obtenidos al resolver la ecuación de difusión de neutrones a pocos grupos de energía en estado estacionario con volúmenes finitos por un lado y por elementos finitos por el otro. Siendo estos dos esquemas numéricos en su naturaleza, es de esperar que su implementación práctica se haga en algún tipo de computadora digital. El paradigma actual de la computación científica al momento de escribir estas líneas (Diciembre de 2013) consiste en la utilización de computadoras con representación binaria, memoria accesible a través de un flat address space con diferentes niveles de cache con direcciones de tamaño fijo [2]. Cada una de estas computadoras digitales posee varios núcleos de microprocesadores tipo CISC y usualmente están conectadas entre sí a través de una red tipo Ethernet formando clústers de cálculo que distribuyen la carga computacional entre sus nodos. Por una serie de razones técnicas que no viene al caso enumerar [2, 3], el sistema operativo estándar para estas configuraciones es alguna variante de GNU/Linux. Ahora bien, además de las razones técnicas sucede que el proyecto GNU—que consiste en escribir un sistema operativo completamente *libre*—está basado en las ideas fundamentales de libertades individuales y colectivas. Particularmente, en términos de software esto implica que, dado un cierto programa, cualquier persona debe poseer simultáneamente las libertades de

0. ejecutarlo para cualquier propósito dentro de las reglas de la ética
1. estudiar cómo funciona y cambiarlo para que haga lo que esa persona necesite que haga
2. redistribuir copias del software original para compartirlo con la comunidad
3. redistribuir copias de las versiones modificadas para compartirlas con la comunidad

Estas cuatro libertades básicas no son linealmente dependientes como podría parecer a primera vista, y tomadas como una base generan el subespacio del software libre, cuyas implicaciones y discusión detallada podemos encontrar en literatura especializada [4]. De todas maneras, el concepto de *free software*—con el adjetivo usado como en *free speech* y no como en *free beer* [5]—es tremendamente importante para el desarrollo de la ciencia y la tecnología como ilustra el humilde ejemplo del código computacional desarrollado y utilizado para resolver los problemas propuestos en esta monografía.

Por otra serie de razones, usualmente los programas de cálculo suelen ser software privativo que, además de estar involucrar costos casi siempre inalcanzables para un estudiante promedio, *priva* a los usuarios de las cuatro libertades básicas. Casi todos los códigos “enlatados” con los que me topé durante mis épocas de estudiante y luego como profesional son entregados en forma binaria para una cierta arquitectura sin que podamos ejecutarlos como quisiera (libertad 0). El código fuente original rara vez es entregado para que podamos estudiar cómo funciona y mucho menos modificarlo para agregarle funcionalidades (libertad 1). No tenemos permitido regalar una copia del programa a algún amigo que podría beneficiarse con él (libertad 3), y de yapa nos califican con un remoquete propio de gente que atraca barcos para robar su cargamento (sic). En estas condiciones, ni siquiera es necesario decir que no podemos distribuir versiones modificadas (libertad 4). Desde el punto de vista académico de distribución del conocimiento existente y de generación de nuevo conocimiento, el software libre toma una importancia central. Cualquier software privativo está justamente privando a la



comunidad del acceso al conocimiento y dificulta la generación de nuevo conocimiento y debe ser evitado—por no usar una palabra más fuerte—por las comunidades académicas, científicas y tecnológicas.

Comencé a desarrollar el código neutrónico milonga desde cero en 2010 como una herramienta computacional de soporte a mi tesis de Doctorado en Ingeniería Nuclear con las premisas del software libre desde la concepción de sus bases de diseño [6]. A mediados de 2011 la primera versión de milonga es liberada bajo la licencia GNU General Public License [7]. En la versión 0.1, milonga es capaz de resolver el problema de difusión de neutrones multigrupo en estado estacionario en una y dos dimensiones usando un método de discretización espacial basado en volúmenes finitos sobre mallas estructuradas (y las bibliotecas PETSc [8, 9] y SLEPc [10] para resolver el problema de autovalores). Cuando en el segundo semestre de 2011 cursé la materia Introducción a Elementos Finitos a cargo del Dr. Sergio Ferro en la Universidad de Buenos Aires como parte de mi plan de formación, el paso siguiente era tan obvio como inevitable: incorporar la capacidad de resolver el problema de difusión de neutrones multigrupo en mallas no estructuradas utilizando el método de elementos finitos. Es esta monografía entonces hija de esta conjunción de condiciones, a saber, un código numérico basado en volúmenes finitos sobre mallas estructuradas y el cursado de una materia que trata sobre elementos finitos en mallas no estructuradas. La consecuencia es, además de un documento de varias páginas en formato A4, la versión 0.2 del código libre milonga. Esta versión será liberada pública y oficialmente una vez que la documentación sea actualizada en forma acorde a los cambios introducidos en el código fuente. En el capítulo 6 discutimos las características deseadas de las versiones futuras de milonga, pero el objetivo es llegar a una versión 1.0 que sea capaz de resolver problemas tridimensionales con geometrías y propiedades de reactores reales de interés industrial. Sin perjuicio de ello, volvemos a repetir que este trabajo se trata de resolver ecuaciones matemáticas utilizando diferentes métodos numéricos para tratar de arribar a conclusiones estrictamente matemáticas. Los juicios ingenieriles están fuera del presente alcance.

Siendo lo que es, un trabajo final de una materia de posgrado, es innegable que la extensión del documento es un tanto excesiva y seguramente su lectura resultará tediosa y aburrida. Sucede que, tal como en la cita con la que comienza el capítulo, me gusta hacer lo que hago, trabajar de lo que trabajo y escribir lo que escribo. Así que lo que comenzó como un escueto trabajo práctico se fue transformando en una larga monografía, rayana a una tesis en sí misma, cuya escritura demandó un año y medio a lo largo del cual fueron apareciendo conceptos e ideas que fui creyendo conveniente incluir. En el ínterin, entre volúmenes y elementos alcancé—y sobrepasé—algunos hitos en mi vida tal como el casamiento con mi hermosa esposa. Sabrá entonces entender el amable lector que estas hojas abrochadas—o el correspondiente archivo electrónico de varios megabytes de tamaño—esconde un trasfondo más complejo que una monografía larga. Algunos pasajes y la aparición de chistes como notas al pie reflejan vaivenes de humor y emociones, propios de un año y medio en la vida de un ingeniero. No sólo cambia el estilo narrativo sino su agudeza y exigencia matemática, por lo que si ciertos párrafos parecen escritos por diferentes personas es porque lo son. Uno fue escrito por un soltero y otro por un casado.

Con el objetivo de intentar hacer lo más autosostenida posible la presentación, en el capítulo 2 introducimos algunas ideas generales y muy básicas sobre la naturaleza del problema a resolver. Luego pasamos a describir los dos esquemas numéricos con los que discretizamos espacialmente la ecuación de difusión de neutrones. En el capítulo 3 desarrollamos un método basado en volúmenes finitos, que es una generalización a mallas no estructuradas del esquema usado en la primera versión de milonga sobre mallas estructuradas. El capítulo 4 es donde aplicamos el método de elementos finitos a la difusión de neutrones. Su longitud fue creciendo



a medida que era necesario justificar algunos pasos matemáticos que fueron implementados en el código, especialmente en relación con las condiciones de contorno. En su forma final, el desarrollo va desde lo general a lo particular introduciendo ejemplos sobre cómo aplicar los conceptos de complejidad creciente que vamos introduciendo para resolver ecuaciones sencillas y puramente matemáticas, es decir, sin una aplicación física o ingenieril directa. En el caso general de sistemas de ecuaciones en varias dimensiones a veces es necesario realizar operaciones sobre escalares, vectores, matrices y tensores lo que acarrea consigo la utilización de índices, subíndices, supraíndices y tipografías especiales para indicar los tipos de operaciones y operandos. Principalmente para lograr que entender yo mismo qué ecuaciones era las que estaba programando en milonga, es que decidí reemplazar las expresiones generales en función de una cantidad de grupos genérica G y una dimensión espacial D arbitraria, tomar casos particulares fijando G y D , escribir explícitamente los elementos de matrices y vectores llevando las expresiones a operaciones escalares para luego generalizar los resultados para cualquier cantidad de grupo y dimensiones espaciales. Esto deja un capítulo 4 bastante largo pero que ha resultado de especial interés por un lado para saber cabalmente qué es lo que se esconde detrás del método de elementos finitos y por otro para utilizar como referencia para cuando hayan pasado algunos años y me pregunte cuál es la relación entre el operador gradiente del flujo neutrónico y la matriz jacobiana de las funciones de forma de los elementos.

Cabe aclarar que si bien el capítulo 3 se llama “Volúmenes finitos” y el capítulo 4 se llama “Elementos finitos”, su contenido no refleja otra cosa más que dos de los muchos posibles esquemas de discretización espacial basados en dichos métodos. Es más, dichos capítulos reflejan ideas y conceptos de lo que yo entiendo por volúmenes y elementos finitos, los cuales deberían ser tomados con un especial juicio crítico.

En el capítulo 5 es donde resolvemos una cantidad de problemas, también de complejidad creciente. Comenzamos con casos canónicos como slabs unidimensionales con propiedades uniformes y vamos avanzando hacia problemas cada vez más complicados. Todos los problemas son resueltos utilizando ambos esquemas y los resultados son comparados a la luz de las ideas desarrolladas en los capítulos 3 y 4. Aunque arribamos a algunos resultados generales y conclusiones interesantes—tal como la notable ecuación (5.7), la figura 5.21 o la discusión de la página 227—aún hay muchas más conclusiones por sacar. Es posible que debido a la muestra de problemas elegidos para resolver y estudiar, muchos de los resultados sean consecuencias de los casos particulares tomados. Quedan aún muchas preguntas abiertas, problemas por resolver y conclusiones por obtener. Incluso solamente hemos tratado problemas en una y dos dimensiones debido a una limitación del código milonga. Es de esperar que en problemas tridimensionales aparezcan efectos que no hemos podido observar en los casos estudiados.

Hemos intentado analizar y discutir los problemas dentro de un alcance razonable. De todas maneras, hay muchas conclusiones que puede ser obtenidas a partir del estudio de las figuras resultantes que quedan a cargo del lector. En lo posible, hemos intentado incluir solamente figuras vectoriales para que sea posible ilustrar los detalles sin pérdida de calidad en medios digitales. De cualquier manera, el diseño del documento fue pensado para representar un soporte en papel por lo que en caso de ser necesario, es posible proveer las figuras en sus formatos originales. O, mejor aún, un conjunto de scripts que utilizando el código milonga, crean no sólo nuevamente las figuras desde cero sino que dan como resultado el capítulo 5 completo tal cual lo presentamos. De hecho fue ésta la forma en la cual lo generamos: un script ejecuta sucesivamente a una serie de programas—incluyendo las llamadas a milonga para resolver los problemas—y genera un input de \LaTeX listo para ser incluido como capítulo de en un documento maestro. Mientras tanto, milonga usa el código gmsh como mallador y genera datos que pueden ser post-procesados con el mismo gmsh o con herramientas de graficación como pyxplot o gnuplot. En todos los casos, se trata de software libre según la discusión anterior.



Por último, en el capítulo 6 discutimos las conclusiones que nos dejaron los tres capítulos anteriores y proponemos una lista de ejes sobre los cuáles deberíamos basar algunos de los trabajos futuros relacionados a resolver más y mejor algunas ecuaciones matemáticas con aplicación directa a problemas de interés industrial.

No es cierto que este trabajo venga a llenar ningún tipo de gap en la literatura ni a suplir deficiencias académicas de índole alguna. Ni siquiera estoy cierto de que las haya. Por el contrario, debe tomarse cabalmente como lo que es: una monografía final de una materia universitaria de introducción al método de elementos finitos a nivel de posgrado escrita por un estudiante que trabaja en la industria nuclear y que al mismo tiempo cursa un doctorado en background. Es por eso que solicito desde un principio piedad en los requerimientos técnicos y académicos por parte del lector. Cualquier enunciado que pertenezca a este texto debe ser entendido como dicho pidiéndole permiso a los que saben. Correcciones, sugerencias y todo tipo de comentarios son bienvenidos.

La ecuación de difusión de neutrones

Mathematics, rightly viewed, possesses not only truth,
but supreme beauty—a beauty cold and austere,
like that of sculpture, without appeal
to any part of our weaker nature,
without the gorgeous trappings of painting or music,
yet sublimely pure, and capable of a stern perfection
such as only the greatest art can show.

Bertrand Russel, The Study of Mathematics, 1902

El presente trabajo consiste básicamente en resolver en forma numérica un problema de autovalores formulado a través de una ecuación en derivadas parciales. Si bien es éste un trabajo estrictamente matemático, la ecuación continua que es luego discretizada y resuelta numéricamente tiene ciertas aplicaciones en ingeniería nuclear que hacen que los desarrollos matemáticos usuales se realicen de una cierta manera particular. Es por eso que para que aquellos lectores que no estén familiarizados con la neutrónica y el análisis de reactores de fisión observen cuáles son las motivaciones, la importancia, el significado físico y las razones de por qué desarrollamos la ecuación de cierta manera y no de otra, antes de pasar a introducir la ecuación y a explicar sus parámetros, la sección siguiente contiene un marco teórico a modo de pequeño prelude a las ecuaciones propiamente dichas. La deducción y el desarrollo de la ecuación central objeto de estudio desde primeros principios está fuera del alcance del trabajo. Los detalles físicos, matemáticos e ingenieriles se dan como referencias [11, 12, 13, 14].

2.1. Marco físico-matemático

El diseño y análisis de reactores nucleares de fisión se basa en estudiar las colisiones que se dan entre los neutrones y los materiales que componen el núcleo (*core*) del reactor. Los neutrones interactúan con los núcleos (*nuclei*) de los átomos de los materiales circundantes de una manera estocástica. Para fijar ideas, supongamos que si un neutrón colisiona con un núcleo de un átomo (*nucleus*), entonces tiene una probabilidad p de ser absorbido y $(1 - p)$ de dispersarse. Sin embargo, dada la enorme población de neutrones que se encuentra normalmente en los núcleos de los reactores nucleares—del orden de 10^{15} —es correcto considerar que se comportan de manera determinista y, del total de neutrones que colisionan con dicho núcleo, exactamente una fracción p son absorbidos y $(1 - p)$ son dispersados.

Un neutrón está caracterizado por su posición espacial $\mathbf{x} \in \mathbb{R}^3$, por su energía E y por la dirección Ω en la que viaja (figura 2.1). La probabilidad—técnicamente definida como una sección eficaz—de que dicho neutrón sufra cierto tipo de interacción con la materia también

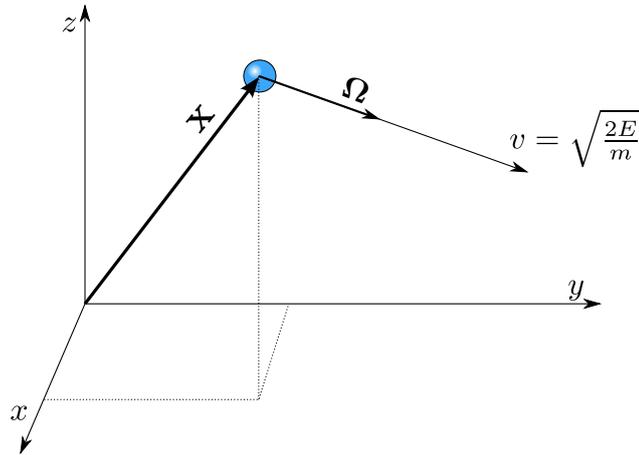


Figura 2.1 – Un neutrón clásico (bola celeste) se puede caracterizar por una posición \mathbf{x} en el espacio, una dirección de vuelo Ω y una energía E , que se relaciona con la velocidad de acuerdo a la fórmula clásica $E = mv^2/2$.

depende de la posición espacial de los núcleos blanco (el uranio ocupa un cierto espacio y el agua pesada otro), de la energía de los neutrones incidentes (el uranio fisiona con mucha más probabilidad si es bombardeado con neutrones lentos que con neutrones rápidos) y de la dirección de los neutrones incidentes, aunque en mucho menor medida.¹

En este sentido, es posible realizar un balance de neutrones con una idea similar a la ecuación de conservación de partículas de Maxwell para la teoría cinética de los gases, teniendo en cuenta los diferentes mecanismos particulares de aparición, desaparición y migración—tanto en el espacio como en el dominio de las energías—de neutrones. La ecuación de transporte resultante describe el comportamiento del campo de neutrones $\varphi(\mathbf{x}, E, \Omega, t)$ en función de siete variables independientes: tres para la posición espacial \mathbf{x} , una para la energía E , dos para la dirección Ω y eventualmente una para el tiempo t en problemas transitorios.

Este trabajo trata sólo con la obtención de estados estacionarios, por lo que en lo que resta del desarrollo no volvemos a hacer referencia al tiempo. Por otro lado, la dependencia en la dirección de vuelo Ω no suele interesar a efectos de los cálculos de reactor, ya que es suficiente conocer sólo los ritmos de reacción en función del espacio dados por el producto del flujo integrado $\phi(\mathbf{x}, E)$ por la sección eficaz macroscópica $\Sigma(\mathbf{x})$ correspondiente a la reacción de interés. Luego, si integramos de la ecuación estacionaria de transporte con respecto al ángulo sólido en 4π obtenemos una versión integral de la ecuación de Maxwell para el flujo integrado $\phi(\mathbf{x}, E)$. En verdad esta magnitud es un artefacto matemático que utilizamos como medio para calcular ritmos de reacción en función de la posición \mathbf{x} . De cualquier manera, esto implica que, a menos que utilicemos métodos continuos como Monte Carlo, debemos discretizar tanto el dominio espacial como el dominio de las energías para obtener resultados numéricos útiles.

Los parámetros que caracterizan la interacción de los neutrones con núcleo individuales se llaman secciones eficaces microscópicas, se simbolizan con la letra griega σ , tienen unidades área y dan una medida de la probabilidad que tiene un cierto núcleo blanco de interactuar con neutrones. Estas secciones eficaces dependen, para cada uno de los posibles isótopos blanco, de la energía del neutrón incidente y de la temperatura del material que es bombardeado. Debido a efectos cuánticos dentro de los núcleos, se producen resonancias a ciertas energías que

¹En la mayoría de los casos de aplicación de la ecuación de difusión de neutrones, incluyendo este trabajo, se asume que los materiales son isótropos y difunden de la misma manera en cualquier dirección. En este caso, se puede utilizar un coeficiente de difusión escalar. En caso de tener materiales anisótropos, se debe formular la ecuación utilizando un tensor de coeficientes de difusión.

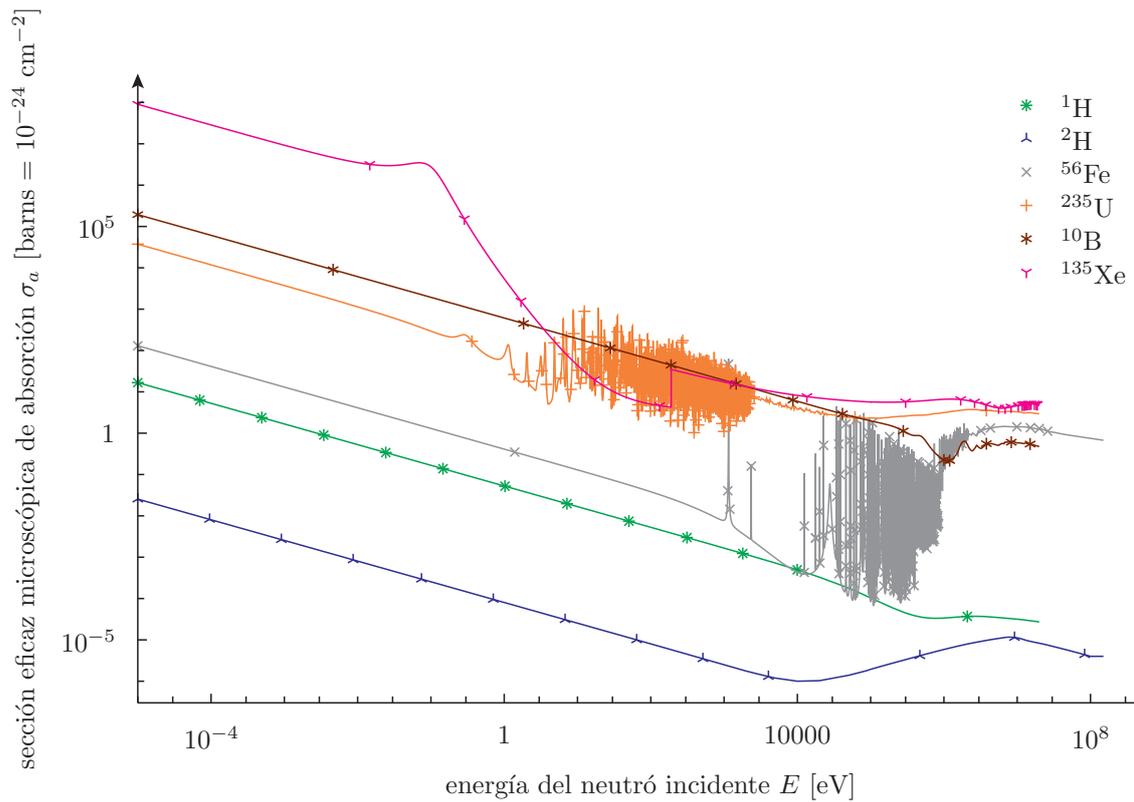


Figura 2.2 – Secciones eficaces microscópicas de absorción para algunos isótopos a 293 K de temperatura en función de la energía del neutrón incidente para el rango de energías típicas en un reactor de fisión térmico. Notar los órdenes de cambio de magnitud tanto para un mismo isótopo en función de la energía como para los diferentes isótopos a la misma energía. Un barn es igual a 10^{-24} cm^2 (as big as a barn!). Datos tomados de la referencia [15].

hacen que esta dependencia en energía y en temperatura sean bastante marcadas e irregulares. A modo de ilustración, la figura 2.2 muestra la sección eficaz microscópicas de absorción para algunos isótopos normalmente encontrados en el núcleo de un reactor nuclear. Existen secciones eficaces para todos los posibles tipos de interacciones neutrón-núcleo, tales como absorción total, absorción con emisión γ , absorción con emisión β , fisión, dispersión (*scattering*) elástica e inelástica, etc.

La ecuación de transporte de Maxwell no utiliza directamente las secciones eficaces microscópicas σ en su formulación, sino que aparecen secciones eficaces macroscópicas Σ que resultan de multiplicar la sección eficaz microscópica por la densidad volumétrica de núcleos blanco, obteniendo así unidades de longitud inversa para Σ .

Ahora bien, el hecho de que dentro del núcleo de un reactor nuclear conviven usualmente un gran número de materiales diferentes, cada uno de ellos normalmente constituido por diferentes clases de átomos—cuyas concentraciones van cambiando a medida que el uranio se quema y aparecen productos de fisión—y a distintas temperaturas, sumado a la compleja dependencia de las secciones eficaces con las variables independientes hace que el cálculo del flujos de neutrones $\phi(\mathbf{x}, E)$ requiera una discretización espacial y en energía tan fina que los sistemas discretos resultantes sean órdenes de magnitud más grandes que lo que pueden resolver hoy en día las más grandes supercomputadoras.² Por lo tanto, el cálculo se divide en dos

²Aunque por supuesto hay gente que lo intenta. Utilizando una supercomputadora con 150.000 núcleos [17] han logrado realizar este tipo de cálculos en Argonne National Laboratory. De todas maneras, la industria nuclear seguirá trabajando de la manera indicada en el presente trabajo por varios años más.



etapas. La primera, llamada cálculo de celda, consiste en obtener secciones eficaces macroscópicas homogeneizadas para ciertas celdas típicas [18]. Esto quiere decir, que suponemos que una celda bidimensional compuesta por ciertos materiales heterogéneos se comporta como si fuera un meta-material caracterizado por un juego de secciones eficaces macroscópicas—una Σ_a de absorción, una Σ_f de fisión, una Σ_s de scattering, etc. La figura 2.3 muestra la geometría de algunas celdas típicas para tres tipos de reactores.

El cálculo de celda utiliza ciertos métodos particulares para resolver el transporte de neutrones bidimensional dentro de la celda teniendo en cuenta bibliotecas de secciones eficaces microscópicas con una discretización fina en espacio (longitudes características del orden del tamaño de la geometría más pequeña que aparezca dentro de la celda) y en energía (del orden de cientos de grupos de energía). El output principal de este paso es la generación de secciones macroscópicas homogeneizadas en el espacio y con una discretización gruesa en energía, en función de las temperaturas y densidades de los diferentes materiales que componen las celdas, que alimentan la siguiente etapa llamada cálculo de núcleo.

En esta segunda etapa, se resuelve la ecuación de transporte—o alguna variación tal como la ecuación de difusión—utilizando las secciones eficaces macroscópicas homogeneizadas obtenidas en el cálculo de celda con discretizaciones gruesas tanto en espaciales (longitudes características del orden del tamaño de la celda) como en energía (un par de grupos de energía). Es sobre el cálculo de núcleo que se aplican los resultados objeto del este trabajo, así que a partir de ahora haremos especial énfasis en esta etapa del cálculo. Suponemos que el cálculo de celda ya fue realizado y todas las secciones eficaces macroscópicas necesarias para realizar el cálculo de núcleo son datos conocidos.

2.2. Difusión de neutrones

Antes de introducir la ecuación de difusión de neutrones continua tanto en el dominio espacial como en el de la energía, discutimos dos particularidades que posee esta ecuación y que influyen en las discretizaciones necesarias para su solución numérica, sea que utilicemos un esquema de volúmenes finitos o uno basado en elementos finitos.

La primera es que, en ausencia de fuentes de neutrones externas, los neutrones nacen solamente a partir de la fisión de ciertos isótopos—como por ejemplo ^{235}U o ^{239}Pu —que, además

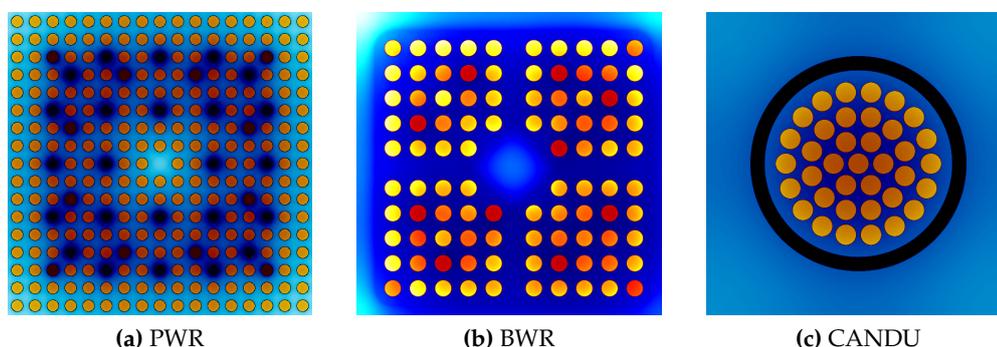


Figura 2.3 – Celdas típicas para un reactor tipo PWR, para un tipo BWR y para un CANDU. La salida del cálculo de celda consiste en secciones eficaces macroscópicas homogeneizadas para la celda completa en función de las temperaturas y densidades de los diferentes materiales que la componen. Imágenes tomadas del sitio web del código de celda basado en Monte Carlo Serpent [16].

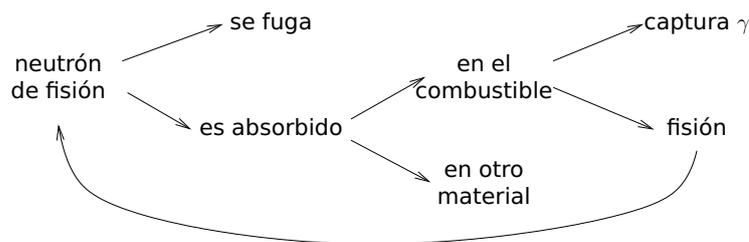


Figura 2.4 – Ilustración simplificada de la vida de un neutrón nacido por fisión. Para obtener un reactor crítico, por cada ν neutrones producidos por fisión, $\nu - 1$ deben fugarse o producir absorciones espurias y uno debe producir una nueva fisión.

de proveer la energía útil que luego se transformará en energía eléctrica en las centrales nucleares, genera una cierta cantidad $2 < \nu < 3$ de neutrones en promedio. Un neutrón nacido en una fisión, tiene una vida que esquematizamos en la figura 2.4: o bien se fuga o bien se absorbe dentro del núcleo. Si se absorbe, puede hacerlo en el combustible o en otro material. Si lo hace en el combustible, puede producir una captura radiativa o generar una nueva fisión. Para lograr un estado estacionario, es necesario que de los ν neutrones nacidos en una fisión, *exactamente* $\nu - 1$ se fuguen o sean absorbidos en forma espuria para que *exactamente* uno genere una nueva fisión. En este estado, decimos que el reactor está crítico y la población neutrónica se mantiene constante en el tiempo, dando lugar a una solución estacionaria.

La consecuencia matemática de esta particularidad física es que, dada una geometría y una cierta distribución de materiales, la probabilidad de que el reactor resultante esté crítico es cero. Una forma de lidiar con este inconveniente es trabajar con lo que se llama el reactor crítico asociado en k , que consiste en dividir todas las fuentes de fisión por un número real k_{eff} —que es también una de las incógnitas—de forma tal de forzar el estado estacionario. De esta manera, si resulta $k_{\text{eff}} < 1$ el reactor real es subcrítico y si $k_{\text{eff}} > 1$ el reactor real es supercrítico. Sin embargo, en cualquier caso, el reactor asociado siempre resulta crítico y es posible encontrar una solución estacionaria para cualquier geometría y distribución de materiales.

La segunda particularidad es que la ecuación de transporte de neutrones tiene una estructura similar a la ecuación de conducción de calor en el sentido de que se trata de una ecuación de conservación de una magnitud principal (la población de neutrones en un caso y la temperatura en el otro) involucrando una magnitud derivada (la corriente neta de neutrones en un caso y el flujo de calor en el otro), por lo que se obtienen dos incógnitas con una sola ecuación. Al dar una relación explícita entre la magnitud principal y la derivada, obtenemos la segunda ecuación necesaria para arribar a un problema matemático bien definido.

En el caso de la ecuación de difusión de neutrones, se propone utilizar una relación similar a la ley de Fick para la difusión de especies químicas en un medio, donde se relaciona la corriente vectorial neta de neutrones $\mathbf{J}(\mathbf{x}, E)$ con el gradiente del flujo integrado $\phi(\mathbf{x}, E)$ mediante un coeficiente de difusión $D(\mathbf{x}, E)$

$$\mathbf{J}(\mathbf{x}, E) = -D(\mathbf{x}, E) \cdot \nabla \phi(\mathbf{x}, E) \quad (2.1)$$

donde el operador gradiente opera sólo sobre las variables espaciales y cuya validez se discute ampliamente en la bibliografía de física de neutrones [19, 20, 21].

Luego de haber discutido esta dos particularidades, introducimos a continuación la ecuación continua de difusión neutrones en estado estacionario, que es la ecuación central del trabajo y es la cual discretizamos y resolvemos en forma numérica:



$$0 = \text{div} \left[D(\mathbf{x}, E) \cdot \text{grad} [\phi(\mathbf{x}, E)] \right] - \Sigma_t(\mathbf{x}, E) \cdot \phi(\mathbf{x}, E) + \int_0^\infty \Sigma_s(\mathbf{x}, E' \rightarrow E) \cdot \phi(\mathbf{x}, E') dE' + \chi(E) \int_0^\infty \frac{\nu \Sigma_f(\mathbf{x}, E')}{k_{\text{eff}}} \cdot \phi(\mathbf{x}, E') dE' \quad (2.2)$$

Nuevamente, los operadores divergencia y gradiente operan sobre las coordenadas espaciales de las funciones argumento. En forma muy simplificada, la explicación de los términos y parámetros que aparecen es la que sigue. Para un mayor detalle, referirse a Henry [11], Glassstone y Bell [12] y Duderstadt y Hamilton [13].

El primer término es el llamado término de fugas, y es el que le da el nombre a la ecuación. Involucra al coeficiente de difusión $D(\mathbf{x}, E)$ y al gradiente del flujo $\phi(\mathbf{x}, E)$. El segundo término es el ritmo total de reacción, igual al producto entre la sección eficaz macroscópica total $\Sigma_t(\mathbf{x}, E)$ y el flujo $\phi(\mathbf{x}, E)$. El tercer término es la fuente de neutrones debido a dispersiones provenientes de otras energías, y se obtiene de integrar la sección eficaz de scattering $\Sigma_s(\mathbf{x}, E' \rightarrow E)$ que matemáticamente es un núcleo (*kernel*) e indica la probabilidad de que un neutrón de energía E' quede con una energía E luego de una colisión.³ Finalmente, el último término representa la fuente de neutrones de fisión. Como hemos discutido, se divide la sección eficaz de fisión $\nu \Sigma_f(\mathbf{x}, E)$ —que ya tiene en cuenta el número promedio de neutrones ν que se generan por cada fisión individual—por un número real k_{eff} , que junto a $\phi(\mathbf{x}, E)$ es una de las incógnitas. El ritmo de fisión se integra en todas las energías y el espectro de fisión $\chi(E)$ indica qué fracción del total nace en la energía E .⁴

Tal como está escrita, la ecuación (2.2) es una ecuación lineal ya que los coeficientes sólo dependen de las variables independientes \mathbf{x} y E —y no del flujo $\phi(\mathbf{x}, E)$ mismo. En la mayoría de los casos de aplicación ingenieril, esto no es cierto y la ecuación se transforma en no-lineal, dado que suele suceder que las secciones eficaces dependan de la temperatura local de los materiales, que a su vez dependen de la potencia local generada, que depende del flujo neutrónico, que nuevamente depende de la sección eficaz. El otro caso típico de no linealidad involucra el isótopo ^{135}Xe , que es un producto de fisión y como tal, su concentración depende del nivel de flujo. Pero como ilustramos en la figura 2.2, una pequeña cantidad de xenón puede generar grandes cambios en las secciones eficaces macroscópicas, por lo que se obtiene una ecuación donde los coeficientes que la definen dependen de la solución misma requiriendo una solución iterativa. Sin embargo, se puede mostrar que es posible tomar la ecuación (2.2) como lineal, al menos en cada una de las iteraciones de un cálculo iterativo [7, cap. 2]. Por lo tanto, estudiar la ecuación de difusión lineal tiene una aplicación práctica incluso aún en casos donde no es aplicable directamente. De todas maneras, para el alcance de este trabajo vamos a suponer en todos los casos que las secciones eficaces dependen solamente de la posición y de la energía en una forma conocida, dando lugar a una ecuación lineal.

Por otro lado, el hecho de que la ecuación (2.2) además de lineal sea homogénea y con condiciones de contorno homogéneas (ver sección 2.2.2), es decir sin términos independientes, hace que si $\phi_1(\mathbf{x}, E)$ es una solución, entonces $\phi_2(\mathbf{x}, E) = \alpha \cdot \phi_1(\mathbf{x}, E)$ también lo sea. Luego, la solución final queda determinada a menos de una constante multiplicativa. Esta característica matemática implica, además de ciertos efectos físicos e ingenieriles que no vienen al caso traer a colación, que es necesario normalizar de alguna manera las soluciones para, primero obtener

³En el sentido matemático estricto deberíamos hablar de distribuciones y tomar intervalos $[E', E' + dE']$ y $[E, E + dE]$ para estudiar densidades de ritmos de reacción, pero no es la intención de este capítulo incurrir en complicaciones que no son necesarias para el objeto del trabajo. En el capítulo dos de la documentación del código milonga [7] se dan ciertos detalles pueden ser complementarias a esta sección.

⁴*Ibid*



Figura 2.5 – Partición del rango de energía $0 < E < E_0$ en G grupos discretos. Figura tomada de la documentación del código milonga [7]

resultados útiles y segundo, para poder comparar soluciones obtenidas con diferentes métodos o discretizaciones, que es justamente el objetivo de este trabajo.

2.2.1. Grupos de energía discreta

Siendo la ecuación (2.2) una expresión en función del espacio \mathbf{x} y la energía E , es necesario discretizar ambas variables para resolver numéricamente la ecuación utilizando elementos o volúmenes finitos. Si bien desde el punto de vista matemático tanto la energía como el espacio son dos dominios equivalentes, el hecho de separar la solución en cálculo de celda y de núcleo hace que los códigos de cada etapa deban utilizar discretizaciones compatibles. Luego, aunque se pueda aplicar el método de elementos finitos para resolver la dependencia con la energía E , en la práctica se utiliza en los códigos de núcleo una discretización que resulta conveniente para los códigos de celda ya que mantiene los ritmos de reacción de las ecuaciones continuas en las ecuaciones discretas. Este proceso consiste en dividir el dominio de la energía en G grupos no necesariamente equiespaciados tales que $0 = E_G < E_{G-1} < \dots < E_1 < E_0$, donde E_0 es la mayor energía esperada. El grupo de energía g es el definido por el intervalo $[E_g, E_{g-1}]$ (figura 2.5).

Siguiendo un desarrollo que se puede encontrar en las referencias [7, 11], podemos obtener la ecuación de difusión de neutrones multigrupo estacionaria

$$\begin{aligned}
 0 = \operatorname{div} \left[D_g(\mathbf{x}) \cdot \operatorname{grad} [\phi_g(\mathbf{x})] \right] - \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) \\
 + \sum_{g'=1}^G \Sigma_{sg' \rightarrow g}(\mathbf{x}) \cdot \phi_{g'}(\mathbf{x}) + \chi(g) \sum_{g'=1}^G \frac{\nu \Sigma_{fg'}(\mathbf{x})}{k_{\text{eff}}} \cdot \phi_{g'}(\mathbf{x})
 \end{aligned} \quad (2.3)$$

que es la que vamos a resolver utilizando esquemas de volúmenes y elementos finitos en la coordenada espacial $\mathbf{x} \in U \in \mathbb{R}^M$. La variable independiente g es un índice entero entre 1 y G inclusive, e indica la naturaleza vectorial de la expresión (2.3). Es decir, en verdad la ecuación de difusión multigrupo son G ecuaciones diferenciales en derivadas parciales sobre el espacio acopladas. Consideramos a los parámetros $D_g(\mathbf{x})$, $\Sigma_{tg}(\mathbf{x})$, $\Sigma_{sg' \rightarrow g}(\mathbf{x})$, $\chi(g)$ y $\nu \Sigma_{fg}(\mathbf{x})$ funciones conocidas para todo $\mathbf{x} \in \mathbb{R}^M$, $g = 1, \dots, G \in \mathbb{N}$ y $g' = 1, \dots, G \in \mathbb{N}$. Las incógnitas del problema son por un lado los G flujos $\phi_g(\mathbf{x})$ sobre algún dominio M -dimensional para $M = 1, 2$ ó 3 y por otro el factor de multiplicación efectivo $k_{\text{eff}} \in \mathbb{R}$.

La ecuación (2.3) es la formulación diferencial o *fuerte* del problema de difusión de neutrones. Puesta de esta manera, la ecuación requiere que la solución sea dos veces diferenciable en el dominio U . De ahí la taxonomía de fuerte, ya que es ésta una imposición bastante restrictiva dado que si el dominio U está compuesto por materiales diferentes—cómo usualmente lo está—es de esperar que las secciones eficaces macroscópicas sean discontinuas en las interfaces. En este caso, la divergencia del primer término del miembro derecho de la ecuación (2.3) no está definida, y no es posible aplicar la ecuación de difusión en los puntos ubicados sobre

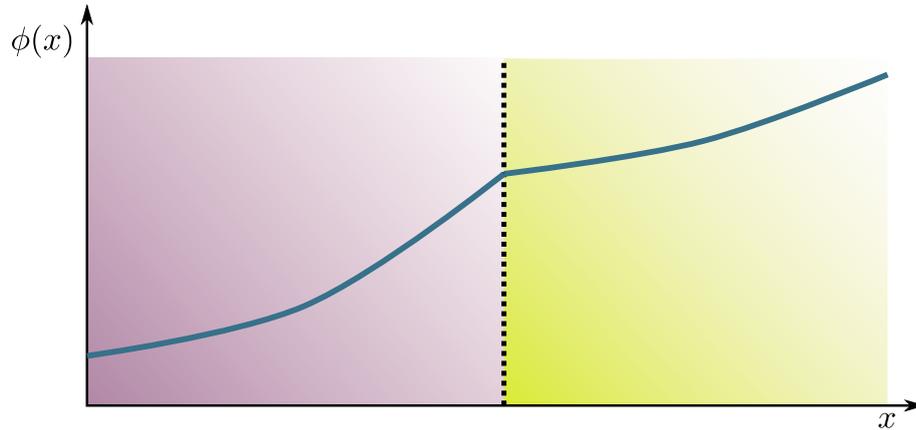


Figura 2.6 – Si dos materiales tienen coeficientes de difusión diferentes en la interfaz, la conservación de corrientes implica que el flujo no es derivable en dicha interfaz.

la interfaz. Se reemplaza entonces, la ecuación de difusión por la condición de que la corriente neta de neutrones calculada con la ecuación (2.1) sea continua sobre la superficie interfaz

$$D_g(\mathbf{x}) \cdot \text{grad } \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in S^+} = D_g(\mathbf{x}) \cdot \text{grad } \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in S^-} \quad (2.4)$$

donde S^+ y S^- representan los dos lados de la superficie interfaz. Notar que si los coeficientes de difusión de ambos materiales son diferentes como se espera que sean, entonces los gradientes a ambos lados de la superficie S también lo serán y los flujos no son dos veces diferenciables (figura 2.6). Este requerimiento matemático implica físicamente que todos los neutrones que atraviesan la interfaz de un lado aparecen por el otro sin generarse o perderse neutrones en el proceso.

2.2.2. Condiciones de contorno

La ecuación (2.3) necesita condiciones de contorno sobre el borde ∂U del dominio espacial U para plantear un problema matemático bien definido. Éstas pueden ser condiciones sobre el flujo (Dirichlet), sobre la corriente (Neumann) o una combinación lineal de ambas (Robin). Más aún, puede darse un tipo de condición de contorno sobre una parte del borde del dominio ∂U_a y otro tipo sobre otra parte ∂U_b de tal forma que $\partial U = \partial U_a \cup \partial U_b$. Sin embargo, en cada porción del borde del dominio, todos los grupos tienen que tener el mismo tipo de condición de contorno. Es decir, si en un cierto punto $\mathbf{x} \in \partial U$ se especifica que $\phi_1 = 0$, no se puede pedir que $\partial \phi_2 / \partial n$ sea cero.

La condición de contorno más general que se puede dar es el valor que toma una cierta combinación lineal del flujo y de la derivada en el borde del dominio

$$u_g(\mathbf{x}) \cdot \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U} + v_g(\mathbf{x}) \cdot \frac{\partial \phi_g}{\partial n} \Big|_{\mathbf{x} \in \partial U} = w_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U}$$

Para un g fijo, de los tres coeficientes reales u_g , v_g y w_g asociados, sólo dos son independientes. Además, el valor w_g que toma esta combinación lineal debe ser igual a cero, ya que la ecuación de difusión sin fuente es homogénea y su solución está definida a menos de una constante. Luego, esta expresión se reduce a

$$\frac{\partial \phi_g}{\partial n} \Big|_{\mathbf{x} \in \partial U} = b_g(\mathbf{x}) \cdot \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U}$$

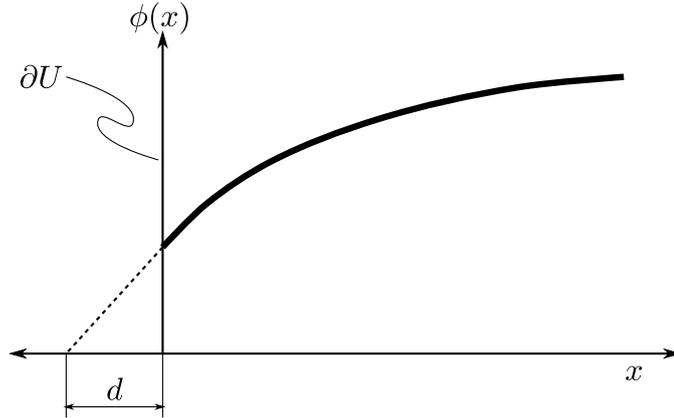


Figura 2.7 – Equivalencia entre la condición de corriente entrante nula dada por la ecuación (2.5) y la condición de flujo nulo en la longitud extrapolada d dada por la ecuación (2.6). Si $d \ll D$, entonces podemos asumir $d = 0$ y tomar una condición de contorno de Dirichlet $\phi(x) = 0$ en lugar de requerir una condición de Robin.

para algún coeficiente b_g . Si $b_g = 0$, obtenemos una condición de contorno de Neumann $\partial\phi_g/\partial n = 0$. Si $b_g = \infty$, el problema tiene una condición de Dirichlet $\phi_g = 0$ y en cualquier otro caso, la condición de contorno es de tipo Robin.

Físicamente, se supone que los neutrones que salen de un dominio convexo U no pueden volver a ingresar al mismo. Según la aproximación de difusión, la corriente parcial que ingresa a una superficie es [21]

$$J_g^-(\mathbf{x}) \approx \frac{1}{4} \cdot \phi_g(\mathbf{x}) + \frac{1}{2} \cdot D_g(\mathbf{x}) \cdot \frac{\partial\phi_g}{\partial n}$$

Si requerimos que esta corriente sea nula, debe ser

$$\left. \frac{\partial\phi_g}{\partial n} \right|_{\mathbf{x} \in \partial U} = -\frac{1}{2D_g(\mathbf{x})} \cdot \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U} \quad (2.5)$$

con lo que $b_g = -(2D_g)^{-1}$. Si extrapolamos linealmente el flujo $\phi_g(\mathbf{x})$ hacia la dirección de la normal externa a la frontera ∂U mediante una expansión de Taylor a primer orden

$$\phi_g(\mathbf{x} + d_g(\mathbf{x}) \cdot \hat{\mathbf{n}}) \Big|_{\mathbf{x} \in \partial U} \approx \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U} + \frac{\partial\phi_g}{\partial n} \cdot d_g(\mathbf{x}) \quad (2.6)$$

vemos que si

$$d_g(\mathbf{x}) = 2D_g(\mathbf{x}) = -\frac{1}{b_g(\mathbf{x})}$$

entonces la extrapolación del flujo se anula a una distancia d de la frontera (figura 2.7). Es decir, la condición de contorno (2.5) es equivalente, a primer orden, a la condición

$$\phi(\mathbf{x} + d_g(\mathbf{x}) \cdot \hat{\mathbf{n}}, g) \Big|_{\mathbf{x} \in \partial U} = 0 \quad (2.7)$$

donde $\hat{\mathbf{n}}$ es el vector normal unitario exterior a la superficie ∂U . En este caso, $d_g(\mathbf{x})$ es la llamada longitud de extrapolación del grupo g , y dependiendo de cómo se realice la estimación de la corriente parcial J^- , puede tomar diferentes valores en el rango

$$D_g(\mathbf{x}) < d_g(\mathbf{x}) < 3D_g(\mathbf{x})$$



Dado que usualmente los coeficientes de difusión son sensiblemente menores que las distancias características que determinan el tamaño del dominio U , se suele aproximar aún más esta condición de contorno y suponer que $d_g(\mathbf{x}) = 0$ para requerir directamente

$$\phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U} = 0$$

cambiando así una condición de contorno de Robin por una condición de Dirichlet.

Condiciones de reflectividad dadas por un coeficiente β_g de albedo también pueden definirse como condiciones de contorno de Robin, definiendo el coeficiente $b_g(\mathbf{x})$ como [21]

$$b_g(\mathbf{x}) = -\frac{1}{2D_g(\mathbf{x})} \frac{1 - \beta_g}{1 + \beta_g}$$

En el caso particular en el que una parte de la frontera ∂U represente una superficie de simetría, entonces $\beta_g = 1$ para todos los grupos g y esta expresión se transforma en una condición de contorno de Neumann de derivada normal nula

$$\frac{\partial \phi_g}{\partial n} \Big|_{\mathbf{x} \in \partial U} = 0$$

Esta condición se suele aplicar a superficies planas de simetría, pero puede aplicarse también a superficies curvas.

En aras de la simplicidad, dado que el coeficiente $b_g(\mathbf{x})$ de las condiciones de contorno discutidas involucra la inversa del coeficiente de difusión del grupo g , utilizamos un nuevo coeficiente para definir condiciones de contorno de Robin

$$a_g(\mathbf{x}) = b_g(\mathbf{x}) \cdot D_g(\mathbf{x})$$

donde para la mayoría de las aplicaciones prácticas de la ecuación de difusión, el coeficiente a no depende ni de la posición \mathbf{x} ni del grupo g y vale $-1/2$ para superficies externas.

Resumiendo, las condiciones de contorno puede ser:

Dirichlet sobre superficies externas se requiere flujo nulo, siempre y cuando las dimensiones del dominio U sean mucho mayores que los coeficientes de difusión

Neumann sobre superficies de simetría se requiere derivada normal nula

Robin sobre superficie externas con condiciones de corriente entrante nula o de reflectividad dada por factores de albedo se requiere un factor que relacione el flujo con la derivada normal

2.2.3. El problema de autovalores

Las G ecuaciones lineales (2.3) tienen la característica de que se pueden escribir en forma matricial en función del vector incógnita

$$\phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_G(\mathbf{x}) \end{bmatrix}$$

como



$$\mathcal{R}(\mathbf{x}) \cdot \phi(\mathbf{x}) = \frac{1}{k_{\text{eff}}} \mathcal{F}(\mathbf{x}) \cdot \phi(\mathbf{x}) \quad (2.8)$$

donde \mathcal{R} y \mathcal{F} son matrices cuadradas de tamaño $G \times G$, cuyos elementos son o bien funciones u operadores diferenciales definidos sobre algún dominio $U \in \mathbb{R}^M$.

Puesta en esta forma, la ecuación de difusión multigrupo continua en el espacio más sus condiciones de contorno define un problema de autovalores y autovectores generalizado. En efecto, podemos escribir la ecuación (2.8) como un problema de autovalores estándar como

$$[\mathcal{R}^{-1}(\mathbf{x}) \cdot \mathcal{F}(\mathbf{x})] \cdot \phi(\mathbf{x}) = k_{\text{eff}} \cdot \phi(\mathbf{x}) \quad (2.9)$$

Es posible demostrar que \mathcal{R}^{-1} existe para cualquier conjunto físicamente posible de secciones eficaces [11], por lo que el problema de autovalores—tanto definido por la ecuación (2.8) como por la ecuación (2.9)—está bien definido. Ahora, en general existen G pares de autovalores y autovectores que satisfacen la formulación problema. Puede darse el caso de que muchos de los autovalores sean negativos, complejos o den lugar a flujos negativos. Sin embargo, para configuraciones físicamente realizables, es posible demostrar [11, sección 3.3] que para el problema estándar (2.9) se cumplen las siguiente propiedades

1. Hay un único autovalor positivo real que es mayor en magnitud que el resto de los autovalores
2. Todos los elementos del autovector correspondiente a dicho autovalor son reales y tienen el mismo signo
3. Todos los otros autovectores o bien tienen al menos un elemento igual a cero o tienen elementos que difieren en su signo

Es justamente este autovalor el factor de multiplicación k_{eff} buscado, y los G elementos de su correspondiente autovector los flujos solución del problema de difusión de neutrones estacionario del reactor crítico asociado en k . Como ya hemos discutido, la naturaleza homogénea del problema hace que el flujo neutrónico esté definido a menos de una constante multiplicativa, que es una propiedad conocida del problema de autovectores.

Para fijar ideas, supongamos que $G = 2$. En este caso, el flujo del grupo uno se llama flujo rápido y el flujo del grupo dos se llama flujo térmico. Entonces la matriz de remoción es

$$\mathcal{R} = \begin{bmatrix} -\nabla[D_1 \cdot \nabla] + \Sigma_{t1} - \Sigma_{s1 \rightarrow 1} & -\Sigma_{s2 \rightarrow 1} \\ -\Sigma_{s1 \rightarrow 2} & -\nabla[D_2 \cdot \nabla] + \Sigma_{t2} - \Sigma_{s2 \rightarrow 2} \end{bmatrix}$$

y la matriz de fisión es

$$\mathcal{F} = \begin{bmatrix} \chi_1 \cdot \nu \Sigma_{f1} & \chi_1 \cdot \nu \Sigma_{f2} \\ \chi_2 \cdot \nu \Sigma_{f1} & \chi_2 \cdot \nu \Sigma_{f2} \end{bmatrix}$$

donde todos los elementos de las matrices son funciones de la posición \mathbf{x} . Notar que la simetría en la notación no implica simetría en las matrices.

Usualmente no se tiene upscattering $\Sigma_{s2 \rightarrow 1} = 0$, todas las fisiones se producen por el flujo térmico $\nu \Sigma_{f1} = 0$ y todos los neutrones de fisión nacen rápidos $\chi_1 = 1$ y $\chi_2 = 0$. Además, para el grupo g , la diferencia entre la sección eficaz total y la de self-scattering es la sección eficaz de remoción



$$\Sigma_{rg}(\mathbf{x}) = \Sigma_{tg}(\mathbf{x}) - \Sigma_{sg \rightarrow g}(\mathbf{x}) = \Sigma_{ag}(\mathbf{x}) + \sum_{g' \neq g}^G \Sigma_{sg \rightarrow g'}(\mathbf{x}) \quad (2.10)$$

Luego, el problema multigrupo continuo se simplifica a

$$\begin{bmatrix} -\nabla[D_1 \cdot \nabla] + \Sigma_{r1} & 0 \\ -\Sigma_{s1 \rightarrow 2} & -\nabla[D_2 \cdot \nabla] + \Sigma_{r2} \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \end{bmatrix} = \frac{1}{k_{\text{eff}}} \begin{bmatrix} 0 & \nu \Sigma_{f2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \end{bmatrix}$$

2.3. La ecuación de difusión discreta

Al discretizar el dominio espacial utilizando algún esquema numérico, en esencia lo que hacemos es representar la dependencia espacial de cada uno de los flujos $\phi_g(\mathbf{x})$ mediante un conjunto finito de pares de valores $[\mathbf{x}_i, \phi_g^i]$ para $i = 1, \dots, I$ sobre una malla de tamaño I . El problema central de la discretización de la ecuación de difusión de neutrones multigrupo—y por cierto de este trabajo—es entonces encontrar entonces dos matrices R y F de tamaño $IG \times IG$ tales que

$$R \cdot \hat{\phi} = \frac{1}{k_{\text{eff}}} \cdot F \cdot \hat{\phi} \quad (2.11)$$

donde el vector de incógnitas discretas es

$$\hat{\phi} = \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_2^1 \\ \vdots \\ \hat{\phi}_G^1 \\ \hat{\phi}_1^2 \\ \vdots \\ \hat{\phi}_G^2 \\ \hat{\phi}_1^3 \\ \vdots \\ \hat{\phi}_G^I \end{bmatrix} \quad (2.12)$$

A lo largo del trabajo denotamos con el sombrero $\hat{}$ los valores asociados a una celda en volúmenes finitos y a un nodo elementos finitos. El subíndice indica el grupo de energía $g = 1, \dots, G$ y el supraíndice el número global de incógnita ($i = 1, \dots, I$ para volúmenes finitos y $k = 1, \dots, K$ para elementos finitos).

Es de esperar⁵ que esquemas numéricos consistentes, convergentes y estables las tres propiedades del problema de autovalores continuo discutidas en la sección 2.2.3 se trasladen al problema discreto. Para tamaños de problema IG de interés—del orden de centenas de miles de incógnitas—es prohibitivo trabajar con el problema de autovalores estándar dado por la ecuación (2.9) ya que no sólo involucra el cálculo de la inversa de una matriz sino que tanto R como F son ralas, mientras que ni R^{-1} ni $R^{-1}F$ lo son. Por lo tanto, la resolución del problema estacionario de difusión de neutrones implica el cálculo del autovalor—y su correspondiente autovector—de menor valor absoluto utilizando la ecuación (2.11), o bien el autovalor—y su correspondiente autovector—de mayor valor absoluto de la ecuación

⁵Me falta una demostración rigurosa de esta afirmación.



$$F \cdot \hat{\phi} = k_{\text{eff}} \cdot R \cdot \hat{\phi} \quad (2.13)$$

En el capítulo 3 utilizamos un esquema basado en volúmenes finitos para encontrar R y F a partir de \mathcal{R} y \mathcal{F} , mientras en que el capítulo 4 hacemos lo propio aplicando el método de elementos finitos. Dependiendo de las características de las matrices involucradas y del método numérico utilizado para obtener la solución, en ciertos casos es conveniente utilizar la formulación dada por la ecuación (2.11) y en otros la dada por la ecuación (2.13). La discusión y análisis de la forma de calcular k_{eff} y ϕ una vez obtenidas las matrices R y F están fuera del alcance de este trabajo y se da como referencia [22, 10, 23]. Sin embargo, es propicio notar lo que sigue.

Según la documentación de la biblioteca SLEPc—que es la utilizada por el código milonga para resolver el problema de autovalores generalizados—todos los eigensolvers provistos son de naturaleza iterativa, lo que quiere decir que (usualmente) las soluciones mejoran en cada paso hasta que son lo suficientemente exactas como para considerarse convergidas. Esta condición de convergencia se define a través de una tolerancia sobre la norma de algún residuo.

El residuo más sencillo es el absoluto, que obtenemos restando los dos miembros de la ecuación (2.11). Dado que en el paso n ni el autovalor $1/k_{\text{eff}}^{(n)}$ ni el autovector $\hat{\phi}^{(n)}$ son iguales a la solución exacta, esta diferencia no es cero sino que es igual a un vector $\mathbf{r}^{(n)}$ que llamamos residuo absoluto

$$\mathbf{r}^{(n)} = \left[R - \frac{1}{k_{\text{eff}}^{(n)}} \cdot F \right] \cdot \hat{\phi}^{(n)}$$

La norma de este vector depende del tamaño de las matrices R y F ya que mientras más grandes sean éstas, más elementos va a tener $\mathbf{r}^{(n)}$. Es decir, si bien al aumentar la nodalización es de esperar que cada elemento del residuo absoluto sea más pequeño, hay más elementos en el residuo y su norma puede aumentar. Es por ello que no es una buena idea definir el criterio de convergencia de la solución del problema de autovalores en función de $\|\mathbf{r}^{(n)}\|$. Una medida apropiada para definir el criterio de convergencia sería calcular la integral en el dominio del problema de la ecuación de difusión multigrupo (2.3)

$$\int_U \left\{ \text{div} \left[D_g(\mathbf{x}) \cdot \text{grad} \left[\tilde{\phi}_g^{(n)}(\mathbf{x}) \right] \right] - \Sigma_{tg}(\mathbf{x}) \cdot \tilde{\phi}_g^{(n)}(\mathbf{x}) \right. \\ \left. + \sum_{g'=1}^G \Sigma_{sg' \rightarrow g}(\mathbf{x}) \cdot \tilde{\phi}_{g'}^{(n)}(\mathbf{x}) + \chi(g) \sum_{g'=1}^G \frac{\nu \Sigma_f(\mathbf{x}, g')}{k_{\text{eff}}} \cdot \tilde{\phi}_{g'}^{(n)}(\mathbf{x}) \right\} d^D \mathbf{x}$$

donde $\tilde{\phi}_g^{(n)}(\mathbf{x})$ es una función continua definida a partir de la solución numérica discreta calculada en el paso p . El valor absoluto de esta integral debería ser independiente del tamaño de la malla espacial. Sin embargo, el cálculo de esta integral es muy demandante, y la evaluación de la condición de convergencia sería computacionalmente muy cara. El código milonga entonces utiliza como medida para comparar contra la tolerancia de convergencia el residuo relativo a las normas de las matrices, definido como

$$\delta^{(n)} = \frac{\|\mathbf{r}^{(n)}\|}{\|R\| + 1/k_{\text{eff}} \cdot \|F\|}$$

que es fácil de evaluar a partir de los estados internos de la biblioteca SLEPc y que además no es tan dependiente de la malla como la normal vector absoluto $\|\mathbf{r}^{(n)}\|$ ya que tanto numerador



como denominador varían con la cantidad de incógnitas de una forma más o menos similar. De esta forma, consideramos la solución del problema de autovectores como convergido si para algún paso p se cumple que

$$\delta^{(n)} < \epsilon$$

para alguna tolerancia ϵ prefijada, del orden de 10^{-8} .

Quedan fuera del alcance de este trabajo otros temas computacionales de vital importancia, tales como la lectura e interpretación de la malla, la detección y resolución de volúmenes vecinos, la lectura de las distribuciones espaciales de secciones eficaces—que en general podrían ser dadas por expresiones algebraicas en función de x , y y z o interpoladas a partir de datos discretos, eventualmente provenientes de otros códigos de cálculo en esquemas acoplados—la forma de almacenar matrices ralas de tamaños de orden $10^6 \times 10^6$, los métodos de integración numérica espacial y la evaluación y presentación de los resultados. Todos estos aspectos son tenidos en cuenta por el código neutrónico milonga desarrollado por el autor de este trabajo en el marco de su tesis de doctorado, pero no son descriptos en esta monografía ya que no están directamente relacionado con la comparación entre esquemas basados en volúmenes y en elementos.

Volúmenes finitos

No me asusta esta complicada integral del núcleo de scattering.
Lo que me asusta es el pequeño término de la divergencia.

*Javier Fernandez, Doctor en Matemática,
luego de mirar por primera vez la ecuación
de difusión de neutrones, 2006*

En este capítulo proponemos un esquema numérico basado en volúmenes finitos para resolver la ecuación de difusión de neutrones multigrupo. Existen varios esquemas posibles, algunos más eficientes que otros para cierto tipo de problemas. La discretización propuesta es sólo una de las tantas formas de encarar el problema utilizando volúmenes finitos. En particular, si dos celdas vecinas están compuestas del mismo material entonces asumimos que, si bien el coeficiente de difusión puede variar con la posición, lo hace en forma continua. Luego, el flujo en la superficie que separa ambas celdas puede ser aproximado por el promedio de los flujos en las celdas y mantener la condición de continuidad de corriente al mismo tiempo. Por el contrario, en caso de que dos celdas adyacentes estén compuestas de materiales diferentes, esperamos que el coeficiente de difusión tenga una discontinuidad en la interfaz. Es por ello, que el flujo en la superficie debe ser calculado de una cierta manera particular para satisfacer la conservación de corriente de neutrones, ya que en estas condiciones su derivada no está definida (pero las corrientes sí). En el primer caso, los coeficientes de la matriz resultantes son más fáciles de calcular y, además, la precisión de la aproximación del operador diferencial es un orden de magnitud mayor que en el segundo caso. Sin embargo, debemos tratar el caso de celdas heterogeneas en forma particular, como ilustramos en el ejemplo de la sección 5.1.2.2 donde observamos que no tener en cuenta este caso conduce a resultados incorrectos.

En general, tanto en este como en el capítulo siguiente, los subíndices están indicados con letras occidentales minúsculas (como por ejemplo i para indicar la celda i -ésima) y las cantidades totales con la correspondiente letra mayúscula (I indica la cantidad total de celdas).

3.1. Esquema basado en volúmenes

La idea básica del método de los volúmenes finitos es tomar el dominio continuo $U \in \mathbb{R}^M$ donde aplica la ecuación en derivadas parciales a resolver, y aproximarlos mediante una yuxtaposición de una cierta cantidad finita I de celdas M -dimensionales definidas por bordes planos, i.e. segmentos en una dimensión, polígonos en dos dimensiones y poliedros en tres dimensiones. En el contexto de volúmenes finitos, la celda i -ésima corresponde a un volumen $V_i \in \mathbb{R}^M$. Los vértices de las I celdas definen K nodos. En la figura 3.1 mostramos un ejemplo de aproximación de un dominio $U \in \mathbb{R}^2$ con celdas cuadrangulares. El análisis del problema del mallado de

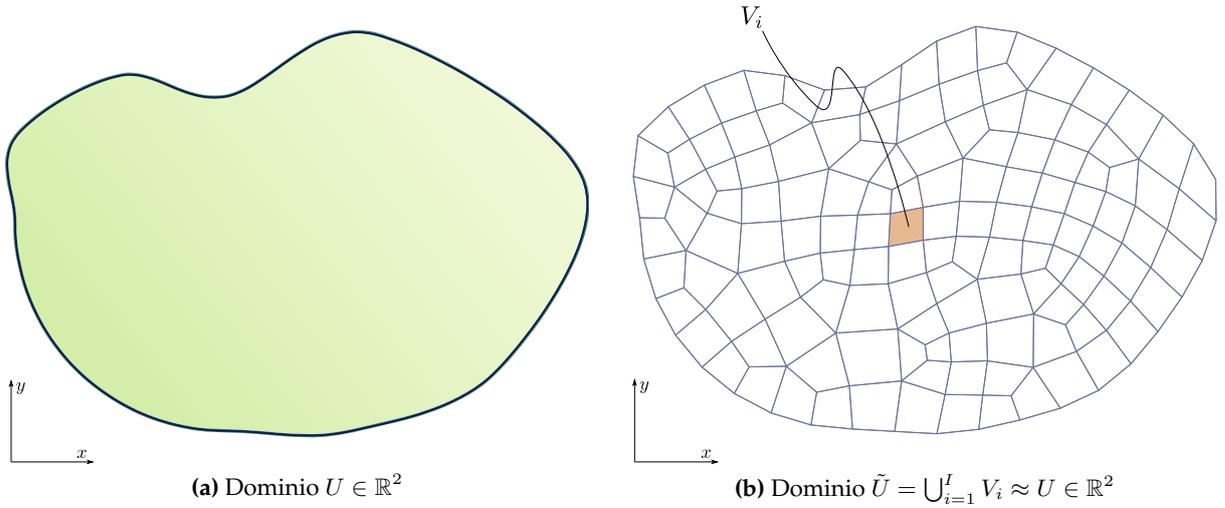


Figura 3.1 – Un dominio bidimensional $U \in \mathbb{R}^2$ es aproximado por la unión de I polígonos yuxtapuestos.

un dominio continuo está fuera del alcance de este trabajo. Suponemos que la discretización en celdas y nodos es un dato conocido del problema de difusión de neutrones.

En el método de los volúmenes finitos las incógnitas del problema son los valores medios de los flujos en las I celdas, a diferencia de lo que sucede en diferencias o elementos finitos donde las incógnitas son los valores que toman los flujos sobre los K nodos. Definimos el flujo del grupo $g = 1, \dots, G$ en el volumen V_i como

$$\hat{\phi}_g^i = \frac{\int_{V_i} \phi_g(\mathbf{x}) d^M \mathbf{x}}{\int_{V_i} d^M \mathbf{x}} \quad (3.1)$$

De esta manera, el problema consiste en encontrar los IG valores $\hat{\phi}_g^i$, que son los elementos del vector incógnita $\hat{\phi}$ definido en la ecuación (2.12). Para ello, comenzamos integrando la ecuación de difusión multigrupo diferencial (2.3)

$$0 = \text{div} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] - \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) + \sum_{g'=1}^G \Sigma_{sg' \rightarrow g}(\mathbf{x}) \cdot \phi_{g'}(\mathbf{x}) + \chi_g \sum_{g'=1}^G \frac{\nu \Sigma_{fg'}(\mathbf{x})}{k_{\text{eff}}} \cdot \phi_{g'}(\mathbf{x}) \quad (2.3)$$

sobre el i -ésimo volumen V_i

$$0 = \int_{V_i} \text{div} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] d^M \mathbf{x} - \int_{V_i} \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x} + \sum_{g'=1}^G \int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) \cdot \phi_{g'}(\mathbf{x}) d^M \mathbf{x} + \chi_g \sum_{g'=1}^G \int_{V_i} \frac{\nu \Sigma_{fg'}(\mathbf{x})}{k_{\text{eff}}} \cdot \phi_{g'}(\mathbf{x}) d^M \mathbf{x} \quad (3.2)$$



y analizamos en las secciones que siguen cada término por separado para obtener expresiones explícitas para los elementos de las matrices F y R del problema discreto, ambas de tamaño $IG \times IG$. Notar que la ecuación (2.3) implica que la ecuación (3.2) es cierta, pero la afirmación inversa no es necesariamente cierta. Luego, la ecuación (3.2) es una formulación integral pero no es una formulación débil equivalente a la formulación fuerte dada por la ecuación (2.3). Volvemos sobre este punto en la sección 4.3.

3.1.1. Reacciones totales

El segundo término del miembro derecho de la ecuación (3.2), además de ser el más sencillo de analizar, representa el ritmo total al cual los neutrones pertenecientes al grupo de energía g interactúan con los núcleos ubicados en todo el volumen V_i . El objetivo es poder escribir este término como el producto de un coeficiente, que aparecerá en algún elemento de la matriz R , por el flujo medio $\hat{\phi}_g^i$, que corresponderá a algún elemento del vector incógnita $\hat{\phi}$. Es decir, queremos escribir

$$\int_{V_i} \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x} = \hat{\Sigma}_{tg}^i V_i \cdot \hat{\phi}_g^i \quad (3.3)$$

Teniendo en cuenta la definición dada por la ecuación (3.1), la sección eficaz macroscópica total media del volumen V_i debería ser

$$\hat{\Sigma}_{tg}^i = \frac{\int_{V_i} \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x}}{\int_{V_i} \phi_g(\mathbf{x}) d^M \mathbf{x}} \quad (3.4)$$

es decir, el valor medio de $\Sigma_{tg}(\mathbf{x})$ en el volumen V_i pesado con la distribución continua de flujo $\phi_g(\mathbf{x})$, que por supuesto no conocemos.

Por un lado, en el caso particular de que $\Sigma_{tg}(\mathbf{x})$ sea uniforme en V_i , entonces la ecuación (3.4) da como resultado dicho valor uniforme. Por otro lado, si es el flujo el que resulta ser uniforme en V_i , la ecuación (3.4) se reduce al valor medio sin pesar

$$\hat{\Sigma}_{tg}^i = \frac{\int_{V_i} \Sigma_{tg}(\mathbf{x}) d^M \mathbf{x}}{\int_{V_i} d^M \mathbf{x}} \quad (3.5)$$

que aplica tanto para el caso de sección eficaz uniforme en el volumen o de flujo uniforme en el volumen. En el caso general, esta ecuación será solo una aproximación. Sin embargo, aún para distribuciones arbitrarias de secciones eficaces, a medida que $V_i \rightarrow 0$ la aproximación de flujo uniforme en el volumen es cada vez más exacta, y por lo tanto la ecuación (3.5) también.

Aceptando entonces que vale la ecuación (3.5) podemos generalizar la expresión que esperamos escribir como un coeficiente por el flujo medio de la celda aproximadamente como

$$\int_{V_i} \Sigma_{tg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x} \approx \left(\int_{V_i} \Sigma_{tg}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_g^i \quad (3.6)$$

De esta manera, evitamos tener que calcular la sección eficaz media del volumen explícitamente para luego tener que volver a multiplicar dicho valor por el volumen V_i , como hicimos en la ecuación (3.3). Dado que la distribución espacial de secciones eficaces es un dato conocido, es posible calcular la integral del miembro derecho de la ecuación (3.6) numéricamente y obtener entonces el coeficiente de $\hat{\phi}_g^i$ correspondiente al término de reacción total del grupo $g = 1, \dots, G$ en el volumen $i = 1, \dots, I$.



3.1.2. Fisiones

El último término del miembro derecho de la ecuación (3.2) representa el ritmo de nacimiento de neutrones de energía correspondiente al grupo g debido a fisiones en el volumen V_i . Para ello, calculamos el ritmo total de nacimientos en V_i sumando las tasas de nu-fisiones $\nu\Sigma_f\phi$ sobre todos los grupos de energía del volumen y luego multiplicando por el espectro de fisión χ_g que indica la fracción de neutrones que nacen con una energía correspondiente al grupo g . Todas las secciones eficaces de fisión se dividen por el factor de multiplicación incógnita k_{eff} para hacer crítico el reactor artificialmente, como discutimos en la sección 2.2. Debido a la linealidad de los operadores suma e integración, podemos extraer $1/k_{\text{eff}}$ factor común de la suma sobre $g' = 1, \dots, G$, por lo que no lo tendremos en cuenta en el análisis que sigue.

De la misma manera que en la sección anterior, queremos escribir los ritmos de reacción como el producto entre una sección eficaz macroscópica media, el volumen V_i de la celda y el flujo incógnita:

$$\int_{V_i} \nu\Sigma_{fg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x} = \nu\hat{\Sigma}_{fg}^i V_i \cdot \hat{\phi}_g^i$$

De manera análoga a la ecuación (3.5), definimos el valor medio de la sección eficaz de nu-fisión en el volumen V_i para el grupo g como

$$\nu\hat{\Sigma}_{fg}^i = \frac{\int_{V_i} \nu\Sigma_{tg}(\mathbf{x}) d^M \mathbf{x}}{\int_{V_i} d^M \mathbf{x}} \quad (3.7)$$

para finalmente escribir la tasa de nu-fisiones debidas a neutrones del grupo g en el volumen V_i aproximadamente como

$$\int_{V_i} \nu\Sigma_{fg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x} \approx \left(\int_{V_i} \nu\Sigma_{fg}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_g^i \quad (3.8)$$

3.1.3. Scattering

El tercer término del miembro derecho de la ecuación (3.2) representa el ritmo de aparición de neutrones en el volumen V_i provenientes de colisiones elásticas de neutrones con núcleos localizados en el volumen V_i de tal manera que la energía resultante del neutrón dispersado corresponde al grupo g . Para ello, sumamos los ritmos de scattering $\Sigma_s\phi$ que tienen como destino el grupo g sobre todos los grupos de energía, incluyendo el ritmo de self-scattering que indica el ritmo de neutrones de energía g que luego de una colisión aún mantienen el grupo de energía. Incluir este término como fuente de neutrones es necesario si escribimos el término de reacción total en función de la sección eficaz total—tal como en la ecuación (3.2)—ya que este término considera cualquier colisión como desaparición de neutrones. También podemos utilizar la sección eficaz de remoción introducida en la sección 2.2.3

$$\Sigma_{rg}(\mathbf{x}) = \Sigma_{tg}(\mathbf{x}) - \Sigma_{sg \rightarrow g}(\mathbf{x}) = \Sigma_{ag}(\mathbf{x}) + \sum_{g' \neq g}^G \Sigma_{sg \rightarrow g'}(\mathbf{x}) \quad (2.10)$$

En cualquier caso, la sección eficaz media de scattering entre el grupo g' y el g en el volumen V_i es

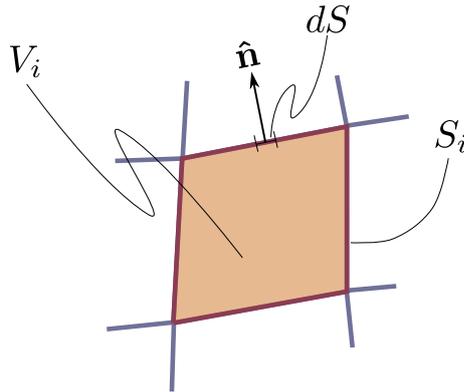


Figura 3.2 – Detalle del volumen bidimensional V_i de la figura 3.1b. La superficie S_i que lo delimita en este caso está compuesta por cuatro segmentos, cada uno con un vector normal $\hat{\mathbf{n}}$ diferente.

$$\hat{\Sigma}_{sg' \rightarrow g}^i = \frac{\int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) d^M \mathbf{x}}{\int_{V_i} d^M \mathbf{x}} \quad (3.9)$$

y el ritmo de dispersiones del grupo g' al g en el volumen V_i es aproximadamente

$$\int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) \cdot \phi_{g'}(\mathbf{x}) d^M \mathbf{x} \approx \left(\int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_{g'}(i) \quad (3.10)$$

3.1.4. Fugas

El primer término del miembro derecho de la ecuación (3.2), con signo negativo, representa el ritmo neto de fugas de neutrones de energía g a través de la superficie que delimita el volumen V_i .¹ Es en este término donde aparece el acople entre celdas espaciales y cobra importancia el método de volúmenes finitos. Utilizando el teorema de la divergencia, podemos escribir

$$\int_{V_i} \text{div} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] d^M \mathbf{x} = \int_{S_i} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}} dS$$

donde S_i representa la superficie que delimita el volumen V_i —que normalmente se compone de varias superficies planas—y $\hat{\mathbf{n}}$ es el vector unitario normal al diferencial de superficie dS que apunta en sentido externo al volumen V_i , como ilustramos en la figura 3.2.

La forma de estimar la contribución a la integral de superficie sobre S_i de cada una de las caras planas que define el volumen V_i depende de qué tipo de celda está yuxtapuesta a éste. Si las dos celdas contienen el mismo material, entonces los parámetros nucleares son continuos en el espacio y vale la ecuación de difusión. Si las dos celdas son de diferente material y el coeficiente de difusión tiene una discontinuidad en la interfaz, entonces no vale la ecuación de difusión (2.3) y hay que plantear continuidad de corrientes dada por la ecuación (2.4). Por otro lado, si una cara del volumen V_i está en el borde del dominio, hay que aplicar las condiciones de contorno para evaluar la integral de superficie. A continuación analizamos cada caso en forma particular.

¹En el marco de la ecuación de difusión, que es el alcance de este trabajo. En teoría de transporte las fugas se calculan utilizando otra expresión.

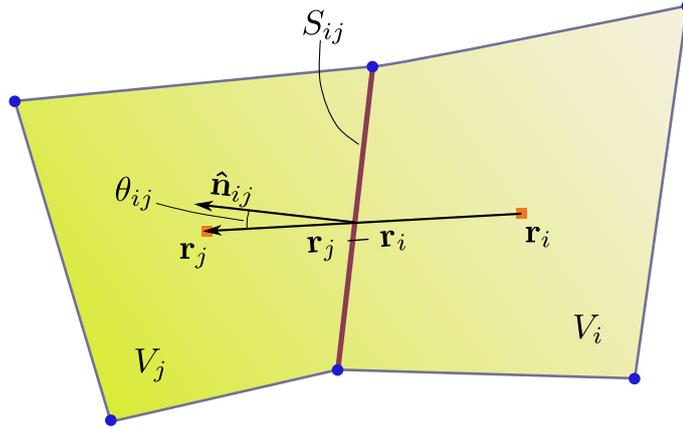


Figura 3.3 – Dos volúmenes vecinos V_i y V_j , la superficie S_{ij} que los separa, la normal externa \mathbf{n}_{ij} con respecto a V_i , los baricentros \mathbf{x}_i y \mathbf{x}_j y el ángulo θ_{ij} entre la normal y el segmento que une los baricentros.

3.1.4.1. Celdas homogéneas

Dados dos volúmenes vecinos V_i y V_j , sea S_{ij} la superficie plana que los separa, como ilustramos en la figura 3.3. En esta superficie, el versor normal $\hat{\mathbf{n}}_{ij}$ es uniforme y se puede calcular a partir de las coordenadas de los nodos que definen los vértices de la superficie S_{ij} (ver apéndice A). Asumimos que los volúmenes V_i y V_j están compuestos del mismo material y a lo sumo tienen una variación continua de propiedades tales como temperatura o densidad, y por lo tanto una variación continua de secciones eficaces macroscópicas. Suponemos que el gradiente del flujo se mantiene uniforme a lo largo de la superficie S_{ij} y tiene un módulo igual a

$$\left\| \text{grad} [\phi_g(\mathbf{x})] \right\|_{\mathbf{x} \in S_{ij}} \approx \frac{\hat{\phi}_g^j - \hat{\phi}_g^i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (3.11)$$

y está en la dirección del vector $\mathbf{x}_j - \mathbf{x}_i$ que une los baricentros \mathbf{x}_j y \mathbf{x}_i de los volúmenes j e i respectivamente (figura 3.3). Esta dirección forma un cierto ángulo θ_{ij} con la normal $\hat{\mathbf{n}}_{ij}$, que podemos calcular geoméricamente a partir de las coordenadas de los nodos que definen la superficie S_{ij} y de las coordenadas de los baricentros \mathbf{x}_i y \mathbf{x}_j como

$$\cos \theta_{ij} = \frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|}$$

Luego la tasa neta de neutrones de energía g que cruzan la superficie S_{ij} es aproximadamente

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}}_{ij} dS \approx \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2} \cdot [\hat{\phi}_g^j - \hat{\phi}_g^i] \quad (3.12)$$

Podemos calcular la integral del coeficiente de difusión numéricamente a partir de la distribución $D_g(\mathbf{x})$ dada como dato, ya que hemos supuesto que $D_g(\mathbf{x})$ es continua en S_{ij} y por lo tanto su integral está bien definida.

Notar que estimar el módulo del gradiente del flujo en el borde del volumen según la ecuación (3.11) es equivalente a suponer que, dada una función continua $\phi(x)$, el punto c que cumple el teorema del valor medio

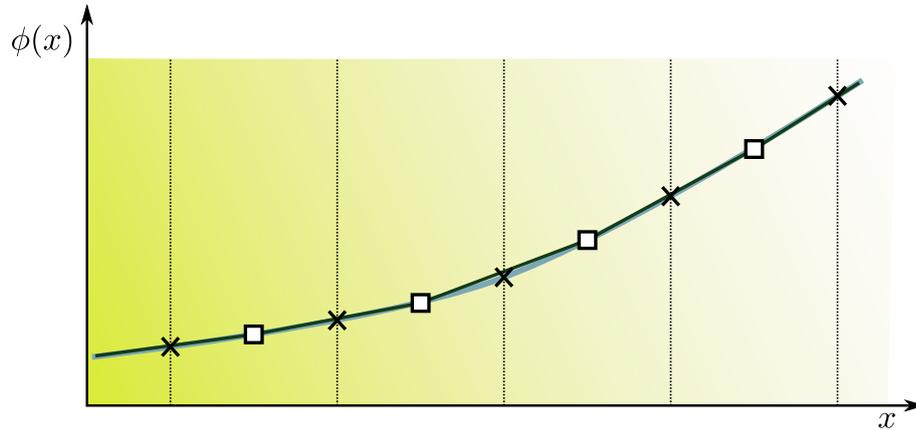


Figura 3.4 – Representación unidimensional de una función continua $\phi(x)$, con valores medios de volúmenes (cuadrados) y valores estimados en borde (cruces). La estimación de la derivada en los bordes a primeros vecinos es de orden superior a la estimación del valor de la función.

$$\phi'(c) = \frac{\phi(b) - \phi(a)}{b - a}$$

es $c \approx (b - a)/2$. Esta aproximación es un orden Δx^2 (es exacta para una parábola), mientras que la aproximación

$$\phi\left(\frac{b - a}{2}\right) \approx \frac{1}{2}[\phi(a) + \phi(b)]$$

es de orden Δx (es exacta para una recta). Ilustramos la situación en la figura 3.4, donde los cuadrados representan los valores medios en los volúmenes y las cruces los valores que tomaría el flujo en los bordes con una interpolación lineal. Aproximar las derivadas en los bordes con los valores medios es más preciso que aproximar los flujos en los bordes con los valores medios de los volúmenes vecinos.

3.1.4.2. Celdas heterogéneas

Si la superficie S_{ij} coincide con una interfaz entre dos materiales que tienen coeficientes de difusión diferentes, entonces necesariamente el flujo debe tener derivada primera discontinua de forma tal de que se cumpla la condición de conservación de corriente dada por la ecuación (2.4). En este caso, aproximar el gradiente del flujo en la interfaz utilizando los valores medios en los volúmenes vecinos arroja una pobre estimación, como ilustramos en la figura 3.5 (y en el ejemplo 5.1.2.2). Además, necesitamos obtener un valor para el gradiente a un lado de la interfaz y otro valor al otro lado de forma tal de satisfacer la ecuación (2.4).

Conviene entonces, evaluar el gradiente del flujo a un lado de la superficie S_{ij} partir del valor medio $\phi(\mathbf{x}_i)$ (cuadrado) y de $\phi_g(\mathbf{x}_{ij}) = \phi(\mathbf{x}_{ji})$ (cruz) y al otro lado—que llamamos S_{ji} —a partir de $\phi_g(\mathbf{x}_j)$ (cuadrado) y de $\phi_g(\mathbf{x}_{ij})$ (cruz). Debido a que el flujo $\phi_g(\mathbf{x})$ debe ser continuo en \mathbf{x} , el valor en la superficie $\phi_g(\mathbf{x}_{ij})$ debe ser único tanto visto desde el volumen i como desde el volumen j .

Proponemos entonces estimar el módulo del gradiente del flujo sobre la superficie S_{ij} ahora como

$$\left\| \text{grad} [\phi_g(\mathbf{x})] \right\|_{\mathbf{x} \in S_{ij}} \approx \frac{\phi_g(\mathbf{x}_{ij}) - \hat{\phi}_g^i}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|} \quad (3.13)$$

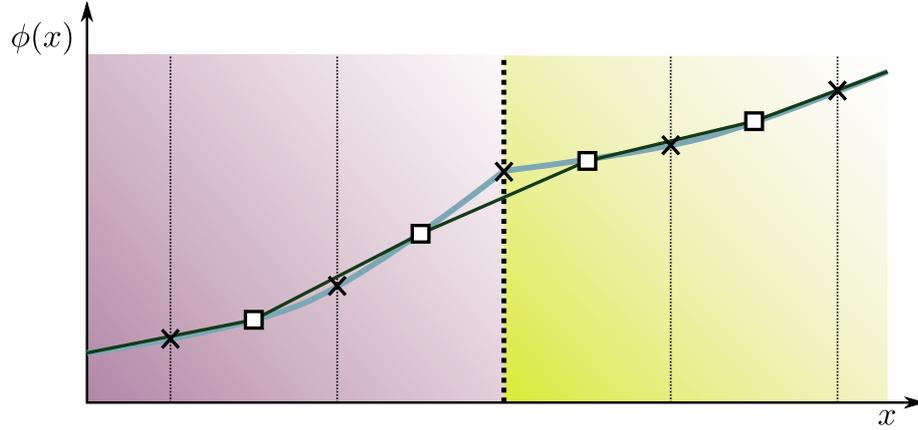


Figura 3.5 – Idem figura 3.4 para una interfaz material. La estimación de la derivada en la interfaz utilizando valores medios (cuadrados) es pobre. Conviene estimar el valor del flujo en la interfaz (cruz) y luego tomar derivadas a cada lado de la superficie.

tomándolo en la dirección del vector $\mathbf{x}_{ij} - \mathbf{x}_i$ que une el baricentro \mathbf{x}_{ij} de la superficie S_{ij} y el baricentro \mathbf{x}_i del volumen i (figura 3.6). En la ecuación (3.13) aparece el valor del flujo sobre la cara $\phi(\mathbf{x}_{ij})$ (cruz), que no es parte de las incógnitas del problema. Tenemos que estimarlo en función de los valores $\hat{\phi}_g^i$ y $\hat{\phi}_g^j$ de tal manera de que se satisfaga la continuidad de corrientes.

Llamemos S_{ij} la superficie orientada cuya normal exterior da al volumen i , y S_{ji} la que da al volumen j . Podemos plantear una ecuación de donde despejar $\phi_g(\mathbf{x}_{ij})$. En efecto, la continuidad de corrientes requiere que

$$\int_{S_{ij}} [D_g(\mathbf{x}) \cdot \text{grad } \phi_g(\mathbf{x})]_i \cdot \hat{\mathbf{n}}_{ij} dS = - \int_{S_{ji}} [D_g(\mathbf{x}) \cdot \text{grad } \phi_g(\mathbf{x})]_j \cdot \hat{\mathbf{n}}_{ji} dS$$

$$\int_{S_{ij}} D_g^i(\mathbf{x}) dS \cdot \frac{\phi_g(\mathbf{x}_{ij}) - \hat{\phi}_g^i}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|} \cos \theta_{ij} = - \int_{S_{ji}} D_g^j(\mathbf{x}) dS \cdot \frac{\phi_g(\mathbf{x}_{ji}, g) - \hat{\phi}_g^j}{\|\mathbf{x}_{ji} - \mathbf{x}_j\|} \cos \theta_{ji}$$

donde entendemos que la integral sobre S_{ij} se tiene que realizar evaluando el coeficiente de difusión sobre el volumen i , y viceversa escribiendo los subíndices i y j en el símbolo $D_g(\mathbf{x})$. Para simplificar el despeje de $\phi_g(\mathbf{x}_{ij}) = \phi(\mathbf{x}_{ji}, g)$ definimos los factores de peso

$$w_{ij} = \left(\int_{S_{ij}} D_g^i(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2} \quad (3.14)$$

$$w_{ji} = \left(\int_{S_{ji}} D_g^j(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ji} - \mathbf{x}_j) \cdot \hat{\mathbf{n}}_{ji}}{\|\mathbf{x}_{ji} - \mathbf{x}_j\|^2} \quad (3.15)$$

Entonces

$$w_{ij} [\phi_g(\mathbf{x}_{ij}) - \hat{\phi}_g^i] = -w_{ji} [\phi_g(\mathbf{x}_{ji}) - \hat{\phi}_g^j]$$

y el flujo $\phi_g(\mathbf{x}_{ij})$ en la interfaz es aproximadamente

$$\phi_g(\mathbf{x}_{ij}) \approx \frac{w_{ij}}{w_{ij} + w_{ji}} \cdot \hat{\phi}_g^i + \frac{w_{ji}}{w_{ij} + w_{ji}} \cdot \hat{\phi}_g^j \quad (3.16)$$

Reemplazando la ecuación (3.16) en la expresión (3.13) propuesta para el módulo del gradiente, obtenemos

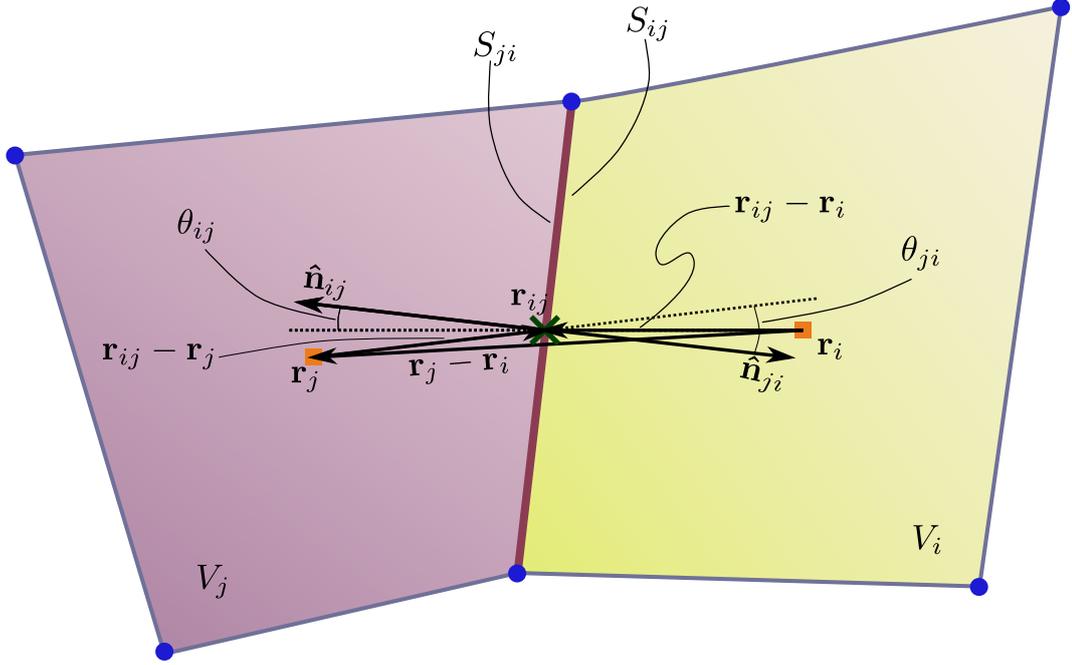


Figura 3.6 – Idem figura 3.3 para el caso en que la superficie S_{ij} corresponda a una interfaz material. El punto \mathbf{x}_{ij} es el baricentro de la superficie S_{ij} , y no necesariamente está sobre la dirección $\mathbf{x}_j - \mathbf{x}_i$.

$$\left\| \text{grad} [\phi_g(\mathbf{x})] \right\|_{\mathbf{x} \in S_{ij}} \approx \left(\frac{w_{ji}}{w_{ij} + w_{ji}} \right) \cdot \left(\frac{\hat{\phi}_g^j - \hat{\phi}_g^i}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|} \right)$$

La tasa neta de cruce de neutrones sobre la superficie S_{ij} cuando ésta coincide con una interfaz material es aproximadamente

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}}_{ij} dS \approx \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \right) \cdot \frac{w_{ji}}{w_{ij} + w_{ji}} \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2} \cdot [\hat{\phi}_g^j - \hat{\phi}_g^i] \quad (3.17)$$

donde la integral del coeficiente de difusión tiene que tomarse sobre la cara orientada de S_{ij} que da al volumen V_i , es decir, con las propiedades del material de la celda i , y los factores de peso w_{ij} y w_{ji} están definidos en las ecuaciones (3.14) y (3.15).

Si bien la ecuación (3.17) puede utilizarse aún cuando los volúmenes V_i y V_j tengan el mismo material para estimar el término de fugas, la ecuación (3.12) es más precisa ya que la aproximación para el gradiente dada por la ecuación (3.11) es de orden Δx^2 , mientras que la aproximación del gradiente dada por la ecuación (3.13) es de orden Δx ya que se basa en estimar el flujo en el borde de los volúmenes, como ilustramos en la figura 3.4. Sin embargo, si S_{ij} coincide con una interfaz y el coeficiente de difusión es discontinuo, la ecuación (3.17) arroja mejores resultados ya que tiene en cuenta el efecto mostrado en la figura 3.5. Luego, para minimizar el error cometido por la discretización basada en el esquema de volúmenes finitos propuesto, si S_{ij} no define una interfaz, utilizamos la ecuación (3.12) para calcular los coeficientes que formarán parte de la matriz de remoción R . De otra manera, utilizamos la ecuación (3.17).



3.1.4.3. Condiciones de contorno nulas

Si la superficie S_{ij} está en el borde del dominio, entonces debemos calcular la integral de superficie

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}} dS$$

teniendo en cuenta las condiciones de contorno del problema. Como discutimos en la sección 2.2.2, sólo se pueden dar condiciones de contorno homogéneas. En particular, si la superficie S_{ij} tiene una condición de contorno de Dirichlet, como mencionamos en la sección 2.2.2, ésta debe ser deflujo nulo. Luego, $\phi_g(\mathbf{x}_{ij}) = 0$ y

$$\left\| \text{grad} [\phi_g(\mathbf{x})] \right\|_{\mathbf{x} \in S_{ij}} \approx \frac{0 - \hat{\phi}_g^i}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|}$$

La tasa neta de cruce de neutrones sobre la superficie S_{ij} cuando ésta coincide con un borde del dominio con condición de contorno de flujo nulo es

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}}_{ij} dS \approx \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2} \cdot \left[-\hat{\phi}_g^i \right] \quad (3.18)$$

3.1.4.4. Condiciones de contorno espejo

Para especificar superficies de simetría, se debe una condición de contorno de Neumann con derivada normal nula (sección 2.2.2). Si S_{ij} coincide con una de estas superficies, entonces el módulo del gradiente es cero y su integral también lo es. Por lo tanto, la tasa neta de cruce de neutrones sobre la superficie S_{ij} es cero y

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}}_{ij} dS = 0 \quad (3.19)$$

3.1.4.5. Condiciones de contorno de Robin

Si la superficie S_{ij} pertenece a una zona de la frontera del dominio ∂U con condiciones de contorno de Robin (sección 2.2.2), entonces

$$\left[\text{grad} [\phi_g(\mathbf{x})] \cdot \hat{\mathbf{n}} \right]_{\mathbf{x} \in \partial U} = \frac{\partial \phi_g}{\partial n} \Big|_{\mathbf{x} \in \partial U} = \frac{a_g(\mathbf{x})}{D_g(\mathbf{x})} \cdot \phi_g(\mathbf{x}) \Big|_{\mathbf{x} \in \partial U}$$

Asumiendo que $\phi_g(\mathbf{x})$ para $\mathbf{x} \in \partial U$ es aproximadamente igual a $\hat{\phi}_g^i$, la tasa neta de cruce de neutrones a través de la superficie S_{ij} es

$$\int_{S_{ij}} \left[D_g(\mathbf{x}) \cdot \text{grad} [\phi_g(\mathbf{x})] \right] \cdot \hat{\mathbf{n}} dS \approx S_{ij} \cdot a_g(\mathbf{x}) \cdot \left[\hat{\phi}_g^i \right] \quad (3.20)$$

donde S_{ij} es el área de la superficie.



3.2. Formulación matricial

Resumiendo las expresiones obtenidas a lo largo del capítulo, para cada celda $i = 1, \dots, I$ y para cada grupo de energía $g = 1, \dots, G$ la ecuación de difusión de neutrones en estado estacionario discretizada con el método de volúmenes finitos propuesto implica que debemos satisfacer simultáneamente las IG ecuaciones

$$\begin{aligned} & \left(\int_{V_i} \Sigma_{tg}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_g^i - \sum_{g'=1}^G \left[\left(\int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_{g'}^i \right] \\ & + \left[\sum_{j \text{ vecinos}} \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \cdot \frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2} \right) \right] \cdot \hat{\phi}_g^i \\ & - \sum_{j \text{ vecinos}} \left[\left(\int_{S_{ij}} D_g(\mathbf{x}) dS \cdot \frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2} \right) \cdot \hat{\phi}_g^j \right] \\ & = \frac{1}{k_{\text{eff}}} \cdot \sum_{g'=1}^G \left[\left(\chi_g \cdot \int_{V_i} \nu \Sigma_{fg'}(\mathbf{x}) d^M \mathbf{x} \right) \cdot \hat{\phi}_{g'}^i \right] \end{aligned}$$

Así escrita, esta ecuación corresponde a una configuración en la que los vecinos de la celda i están compuestos del mismo material. En caso de que algún vecino sea de otro material o corresponda a una condición de contorno, debemos reemplazar los términos tres y cuatro por expresiones apropiadas según discutimos a lo largo del capítulo. En cualquier caso, lo que queremos mostrar es que estas IG ecuaciones son lineales para los flujos $\hat{\phi}_g^i$ (si las secciones eficaces dependen solamente de la posición \mathbf{x} y no del flujo, que es lo que supusimos) y que podemos escribir el problema discretizado en forma matricial como

$$R \cdot \hat{\phi} = \frac{1}{k_{\text{eff}}} \cdot F \cdot \hat{\phi} \quad (2.11)$$

para algún vector incógnita $\hat{\phi}$ de tamaño $L = IG$ que contenga como elementos $\hat{\phi}_g^i$ para los índices $i = 1, \dots, I$ y $g = 1, \dots, G$. Si en particular ordenamos los flujos como sigue

$$\hat{\phi} = \begin{bmatrix} \phi_1^1 \\ \phi_2^1 \\ \vdots \\ \phi_G^1 \\ \phi_1^2 \\ \vdots \\ \phi_G^2 \\ \phi_1^3 \\ \vdots \\ \phi_G^I \end{bmatrix} \quad (2.12)$$

entonces el flujo incógnita $\hat{\phi}_g^i$ está en la posición

$$\ell(i, g) = g + G \cdot (i - 1) \quad (3.21)$$

del vector $\hat{\phi}$ de tal forma que



$$\hat{\phi}_{\ell(i,g)} = \hat{\phi}_g^i$$

De la misma manera, con $R_{p,q}$ y $F_{p,q}$ denotamos los elementos ubicados en la fila p y en la columna q de las matrices R y F respectivamente. En la figura 3.7 mostramos el algoritmo que utilizamos para construir las matrices R y F según las ecuaciones desarrolladas en este capítulo, teniendo en cuenta las posibles combinaciones de tipos de vecinos que puede tener cada celda, i.e. celdas homogéneas, celdas heterogéneas, condición de contorno de flujo nulo, condición de contorno de simetría o condición de contorno de Robin.

En la evaluación del término de fugas para celdas vecinas heterogéneas, denotamos con el supraíndice i en la expresión $D_g^i(\mathbf{x})$ que el coeficiente de difusión correspondiente al material de la celda i y con $D_g^j(\mathbf{x})$ el de la celda j . Si las celdas fuesen homogéneas la distinción no es necesaria.

Podemos reemplazar las integrales sobre las celdas por el producto entre el valor del integrando en el baricentro \mathbf{x}_i y el volumen V_i de la celda i -ésima

$$\int_{V_i} f(\mathbf{x}) d^M \mathbf{x} \approx f(\mathbf{x}_i) \cdot V_i \quad (3.22)$$

con una gran ganancia de eficiencia a costa de una pequeña pérdida de precisión. De la misma manera, podemos reemplazar las integrales sobre las superficies de las celdas por el producto entre el coeficiente de difusión evaluado en el centro de la cara multiplicado por el área de la misma, es decir

$$\int_{S_{ij}} D_g(\mathbf{x}) dS \approx D_g(\mathbf{x}_{ij}) \cdot S_{ij} \quad (3.23)$$

Las aproximaciones de las ecuaciones (3.22) y (3.23) son equivalentes a utilizar un único punto de Gauss en la cuadratura numérica (ver Apéndice C) pero su cómputo es más eficiente ya que—al menos en una y dos dimensiones—los volúmenes de las celdas—que equivalen a segmentos y a superficies planas respectivamente—pueden ser calculados fácilmente a partir de las coordenadas de los puntos que las definen (Apéndice A). De hecho, si quisiéramos usar más de un punto de Gauss deberíamos recurrir a la evaluación de jacobianos de funciones de forma similares a las asociadas al método de elementos finitos, aumentando sensiblemente el esfuerzo computacional necesario sin asegurar siquiera una mejora en la estimación del valor de la integral.

En problemas unidimensionales—a menos que se especifique lo contrario en el archivo de entrada—las integrales volumétricas que aparecen en el algoritmo de la figura 3.7 en una dimensión son estimadas por el código milonga como el producto del integrando evaluado en el centro de la celda por la longitud de ésta. De la misma manera, en dos dimensiones el código evalúa el integrando en el baricentro de la celda y lo multiplica por el área, calculada según los resultados geométricos discutidos en el apéndice A. En una dimensión, todas las superficies que separan celdas contiguas se asumen de área uno y en dos dimensiones el área resultante es igual a la longitud de un segmento cuya longitud es trivial. Como indicamos en el capítulo 6, los problemas tridimensionales al momento de escribir este trabajo están en el TODO-list y quedan planteados como ejercicio a futuro.



inicializar $R \leftarrow 0, F \leftarrow 0$

para cada celda $i = 1, \dots, I$

para cada grupo $g = 1, \dots, G$

$p \leftarrow \ell(i, g)$ posición en $\hat{\phi}$ ec. (3.21)

$R_{p,p} \leftarrow \int_{V_i} \Sigma_{tg}(\mathbf{x}) d^M \mathbf{x}$ absorción ec. (3.6)

para cada grupo $g' = 1, \dots, G$

$q \leftarrow \ell(i, g')$ posición en $\hat{\phi}$ ec. (3.21)

$R_{p,q} \leftarrow R_{p,q} - \int_{V_i} \Sigma_{sg' \rightarrow g}(\mathbf{x}) d^M \mathbf{x}$ scattering ec. (3.10)

$F_{p,q} \leftarrow \chi_g \cdot \int_{V_i} \nu \Sigma_{fg'}(\mathbf{x}) d^M \mathbf{x}$ fisión ec. (3.8)

para cada superficie S_{ij} de V_i

si $\exists V_j$ **entonces**

$q \leftarrow \ell(j, g)$ posición en $\hat{\phi}$ ec. (3.21)

si V_i y V_j tienen el mismo material **entonces**

$\xi \leftarrow \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}$ fugas ec. (3.12)

sino

$w_{ij} \leftarrow \left(\int_{S_{ij}} D_g^i(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2}$

$w_{ji} \leftarrow - \left(\int_{S_{ji}} D_g^j(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ji} - \mathbf{x}_j) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ji} - \mathbf{x}_j\|^2}$

$\xi \leftarrow \left(\int_{S_{ij}} D_g^i(\mathbf{x}) dS \right) \cdot \frac{w_{ji}}{w_{ij} + w_{ji}} \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2}$ fugas ec. (3.17)

$R_{p,p} \leftarrow R_{p,p} + \xi$

$R_{p,q} \leftarrow -\xi$

sino

si S_{ij} tiene condición de contorno $\phi = 0$ **entonces** fugas ec. (3.18)

$R_{p,p} \leftarrow R_{p,p} + \left(\int_{S_{ij}} D_g(\mathbf{x}) dS \right) \cdot \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot \hat{\mathbf{n}}_{ij}}{\|\mathbf{x}_{ij} - \mathbf{x}_i\|^2}$

sino si S_{ij} tiene condición $\partial\phi/\partial n = 0$ **entonces** fugas ec. (3.19)

 no hay que hacer nada

sino si S_{ij} tiene condición $\partial\phi/\partial n = a_g(\mathbf{x})\phi$ **entonces** fugas ec. (3.20)

$R_{p,p} \leftarrow R_{p,p} + S_{ij} \cdot a_g(\mathbf{x})$

Figura 3.7 – Construcción de las matrices R y F en volúmenes finitos

Elementos finitos

An alleged scientific discovery has no merit unless it can be explained to a barmaid.

Ernest Rutherford

Variación de la mal atribuida a Einstein pero superior cita:
“You do not really understand something unless you can explain it to your grandmother”

En aquellos tiempos en los que éramos simpáticos estudiantes de grado, teníamos con mi compañero de estudios una metodología para mejor incorporar las nociones que infructuosamente trataban de inculcarnos nuestros desdichados profesores. Basándonos en la cita con la que este iniciamos este capítulo, consistía ésta en explicarnos mutuamente cada uno una bolilla diferente. De esta manera, más que acercarse a los conceptos el que recibe la explicación, es el sujeto de enunciación el que ejercita sus entendederas hasta el punto de internalizar los conocimientos. Es en este sentido en el que—dado que el tema central del presente trabajo consiste en aplicar el método de elementos finitos para resolver una cierta ecuación—en este capítulo además de describir el esquema numérico elegido para la ecuación de difusión de neutrones, introducimos con bastante detalle los razonamientos y ecuaciones que llevan a la formulación discretizada. Suponemos que el lector no está familiarizado con los métodos matemáticos usuales de los elementos finitos ni con la notación usual que combina operaciones vectoriales y matriciales utilizada en la literatura, a favor de que el autor vuelva a derivar lo que otros ya han hecho para—tal como sucedía en pretéritas épocas mejores—obligarlo a entender la matemática involucrada a un nivel suficiente como para que sea capaz de explicársela a su abuelita—particularmente la forma de aplicar las condiciones de contorno del problema a la formulación basada en elementos finitos.

El método de elementos finitos es un caso particular de una familia de herramientas matemáticas desarrolladas para atacar problemas planteados en ecuaciones diferenciales llamados residuos ponderados. Comenzamos entonces planteando las bases matemáticas que usamos más adelante al abordar la resolución de la ecuación de difusión de neutrones usando un esquema en elementos finitos, introduciendo las complicaciones inherentes del problema una a una.

Tal como en el capítulo anterior, indicamos los subíndices con letras minúsculas (esta vez i es el elemento i -ésimo) y los totales con la correspondiente mayúscula (la letra I indica el número total de elementos). Debemos notar que, aunque hemos intentado tener especial cuidado, esta vez la utilización—y mezcla—de diferentes subíndices puede llegar a ser bastante confusa. La teoría y las ecuaciones que introducimos en este capítulo que sirven de base de las que están programadas en milonga aparecen en forma similar a la resolución de un problema implícito.



Pueden aparecer ideas que se necesitan utilizar sin antes poder desarrollarlas en detalle, ya que este desarrollo puede necesitar conceptos que a su vez dependen de las primeras. Sirva-se el lector de releer el capítulo varias veces y si, luego de n iteraciones aún quedan ideas sin cerrar, por favor repórtele los correspondientes errores al autor de la monografía.

4.1. Residuos ponderados

Sea $\mathcal{A}[\phi(\mathbf{x})]$ un operador diferencial vectorial que representa un cierto sistema de G ecuaciones diferenciales cuya solución es una función vectorial $\phi(\mathbf{x})$ tal que

$$\mathcal{A}[\phi(\mathbf{x})] = \begin{bmatrix} \mathcal{A}_1[\phi(\mathbf{x})] \\ \mathcal{A}_2[\phi(\mathbf{x})] \\ \vdots \\ \mathcal{A}_G[\phi(\mathbf{x})] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.1)$$

en un dominio M -dimensional U tal que $\mathbf{x} \in U \in \mathbb{R}^M$. Queremos encontrar otra función vectorial $\tilde{\phi}(\mathbf{x}, \hat{\phi})$ que aproxime la solución exacta $\phi(\mathbf{x})$ en función de una cantidad finita K de parámetros ajustables $\hat{\phi}_k$. Dado que usualmente existen $G > 1$ ecuaciones diferenciales acopladas y el dominio U es usualmente es multidimensional con $M > 1$, para no confundir ni al lector ni al autor abusando de la notación matricial y vectorial—incluso a veces hasta tensorial—conviene en este caso ir de lo particular a lo general. Comenzamos estudiando el caso más sencillo con $G = 1$ y $M = 1$ para luego completar la formulación general una vez comprendidos los fundamentos del método. Así mismo, para fijar ideas recurrimos a la introducción de ejemplos sencillos que ilustran ciertas particularidades que pueden ayudar a entender—principalmente al autor—los conceptos básicos.

4.1.1. Ecuación escalar en una dimensión

Sea la ecuación diferencial a resolver

$$\mathcal{A}[\phi(x)] = 0 \quad (4.2)$$

en el dominio $a < x < b$, y donde por ahora suponemos que el problema tiene condiciones de contorno de Dirichlet (no necesariamente homogéneas) en $x = a$ y en $x = b$, es decir

$$\begin{cases} \phi(a) = \phi_a \\ \phi(b) = \phi_b \end{cases} \quad (4.3)$$

El método de los residuos ponderados consiste en aproximar la solución real $\phi(x)$ por otra función $\tilde{\phi}(x)$ que pueda escribirse en función de K parámetros ajustables $\hat{\phi}_k$ como

$$\tilde{\phi}(x) = h_0(x) + \sum_{k=1}^K h_k(x) \cdot \hat{\phi}_k \quad (4.4)$$

donde las $h_k(x)$ son funciones de forma conocidas que deben cumplir ciertas condiciones particulares que discutimos más adelante, pero que además exigimos que junto con la $h_0(x)$ sean tales que las condiciones de contorno de Dirichlet dadas en la ecuación (4.3) se cumplan para cualquier conjunto de valores $\hat{\phi}_k$.

Dado que en general $\phi(x) \neq \tilde{\phi}(x)$ y la solución de la ecuación diferencial (4.2) es única, entonces $\mathcal{A}[\tilde{\phi}(x)] \neq 0$. Definimos este último valor como el error residual cometido en el punto x . El método de los residuos ponderados propone obtener K ecuaciones de donde despejar



los parámetros $\tilde{\phi}_k$ multiplicando el residuo por K funciones de peso $w_k(x)$ y requiriendo la integral en el intervalo $[a, b]$ de cada una de estas expresiones sea cero, es decir

$$\int_a^b w_k(x) \cdot \mathcal{A} \left[h_0(x) + \sum_{k=1}^K h_k(x) \cdot \hat{\phi}_k \right] dx = 0 \quad \text{para } k = 1, \dots, K$$

Ilustramos la utilización del método, la elección de las funciones de forma, la importancia de las funciones de peso y una particularidad de la formulación integral en los ejemplos que siguen.

Ejemplo 1 Resolver la ecuación diferencial

$$\frac{d^2 \phi}{dx^2} - 1 = 0$$

en el intervalo $-1 < x < 1$ con las condiciones de contorno

$$\begin{cases} \phi(-1) = 0 \\ \phi(+1) = 0 \end{cases}$$

aplicando el método de los residuos ponderados proponiendo una solución aproximada con un único parámetro $\hat{\phi}_1$

$$\tilde{\phi}(x) = h_1(x) \cdot \hat{\phi}_1$$

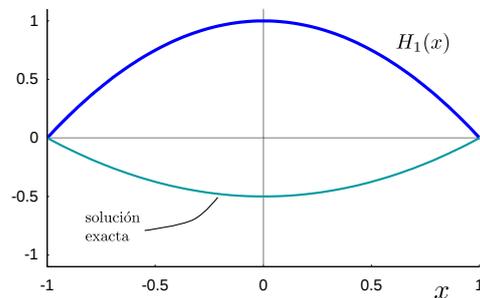
con una única la función de prueba

$$h_1(x) = 1 - x^2$$

que satisface las condiciones de contorno y además $h_1(0) = 1$ y $h_1(x) < 1$ en el resto del dominio. De esta forma, el parámetro $\hat{\phi}_1$ es el valor de la función $\tilde{\phi}(x)$ en un cierto valor de x , en este caso $x = 0$. Esta característica de las funciones de forma es ciertamente útil ya que los parámetros $\hat{\phi}_i$ pasan a ser valores nodales de la función incógnita.

▷ Requerimos que la integral pesada con una cierta función $w(x)$ del residuo sea nula. Entonces

$$\begin{aligned} 0 &= \int_{-1}^1 w(x) \cdot \left[\frac{d^2 \tilde{\phi}}{dx^2} - 1 \right] dx \\ 0 &= \int_{-1}^1 w(x) \cdot \left[\frac{d^2 h_1}{dx^2} \cdot \hat{\phi}_1 - 1 \right] dx \\ 0 &= \int_{-1}^1 w(x) \cdot \left[-2\hat{\phi}_1 - 1 \right] dx \end{aligned}$$



Como la integral debe ser nula para cualquier función de peso $w(x)$, entonces el corchete debe anularse idénticamente. Luego $\phi_1 = -1/2$ para cualquier función de peso $w(x)$, y recuperamos la solución exacta de la ecuación diferencial.

El ejemplo 1 ilustra—no demuestra—que si la elección de las funciones de forma es tal que



la solución aproximada escrita de la forma (4.4) admite recuperar la solución exacta, ésta se recupera independientemente de las funciones de peso. Es decir, el resultado es independiente de la elección de las $w_i(x)$. Sin embargo, en general no es posible escribir la solución exacta de la forma (4.4), por lo que el resultado depende de las funciones de peso elegidas, como vemos en el siguiente ejemplo.

Ejemplo 2 Resolver ahora la misma ecuación diferencial y las mismas condiciones de contorno pero con la función continua a trozos

$$h_1(x) = 1 - |x|$$

a modo de función de prueba.

▷ Este tipo de funciones es muy utilizada ya que resulta la interpolación más simple entre los valores nodales de las funciones solución. Sin embargo, en $x = 0$ resulta que $h_1(x)$ no tiene ni derivada primera ni derivada segunda, por lo que el residuo no se puede evaluar en $x = 0$. Pero, a modo de retruque del adversativo, la formulación integral de los residuos ponderados hace posible encontrar una solución en este caso, ya que si bien la estrictamente derivada segunda no se puede evaluar, podemos escribirla en función de la delta de Dirac como

$$\frac{d^2 h_1}{dx^2} = -2\delta(x)$$

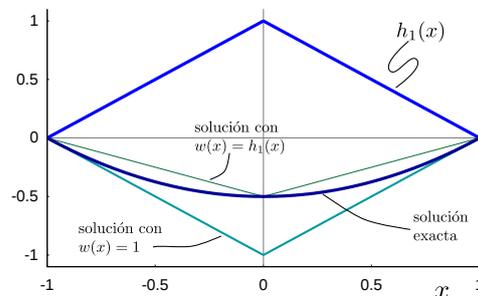
ya que la derivada primera vale -1 para $x > 0$ y $+1$ para $x < 0$, obteniendo una función que puede ser integrada y pesada

$$0 = \int_{-1}^1 w(x) \cdot [-2\hat{\phi}_1 \cdot \delta(x) - 1] dx$$

$$0 = -2\hat{\phi}_1 \cdot w(0) - \int_{-1}^1 w(x) dx$$

para obtener

$$\hat{\phi}_1 = -\frac{\int_{-1}^1 w(x) dx}{2 \cdot w(0)} \quad (4.5)$$



de donde vemos que la solución depende de la elección de la función de peso $w(x)$. Por ejemplo, si $w(x)$ es idénticamente igual a uno, resulta $\hat{\phi}_1 = -1$ y si $w(x) = h_1(x)$ entonces $\hat{\phi}_1 = -1/2$.

En forma alternativa—y ligeramente más formal—podemos integrar por partes el término de la derivada segunda

$$0 = \int_{-1}^1 w(x) \cdot \left[\frac{d^2 h_1}{dx^2} \cdot \hat{\phi}_1 - 1 \right] dx$$

$$0 = - \int_{-1}^1 \frac{dw}{dx} \frac{dh_1}{dx} \cdot \hat{\phi}_1 dx + \left[w(x) \cdot \frac{dh_1}{dx} \cdot \hat{\phi}_1 \right]_{-1}^1 - \int_{-1}^1 w(x) dx \quad (4.6)$$



para observar explícitamente que se requiere que la función de prueba sea derivable sólo una vez en el dominio y puede no tener derivada definida en un conjunto de medida cero.

Este último ejemplo ilustra varios puntos interesantes del método de residuos ponderados que son heredados por el método de elementos finitos, como discutimos más adelante. Por un lado, la formulación integral permite que los requerimientos de continuidad sobre las funciones de prueba sean más laxos que en la formulación diferencial. Hemos ejemplificado esta característica en el caso escalar unidimensional o bien integrando una delta de Dirac o bien realizando una integración por partes. Es posible generalizar estos procedimientos a más dimensiones. Esta característica de las formulaciones integrales o débiles es particularmente útil en el caso de la ecuación de difusión de neutrones cuando existen discontinuidades espaciales en los coeficientes de difusión.

Por otro lado, el ejemplo 2 muestra que los parámetros ajustables que arroja el método en general dependen de la elección de las funciones de peso usadas para ponderar los residuos. En particular, si utilizamos las mismas funciones de prueba $h_k(x)$ como funciones de peso $w_k(x)$, sucede que el residuo resultante es perpendicular a todas ellas en el espacio vectorial de todas las posibles funciones continuas. Mientras más funciones de prueba usemos, más pequeño—en el sentido geométrico de dicho espacio vectorial—resulta el residuo. Por esta y por otras propiedades [24], es ésta la elección usual para las funciones de peso. Este caso tiene nombre propio, y cuando $w_k(x) = h_k(x)$ decimos que estamos aplicando el método de Galerkin.

Finalmente, vemos en el ejemplo 2 que las condiciones de contorno aparecen directamente en la formulación fuerte, o bien en el coeficiente de la delta o en el término de superficie al realizar la integración por partes, sin necesidad de incorporarlas como ecuaciones anexas. El siguiente ejemplo ilustra con un poco más de detalle cómo se incorporan las condiciones de contorno del problema, que es uno de los conceptos más sutiles del método de elementos finitos. De hecho fue el papel que juegan las condiciones de contorno en la formulación débil la que generó la redacción de este prolegómeno teórico-divagativo antes de introducir la discretización espacial de la ecuación de difusión de neutrones multigrupo en un esquema en elementos finitos. Ilustramos la diferencia entre el manejo de condiciones de contorno de Dirichlet y de Neumann, y además aprovechamos para introducir una primera notación matricial para manejar el correspondiente sistema de ecuaciones que aparece al tener más de una función de prueba.

Ejemplo 3 Resolver una vez más la ecuación diferencial

$$\frac{d^2 \phi}{dx^2} - 1 = 0$$

pero ahora con las condiciones de contorno

$$\begin{cases} \phi(-1) & = 1 \\ \left. \frac{d\phi}{dx} \right|_{x=1} & = 1 \end{cases} \quad (4.7)$$

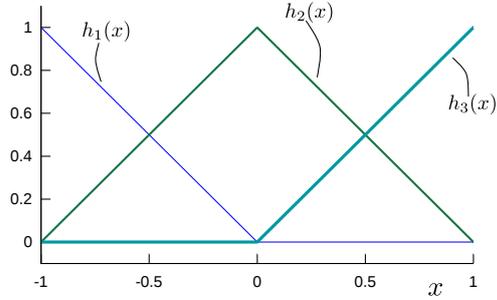
▷ Según la ecuación (4.4), las funciones de prueba deben ser tales que se satisfagan las condiciones de Dirichlet del problema. En este caso, $\phi(-1) = 1$. Para resolver el problema,



vamos a suponer que tenemos tres funciones de prueba y tres parámetros, pero en verdad hay uno que no es ajustable sino que lo fijamos de forma tal de que se satisfaga la condición de contorno en $x = -1$. Lo que hacemos entonces es trabajar como si tuviésemos tres incógnitas—por lo que necesitamos tres ecuaciones—y antes de resolver el problema, fijamos el valor de una de ella para satisfacer la condición de Dirichlet.

Proponemos

$$\begin{aligned}\tilde{\phi}(x) &= h_1(x)\hat{\phi}_1 + h_2(x)\hat{\phi}_2 + h_3(x)\hat{\phi}_3 \\ &= [h_1(x) \quad h_2(x) \quad h_3(x)] \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} \\ &= H \cdot \hat{\phi}\end{aligned}$$



La derivada es

$$\begin{aligned}\frac{d\tilde{\phi}}{dx} &= \frac{dh_1}{dx} \cdot \hat{\phi}_1 + \frac{dh_2}{dx} \cdot \hat{\phi}_2 + \frac{dh_3}{dx} \cdot \hat{\phi}_3 \\ &= \begin{bmatrix} \frac{dh_1}{dx} & \frac{dh_2}{dx} & \frac{dh_3}{dx} \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} \\ &= B \cdot \hat{\phi}\end{aligned}$$

y las tres ecuaciones de los residuos ponderados a cumplir son

$$\begin{aligned}0 &= \int_{-1}^1 w_k(x) \cdot \left[\frac{d^2\tilde{\phi}}{dx^2} - 1 \right] dx \\ &= - \int_{-1}^1 \frac{dw_k}{dx} \cdot \frac{d\tilde{\phi}}{dx} dx + \left[w_k(x) \cdot \frac{d\tilde{\phi}}{dx} \right]_{-1}^{+1} - \int_{-1}^1 w_k(x) dx\end{aligned}$$

para $k = 1, 2, 3$. En el método de Galerkin, $w_k(x) = h_k(x)$ y podemos escribirlas en notación matricial como

$$\begin{aligned}0 &= - \int_{-1}^1 \begin{bmatrix} h'_1 \\ h'_2 \\ h'_3 \end{bmatrix} [h'_1 \quad h'_2 \quad h'_3] \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} dx + \left(\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \frac{d\tilde{\phi}}{dx} \right)_{-1}^{+1} - \int_{-1}^1 \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} dx \\ 0 &= - \left(\int_{-1}^1 \begin{bmatrix} h'_1(x)h'_1(x) & h'_1(x)h'_2(x) & h'_1(x)h'_3(x) \\ h'_2(x)h'_1(x) & h'_2(x)h'_2(x) & h'_2(x)h'_3(x) \\ h'_3(x)h'_1(x) & h'_3(x)h'_2(x) & h'_3(x)h'_3(x) \end{bmatrix} dx \right) \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} \\ &\quad + \begin{bmatrix} h_1(1) \\ h_2(1) \\ h_3(1) \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=1} - \begin{bmatrix} h_1(-1) \\ h_2(-1) \\ h_3(-1) \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=-1} - \int_{-1}^1 \begin{bmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \end{bmatrix} dx\end{aligned}$$



$$0 = \left(- \int_{-1}^1 B^T(x) \cdot B(x) dx \right) \hat{\phi} + \left[H^T(x) \cdot \frac{d\tilde{\phi}}{dx} \right]_{-1}^{+1} - \int_{-1}^1 H^T(x) dx$$

En particular resulta

$$\begin{aligned} 0 = & \left(- \int_{-1}^0 \begin{bmatrix} (-1)(-1) & (-1)(+1) & (-1) \cdot 0 \\ (+1)(-1) & (+1)(+1) & (+1) \cdot 0 \\ 0 \cdot (-1) & 0 \cdot (+1) & 0 \cdot 0 \end{bmatrix} dx \right. \\ & \left. - \int_0^1 \begin{bmatrix} 0 \cdot 0 & 0 \cdot (-1) & 0 \cdot (+1) \\ (-1) \cdot 0 & (-1)(-1) & (-1) \cdot (+1) \\ (+1) \cdot 0 & (+1)(-1) & (+1) \cdot (+1) \end{bmatrix} dx \right) \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} \\ & + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=1} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=-1} - \int_{-1}^0 \begin{bmatrix} 1-x \\ 1+x \\ 0 \end{bmatrix} dx - \int_0^1 \begin{bmatrix} 0 \\ 1-x \\ x \end{bmatrix} dx \end{aligned}$$

y el sistema de ecuaciones de donde despejar los parámetros ajustables es

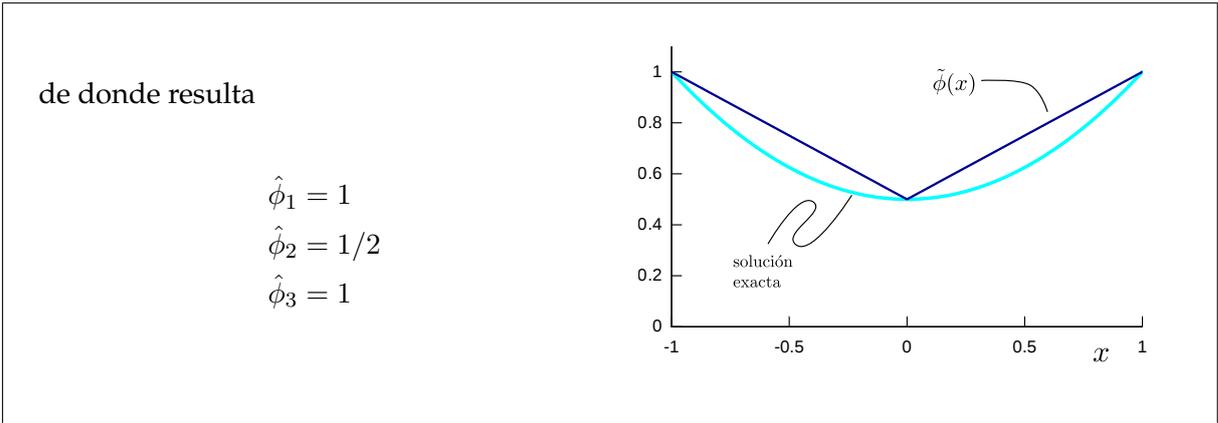
$$\begin{bmatrix} +1 & -1 & 0 \\ -1 & +2 & -1 \\ 0 & -1 & +1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} - \frac{d\tilde{\phi}}{dx} \Big|_{x=-1} - \frac{1}{2} \\ -\frac{1}{2} - \frac{1}{2} \\ + \frac{d\tilde{\phi}}{dx} \Big|_{x=1} - \frac{1}{2} \end{bmatrix}$$

Como dijimos al principio, para satisfacer la condición de contorno de Dirichlet en $x = -1$ debe ser $\hat{\phi}_1 = 1$. Entonces reemplazamos la primera fila de la matriz por un uno en la diagonal y ceros en el resto de las columnas y el primer elemento del vector del miembro derecho por el valor que haga que se satisfaga la condición de contorno de Dirichlet (en este caso 1), de tal modo de reemplazar la ecuación del residuo ponderado correspondiente a la función de forma $h_1(x)$ por la ecuación $\hat{\phi}_1 = +1$:

$$\begin{bmatrix} +1 & 0 & 0 \\ -1 & +2 & -1 \\ 0 & -1 & +1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ \frac{d\tilde{\phi}}{dx} \Big|_{x=1} - \frac{1}{2} \end{bmatrix} \quad (4.8)$$

Finalmente, reemplazamos $\frac{d\tilde{\phi}}{dx} \Big|_{x=1}$ en el tercer elemento del vector independiente por el valor dado por la condición de contorno de Neumann (en este caso 1), para obtener

$$\begin{bmatrix} +1 & 0 & 0 \\ -1 & +2 & -1 \\ 0 & -1 & +1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ +\frac{1}{2} \end{bmatrix}$$



Tal como habíamos prometido, en este ejemplo ilustramos principalmente cómo lidiar con las condiciones de contorno. El primer comentario es que las condiciones de contorno de Dirichlet se satisfacen o bien con la elección de las funciones de prueba como en los ejemplos 1 y 2 o bien reemplazando la ecuación que calcula los parámetros ajustables que definen el valor de la solución en el contorno del dominio que tiene condiciones de Dirichlet. Estas dos maneras de atacar el problema son esencialmente iguales, sólo que la última tiene ventajas prácticas ya que no requiere encontrar explícitamente la función $h_0(x)$ en la ecuación (4.4) de antemano, y matemáticamente queda definida como una combinación lineal de las funciones $h_k(x)$ para los valores de k cuyos parámetros ajustables $\hat{\phi}_k$ son fijados. La función aproximada $\tilde{\phi}(x)$ cumple exactamente las condiciones de contorno de Dirichlet del problema. En este marco, llamamos a este tipo de condiciones de contorno condiciones esenciales.

El segundo comentario se refiere a las condiciones de contorno de Neumann, que llamamos condiciones naturales. Éstas aparecen en el término de superficie de la integración por partes con un símbolo $d\tilde{\phi}/dx$ que denota la derivada de la función incógnita evaluada en la superficie. Sin embargo, como vemos a continuación, este símbolo debe tratarse diferente cuando aparece en un término de superficie y cuando aparece en el integrando de una integral sobre el dominio. Detengámonos por un momento en esta idea.

En el ejemplo 3, el valor de la derivada en $x = -1$ desapareció de la formulación al reemplazar la ecuación de la condición esencial en ese punto, mientras que el valor de la derivada en $x = 1$ era un dato del problema y lo incorporamos a las ecuaciones. Con respecto a las condiciones de contorno naturales, debemos notar dos conceptos importantes, y que parecen sutiles pero hacen a la formulación general del método de elementos finitos. Por un lado, cuando aparezcan expresiones conteniendo la derivada de la función incógnita evaluada en una frontera del dominio que tenga condiciones esenciales, no debemos prestar mayor atención ya que eventualmente la ecuación donde ésta aparece será reemplazada por la ecuación de la condición natural. Por otro lado, cuando aparezca una expresión conteniendo la derivada de la función incógnita evaluada en una frontera del dominio que tenga condiciones naturales, debemos reemplazar dicha expresión por el valor de la condición natural del problema y no por la expresión de la derivada de $\tilde{\phi}(x)$ en función de las funciones de prueba y de los parámetros ajustables.

Resumiendo el confuso párrafo anterior, cuando aparezca $d\tilde{\phi}/dx$ evaluada en una superficie con condiciones de contorno esenciales, podemos ignorar su valor ya que no formará parte de la formulación. Y cuando aparezca $d\tilde{\phi}/dx$ en una superficie con condiciones de contorno naturales, debemos reemplazarla por el valor numérico requerido en la condición de Neumann, y no por $B(x)\hat{\phi}$ como sí debemos hacer cuando la derivada aparece en algún integrando de



la formulación débil. Más adelante demostramos este resultado para el caso particular de la ecuación de difusión de neutrones multigrupo.

Por ejemplo, si en la ecuación (4.8) en lugar de reemplazar $d\tilde{\phi}/dx$ en $x = 1$ por uno, que es el valor de la condición natural, hubiésemos escrito

$$\begin{aligned} \left. \frac{d\tilde{\phi}}{dx} \right|_{x=1} &= \left. \frac{dh_1}{dx} \right|_{x=1} \cdot \hat{\phi}_1 + \left. \frac{dh_2}{dx} \right|_{x=1} \cdot \hat{\phi}_2 + \left. \frac{dh_3}{dx} \right|_{x=1} \cdot \hat{\phi}_3 \\ &= -\hat{\phi}_2 + \hat{\phi}_3 \end{aligned}$$

entonces la ecuación (4.8) quedaría

$$\begin{bmatrix} +1 & 0 & 0 \\ -1 & +2 & -1 \\ 0 & -1 & +1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & +1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} \quad (4.9)$$

lo que daría lugar a un sistema de ecuaciones indeterminado, que es la versión aproximada de la indeterminación de una ecuación diferencial continua a la que le faltan condiciones de contorno.

Debemos notar que las condiciones de contorno naturales no se cumplen exactamente. Si bien los valores nodales—en el ejemplo $3 \hat{\phi}_1$ en $x = -1$, $\hat{\phi}_2$ en $x = 0$ y $\hat{\phi}_3$ en $x = 1$ —de la función aproximada $\hat{\phi}(x)$ coinciden con la solución exacta

$$\phi(x) = \frac{1}{2}(1 + x^2)$$

la derivada en $x = 1$ de $\hat{\phi}(x)$ no es igual a uno sino a un medio. Esta es una característica del método de residuos ponderados: las condiciones esenciales se satisfacen exactamente—a menos de la precisión numérica con la que podemos resolver el sistema de ecuaciones para los parámetros—y las condiciones naturales se satisfacen aproximadamente, de la misma manera que la función obtenida es sólo una aproximación de la solución real en el dominio.

4.1.2. Ecuación escalar en varias dimensiones

Estudiemos ahora el caso en el que la función incógnita es un campo escalar de varias variables. En este caso

$$\mathcal{A}[\phi(\mathbf{x})] = 0$$

sobre algún dominio $U \in \mathbb{R}^M$, con condiciones de contorno—o bien de Dirichlet o bien de Neumann—en la frontera ∂U . De la misma manera que en la sección anterior, aproximamos la función solución por una expresión aproximada

$$\tilde{\phi}(\mathbf{x}) = h_0(\mathbf{x}) + \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_k$$

donde introducimos $h_0(\mathbf{x})$ para satisfacer las condiciones de contorno de Dirichlet del problema, aunque la podemos incorporar a las $h_k(\mathbf{x})$ reemplazando algunos de los parámetros $\hat{\phi}_k$ por los valores que satisfacen dichas condiciones de contorno. Nuevamente ilustramos la aplicación del método de residuos ponderados con un ejemplo.



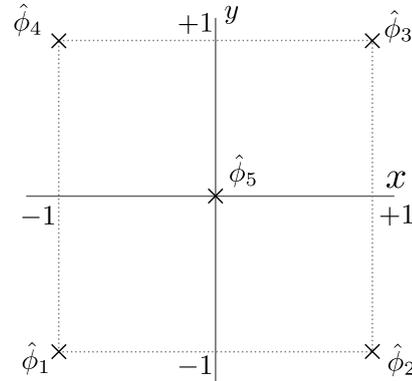
Ejemplo 4 Plantear el método de Galerkin a la ecuación diferencial de Poisson

$$\nabla^2 \phi(x, y) - 1 = 0$$

en el dominio $U = [-1, 1] \times [-1, 1] \in \mathbb{R}^2$ con condiciones de contorno de Dirichlet

$$\phi(\mathbf{x}) = \phi_a \quad \text{si } \mathbf{x} \in \partial U$$

utilizando cinco puntos de colocación tal como ilustra la figura.



Como los cuatro puntos que están en la frontera son singulares y no tienen normal, evitar los problemas aparejados a las condiciones de Neumann y dejar el análisis de la incorporación de este tipo de condiciones de contorno para más adelante cuando aparezcan dominios más suaves.

▷ Sean $h_k(x, y)$ para $k = 1, \dots, 5$ funciones de forma cuya expresión explícita aún no discutimos en esta sección pero que asumimos son apropiadas para resolver el problema. Entonces proponemos

$$\begin{aligned} \tilde{\phi}(x, y) &= \sum_{k=1}^5 h_k(x, y) \cdot \hat{\phi}_k \\ &= [h_1(x, y) \quad h_2(x, y) \quad h_3(x, y) \quad h_4(x, y) \quad h_5(x, y)] \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} \\ &= H \cdot \hat{\phi} \end{aligned}$$

Notamos que $\hat{\phi} \in \mathbb{R}^5$ y $H \in \mathbb{R}^{1 \times 5}$. El método de Galerkin provee las cinco ecuaciones que necesitamos para despejar los parámetros ajustables $\hat{\phi}_k$ utilizando como funciones de peso $w_k(x, y)$ las cinco funciones de forma $h_k(x, y)$:

$$0 = \int_{-1}^1 \int_{-1}^1 h_k(x, y) \cdot [\nabla^2 \tilde{\phi}(x, y) - 1] dx dy \quad \text{para } i = 1, \dots, 5$$

Podemos aprovechar la formulación integral para reducir en uno el orden del operador diferencial que aparece en la ecuación utilizando las fórmulas de Green (apéndice B) como sigue

$$\begin{aligned} 0 &= \int_{-1}^1 \int_{-1}^1 h_k(x, y) \cdot \nabla^2 \tilde{\phi}(x, y) dx dy - \int_{-1}^1 \int_{-1}^1 h_k(x, y) dx dy \\ &= - \int_{-1}^1 \int_{-1}^1 \nabla h_k(x, y) \cdot \nabla \tilde{\phi}(x, y) dx dy + \int_{\partial U} h_k(x, y) [\nabla \tilde{\phi}(x, y) \cdot \hat{\mathbf{n}}] dS \end{aligned}$$



$$- \int_{-1}^1 \int_{-1}^1 h_k(x, y) dx dy$$

donde debemos tener en cuenta que en este problema en particular no hay condiciones de Neumann, por lo que—según aprendimos en la sección anterior—tenemos que reemplazar el símbolo $\nabla \tilde{\phi}$ por

$$\begin{aligned} \nabla \tilde{\phi}(x, y) &= \begin{bmatrix} \partial \tilde{\phi} / \partial x \\ \partial \tilde{\phi} / \partial y \end{bmatrix} \\ &= \begin{bmatrix} \partial h_1 / \partial x & \partial h_2 / \partial x & \partial h_3 / \partial x & \partial h_4 / \partial x & \partial h_5 / \partial x \\ \partial h_1 / \partial y & \partial h_2 / \partial y & \partial h_3 / \partial y & \partial h_4 / \partial y & \partial h_5 / \partial y \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} \\ &= B \cdot \hat{\phi} \end{aligned}$$

Notamos que $B \in \mathbb{R}^{2 \times 5}$ y que además podemos escribir el producto escalar entre los gradientes $\nabla h_k(x, y)$ y $\nabla \tilde{\phi}(x, y)$ como un producto matricial

$$\begin{aligned} \nabla h_k(x, y) \cdot \nabla \tilde{\phi}(x, y) &= [\partial h_k / \partial x \quad \partial h_k / \partial y] \cdot \begin{bmatrix} \partial \tilde{\phi} / \partial x \\ \partial \tilde{\phi} / \partial y \end{bmatrix} \\ &= [\partial h_k / \partial x \quad \partial h_k / \partial y] \cdot (B \cdot \hat{\phi}) \end{aligned}$$

Entonces la k -ésima ecuación del sistema de cinco ecuaciones es

$$\begin{aligned} 0 &= - \int_{-1}^1 \int_{-1}^1 [\partial h_k / \partial x \quad \partial h_k / \partial y] \cdot (B \cdot \hat{\phi}) dx dy \\ &\quad + \int_{\partial U} h_k(x, y) \left\{ (B \cdot \hat{\phi}) \cdot \hat{\mathbf{n}} \right\} dS - \int_{-1}^1 \int_{-1}^1 h_k(x, y) dx dy \end{aligned}$$

Podemos escribir las cinco ecuaciones en una sola expresión matricial como

$$\begin{aligned} 0 &= - \int_{-1}^1 \int_{-1}^1 \begin{bmatrix} \partial h_1 / \partial x & \partial h_1 / \partial y \\ \partial h_2 / \partial x & \partial h_2 / \partial y \\ \partial h_3 / \partial x & \partial h_3 / \partial y \\ \partial h_4 / \partial x & \partial h_4 / \partial y \\ \partial h_5 / \partial x & \partial h_5 / \partial y \end{bmatrix} \cdot (B \cdot \hat{\phi}) dx dy \\ &\quad + \int_{\partial U} \begin{bmatrix} h_1(x, y) \\ h_2(x, y) \\ h_3(x, y) \\ h_4(x, y) \\ h_5(x, y) \end{bmatrix} \left\{ (B \cdot \hat{\phi}) \cdot \hat{\mathbf{n}} \right\} dS - \int_{-1}^1 \int_{-1}^1 \begin{bmatrix} h_1(x, y) \\ h_2(x, y) \\ h_3(x, y) \\ h_4(x, y) \\ h_5(x, y) \end{bmatrix} dx dy \end{aligned}$$



donde identificamos las transpuestas de las matrices H y B , con lo que escribimos en forma compacta

$$0 = - \int_{-1}^1 \int_{-1}^1 B^T \cdot (B \cdot \hat{\phi}) \, dx \, dy + \int_{\partial U} H^T \cdot \{ (B \cdot \hat{\phi}) \cdot \hat{\mathbf{n}} \} \, dS - \int_{-1}^1 \int_{-1}^1 H^T \, dx \, dy$$

Notando que la multiplicación de matrices es asociativa, podemos escribir el integrando del primer término, que es un vector de tamaño cinco, como

$$B^T \cdot (B \cdot \hat{\phi}) = (B^T \cdot B) \cdot \hat{\phi}$$

El integrando del segundo término tiene cierta complejidad ya que involucra un producto escalar entre los vectores $B\hat{\phi}$ y $\hat{\mathbf{n}}$, resultado escalar que a su vez multiplica a un vector de tamaño cinco. El producto escalar en forma de producto matricial es

$$(B \cdot \hat{\phi}) \cdot \hat{\mathbf{n}} = \hat{\mathbf{n}}^T \cdot (B \cdot \hat{\phi})$$

y el integrando queda

$$\{ (B \cdot \hat{\phi}) \cdot \hat{\mathbf{n}} \} = (H^T \cdot \hat{\mathbf{n}}^T \cdot B) \cdot \hat{\phi}$$

Como el vector $\hat{\phi}$ no depende de la posición \mathbf{x} entonces puede salir fuera de las integrales, con lo que obtenemos un sistema lineal de cinco ecuaciones con cinco incógnitas

$$\left(- \int_{-1}^1 \int_{-1}^1 B^T \cdot B \, dx \, dy + \int_{\partial U} H^T \cdot \hat{\mathbf{n}}^T \cdot B \, dS \right) \cdot \hat{\phi} = \int_{-1}^1 \int_{-1}^1 H^T \, dx \, dy$$

al que aún le falta ponerle las condiciones de contorno del problema reemplazando las filas de la matriz y los elementos del vector independiente que corresponden a aquellos parámetros que están en la frontera del dominio por filas canónicas y por los valores de la condición de Dirichlet, respectivamente. En el caso del problema del ejemplo, debemos reemplazar las primeras cuatro filas y los primeros cuatro elementos del vector, quedando sólo la última fila de la matriz y el último elemento del vector según la formulación obtenida. Por lo tanto, de la segunda integral, sólo debemos prestar atención a la ecuación que involucra a $h_5(x, y)$ e ignorar las primeras. Pero sucede además—como discutimos en la sección 4.2.2—que es condición necesaria que $h_5(x, y)$ sea idénticamente cero en ∂U . Luego podemos ignorar la contribución de la integral sobre la frontera porque las primeras cuatro filas del integrando serán luego reemplazadas al poner las condiciones de contorno, y la última fila contiene cinco ceros porque $h_5(x, y) = 0$ en ∂U .

Finalmente, el sistema de ecuaciones a resolver es

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & u_{5,5} \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = \begin{bmatrix} \phi_a \\ \phi_a \\ \phi_a \\ \phi_a \\ v_5 \end{bmatrix}$$

donde



$$u_{k,k'} = - \int_{-1}^1 \int_{-1}^1 \left(\frac{\partial h_k}{\partial x} \frac{\partial h_{k'}}{\partial x} + \frac{\partial h_k}{\partial y} \frac{\partial h_{k'}}{\partial y} \right) dx dy$$

y

$$v_k = \int_{-1}^1 \int_{-1}^1 h_k(x, y) dx dy$$

Como vemos, las ecuaciones y los desarrollos se hacen más engorrosos y complicados a medida que agregamos complejidades a los problemas a resolver. Desde el humilde punto de vista de este que escribe, la dificultad que más atención necesita es discernir entre las diferentes operaciones matriciales y vectoriales que aparecen, y en identificar qué expresiones son escalares, cuáles son vectores y cuáles son matrices. Es por eso que la escritura explícita de todos los elementos en forma de matrices, si bien es tediosa y ocupa mucho lugar, ayuda a entender las expresiones que aparecen en forma elegante y compacta cuando uno intenta leer libros sobre el método de elementos finitos.

En el ejemplo 4 obtenemos un sistema lineal de 5×5 donde cuatro ecuaciones son triviales y corresponden a las condiciones de contorno. En problemas de interés con decenas de miles de nodos, las ecuaciones que corresponden a las condiciones de contorno son muchas menos que las que correspondan a la ecuación diferencial en el interior del dominio. Sin embargo, si no incorporamos las condiciones de contorno de esta manera, las matrices que representan el sistema de ecuaciones resultan singulares y el problema discretizado no tiene solución.

Notamos también que la incorporación de las condiciones de contorno en la forma en que lo hicimos en el ejemplo rompe la simetría de la matriz $B^T B$, y de alguna manera hace que para resolver el problema se necesite más esfuerzo computacional del que se necesitaría si efectivamente el problema fuese simétrico. Sin embargo, como el problema objeto del presente trabajo es la ecuación de difusión de neutrones multigrupo que en general no es simétrica, el proceso de fijar los valores de los parámetros ajustables que corresponde a nodos en la frontera no introduce mayores inconvenientes, así que utilizamos la forma de definir las condiciones de contorno de Dirichlet tal como lo hicimos en el ejemplo 4 en lugar de complicar aún más el esquema numérico con el único objetivo de obtener matrices globales simétricas.

Finalmente, observamos que el resultado obtenido en la sección 4.1.1 concerniente a las condiciones de contorno también vale para el caso multidimensional. Por un lado, no es necesario calcular la contribución de los términos que correspondan a parámetros que están sobre la frontera del dominio que tiene condiciones de Dirichlet a la integral de superficie que resulta de aplicar la fórmula de Green, ya que estas contribuciones serán luego reemplazadas por el proceso ilustrado de fijación de parámetros sobre dicha superficie. Y, además, la contribución de los términos que corresponden a parámetros que no están sobre la superficie es nula ya que las funciones de forma asociadas a dichos parámetros debe ser cero en la frontera. Luego, el término de superficie de la fórmula de Green sobre la parte de ∂U que tiene condiciones de Dirichlet usualmente puede ignorarse al aplicar el método de Galerkin. Es este un detalle sutil pero particularmente importante, que no es obvio y que es difícil de demostrar en forma general.

4.1.3. Ecuación vectorial en varias dimensiones

Volvamos a considerar ahora el caso general planteado al principio del capítulo en la ecuación (4.1)



$$\mathcal{A}[\phi(\mathbf{x})] = \begin{bmatrix} \mathcal{A}_1[\phi(\mathbf{x})] \\ \mathcal{A}_2[\phi(\mathbf{x})] \\ \vdots \\ \mathcal{A}_G[\phi(\mathbf{x})] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.1)$$

en el que $M > 1, G > 1$ y

$$\phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_G(\mathbf{x}) \end{bmatrix}$$

son las funciones incógnitas. Proponemos una función vectorial aproximada, en la que cada componente está escrita en la forma (4.4)

$$\tilde{\phi}(\mathbf{x}) = \begin{bmatrix} \tilde{\phi}_1(\mathbf{x}) \\ \tilde{\phi}_2(\mathbf{x}) \\ \vdots \\ \tilde{\phi}_G(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_1^k \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_2^k \\ \vdots \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_G^k \end{bmatrix} \quad (4.10)$$

donde $\hat{\phi}_g^k$ es el k -ésimo parámetro ajustable correspondiente a la k -ésima función incógnita, con $k = 1, \dots, K$ y $g = 1, \dots, G$, donde ya supusimos que no necesitamos ninguna función de forma $h_0(x)$ para satisfacer las condiciones de Dirichlet sino que las tenemos en cuenta en los parámetros ajustables $\hat{\phi}_g^k$ que fijamos antes de resolver el sistema de ecuaciones. Para escribir el vector aproximado de funciones incógnitas en forma matricial y poder operar de la misma manera que en la sección anterior, debemos construir un único vector de parámetros incógnita $\hat{\phi}$ de tamaño $L = KG$ que contenga—en algún orden—todos los parámetros $\hat{\phi}_g^k$ de todas las funciones. Por ejemplo, si ordenamos las componentes del vector incógnita tal como lo hicimos en la ecuación (2.12) en el capítulo 2

$$\hat{\phi} = \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_1^2 \\ \vdots \\ \hat{\phi}_1^G \\ \hat{\phi}_2^1 \\ \hat{\phi}_2^2 \\ \vdots \\ \hat{\phi}_2^G \\ \vdots \\ \hat{\phi}_K^1 \\ \hat{\phi}_K^2 \\ \vdots \\ \hat{\phi}_K^G \end{bmatrix} \quad (2.12)$$

entonces podemos escribir la función vectorial aproximada como un producto matriz vector como sigue:



$$\tilde{\phi}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) & 0 & \cdots & h_2(\mathbf{x}) & 0 & \cdots & h_G(\mathbf{x}) & 0 & \cdots & 0 \\ 0 & h_1(\mathbf{x}) & \cdots & 0 & h_2(\mathbf{x}) & \cdots & 0 & h_G(\mathbf{x}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & h_G(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_2^1 \\ \vdots \\ \hat{\phi}_G^1 \\ \hat{\phi}_1^2 \\ \hat{\phi}_2^2 \\ \vdots \\ \hat{\phi}_G^K \end{bmatrix}$$

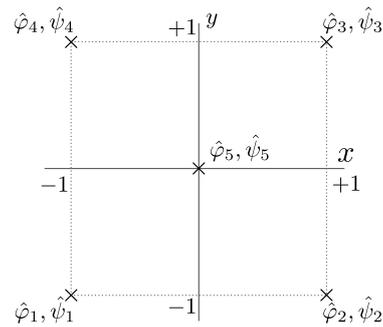
Este planteo, que es un verdadero abuso de notación vectorial, queda ilustrado con el siguiente ejemplo.

Ejemplo 5 Plantear la solución del sistema de ecuaciones diferenciales

$$\begin{cases} 0 &= \nabla^2 \varphi(x, y) + \psi(x, y) \\ 0 &= \nabla^2 \psi(x, y) - 1 \end{cases}$$

en el dominio $[-1, 1] \times [-1, 1] \in \mathbb{R}^2$ con condiciones de contorno de Dirichlet

$$\begin{cases} \varphi(\mathbf{x}) = \varphi_a & \text{si } \mathbf{x} \in \partial U \\ \psi(\mathbf{x}) = \psi_a & \text{si } \mathbf{x} \in \partial U \end{cases}$$



utilizando cinco puntos de colocación tal como ilustra la figura.

▷ Sean $h_k(x, y)$ para $k = 1, \dots, 5$ funciones de forma apropiadas, que pueden ser las mismas que las del ejemplo 4 ya que la elección de las funciones de forma depende—excepto en casos muy particulares—sólo de la geometría del dominio U y no de la ecuación en sí a resolver. El vector solución aproximado $\tilde{\phi}$ escrito como producto matriz-vector es

$$\begin{bmatrix} \tilde{\varphi}(x, y) \\ \tilde{\psi}(x, y) \end{bmatrix} = \begin{bmatrix} h_1 & 0 & h_2 & 0 & h_3 & 0 & h_4 & 0 & h_5 & 0 \\ 0 & h_1 & 0 & h_2 & 0 & h_3 & 0 & h_4 & 0 & h_5 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \psi_1 \\ \varphi_2 \\ \psi_2 \\ \varphi_3 \\ \psi_3 \\ \varphi_4 \\ \psi_4 \\ \varphi_5 \\ \psi_5 \end{bmatrix} \quad (4.11)$$

$$\tilde{\phi} = H \cdot \hat{\phi}$$



Notamos que si $G = 2$ es la cantidad de funciones incógnita (igual a la cantidad de ecuaciones del sistema) y $K = 5$ es la cantidad de puntos de colocación del dominio, entonces $\hat{\phi} \in \mathbb{R}^{KG}$ es un vector de dimensión 10 y $H \in \mathbb{R}^{G \times KG}$ es una matriz de 2×10 . Tenemos entonces diez incógnitas y necesitamos diez ecuaciones. Según Galerkin, éstas son

$$\begin{aligned} 0 &= \int_{-1}^1 \int_{-1}^1 h_k(x, y) \cdot [\nabla^2 \tilde{\varphi}(x, y) + \tilde{\psi}(x, y)] dx dy && \text{para } i = k, \dots, 5 \\ 0 &= \int_{-1}^1 \int_{-1}^1 h_k(x, y) \cdot [\nabla^2 \tilde{\psi}(x, y) - 1] dx dy && \text{para } i = k, \dots, 5 \end{aligned}$$

que, según el ya usual paso de aplicar la fórmula de Green y no tener en cuenta el término de superficie ya que no tenemos condiciones de Neumann, podemos escribir como

$$\begin{aligned} 0 &= \int_{-1}^1 \int_{-1}^1 [\nabla h_k(x, y) \cdot \nabla \tilde{\varphi}(x, y) + h_k(x, y) \cdot \tilde{\psi}(x, y)] dx dy && \text{para } i = k, \dots, 5 \\ 0 &= \int_{-1}^1 \int_{-1}^1 [\nabla h_k(x, y) \cdot \nabla \tilde{\psi}(x, y) - h_k(x, y)] dx dy && \text{para } i = k, \dots, 5 \end{aligned}$$

La idea es encontrar una forma de escribir estas diez ecuaciones en forma matricial en función del vector de parámetros ajustables $\hat{\phi}$. En particular, prestamos atención al término del producto interno de los gradientes de las $h_k(\mathbf{x})$ y de las funciones incógnita $\tilde{\varphi}(\mathbf{x})$ y $\tilde{\psi}(\mathbf{x})$. Para simplificar el desarrollo que sigue, vamos a suponer que $k = 1, 2$ en lugar de ir de uno hasta cinco. Queremos encontrar una expresión compacta para el vector

$$\begin{aligned} \begin{bmatrix} \nabla h_1 \cdot \nabla \tilde{\varphi} \\ \nabla h_1 \cdot \nabla \tilde{\psi} \\ \nabla h_2 \cdot \nabla \tilde{\varphi} \\ \nabla h_2 \cdot \nabla \tilde{\psi} \end{bmatrix} &= \begin{bmatrix} h'_{1x} \cdot \tilde{\varphi}'_x + h'_{1y} \cdot \tilde{\varphi}'_y \\ h'_{1x} \cdot \tilde{\psi}'_x + h'_{1y} \cdot \tilde{\psi}'_y \\ h'_{2x} \cdot \tilde{\varphi}'_x + h'_{2y} \cdot \tilde{\varphi}'_y \\ h'_{2x} \cdot \tilde{\psi}'_x + h'_{2y} \cdot \tilde{\psi}'_y \end{bmatrix} \\ &= \begin{bmatrix} h'_{1x} \cdot (h'_{1x} \hat{\varphi}_1 + h'_{2x} \hat{\varphi}_2) + h'_{1y} \cdot (h'_{1y} \hat{\varphi}_1 + h'_{2y} \hat{\varphi}_2) \\ h'_{1x} \cdot (h'_{1x} \hat{\psi}_1 + h'_{2x} \hat{\psi}_2) + h'_{1y} \cdot (h'_{1y} \hat{\psi}_1 + h'_{2y} \hat{\psi}_2) \\ h'_{2x} \cdot (h'_{1x} \hat{\varphi}_1 + h'_{2x} \hat{\varphi}_2) + h'_{2y} \cdot (h'_{1y} \hat{\varphi}_1 + h'_{2y} \hat{\varphi}_2) \\ h'_{2x} \cdot (h'_{1x} \hat{\psi}_1 + h'_{2x} \hat{\psi}_2) + h'_{2y} \cdot (h'_{1y} \hat{\psi}_1 + h'_{2y} \hat{\psi}_2) \end{bmatrix} \\ &= \begin{bmatrix} (h'_{1x} h'_{1x} + h'_{1y} h'_{1y}) \tilde{\varphi}_1 + (h'_{1x} h'_{2x} + h'_{1y} h'_{2y}) \tilde{\varphi}_2 \\ (h'_{1x} h'_{1x} + h'_{1y} h'_{1y}) \tilde{\psi}_1 + (h'_{1x} h'_{2x} + h'_{1y} h'_{2y}) \tilde{\psi}_2 \\ (h'_{2x} h'_{1x} + h'_{2y} h'_{1y}) \tilde{\varphi}_1 + (h'_{2x} h'_{2x} + h'_{2y} h'_{2y}) \tilde{\varphi}_2 \\ (h'_{2x} h'_{1x} + h'_{2y} h'_{1y}) \tilde{\psi}_1 + (h'_{2x} h'_{2x} + h'_{2y} h'_{2y}) \tilde{\psi}_2 \end{bmatrix} = \\ &\begin{bmatrix} h'_{1x} h'_{1x} + h'_{1y} h'_{1y} & 0 & h'_{1x} h'_{2x} + h'_{1y} h'_{2y} & 0 \\ 0 & h'_{1x} h'_{1x} + h'_{1y} h'_{1y} & 0 & h'_{1x} h'_{2x} + h'_{1y} h'_{2y} \\ h'_{2x} h'_{1x} + h'_{2y} h'_{1y} & 0 & h'_{2x} h'_{2x} + h'_{2y} h'_{2y} & 0 \\ 0 & h'_{2x} h'_{1x} + h'_{2y} h'_{1y} & 0 & h'_{2x} h'_{2x} + h'_{2y} h'_{2y} \end{bmatrix} \begin{bmatrix} \tilde{\varphi}_1 \\ \tilde{\psi}_1 \\ \tilde{\varphi}_2 \\ \tilde{\psi}_2 \end{bmatrix} \\ &= \begin{bmatrix} h'_{1x} & 0 & h'_{1y} & 0 \\ 0 & h'_{1x} & 0 & h'_{1y} \\ h'_{2x} & 0 & h'_{2y} & 0 \\ 0 & h'_{2x} & 0 & h'_{2y} \end{bmatrix} \begin{bmatrix} h'_{1x} & 0 & h'_{2x} & 0 \\ 0 & h'_{1x} & 0 & h'_{2x} \\ h'_{1y} & 0 & h'_{2y} & 0 \\ 0 & h'_{1y} & 0 & h'_{2y} \end{bmatrix} \begin{bmatrix} \tilde{\varphi}_1 \\ \tilde{\psi}_1 \\ \tilde{\varphi}_2 \\ \tilde{\psi}_2 \end{bmatrix} \end{aligned}$$



con lo que finalmente, podemos escribir los términos del producto escalar del gradiente de las funciones de forma y las funciones incógnita aproximadas como

$$= B^T \cdot B \cdot \hat{\phi}$$

donde B es, en el caso del ejemplo reducido para $i = 1, 2$ es la matriz de 4×4

$$B = \begin{bmatrix} h'_{1x} & 0 & h'_{2x} & 0 \\ 0 & h'_{1x} & 0 & h'_{2x} \\ h'_{1y} & 0 & h'_{2y} & 0 \\ 0 & h'_{1y} & 0 & h'_{2y} \end{bmatrix}$$

y, en el caso del ejemplo original con cinco puntos, es la matriz de 4×10

$$B = \begin{bmatrix} h'_{1x} & 0 & h'_{2x} & 0 & h'_{3x} & 0 & h'_{4x} & 0 & h'_{5x} & 0 \\ 0 & h'_{1x} & 0 & h'_{2x} & 0 & h'_{3x} & 0 & h'_{4x} & 0 & h'_{5x} \\ h'_{1y} & 0 & h'_{2y} & 0 & h'_{3y} & 0 & h'_{4y} & 0 & h'_{5y} & 0 \\ 0 & h'_{1y} & 0 & h'_{2y} & 0 & h'_{3y} & 0 & h'_{4y} & 0 & h'_{5y} \end{bmatrix}$$

de forma tal que se cumple que

$$B \cdot \hat{\phi} = \begin{bmatrix} \partial \tilde{\varphi} / \partial x \\ \partial \tilde{\psi} / \partial x \\ \partial \tilde{\varphi} / \partial y \\ \partial \tilde{\psi} / \partial y \end{bmatrix}$$

Para completar el ejemplo, la formulación matricial antes de poner las condiciones de contorno de Dirichlet queda

$$\left[\int_{-1}^1 \int_{-1}^1 B^T B \, dx \, dy + \int_{-1}^1 \int_{-1}^1 F \, dx \, dy \right] \cdot \hat{\phi} = \int_{-1}^1 \int_{-1}^1 \mathbf{R} \, dx \, dy$$

donde F es una matriz de 10×10

$$F = \begin{bmatrix} h_1 h_1 & 0 & h_1 h_2 & 0 & \dots & h_1 h_5 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ h_2 h_1 & 0 & h_2 h_2 & 0 & \dots & h_2 h_5 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_5 h_1 & 0 & h_5 h_2 & 0 & \dots & h_5 h_5 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

y \mathbf{R} es el vector de tamaño 10

$$\mathbf{R} = \begin{bmatrix} 0 \\ h_1 \\ 0 \\ h_2 \\ \vdots \\ h_5 \end{bmatrix}$$

recordando que las $h_k(\mathbf{x})$ son funciones del espacio. Luego de integrar en el dominio y reemplazar las filas correspondientes a los nodos que tienen condición de contorno, obtene-



mos un sistema lineal de ecuaciones de 10×10 de donde podemos calcular los parámetros ajustables $\hat{\phi}_k$ y $\hat{\psi}_k$ para $k = 1, \dots, 5$.

Lo que quisimos ilustrar con el ejemplo es que el caso de varias ecuaciones diferenciales multidimensionales acopladas—que es justamente el caso de la difusión de neutrones multigrupo—es tan general que es no es posible dar una formulación genérica para todos los problemas, ya que la forma de ordenar los parámetros ajustables en el vector incógnita no es única, como tampoco lo es la forma en la que aparecen las derivadas de las funciones escalares en las ecuaciones diferenciales. La mejor opción de discretizar un problema vectorial multidimensional depende de cada problema en particular.

De todas maneras, en esta serie de ejemplos hemos tratado de evaluar problemas topológicamente similares al problema de difusión de neutrones, por lo que en la formulación en elementos finitos, que es una generalización del método de Galerkin para mallas no estructuradas, de la ecuación de difusión multigrupo descrita en lo que resta de este capítulo utilizamos los conceptos obtenidos luego de discutir los ejemplos de esta sección. Debemos notar el hecho nada casual que las expresiones matriciales a las cuales arribamos en todos los ejemplos tienen la misma forma funcional.

4.2. El método de elementos finitos

Tal como hicimos en el capítulo anterior con volúmenes finitos, a continuación proponemos una de las tantas maneras en la que es posible discretizar el problema de difusión de neutrones a varios grupos de energía utilizando el método de elementos finitos. En particular, usando términos de la literatura de elementos finitos, utilizamos elementos isoparamétricos estándar de primer orden [24, 1, 25]. Volviendo una vez más a la ecuación (4.1), supongamos que tenemos definido un problema vectorial multidimensional sobre un cierto dominio $U \in \mathbb{R}^M$:

$$\mathcal{A}[\phi(\mathbf{x})] = \begin{bmatrix} \mathcal{A}_1[\phi(\mathbf{x})] \\ \mathcal{A}_2[\phi(\mathbf{x})] \\ \vdots \\ \mathcal{A}_G[\phi(\mathbf{x})] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.1)$$

con

$$\phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_G(\mathbf{x}) \end{bmatrix}$$

Al igual que en volúmenes finitos, en el método de elementos finitos aproximamos el dominio continuo $U \in \mathbb{R}^M$ por una yuxtaposición de una cantidad finita I de celdas de dimensión M definidas por hiperplanos. Los vértices de las celdas definen K nodos. El proceso de mallado es completamente análogo al método de volúmenes finitos. Sin embargo, por una cuestión de consistencia en la nomenclatura, llamamos a la celda i -ésima ahora elemento e_i (figura 4.1). La idea es utilizar funciones de forma locales a cada nodo sobre los elementos a los que define y tomar como parámetros ajustables los valores que toma la función incógnita en los K nodos.

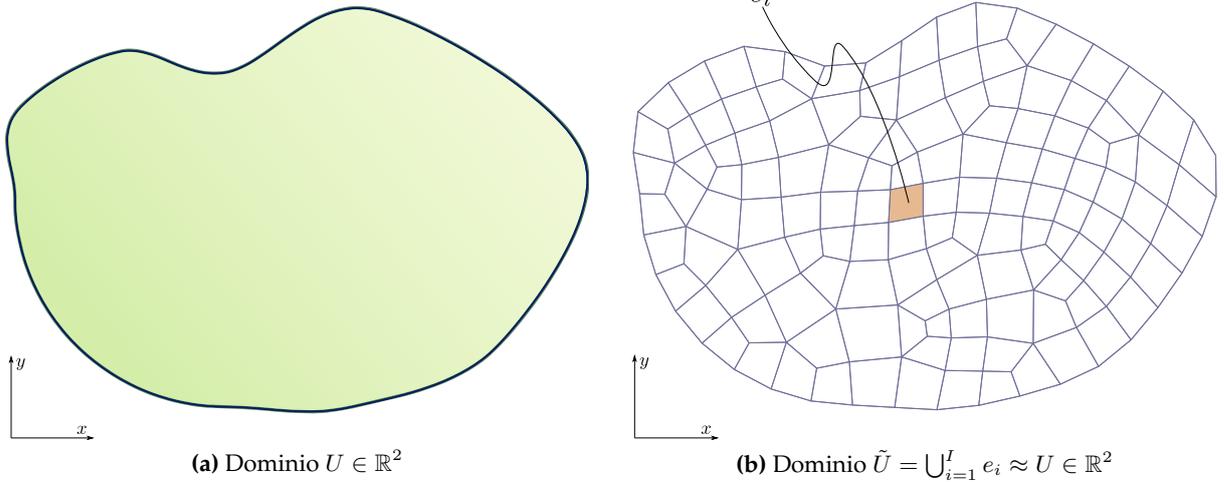


Figura 4.1 – Un dominio bidimensional $U \in \mathbb{R}^2$ es aproximado por la unión de I polígonos yuxtapuestos.

Como hemos dividido el dominio U en elementos yuxtapuestos, cualquier integral sobre todo el dominio es igual a la suma de las integrales sobre los elementos, a menos del error cometido al aproximar los bordes continuos por una cantidad finita de caras planas:

$$\int_U f(\mathbf{x}) d^M \mathbf{x} \approx \sum_{i=1}^I \int_{e_i} f(\mathbf{x}) d^M \mathbf{x} \quad (4.12)$$

Por otro lado, asumimos que podemos expresar las funciones incógnita aproximadas como una combinación lineal de K funciones de forma cuyos coeficientes son parámetros ajustables

$$\tilde{\phi}_g(\mathbf{x}) = \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_g^k$$

Más aún, asignamos a cada uno de los K nodos una función de forma y un parámetro ajustable. Requerimos que la función de forma $h_k(\mathbf{x}) = 1$ si \mathbf{x} coincide con la posición del nodo k y que $h_k(\mathbf{x}) = 0$ fuera de los elementos adyacentes al nodo k . Incluso, aún en los elementos adyacentes, $h_k(\mathbf{x})$ debe ser cero en los bordes opuestos al nodo k . En la figura 4.2 ilustramos una función de forma válida para un cierto nodo. En la sección 4.2.2 discutimos en detalle las propiedades de las funciones de forma y cómo calcularlas para una cierta malla en particular.

Como estamos interesados en realizar integrales—por ejemplo de residuos pesados—sobre todo el dominio aprovechando la ecuación (4.12), queremos poder separar la contribución de cada elemento a las funciones de forma $h_k(\mathbf{x})$. Cada una de las funciones de forma es entonces diferente de cero sólo en los elementos que contienen al nodo asignado. Podemos suponer entonces que la función de forma $h_k(\mathbf{x})$ está compuesta a su vez por funciones de forma elementales asociadas a un elemento en particular de forma tal que cada una de éstas es diferente de cero sólo en el elemento al que está asociada. Y, a la inversa, podemos pensar que cada elemento posee una cantidad de funciones de forma elementales, cada una de ellas asociada a cada uno de los nodos que lo definen. En la figura 4.3 mostramos, en una dimensión, cómo la función de forma $h_k(\mathbf{x})$ se puede pensar como la suma de dos funciones asociadas por un lado al mismo nodo k pero por otro a dos elementos i e i' diferentes (pero adyacentes).

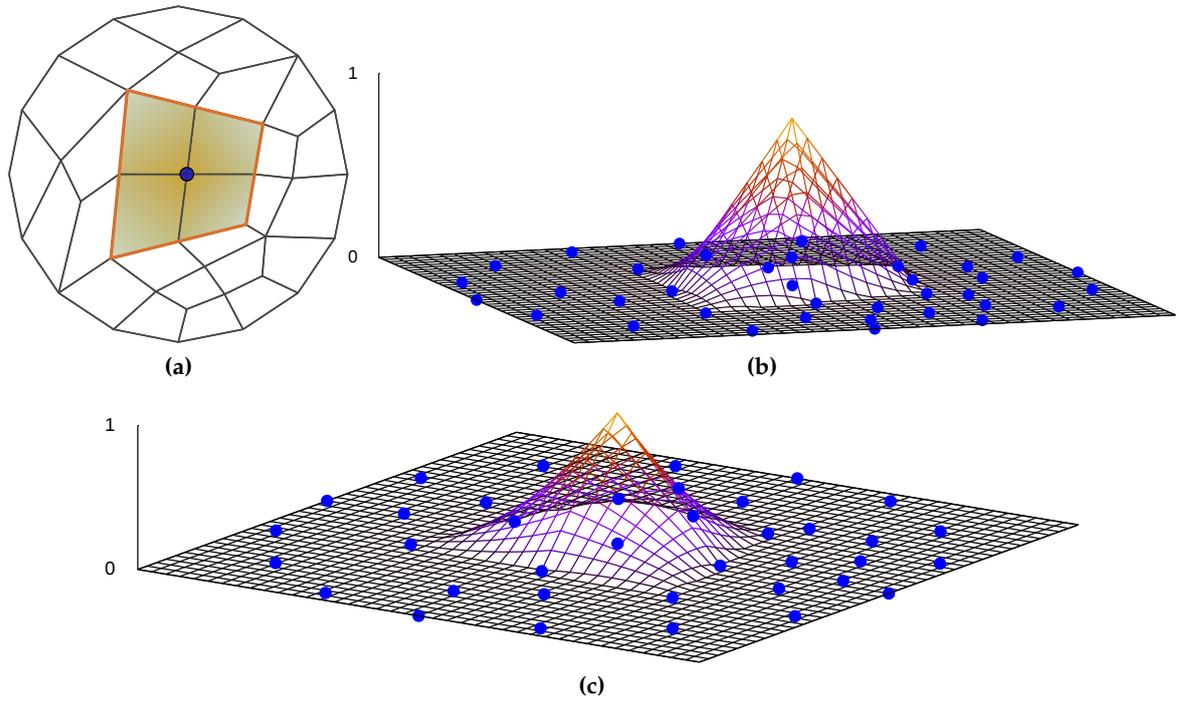


Figura 4.2 – Función de forma asociada a un nodo en un problema bidimensional. Dado un dominio discretizado (a) la función de forma $h_k(x, y)$ para el índice k del nodo central es sólo diferente de cero en los cuatro elementos sombreados. Más aún, debe ser cero en los bordes marcados. Las subfiguras (b) y (c) muestran una posible función de forma $h_k(x, y)$ para el nodo central, que vale uno en el nodo central y cero en todos los otros. La función incógnita buscada por el método de elementos finitos es una combinación lineal de K funciones de forma de este tipo que aproxime la solución real del problema diferencial continuo.

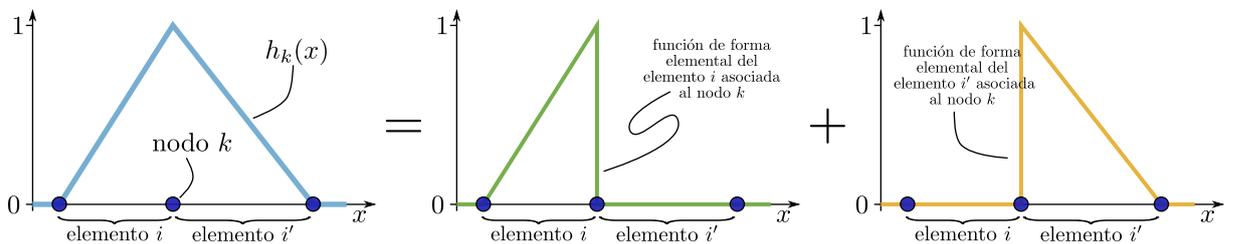


Figura 4.3 – La función de forma $h_k(x)$ puede ser pensada como la suma de varias funciones de forma elementales que son sólo diferente de cero dentro de cada uno de los elementos que tiene al nodo k como frontera. Por ejemplo, la función de forma unidimensional $h_k(x)$ mostrada en la figura puede ser pensada como la suma de dos funciones de forma elementales, una asociada al nodo k desde el elemento i y otra asociada al nodo k desde el elemento i' .



Para poder escribir cada una de las K funciones de forma $h_k(\mathbf{x})$ como suma de funciones de forma elementales en función de otro subíndice—i.e. ponerles nombres a las funciones verde y a la naranja de la figura 4.3—debemos definir antes una nomenclatura para los nodos dentro de cada uno de los elementos y cómo se relacionan esta numeración local con la denominación global dada por el índice k . Para fijar ideas, prestemos atención al caso de la figura 4.4. Tenemos allí un dominio bidimensional discretizado en $I = 7$ elementos indicados por el subíndice $i = 1, \dots, 7$ y $K = 13$ nodos indicados por el subíndice $k = 1, \dots, 13$. A su vez, cada elemento está definido por $J_i = 4$ nodos, que en este caso es la misma cantidad para todos pero no necesariamente todos los elementos tienen que tener la misma cantidad de nodos, por eso indicamos esta cantidad con el subíndice i ya que puede variar de elemento a elemento. Usamos entonces el subíndice $j = 1, \dots, n_i$ para hacer referencia al número de nodo local al elemento i que le corresponde al nodo global k . Por ejemplo, al nodo global $k = 12$ en el elemento $i = 4$ le corresponde $j = 4$, pero en el elemento $i = 5$ le corresponde $j = 3$. De los tres índices i, j y k sólo dos son independientes. Es decir, una vez fijados cualesquiera dos de ellos, el tercero queda automáticamente definido.

En este sentido, podemos escribir que cada una de las K funciones de forma $h_k(\mathbf{x})$ es igual a una suma de funciones elementales

$$h_k(\mathbf{x}) = \sum_i \mathfrak{h}_i^j(\mathbf{x}) \quad (4.13)$$

donde el índice i sólo toma los número de los elementos globales que contienen al nodo k . De este modo, cada elemento contribuye a la k -ésima función de forma a lo sumo con una única función de forma elemental. Por ejemplo, la función de forma correspondiente al nodo $k = 12$ de la figura 4.4 es igual a

$$h_{12}(x, y) = \mathfrak{h}_4^4(x, y) + \mathfrak{h}_3^3(x, y) + \mathfrak{h}_5^3(x, y) + \mathfrak{h}_6^2(x, y)$$

Si el problema tiene un sólo grado de libertad $G = 1$, entonces ya podemos escribir la función incógnita aproximada $\tilde{\phi}(\mathbf{x})$ como

$$\tilde{\phi}(\mathbf{x}) = \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_k \quad (4.14)$$

que podemos fácilmente asignar a un producto escalar entre un vector que contiene las K funciones de forma y otro que contiene los K parámetros ajustables. Sin embargo, si $G > 1$ entonces debemos incorporar un nuevo índice para tener en cuenta sobre qué grado de libertad aplica cada uno de los parámetros ajustables.

En efecto, como vimos en la sección 4.1.3, debemos construir un único vector $\hat{\phi}$ que contenga los $L = KG$ parámetros ajustables, ordenados de alguna manera. Por ejemplo, si

$$\hat{\phi} = \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_2^1 \\ \hat{\phi}_1^2 \\ \hat{\phi}_2^2 \\ \vdots \\ \hat{\phi}_k^g \\ \vdots \\ \hat{\phi}_M^G \end{bmatrix} \quad (4.15)$$

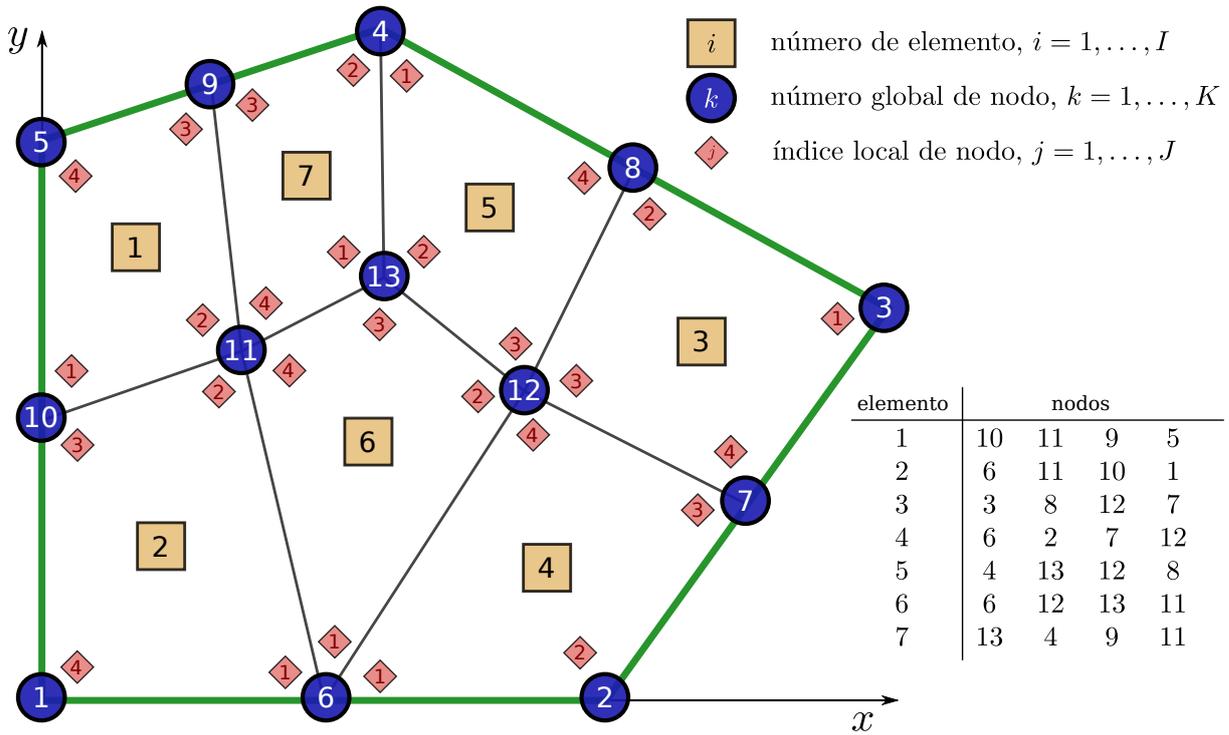


Figura 4.4 – Dominio bidimensional discretizado en trece nodos que definen siete elementos. Se muestra la numeración de los elementos $i = 1, \dots, 7$ y la numeración de los nodos $k = 1, \dots, 13$. Dentro de cada elemento, se indica el índice local asignado a cada nodo $j = 1, \dots, 4$. Se incluye la tabla de índices que indica, para cada elemento, los cuatro nodos que lo definen en su orden. El dominio fue discretizado utilizando el software libre gmsh [26], y tanto la geometría, como la numeración mostrada y la tabla de índices es la arrojada por gmsh.

i	j	g	k	ℓ	i	j	g	k	ℓ	i	j	g	k	ℓ	i	j	g	k	ℓ	i	j	g	k	ℓ
1	1	1	10	19	2	3	1	10	19	4	1	1	6	11	5	3	1	12	23	7	1	1	13	25
1	1	2	10	20	2	3	2	10	20	4	1	2	6	12	5	3	2	12	24	7	1	2	13	26
1	2	1	11	21	2	4	1	1	1	4	2	1	2	3	5	4	1	8	15	7	2	1	4	7
1	2	2	11	22	2	4	2	1	2	4	2	2	2	4	5	4	2	8	16	7	2	2	4	8
1	3	1	9	17	3	1	1	3	5	4	3	1	7	13	6	1	1	6	11	7	3	1	9	17
1	3	2	9	18	3	1	2	3	6	4	3	2	7	14	6	1	2	6	12	7	3	2	9	18
1	4	1	5	9	3	2	1	8	15	4	4	1	12	23	6	1	1	12	23	7	4	1	11	21
1	4	2	5	10	3	2	2	8	16	4	4	2	12	24	6	1	2	12	24	7	4	2	11	22
2	1	1	6	11	3	3	1	12	23	5	1	1	4	7	6	1	1	13	25					
2	1	2	6	12	3	3	2	12	24	5	1	2	4	8	6	1	2	13	26					
2	2	1	11	21	3	4	1	7	13	5	2	1	13	25	6	1	1	11	21					
2	2	2	11	22	3	4	2	7	14	5	2	2	13	26	6	1	2	11	22					

Tabla 4.1 – Tabla con la relación entre los índices i (número de elemento), j (número local de nodo), g (número de grado de libertad), k (número global de nodo) y ℓ (posición del parámetro ajustable en el vector de incógnita) para la discretización de la figura 4.4 y el orden propuesto en la ecuación (4.15) con $G = 2$ grados de libertad por nodo.



entonces la función incógnita aproximada asignada al grado de libertad g es ahora

$$\tilde{\phi}_g(\mathbf{x}) = \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_g^k = \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_\ell \quad (4.16)$$

donde ahora el subíndice ℓ —que depende de k y de g —indica la posición del parámetro ajustable $\hat{\phi}_g^k$ en el vector $\hat{\phi}$. Si $G = 1$, entonces $\ell = k$ y la ecuación (4.16) se transforma en la ecuación (4.14). Si $G > 1$, para el ordenamiento de la ecuación (4.15) existe una relación algebraica entre ℓ, k y g , a saber

$$\ell(k, g) = G \cdot (k - 1) + g$$

donde a su vez, k depende de i y de j . La tabla 4.1 muestra la relación entre los índices i, j, k, g y ℓ para la malla de la figura 4.4 con $G = 2$.

El vector de las G de funciones incógnita aproximadas es entonces

$$\tilde{\phi}(\mathbf{x}) = \begin{bmatrix} \tilde{\phi}_1(\mathbf{x}) \\ \tilde{\phi}_2(\mathbf{x}) \\ \vdots \\ \tilde{\phi}_G(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_{\ell(k,1)} \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_{\ell(k,2)} \\ \dots \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_{\ell(k,G)} \end{bmatrix} = H \cdot \hat{\phi}$$

donde H es una matriz de G filas y KG columnas

$$H = \begin{bmatrix} h_1 & 0 & \dots & 0 & h_2 & 0 & \dots & \dots & 0 \\ 0 & h_1 & \dots & 0 & 0 & h_2 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & h_1 & 0 & 0 & \dots & \dots & h_K \end{bmatrix}$$

es decir, conteniendo como elemento en la fila g y en la columna ℓ :

$$H_{g,\ell} = \begin{cases} h_k(\mathbf{x}) & \text{para el } k \text{ que corresponda al índice } \ell, \text{ si el índice } \ell \text{ corresponde a } g \\ 0 & \text{si el índice } \ell \text{ no corresponde al índice } g \end{cases}$$

Como según la ecuación (4.13) cada una de las $h_k(\mathbf{x})$ puede escribirse como suma de contribuciones elementales $h_i^j(\mathbf{x})$, entonces también podemos decir que la matriz H es una suma de contribuciones elementales

$$H = \sum_{i=1}^I \mathcal{H}_i \quad (4.17)$$

donde las matrices elementales \mathcal{H}_i también son de tamaño $G \times KG$ y sus elementos son

$$\mathcal{H}_{i,g,\ell} = \begin{cases} h_i^j(\mathbf{x}) & \text{para el } j \text{ que corresponda a los índices } \ell \text{ e } i, \text{ si } \ell \text{ corresponde a } g \\ 0 & \text{de lo contrario} \end{cases}$$

Luego

$$\tilde{\phi}(\mathbf{x}) = \left(\sum_{i=1}^I \mathcal{H}_i \right) \cdot \hat{\phi}$$



Podemos escribir entonces el residuo del operador diferencial vectorial \mathcal{A} del problema, ponderado con la función de peso $w(\mathbf{x})$ como

$$\int_U w(\mathbf{x}) \cdot \left\{ \mathcal{A} \left[\tilde{\phi}(\mathbf{x}) \right] \right\} d^M \mathbf{x} = \sum_{i=1}^I \int_{e_i} w(\mathbf{x}) \cdot \left\{ \mathcal{A} \left[\mathcal{H}_i \cdot \hat{\phi} \right] \right\} d^M \mathbf{x} = \mathbf{0}$$

De esta manera, cada función de peso $w(\mathbf{x})$ arroja G ecuaciones escalares. Como tenemos KG incógnitas, necesitamos elegir K funciones de peso independientes para obtener entonces un sistema de KG ecuaciones. Si utilizamos el método de Galerkin y tomamos como funciones de peso las funciones de forma, entonces tenemos

$$\mathbf{0} = \sum_{i=1}^I \int_{e_i} h_k(\mathbf{x}) \cdot \left\{ \mathcal{A} \left[\mathcal{H}_i \cdot \hat{\phi} \right] \right\} d^M \mathbf{x} \quad \text{para } k = 1, \dots, K$$

Pero como según la ecuación (4.13) el elemento i -ésimo contribuye a lo más con una única función de forma elemental, entonces la integral de $h_k(\mathbf{x})$ sobre e_i es igual o bien a cero si el nodo $k \notin e_i$ o bien a la integral de la función de forma elemental $h_i^j(\mathbf{x})$ para el j que corresponda al nodo $k \in e_i$. Luego las KG ecuaciones del método de elementos finitos de donde despejar los $L = KG$ parámetros ajustables $\hat{\phi}_\ell$ son

$$\mathbf{0} = \sum_i \int_{e_i} h_i^{j(i,k)}(\mathbf{x}) \cdot \left\{ \mathcal{A} \left[\mathcal{H}_i \cdot \hat{\phi} \right] \right\} d^M \mathbf{x} \quad \text{para } k = 1, \dots, M \quad (4.18)$$

donde la suma sobre i , al igual que en la ecuación (4.13), debemos tomar sólo sobre los elementos que contengan al nodo k .

Las K ecuaciones vectoriales (4.18), cada una de tamaño G , están escritas como suma de integrales sobre elementos e_i , cada uno de cuyos integrandos sólo depende de funciones elementales locales al elemento i . Esta expresión refleja que es posible escribir los residuos ponderados con el método de Galerkin sumando contribuciones elementales cada una de las cuales es calculada con información local a cada elemento. Si bien hemos deducido la ecuación (4.18) a partir de la formulación fuerte (diferencial) del problema dada en la ecuación (4.1), es posible obtener expresiones similares partiendo de una formulación débil (integral) del problema, que es lo que hacemos en la sección 4.4 con la ecuación de difusión de neutrones. Además, no hemos dicho aún cómo calcular las funciones de forma elementales $h_i^j(\mathbf{x})$ ni las matrices elementales \mathcal{H}_i . Pero la importancia de la expresión (4.18) radica en mostrar que, en general, es posible separar el método de Galerkin en contribuciones elementales independientes.

Notar que en volúmenes finitos, los G grados de libertad se calculan en las I celdas, mientras que en elementos finitos las incógnitas se calculan en los K nodos. Si bien I y K son del mismo orden, el tamaño L del problema a resolver en cada caso es ligeramente diferente.

4.2.1. Ensamble de matrices

En la sección anterior hemos intentado demostrar que el método de elementos finitos se basa en escribir el método de Galerkin como suma de contribuciones elementales, que en principio son independientes y podemos calcular siguiendo reglas bien definidas, que es la gran ventaja del método. Luego de calcular las matrices elementales del problema, debemos recuperar la suma sobre el dominio original para encontrar la solución al problema.



Si bien hasta ahora hemos dicho que el problema total es la suma de los problemas elementales, matemáticamente hablando una suma de matrices implica que todos los sumandos deben tener el mismo tamaño por lo que si escribimos la matriz global como suma de diez mil matrices de cien mil por cien mil cada una entonces no habría ninguna ganancia clara. Como afortunadamente la mayoría de los elementos de estas matrices son ceros, directamente es posible trabajar con matrices elementales pequeñas—del orden de diez por diez—y después aplicamos un proceso de ensamble para obtener la matriz del problema completo. Una explicación satisfactoria del proceso de ensamble necesita apoyarse en conceptos de la construcción de las matrices elementales, y viceversa, configurando una condición tautológica de la que se sale o bien leyendo iterativamente el capítulo completo hasta que haya quedado claro lo que el autor quiso decir o bien hasta que la cantidad de índices introducidos dispare una condición lateral para salir del bucle.

Como veremos más adelante, un problema lineal escrito como residuos ponderados toma la forma

$$\hat{\varphi} \cdot A \cdot \hat{\phi} = 0 \quad (4.19)$$

donde $\hat{\varphi}$ y $\hat{\phi}$ son vectores cuyo tamaño es el producto entre la cantidad K de nodos del dominio y de la cantidad G de grados de libertad del problema, y A es una matriz cuadrada de tamaño $KG \times KG$. El vector $\hat{\phi}$ contiene las incógnitas $\hat{\phi}_g^k$ del problema discretizado, ordenadas de alguna cierta manera, digamos según la ecuación (4.15). La ecuación (4.19) debe satisfacerse para cualesquiera valores que puedan tomar los elementos del vector $\hat{\varphi}$, que contiene valores asociados a nodos y a grados de libertad ordenados de la misma manera que el vector $\hat{\phi}$. Es decir, el elemento ℓ -ésimo de $\hat{\phi}$ es $\hat{\phi}_g^k$, con

$$\ell(k, g) = G \cdot (k - 1) + g$$

que es la relación entre los índices ℓ , g y k del ejemplo de la figura 4.4.

Como también ilustramos en los ejemplos que siguen y más adelante en la formulación en elementos del problema de difusión de neutrones (sección 4.4), en problemas lineales podemos escribir la contribución del elemento i como el producto

$$\hat{\varphi}_i \cdot A_i \cdot \hat{\phi}_i \quad (4.20)$$

donde $\hat{\varphi}_i$ y $\hat{\phi}_i$ son vectores cuyo tamaño es el producto entre la cantidad J de nodos del elemento i y de la cantidad G de grados de libertad del problema, y A_i es una matriz cuadrada de $N \times N$ con $N = JG$. Ahora el vector elemental $\hat{\phi}_i$ contiene las J incógnitas $\hat{\phi}_j^g$ del elemento i para $j = 1, \dots, J$ y $g = 1, \dots, G$, en algún cierto orden. De la misma manera que para los vectores globales, el elemento n -ésimo del vector $\hat{\phi}_i$ contiene la incógnita $\hat{\phi}_j^g$ del elemento i . Como ilustramos en la figura 4.4, cada nodo local identificado con un par de índices i, j tiene un nodo global asociado con un único índice k . De la misma manera, la posición n dentro de cada vector elemental tiene una posición ℓ global asociada. En general no existe una expresión algebraica que relacione ℓ y n sino que esta información la provee el mallador a partir de la decisión de cómo ordenar los nodos y los elementos en los que divide el dominio continuo original. De cualquier manera, denotamos con la expresión $\ell(i, n)$ a la relación entre el índice global ℓ a partir del índice local n del elemento i provista por el mallador bajo la forma de una tabla.

El proceso de ensamble consiste en construir la matriz A a partir de las I matrices A_i . Para ello, notamos que en la ecuación (4.19) el elemento de A_i en la fila p y en la columna q está multiplicado a izquierda por el elemento p del vector $\hat{\varphi}_i$ y a derecha por el elemento q del vector $\hat{\phi}_i$, es decir, resulta en el escalar

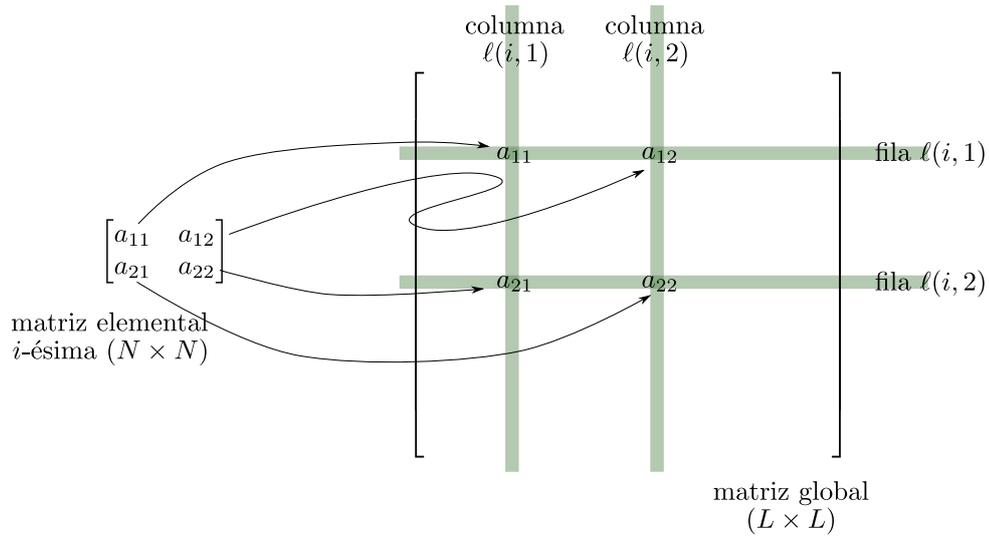


Figura 4.5 – Ensamble de matrices elementales A_i de 2×2 para construir la matriz A del problema. La función $\ell(i, n)$ indica el índice global $\ell = 1, \dots, L$ que corresponde al índice local $n = 1, \dots, N$ del elemento i . Cada uno de los índices tiene en cuenta el nodo y el grado de libertad que corresponden a la posición dentro del vector de incógnitas, tanto locales $\hat{\phi}_i$ como globales $\hat{\phi}$.

$$\hat{\phi}_p \cdot A_{i,p,q} \cdot \hat{\phi}_q$$

por lo que para mantener la formulación, el elemento $A_{i,p,q}$ debe aparecer en la fila $\ell(i, p)$ y en la columna $\ell(i, n)$ en la matriz global A . En la figura 4.5 ilustramos muy esquemáticamente el proceso de ensamblado de matrices elementales, que es mucho más difícil de explicar que de implementar computacionalmente.

4.2.2. Funciones de forma

En la sección 4.2 demostramos que es posible escribir la formulación integral de Galerkin de un problema diferencial como suma de términos que dependen sólo de funciones locales a cada elemento. Como los elementos son adyacentes y disjuntos, esta suma es una unión de N integrales independientes. Es por eso que podemos dividir tanto el problema como las funciones de forma y las funciones incógnita en contribuciones elementales locales a cada elemento e independientes del resto.

En cualquier caso, las funciones de forma deben satisfacer algunas condiciones básicas. La primera es que tienen que ser tales que el parámetro ajustable $\hat{\phi}_\ell = \hat{\phi}_g^k$ correspondiente al nodo k y al grado de libertad g , debe ser igual al valor que toma la función incógnita en la posición \mathbf{x}_k del nodo k . La segunda es que la función de forma asociada a un nodo debe ser diferente de cero sólo en los elementos que contienen a dicho nodo, y debe ser cero en los otros nodos, aún cuando estos estén en el mismo elemento. Estas dos propiedades implican que la función de forma $h_{k_0}(\mathbf{x})$ evaluada en el nodo particular k_0 debe ser igual a uno, ya que por un lado requerimos que

$$\tilde{\phi}_g(\mathbf{x}_{k_0}) = \hat{\phi}_g^{k_0}$$

y por otro



$$\begin{aligned}\tilde{\phi}_g(\mathbf{x}_{k_0}) &= \sum_{k=1}^K h_k(\mathbf{x}_{k_0}) \cdot \hat{\phi}_g^k \\ &= h_{k_0}(\mathbf{x}_{k_0}) \cdot \hat{\phi}_g^{k_0}\end{aligned}$$

ya que $h_k(\mathbf{x}_{k_0}) = 0$ si $k \neq k_0$. Luego

$$h_k(\mathbf{x}_{k'}) = \delta_{kk'}$$

La siguiente propiedad que se le pide a las funciones de forma es que si los parámetros ajustables son todos iguales a un cierto valor a , la función interpolada para cualquier posición \mathbf{x} debe evaluarse al valor a . Esto es,

$$\begin{aligned}\tilde{\phi}_g(\mathbf{x}) &= a \\ &= \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_g^k \\ &= \sum_{k=1}^K h_k(\mathbf{x}) \cdot a \\ &= \left(\sum_{k=1}^K h_k(\mathbf{x}) \right) \cdot a\end{aligned}$$

lo que implica que

$$\sum_{k=1}^J h_k(\mathbf{x}) = 1$$

para todo \mathbf{x} . Esta propiedad se conoce como partición de la unidad [24], y proviene del problema elástico donde se requiere que un desplazamiento de cuerpo rígido no cause campos de deformación. Como de costumbre, empezamos investigando el caso más sencillo que corresponde a elementos unidimensionales y luego generalizamos los conceptos para más dimensiones.

4.2.2.1. Elementos unidimensionales

En un problema unidimensional, un elemento es un segmento definido por dos nodos adyacentes ubicados sobre un eje, que sin perder generalidad suponemos es el x . En una dimensión, usualmente nodos consecutivos reciben índices también consecutivos, de la misma manera que elementos adyacentes tienen índices consecutivos como ilustramos en la figura 4.6.

Para obtener funciones de forma genéricas aplicables a para cada uno de los nodos locales a cada elemento proponemos un mapeo invertible $r \mapsto x$ entre la coordenada global $x \in [x_k, x_{k+1}]$ y una coordenada local $r \in [-1, 1]$, de forma tal que podamos escribir la integral de alguna función $f(x)$ sobre el elemento i como

$$\int_{e_i} f(x) dx = \int_{x_k}^{x_{k+1}} f(x) dx = \int_{-1}^1 f(r) \frac{dx}{dr} dr$$

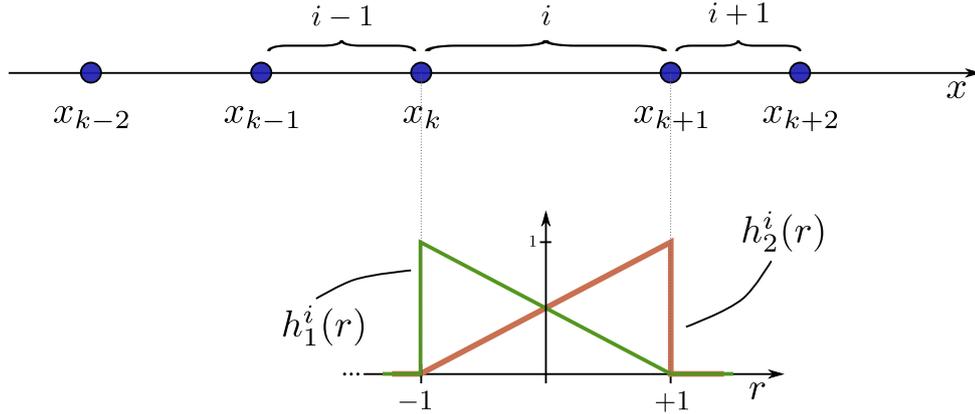


Figura 4.6 – Elemento isoparamétrico estándar unidimensional. Mapeo de la coordenada global x a la coordenada local r y funciones de forma locales del elemento i .

En el marco de referencia local, las funciones de forma locales de un elemento unidimensional lineal estándar son

$$\begin{aligned} h_1(r) &= \frac{1}{2}(1 - r) \\ h_2(r) &= \frac{1}{2}(1 + r) \end{aligned} \quad (4.21)$$

donde utilizamos la tipografía regular h en lugar de la alternativa \mathfrak{h} con la que diferenciábamos las funciones de forma globales de las elementales. Ahora la diferenciación la hacemos de acuerdo a los argumentos de la función: si el argumento es una coordenada local r la función de forma es local, y viceversa.¹

En efecto, las ecuaciones (4.21) evaluadas en $r = -1$ y en $r = +1$ dan uno y cero, respectivamente como requeríamos en la sección (4.2.2). Además,

$$h_1(r) + h_2(r) = \frac{(1 - r) + (1 + r)}{2} = 1 \quad \forall r$$

que es la otra propiedad requerida.

De todos los posibles mapeos inversibles $x \mapsto r$ elegimos aquel que hace que las funciones de forma (4.21) interpolen la coordenada x correctamente en el intervalo $[-1, 1]$ de la coordenada r . A riesgo de abusar de la notación, sean ahora $x_i^1 < x_i^2$ las coordenadas globales de los dos nodos que definen el elemento i . En el caso de la figura 4.6, $x_i^1 = x_k$ y $x_i^2 = x_{k+1}$. La condición de interpolar la coordenada x de la misma manera que la función incógnita

$$x = h_1(r) \cdot x_i^1 + h_2(r) \cdot x_i^2$$

es la característica isoparamétrica de los elementos utilizados.

En el caso unidimensional podemos escribir analíticamente el mapeo $x \mapsto r$ en ambas direcciones. En efecto,

¹Este nuevo abuso de notación proviene por un lado de la *gloriosa discretitud de nuestro alfabeto* [27] y por otro por la escasa disponibilidad de símbolos para representar funciones [28].



$$\begin{aligned}
x(r) &= \frac{1}{2}(1-r) \cdot x_i^1 + \frac{1}{2}(1+r) \cdot x_i^2 \\
&= \frac{1}{2}(1-r) \cdot x_k + \frac{1}{2}(1+r) \cdot x_{k+1} \\
&= \frac{1}{2}(x_k + x_{k+1}) + \frac{1}{2}(x_{k+1} - x_k) \cdot r
\end{aligned} \tag{4.22}$$

y el mapeo es lineal, cuya inversa analítica podemos calcular como

$$r(x) = \frac{2x - (x_k + x_{k+1})}{x_{k+1} - x_k} \tag{4.23}$$

Estas ideas quedan ilustradas en el siguiente ejemplo.

Ejemplo 6 Encontrar la expresión de la contribución del elemento i a las ecuaciones de Galerkin correspondientes a la expresión diferencial

$$\frac{d^2\phi}{dx^2} = 0$$

▷ Si $w(x)$ es la función de peso, el enunciado pide evaluar la expresión

$$\int_{e_i} w(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx$$

Integrando por partes, tenemos

$$- \int_{e_i} \frac{dw}{dx} \cdot \frac{d\tilde{\phi}}{dx} dx + w(x) \cdot \frac{d\tilde{\phi}}{dx} \Big|_{\partial e_i}$$

El término de superficie no es de interés, ya que en el eventual paso de sumar sobre todos los elementos, estos términos se cancelan de a pares y sólo sobreviven los términos de superficie correspondiente a los nodos que están en el borde del dominio y donde se deben aplicar las condiciones de contorno tal como discutimos en la sección (4.1), por lo que no los tenemos en cuenta en este ejemplo.

La función aproximada en función de la coordenada r es

$$\begin{aligned}
\tilde{\phi}(r) &= h_1(r) \cdot \hat{\phi}_1^i + h_2(r) \cdot \hat{\phi}_2^i \\
&= [h_1(r) \quad h_2(r)] \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} \\
&= H(r) \cdot \hat{\phi}
\end{aligned}$$

y su derivada con respecto a la coordenada global x es

$$\begin{aligned}
\frac{d\tilde{\phi}}{dx} &= \frac{dh_1}{dx} \cdot \hat{\phi}_1^i + \frac{dh_2}{dx} \cdot \hat{\phi}_2^i \\
&= [dh_1/dx \quad dh_2/dx] \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} \\
&= B_x(r) \cdot \hat{\phi}
\end{aligned}$$



Utilizando la regla de la cadena, podemos poner esta expresión en función de la derivada de las funciones de forma con respecto a la coordenada local r , para poder evaluarlas directamente a partir de las ecuaciones (4.21):

$$\begin{aligned}\frac{d\tilde{\phi}}{dx} &= \frac{dh_1}{dr} \frac{dr}{dx} \cdot \hat{\phi}_1^i + \frac{dh_2}{dr} \frac{dr}{dx} \cdot \hat{\phi}_2^i \\ &= [dh_1/dr \quad dh_2/dr] \cdot \frac{dr}{dx} \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} \\ &= B_r(r) \cdot \frac{dr}{dx} \cdot \hat{\phi}\end{aligned}$$

donde identificamos las matrices B_x y B_r , cuya relación es

$$B_x = B_r \cdot \frac{dr}{dx}$$

Poniendo como funciones de peso las funciones de forma $h_1(r)$ y $h_2(r)$, la expresión pedida es

$$\begin{bmatrix} \int_{e_i} h_1(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \\ \int_{e_i} h_2(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \end{bmatrix} = \begin{bmatrix} - \int_{e_i} \frac{dh_1}{dx} \cdot \frac{d\tilde{\phi}}{dx} dx \\ - \int_{e_i} \frac{dh_2}{dx} \cdot \frac{d\tilde{\phi}}{dx} dx \end{bmatrix} = - \int_{e_i} \begin{bmatrix} dh_1/dx \\ dh_2/dx \end{bmatrix} \cdot \frac{d\tilde{\phi}}{dx} dx$$

Transformando la integral sobre la variable global x a la variable local r , recordando que

$$\frac{d\tilde{\phi}}{dx} = B_x \cdot \hat{\phi}$$

tenemos

$$- \int_{e_i} \begin{bmatrix} dh_1/dx \\ dh_2/dx \end{bmatrix} \cdot \frac{d\tilde{\phi}}{dx} dx = - \int_{-1}^1 B_x^T B_x \cdot \hat{\phi} \cdot \frac{dx}{dr} dr$$

Como el vector $\hat{\phi}$ no depende del espacio, podemos sacarlo fuera de la integral para obtener finalmente

$$\begin{bmatrix} \int_{e_i} h_1(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \\ \int_{e_i} h_2(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \end{bmatrix} = - \left(\int_{-1}^1 B_x^T B_x \cdot \frac{dx}{dr} dr \right) \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix}$$

En forma alternativa, podemos escribir la expresión en función de la matrix B_r , con lo que obtenemos

$$\begin{bmatrix} \int_{e_i} h_1(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \\ \int_{e_i} h_2(x) \cdot \frac{d^2\tilde{\phi}}{dx^2} dx \end{bmatrix} = - \left(\int_{-1}^1 B_r^T B_r \cdot \frac{dr}{dx} dr \right) \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix}$$

que es más fácil de evaluar analíticamente ya que, para las funciones de forma (4.21), tenemos $dh_1/dr = -1/2$, $dh_2/dr = 1/2$ y a partir de la ecuación (4.23)



$$\frac{dr}{dx} = \frac{2}{x_2^i - x_1^i} = \frac{2}{\Delta x_i}$$

donde Δx_i es la longitud del elemento i . Finalmente,

$$\begin{aligned} - \left(\int_{-1}^1 B_r^T B_r \cdot \frac{dr}{dx} dr \right) \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} &= - \left(\int_{-1}^1 \begin{bmatrix} -1/2 \\ 1/2 \end{bmatrix} [-1/2 \quad 1/2] \cdot \frac{2}{\Delta x_i} dr \right) \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} \\ &= \frac{1}{\Delta x_i} \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} \end{aligned}$$

Este ejemplo, además de generalizar los ejemplos de la sección 4.1, muestra que el mapeo $x \mapsto r$ implica dos transformaciones. La primera conlleva expresar las derivadas espaciales con respecto a la coordenada global x —que es en las coordenadas en la que está planteado el problema diferencial original—en función de la coordenada local r . La segunda incluye cambiar el dominio de integración sobre la coordenada global a la coordenada local. En ambos casos, el factor de conversión involucra el jacobiano del mapeo inversible—o bien dr/dx o bien dx/dr —que en el caso de una dimensión es un escalar y que, por ser conforme, cumple

$$\frac{dr}{dx} = \left(\frac{dx}{dr} \right)^{-1}$$

En particular, dado que el mapeo propuesto para los elementos isoparamétricos es lineal, su jacobiano es una constante que podemos evaluar utilizando las ecuaciones (4.22) y (4.23).

Ejemplo 7 Resolver la ecuación diferencial

$$\frac{d^2 \phi}{dx^2} + 1 = 0$$

en el intervalo $0 < x < 1$ con las condiciones de contorno

$$\begin{aligned} \phi(0) &= \frac{1}{3} \\ \frac{d\phi}{dx} \Big|_{x=1} &= -\frac{2}{3} \end{aligned}$$

utilizando cinco nodos equiespaciados que definen cuatro elementos de longitud $L = 1/4$ (figura 4.7) utilizando elementos finitos, diferencias finitas y volúmenes finitos, cuya solución analítica es

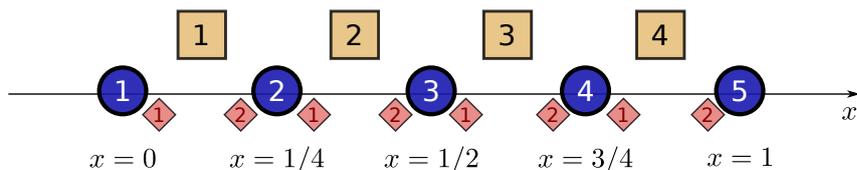


Figura 4.7 – Nodalización del intervalo $0 < x < 1$ para el ejemplo 7.



$$\phi(x) = -\frac{1}{2}x^2 + \frac{1}{3}x + \frac{1}{3}$$

▷ Si

$$\hat{\phi}^i = \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix}$$

es el vector elemental que contiene los valores nodales del elemento e_i , entonces la integral del método de Galerkin sobre el elemento i es

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \int_{-1}^1 \begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \left(\frac{d^2\phi}{dx^2} + 1 \right) \frac{dx}{dr} dr \\ &= \left[- \int_{-1}^1 (B_r^T \cdot B_r) \frac{dr}{dx} dr \right] \cdot \hat{\phi}^i + \left[\begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \cdot \frac{d\phi}{dx} \right]_{\partial e_i} + \int_{-1}^1 \begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \cdot \frac{dx}{dr} dr \end{aligned}$$

Al sumar sobre todos los elementos, los términos de superficie se cancelan de a pares excepto para $x = 0$ ($r = -1$ en el elemento 1) y para $x = 1$ ($r = +1$ en el elemento 4). Volvemos sobre estos términos más adelante. Podemos evaluar fácilmente la integral del último término de la ecuación ya que $dx/dr = \Delta x_i/2$ y la integral de cualquiera de las dos funciones de forma con respecto a r en el intervalo $[-1, 1]$ es igual a la unidad, ya que el área bajo la curva corresponde a la de un triángulo de base dos y altura uno. Luego

$$\int_{-1}^1 \begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \cdot \frac{dx}{dr} dr = \Delta x_i \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

Sin tener en cuenta por ahora los términos de superficie, podemos escribir la integral de Galerkin para cada uno de los elementos como

$$\frac{1}{\Delta x_i} \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1^i \\ \hat{\phi}_2^i \end{bmatrix} = -\Delta x_i \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

Al sumar las contribuciones de cada elemento y expresarlas en función del vector de incógnitas globales

$$\hat{\phi} = \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix}$$

debemos ensamblar una matriz global y un vector global que representen la formulación integral en todo el dominio (sección 4.2.1). De acuerdo a las reglas de numeración de nodos y de elementos elegida, el índice k —que coincide con el índice ℓ porque $G = 1$ —global del nodo j en el elemento i es

$$k = (i - 1) + j$$

Si R_j^i es el elemento ubicado en la columna j de algún vector en la contribución del elemento i , entonces R_j^i debe aparecer sumando en la fila $(i - 1) + j$ del vector global



$$R_j^i \mapsto R_{(i-1)+j} \quad (4.24)$$

De la misma manera, si $K_{p,q}^i$ es el elemento ubicado en la fila p y en la columna q de alguna matriz en la contribución del elemento i , entonces $K_{p,q}^i$ se debe sumar al elemento

$$K_{p,q}^i \mapsto K_{(i-1)+p,(i-1)+q} \quad (4.25)$$

de la matriz global correspondiente.

Con esto en mente, la formulación de Galerkin en todo el dominio suponiendo que todas los elementos tienen la misma longitud Δx y exceptuando por ahora los términos de superficie es

$$\frac{1}{\Delta x} \begin{bmatrix} -1 & +1 & 0 & 0 & 0 \\ +1 & -2 & +1 & 0 & 0 \\ 0 & +1 & -2 & +1 & 0 \\ 0 & 0 & +1 & -2 & +1 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = -\Delta x \begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix}$$

Los dos términos de superficie son, para $x = 0$ ($r = -1$ en el elemento 1)

$$\left(\begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \cdot \frac{d\tilde{\phi}}{dx} \right)_{x=0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=0} = \begin{bmatrix} \phi'(0) \\ 0 \end{bmatrix}$$

y para $x = 1$ ($r = +1$ en el elemento 4)

$$\left(\begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} \cdot \frac{d\tilde{\phi}}{dx} \right)_{x=1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{d\tilde{\phi}}{dx} \Big|_{x=1} = \begin{bmatrix} 0 \\ -2/3 \end{bmatrix}$$

Estos dos términos contribuyen al término independiente de la formulación en $\ell = 1$ y en $\ell = 5$, para obtener

$$\frac{1}{\Delta x} \begin{bmatrix} -1 & +1 & 0 & 0 & 0 \\ +1 & -2 & +1 & 0 & 0 \\ 0 & +1 & -2 & +1 & 0 \\ 0 & 0 & +1 & -2 & +1 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = -\Delta x \begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix} + \begin{bmatrix} \phi'(0) \\ 0 \\ 0 \\ 0 \\ +2/3 \end{bmatrix}$$

Aunque aparece el símbolo $\phi'(0)$ que representa la derivada de la función incógnita en $x = 0$, aún resta por poner la condición de contorno de Dirichlet en $x = 0$, que corresponde a $\ell = 1$. Tal como vimos en los ejemplos de la sección 4.1, la ecuación donde aparece este símbolo desconocido es sobrescrita por la condición de contorno de Dirichlet y desaparece de la formulación. Efectivamente, si requerimos que $\hat{\phi}_1 = 1/3$, entonces el sistema de ecuaciones queda

$$\begin{bmatrix} +1 & 0 & 0 & 0 & 0 \\ +1 & -2 & +1 & 0 & 0 \\ 0 & +1 & -2 & +1 & 0 \\ 0 & 0 & +1 & -2 & +1 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = \begin{bmatrix} 1/3 \\ -\Delta x^2 \\ -\Delta x^2 \\ -\Delta x^2 \\ -\Delta x^2/2 + 2/3 \cdot \Delta x \end{bmatrix}$$

de donde, para $\Delta x = 1/4$, podemos calcular los valores numéricos de los cinco parámetros ajustables que definen la función incógnita $\tilde{\phi}(x)$, que resultan



$$\hat{\phi} = \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = \begin{bmatrix} 0,33333 \\ 0,38542 \\ 0,37500 \\ 0,30208 \\ 0,16667 \end{bmatrix}$$

Si utilizamos diferencias finitas para resolver el mismo problema, obtenemos la siguiente formulación matricial:

$$\begin{bmatrix} +1 & 0 & 0 & 0 & 0 \\ +1 & -2 & +1 & 0 & 0 \\ 0 & +1 & -2 & -1 & 0 \\ 0 & 0 & +1 & -2 & +1 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \\ \hat{\phi}_5 \end{bmatrix} = \begin{bmatrix} 1/3 \\ -\Delta x^2 \\ -\Delta x^2 \\ -\Delta x^2 \\ +2/3 \cdot \Delta x \end{bmatrix}$$

y para volúmenes finitos, el problema discretizado queda

$$\begin{bmatrix} -3 & +1 & 0 & 0 \\ +1 & -2 & +1 & 0 \\ 0 & +1 & -2 & +1 \\ 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\phi}^1 \\ \hat{\phi}^2 \\ \hat{\phi}^3 \\ \hat{\phi}^4 \end{bmatrix} = \begin{bmatrix} -\Delta x^2 - 2 \cdot 1/3 \\ -\Delta x^2 \\ -\Delta x^2 \\ -\Delta x^2 + 2/3 \Delta x \end{bmatrix}$$

donde en este caso las incógnitas son los valores medios sobre cada una de las cuatro celdas (elementos) del dominio.

Este ilustrativo ejemplo muestra una primera aproximación al núcleo del trabajo, que es la comparación entre volúmenes y elementos finitos. Además, introducimos por primera vez el concepto de ensamblaje, que es clave en la implementación de elementos finitos y cuya discusión dejamos para la sección 4.2.1. De cualquier manera, no tiene sentido emplear elementos o volúmenes para resolver una ecuación escalar unidimensional con coeficientes constantes en una malla uniforme. La importancia de los métodos integrales reside en su capacidad de representar satisfactoriamente por un lado casos donde los operadores de la ecuación diferencial dependen del espacio y por otro mallas con geometrías no triviales. En este caso en particular vemos que, si no fuese por las condiciones de contorno de Neumann (en verdad la función que la satisface es la interpolación lineal de los puntos discretos que arroja el método de diferencias finitas) en $x = 1$, la discretización en elementos finitos isoparamétricos lineales coincide con diferencias finitas. Vemos la discrepancia en la última ecuación, donde en diferencias se utiliza toda la ecuación para pedir que la derivada discreta sea igual a la condición de contorno, mientras que en elementos se reparte una parte de la ecuación para la condición de contorno y otra parte para satisfacer la ecuación diferencial en el elemento. Esto resulta en que estrictamente, la solución en diferencias satisface la condición de Neumann mientras que la solución en elementos no lo hace. Sin embargo, es claro que la solución en elementos es superior a la solución en diferencias en este caso sencillo. De todas formas, la calidad de la solución en diferencias aumenta con la cantidad de nodos utilizados y dicha solución se aproxima asintóticamente a la analítica. La solución en volúmenes da como resultado los valores medios en las celdas, que se suponen constantes dentro de cada una. Los valores medios se acercan al valor de la solución analítica en el centro de la celda y el promedio de dos celdas vecinas, que podemos asignar al

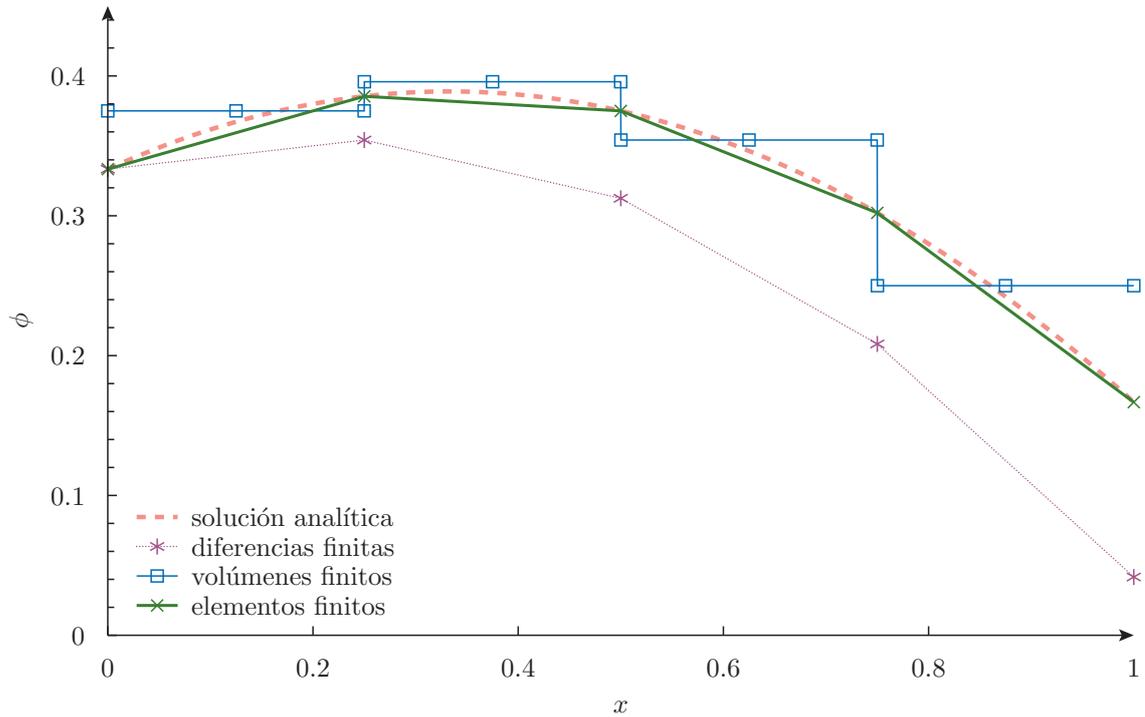


Figura 4.8 – Resultados del ejemplo 7. La solución en diferencias no indica explícitamente cuál es la solución del problema entre los nodos, pero asumimos una interpolación lineal entre ellos. La solución en elementos no satisface exactamente la condición de contorno de Neumann, pero aún con pocos nodos arroja resultados satisfactorios al compararla con la solución analítica. La solución en volúmenes asume que el flujo es uniforme en cada celda, pero tanto su valor medio como el promedio entre los valores de celdas vecinas se acercan a la solución analítica.

valor de la función en el borde, son similares al valor que toma la función analítica en dichos puntos. En el capítulo 5 desarrollamos estas ideas en profundidad.

4.2.2.2. Elementos bidimensionales

En dos dimensiones, los elementos son figuras planas definidas por polígonos de una cantidad arbitraria de lados. En esta monografía, sólo trabajamos con elementos triangulares y cuadrangulares. Comenzamos estudiando el caso de los cuadrángulos, cuyo mapeo al intervalo $[-1, -1] \times [+1, +1]$ es más sencillo.

Cuadrángulos

Dado un elemento cuadrangular i , definimos un mapeo inversible entre el plano de coordenadas globales x, y y el plano de coordenadas locales r, s (figura 4.9) de forma tal que

$$\begin{aligned}
 x(r, s) &= \sum_{j=1}^4 h_j(r, s) \cdot x_j \\
 y(r, s) &= \sum_{j=1}^4 h_j(r, s) \cdot y_j
 \end{aligned}
 \tag{4.26}$$

con las funciones de forma locales

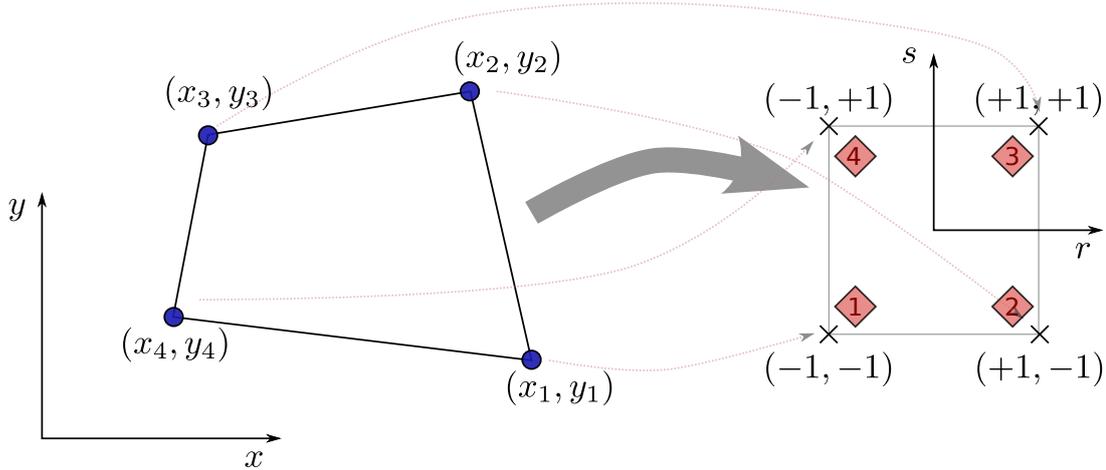


Figura 4.9 – Elemento isoparamétrico estándar cuadrangulo y mapeo de las coordenadas globales x, y a las coordenada locales r, s .

$$\begin{aligned}
 h_1(r, s) &= \frac{1}{4}(1 - r)(1 - s) \\
 h_2(r, s) &= \frac{1}{4}(1 + r)(1 - s) \\
 h_3(r, s) &= \frac{1}{4}(1 + r)(1 + s) \\
 h_4(r, s) &= \frac{1}{4}(1 - r)(1 + s)
 \end{aligned} \tag{4.27}$$

que cumplen los requerimientos de que su suma sea idénticamente igual a uno en el intervalo $[-1, -1] \times [+1, +1]$ y que cada una de ellas valga uno en el nodo j y cero en los otros tres. En la figura 4.10 ilustramos las funciones $h_1(r, s)$ y $h_4(r, s)$.

Con este mapeo estamos requiriendo que las coordenadas espaciales se interpolen de la misma manera que las funciones incógnita, dando lugar a la designación de elementos isoparamétricos [24]. Combinando las ecuaciones (4.26) y (4.27) obtenemos una expresión explícita para el mapeo $(r, s) \mapsto (x, y)$

$$\begin{aligned}
 x(r, s) &= \frac{1}{4} \left[(x_1 + x_2 + x_3 + x_4) + s \cdot (-x_1 - x_2 + x_3 + x_4) \right. \\
 &\quad \left. + r \cdot (-x_1 + x_2 + x_3 - x_4) + rs \cdot (x_1 - x_2 + x_3 - x_4) \right] \\
 y(r, s) &= \frac{1}{4} \left[(y_1 + y_2 + y_3 + y_4) + s \cdot (-y_1 - y_2 + y_3 + y_4) \right. \\
 &\quad \left. + r \cdot (-y_1 + y_2 + y_3 - y_4) + rs \cdot (y_1 - y_2 + y_3 - y_4) \right]
 \end{aligned} \tag{4.28}$$

Que este mapeo es inversible en el intervalo $[-1, 1] \times [-1, 1]$ no es evidente. Para que esto ocurra, el jacobiano de x e y con respecto a r y a s debe tener inversa para cualquier $(r, s) \in [-1, 1] \times [-1, 1]$. Dichas derivadas depende de las coordenadas x_j e y_j de los nodos que definen el elemento, por lo que básicamente es responsabilidad del proceso de mallado generar nodos tales que el determinante del jacobiano no se anule en el intervalo de interés. De cualquier manera, el cálculo de esta inversa no es necesaria para emplear el método de elementos finitos.

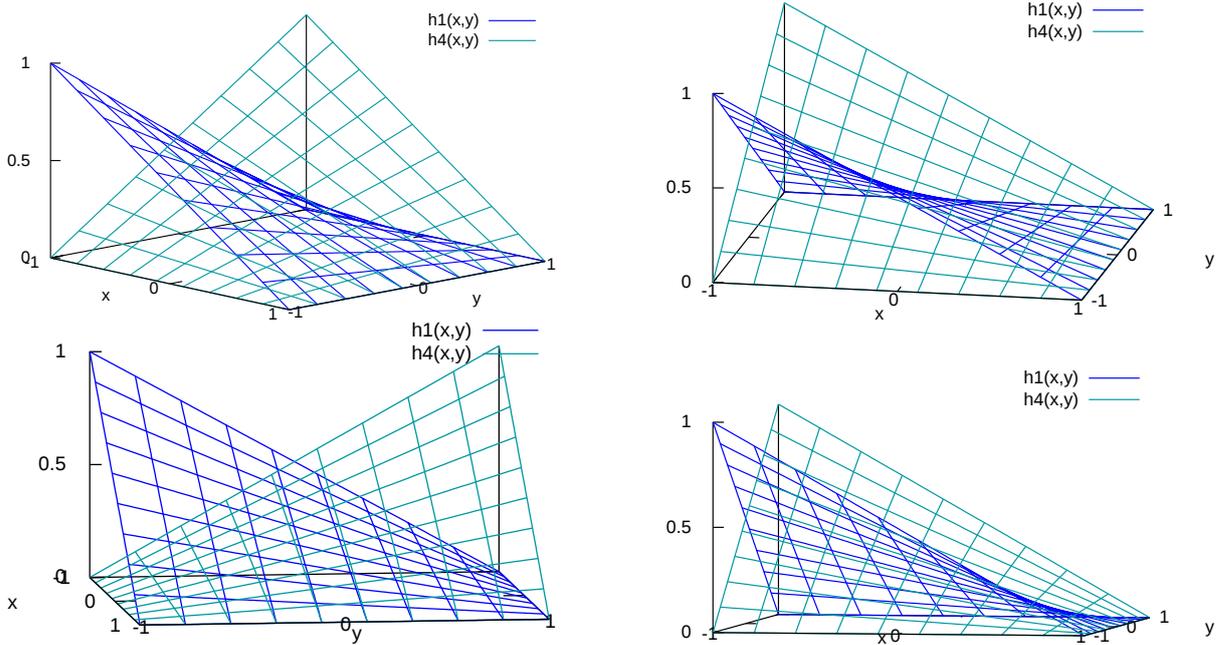


Figura 4.10 – Funciones de forma $h_1(r, s)$ y $h_4(r, s)$ en el intervalo $[-1, -1] \times [+1, +1]$.

Incluso tampoco es necesario calcular explícitamente la transformación directa utilizando las ecuaciones (4.28). En efecto, podemos escribir la integral de alguna función $f(x, y)$ sobre el elemento i como

$$\int_{e_i} f(x, y) \, dx \, dy = \int_{-1}^1 \int_{-1}^1 f(x(r, s), y(r, s)) \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| \, dr \, ds$$

donde $\left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right]$ es el jacobiano de la transformación $(x, y) \mapsto (r, s)$

$$\left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^4 \frac{\partial h_j}{\partial r} \cdot x_j & \sum_{j=1}^4 \frac{\partial h_j}{\partial s} \cdot x_j \\ \sum_{j=1}^4 \frac{\partial h_j}{\partial r} \cdot y_j & \sum_{j=1}^4 \frac{\partial h_j}{\partial s} \cdot y_j \end{bmatrix} \quad (4.29)$$

En caso de que aparezcan derivadas de las coordenadas locales con respecto a las coordenadas globales dentro de la función $f(x, y)$ —tal como aparecía la expresión dr/dx en la sección 4.2.2.1—podemos aún calcularlas sin necesidad de conocer la transformación inversa $(x, y) \mapsto (r, s)$ —como sí la conocíamos en la sección 4.2.2.1 a través de la ecuación (4.23)—notando que, por un lado

$$\begin{bmatrix} dr \\ ds \end{bmatrix} = \begin{bmatrix} \partial r / \partial x & \partial r / \partial y \\ \partial s / \partial x & \partial s / \partial y \end{bmatrix} \cdot \begin{bmatrix} dx \\ dy \end{bmatrix}$$

y por otro

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \partial x / \partial r & \partial x / \partial s \\ \partial y / \partial r & \partial y / \partial s \end{bmatrix} \cdot \begin{bmatrix} dr \\ ds \end{bmatrix}$$

con lo que



$$\begin{bmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial x} & \frac{\partial s}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1}$$

Utilizando la conocida fórmula de la inversa de una matriz de 2×2

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (4.30)$$

entonces podemos calcular las derivadas de las coordenadas locales con respecto a las globales (que son difíciles de calcular explícitamente) utilizando sólo las derivadas de x e y con respecto a r y a s (que son fáciles de calcular) como

$$\begin{aligned} \frac{\partial r}{\partial x} &= \frac{\frac{\partial y}{\partial s}}{\frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial x}{\partial s} \frac{\partial y}{\partial r}} \\ \frac{\partial r}{\partial y} &= \frac{-\frac{\partial x}{\partial s}}{\frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial x}{\partial s} \frac{\partial y}{\partial r}} \\ \frac{\partial s}{\partial x} &= \frac{-\frac{\partial y}{\partial r}}{\frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial x}{\partial s} \frac{\partial y}{\partial r}} \\ \frac{\partial s}{\partial y} &= \frac{\frac{\partial x}{\partial r}}{\frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial x}{\partial s} \frac{\partial y}{\partial r}} \end{aligned} \quad (4.31)$$

Ejemplo 8 Transformar la integral

$$\int_{e_i} \nabla \tilde{\varphi} \cdot \nabla \tilde{\phi} \, dx \, dy$$

sobre un elemento e_i cuadrangular al intervalo $-1 < r < 1 \times -1 < s < 1$, donde $\tilde{\varphi}(x, y)$ y $\tilde{\phi}(x, y)$ son funciones que toman los valores nodales $\tilde{\varphi}_j$ y $\tilde{\phi}_j$ interpolados por las mismas funciones de forma $h_j(r, s)$ con $j = 1, \dots, 4$ para ambas funciones

▷ En notación matricial, podemos escribir



$$\tilde{\varphi}(x, y) = \tilde{\varphi}(r, s) = \begin{bmatrix} h_1(r, s) & h_2(r, s) & h_3(r, s) & h_4(r, s) \end{bmatrix} \begin{bmatrix} \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \hat{\varphi}_3 \\ \hat{\varphi}_4 \end{bmatrix}$$

$$= H \cdot \hat{\varphi}$$

$$\tilde{\phi}(x, y) = \tilde{\phi}(r, s) = \begin{bmatrix} h_1(r, s) & h_2(r, s) & h_3(r, s) & h_4(r, s) \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \end{bmatrix}$$

$$= H \cdot \hat{\phi}$$

$$\nabla \tilde{\varphi} = \begin{bmatrix} \partial \varphi / \partial x \\ \partial \varphi / \partial y \end{bmatrix} = \begin{bmatrix} \partial h_1 / \partial x & \partial h_2 / \partial x & \partial h_3 / \partial x & \partial h_4 / \partial x \\ \partial h_1 / \partial y & \partial h_2 / \partial y & \partial h_3 / \partial y & \partial h_4 / \partial y \end{bmatrix} \cdot \begin{bmatrix} \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \hat{\varphi}_3 \\ \hat{\varphi}_4 \end{bmatrix}$$

$$= B_x \cdot \hat{\varphi}$$

$$\nabla \tilde{\phi} = \begin{bmatrix} \partial \phi / \partial x \\ \partial \phi / \partial y \end{bmatrix} = \begin{bmatrix} \partial h_1 / \partial x & \partial h_2 / \partial x & \partial h_3 / \partial x & \partial h_4 / \partial x \\ \partial h_1 / \partial y & \partial h_2 / \partial y & \partial h_3 / \partial y & \partial h_4 / \partial y \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \end{bmatrix}$$

$$= B_x \cdot \hat{\phi}$$

El diferencial de superficie $dx dy$ en función de $dr ds$ es

$$\begin{aligned} dx dy &= \left| \det \begin{pmatrix} \partial x / \partial r & \partial x / \partial s \\ \partial y / \partial r & \partial y / \partial s \end{pmatrix} \right| dr ds \\ &= \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| \cdot dr ds \end{aligned}$$

donde tomamos el valor absoluto para evitar diferenciales de superficie negativos que pueden llegar a aparecer si la numeración de los nodos en el plano global xy tiene una orientación contraria a la del plano local rs . Con estas definiciones, escribimos la integral como

$$\begin{aligned} \int_{e_i} \nabla \tilde{\varphi} \cdot \nabla \tilde{\phi} dx dy &= \int_{-1}^1 \int_{-1}^1 (B_x \hat{\varphi})^T \cdot (B_x \hat{\phi}) \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| dr ds \\ &= \int_{-1}^1 \int_{-1}^1 \hat{\varphi} \cdot B_x^T B_x \cdot \hat{\phi} \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| dr ds \\ &= \hat{\varphi} \cdot \left(\int_{-1}^1 \int_{-1}^1 B_x^T B_x \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| dr ds \right) \cdot \hat{\phi} \end{aligned}$$

De la misma manera que en el ejemplo 6, la matrix B_x involucra derivadas con respecto a las coordenadas globales—ya que proviene de discretizar una ecuación diferencial que contiene un gradiente con respecto a x y a y , que son las variables del problema—que son difíciles de calcular. Pero podemos avanzar un paso más y notar que



$$\begin{aligned}
B_x &= \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_2}{\partial x} & \frac{\partial h_3}{\partial x} & \frac{\partial h_4}{\partial x} \\ \frac{\partial h_1}{\partial y} & \frac{\partial h_2}{\partial y} & \frac{\partial h_3}{\partial y} & \frac{\partial h_4}{\partial y} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial h_1}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial h_1}{\partial s} \frac{\partial s}{\partial x} & \frac{\partial h_2}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial h_2}{\partial s} \frac{\partial s}{\partial x} & \dots \\ \frac{\partial h_1}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial h_1}{\partial s} \frac{\partial s}{\partial y} & \frac{\partial h_2}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial h_2}{\partial s} \frac{\partial s}{\partial y} & \dots \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial h_1}{\partial r} & \frac{\partial h_2}{\partial r} & \frac{\partial h_3}{\partial r} & \frac{\partial h_4}{\partial r} \\ \frac{\partial h_1}{\partial s} & \frac{\partial h_2}{\partial s} & \frac{\partial h_3}{\partial s} & \frac{\partial h_4}{\partial s} \end{bmatrix} \\
&= \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \right)^T \cdot B_r
\end{aligned}$$

con lo que podemos re-escribir

$$\begin{aligned}
B_x^T \cdot B_x &= \left(\left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \right)^T \cdot B_r \right)^T \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \right)^T B_r \\
&= B_r^T \cdot \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \right)^T \cdot B_r
\end{aligned}$$

para obtener la integral como

$$\hat{\varphi} \cdot \left(\int_{-1}^1 \int_{-1}^1 B_r^T \cdot \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}^{-1} \right)^T \cdot B_r \cdot \left| \det \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix} \right| dr ds \right) \cdot \hat{\varphi}$$

en función de la matriz de derivadas con respecto a las variables locales B_r , de la matriz jacobiana directa $\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{bmatrix}$ y de su inversa, todas fáciles de calcular ya que las derivadas de las \mathbf{x} con respecto a las \mathbf{r} son combinaciones lineales de las funciones de forma $h_j(\mathbf{x})$ y las matrices a invertir tienen un tamaño igual a $M \times M$ con $M = 1, 2$ ó 3 .

Triángulos

Para manejar elementos triangulares, utilizamos las funciones de forma que resultan de colapsar los nodos locales 3 y 4 de un elemento cuadrangular en un único nodo, como ilustramos en la figura 4.11. En este caso, en el plano (r, s) la recta $r = +1$ deja de ser vertical y desaparece la recta $s = +1$. Cualquier punto de la recta $s = +1$ es singular. Sin embargo, dado que sólo intervienen las integrales de expresiones de r y de s , cuyo valor numérico podemos calcular tomando puntos dentro del intervalo $[-1, 1] \times [-1, 1]$ sin llegar a la necesidad de evaluar los integrandos en los valores extremos.

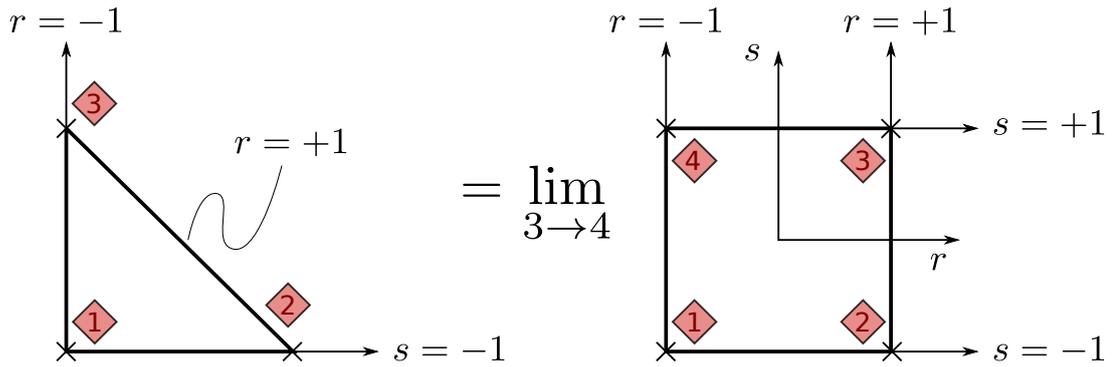


Figura 4.11 - Obtención de las funciones de forma para un elemento triangular mediante el colapso de dos nodos de un elemento cuadrangular.

De esta manera, las funciones de forma para los nodos locales 1 y 2 son iguales que para el caso de cuadrángulos dadas por las ecuaciones (4.27)

$$h_1(r, s) = \frac{1}{4}(1 - r)(1 - s)$$

$$h_2(r, s) = \frac{1}{4}(1 + r)(1 - s)$$

Para el nodo local 3, la función de forma es la suma de las funciones de forma para los nodos 3 y 4 de los cuadrángulos:

$$h_3(r, s) = \frac{1}{4}(1 - r)(1 + s) + \frac{1}{4}(1 + r)(1 + s) \quad (4.32)$$

$$= \frac{1}{2}(1 + s) \quad (4.33)$$

Es trivial verificar que estas tres funciones de forma cumplen las condiciones de partición de la unidad y de representar correctamente los valores nodales

$$\sum_{j=1}^3 h_j(r, s) = 1$$

$$h_j(r_{j'}, s_{j'}) = \delta_{jj'}$$

4.3. Formulación débil de la ecuación de difusión de neutrones

La ecuación de difusión de neutrones multigrupo (2.3) introducida en el capítulo 2

$$0 = \text{div} \left[D(\mathbf{x}, g) \cdot \text{grad} [\phi(\mathbf{x}, g)] \right] - \Sigma_t(\mathbf{x}, g) \cdot \phi(\mathbf{x}, g)$$

$$+ \sum_{g'=1}^G \Sigma_s(\mathbf{x}, g' \rightarrow g) \cdot \phi(\mathbf{x}, g') + \chi(g) \sum_{g'=1}^G \frac{\nu \Sigma_f(\mathbf{x}, g')}{k_{\text{eff}}} \cdot \phi(\mathbf{x}, g') \quad (2.3)$$

es un sistema de G ecuaciones lineales en derivadas parciales de segundo orden. Como discutimos en la sección 2.2.1, si escrita de esta manera, decimos que la ecuación está formulada en su



forma fuerte. Esta familia de ecuaciones admite también una formulación débil, que usualmente se la asocia con principios variacionales que implican una serie de interesantes interpretaciones filosóficas [29, 30]. En este trabajo, nos concentramos sólo en las características prácticas de la matemática asociada a la formulación débil y su aplicación a la discretización basada en elementos finitos. A continuación proponemos una formulación débil con la cual arribaremos a una formulación que, como demostramos en la sección 4.5, es equivalente al método de Galerkin.

Sean G funciones arbitrarias $\varphi(\mathbf{x}, g)$ para $g = 1, \dots, G$ definidas sobre el dominio espacial $U \in \mathbb{R}^M$ donde vamos a resolver la ecuación de difusión multigrupo (2.3). Primero multiplicamos la ecuación de difusión del grupo g por $\varphi(\mathbf{x}, g)$, luego sumamos término a término las G ecuaciones y finalmente integramos sobre el dominio U para obtener

$$0 = \int_U \left\{ \sum_{g=1}^G \varphi(\mathbf{x}, g) \cdot \left[\operatorname{div} \left(D(\mathbf{x}, g) \cdot \operatorname{grad} \phi(\mathbf{x}, g) \right) - \Sigma_t(\mathbf{x}, g) \cdot \phi(\mathbf{x}, g) + \sum_{g'=1}^G \Sigma_s(\mathbf{x}, g' \rightarrow g) \cdot \phi(\mathbf{x}, g') + \chi(g) \sum_{g'=1}^G \frac{\nu \Sigma_f(\mathbf{x}, g')}{k_{\text{eff}}} \cdot \phi(\mathbf{x}, g') \right] \right\} d^M \mathbf{x} \quad (4.34)$$

Dado que cada uno de los G corchetes es idénticamente cero para cualquier $\mathbf{x} \in U$, entonces la suma y la integral también lo es. Pero, a la vez, podemos afirmar que si la ecuación (4.34) se cumple para cualesquiera funciones continuas $\varphi_g(\mathbf{x})$, entonces las G ecuaciones de difusión (2.3) se deben cumplir en todos los puntos del dominio. En efecto, si para algún subconjunto de U —incluyendo un punto aislado—alguno de los corchetes no fuese cero, entonces podríamos encontrar fácilmente al menos una función $\varphi_g(\mathbf{x})$ tal que la integral de la ecuación (4.34) sea no nula [24]. Luego, la ecuación débil (4.34) es matemáticamente equivalente a la formulación fuerte (2.3), situación que no se daba en el capítulo 3 entre las ecuaciones (2.3) y (3.2).

Esta formulación es débil en el sentido que, como mostramos más abajo, es posible reducir en uno el orden de los operadores diferenciales que aparecen, lo que hace que las restricciones sobre la continuidad de la función solución sean más laxas y sea posible resolver directamente casos con coeficientes de difusión discontinuos. La formulación débil escrita de la forma (4.34) parece elegante y concisa, pero no ilustra la utilidad que representa el haber reemplazado un sistema de ecuaciones diferenciales vectoriales por una única ecuación integro-diferencial escalar en función de G nuevas funciones arbitrarias.

Por otra parte, aunque podamos operar sobre la ecuación (4.34) utilizando sumatorias y subíndices genéricos, las expresiones involucradas presentan ciertas complejidades que pueden inducir al lector—y especialmente al autor—a interpretaciones incorrectas. Es por eso que procedemos a continuación trabajando con el problema bidimensional $M = 2$ a dos grupos de energía con $G = 2$, escribiendo explícitamente todos los términos de las sumatorias involucradas. Luego generalizamos los resultados obtenidos nuevamente para un número genérico de grupos de energía G .

En dos dimensiones y a dos grupos de energía, la ecuación de difusión de neutrones estacionaria para el grupo rápido² es

²En la jerga de análisis de reactores, en cálculos a dos grupos de energía se llama grupo rápido al grupo $g = 1$ y grupo térmico al grupo $g = 2$.



$$0 = \operatorname{div} [D_1(x, y) \cdot \operatorname{grad} \phi_1(x, y)] - \Sigma_{t1}(x, y) \cdot \phi_1(x, y) \quad (4.35)$$

$$+ \Sigma_{s1 \rightarrow 1} \cdot \phi_1(x, y) + \Sigma_{s2 \rightarrow 1} \cdot \phi_2(x, y) + \chi_1 \frac{\nu \Sigma_{f1}}{k_{\text{eff}}} \cdot \phi_1(x, y) + \chi_1 \frac{\nu \Sigma_{f2}}{k_{\text{eff}}} \cdot \phi_2(x, y)$$

y la ecuación para el grupo térmico es

$$0 = \operatorname{div} [D_2(x, y) \cdot \operatorname{grad} \phi_2(x, y)] - \Sigma_{t2}(x, y) \cdot \phi_2(x, y) \quad (4.36)$$

$$+ \Sigma_{s1 \rightarrow 2} \cdot \phi_1(x, y) + \Sigma_{s2 \rightarrow 2} \cdot \phi_2(x, y) + \chi_2 \frac{\nu \Sigma_{f1}}{k_{\text{eff}}} \cdot \phi_1(x, y) + \chi_2 \frac{\nu \Sigma_{f2}}{k_{\text{eff}}} \cdot \phi_2(x, y)$$

Sean $\varphi_1(x, y)$ y $\varphi_2(x, y)$ dos funciones continuas—y por lo tanto integrables—definidas sobre el dominio $U \in \mathbb{R}^2$. Como hemos propuesto arriba, multiplicamos la ecuación (4.35) por la función $\varphi_1(x, y)$ y la ecuación (4.36) por $\varphi_2(x, y)$ y sumamos término a término las dos expresiones

$$0 = \varphi_1 \operatorname{div} (D_1 \operatorname{grad} \phi_1) + \varphi_2 \operatorname{div} (D_2 \operatorname{grad} \phi_2) - \varphi_1 \Sigma_{t1} \phi_1 - \varphi_2 \Sigma_{t2} \phi_2$$

$$+ \varphi_1 \Sigma_{s1 \rightarrow 1} \phi_1 + \varphi_1 \Sigma_{s2 \rightarrow 1} \phi_2 + \varphi_2 \Sigma_{s1 \rightarrow 2} \phi_1 + \varphi_2 \Sigma_{s2 \rightarrow 2} \phi_2$$

$$+ \frac{1}{k_{\text{eff}}} \left(\varphi_1 \chi_1 \nu \Sigma_{f1} \phi_1 + \varphi_1 \chi_1 \nu \Sigma_{f2} \phi_2 + \varphi_2 \chi_2 \nu \Sigma_{f1} \phi_1 + \varphi_2 \chi_2 \nu \Sigma_{f2} \phi_2 \right)$$

Integramos sobre el dominio U y reorganizamos los miembros

$$- \int_U \varphi_1 \cdot \operatorname{div} (D_1 \cdot \operatorname{grad} \phi_1) dx dy - \int_U \varphi_2 \cdot \operatorname{div} (D_2 \cdot \operatorname{grad} \phi_2) dx dy$$

$$+ \int_U \left[\varphi_1 \cdot \Sigma_{t1} \cdot \phi_1 + \varphi_2 \cdot \Sigma_{t2} \cdot \phi_2 \right] dx dy$$

$$- \int_U \left[\varphi_1 \Sigma_{s1 \rightarrow 1} \phi_1 + \varphi_1 \Sigma_{s2 \rightarrow 1} \phi_2 + \varphi_2 \Sigma_{s1 \rightarrow 2} \phi_1 + \varphi_2 \Sigma_{s2 \rightarrow 2} \phi_2 \right] dx dy$$

$$= \frac{1}{k_{\text{eff}}} \int_U \left[\varphi_1 \chi_1 \nu \Sigma_{f1} \phi_1 + \varphi_1 \chi_1 \nu \Sigma_{f2} \phi_2 + \varphi_2 \chi_2 \nu \Sigma_{f1} \phi_1 + \varphi_2 \chi_2 \nu \Sigma_{f2} \phi_2 \right] dx dy \quad (4.37)$$

donde hemos agrupado los términos de fugas, los términos de ritmo de reacción total, los términos de scattering y los términos de fisión. Notar que todos los factores—excepto χ_1 y χ_2 —dependen de la posición \mathbf{x} . La ecuación (4.37) es la formulación débil del problema de difusión de neutrones bidimensional a dos grupos de energía. Como hemos discutido, debe cumplirse para cualesquier par de funciones continuas $\varphi_1(x, y)$ y $\varphi_2(x, y)$, que por analogía al principio de los trabajos virtuales [25] y al de las temperaturas virtuales [1, capítulo 7], llamamos *flujos virtuales*. Dejamos las interpretaciones teleológicas sobre la naturaleza del comportamiento de los neutrones para futuros trabajos.

Es importante también notar que $\varphi_1(x, y)$ y $\varphi_2(x, y)$ no representan los flujos adjuntos, como la similitud de la ecuación (4.37) con la ecuación de la importancia neutrónica puede llegar a sugerir. Los flujos virtuales son funciones continuas, pero por lo demás arbitrarias y no constituyen una incógnita del problema sino que forman parte de la formulación débil como funciones arbitrarias. De hecho, en la formulación del problema tal como escrita en la ecuación (4.37) ni siquiera se les requiere que cumplan ninguna clase de condiciones de contorno sobre la frontera del dominio.



Podemos escribir, a menos de un signo, los integrandos de los términos de ritmo de reacción total, de scattering y de fisión respectivamente en forma matricial como

$$\begin{aligned} \varphi_1 \cdot \Sigma_{t1} \cdot \phi_1 + \varphi_2 \cdot \Sigma_{t2} \cdot \phi_2 &= [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & \Sigma_{t2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \\ \varphi_1 \Sigma_{s1 \rightarrow 1} \phi_1 + \varphi_1 \Sigma_{s2 \rightarrow 1} \phi_2 \\ + \varphi_2 \Sigma_{s1 \rightarrow 2} \phi_1 + \varphi_2 \Sigma_{s2 \rightarrow 2} \phi_2 &= [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{s1 \rightarrow 1} & \Sigma_{s2 \rightarrow 1} \\ \Sigma_{s1 \rightarrow 2} & \Sigma_{s2 \rightarrow 2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \\ \varphi_1 \chi_1 \nu \Sigma_{f1} \phi_1 + \varphi_1 \chi_1 \nu \Sigma_{f2} \phi_2 \\ + \varphi_2 \chi_2 \nu \Sigma_{f1} \phi_1 + \varphi_2 \chi_2 \nu \Sigma_{f2} \phi_2 &= [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \chi_1 \nu \Sigma_{f1} & \chi_1 \nu \Sigma_{f2} \\ \chi_2 \nu \Sigma_{f1} & \chi_2 \nu \Sigma_{f2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \end{aligned}$$

En problemas de análisis de reactores, es usual combinar las dos primeras ecuaciones correspondientes al ritmo de reacción y al ritmo de scattering como

$$\begin{aligned} [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & \Sigma_{t2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} - [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{s1 \rightarrow 1} & \Sigma_{s2 \rightarrow 1} \\ \Sigma_{s1 \rightarrow 2} & \Sigma_{s2 \rightarrow 2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \\ = [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{r1} & -\Sigma_{s2 \rightarrow 1} \\ -\Sigma_{s1 \rightarrow 2} & \Sigma_{r2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \end{aligned} \quad (4.38)$$

donde para simplificar la notación utilizamos la sección eficaz macroscópica de remoción introducida en la sección 2.2.3

$$\Sigma_{rg}(\mathbf{x}) = \Sigma_{tg}(\mathbf{x}) - \Sigma_{sg \rightarrow g}(\mathbf{x}) = \Sigma_{ag}(\mathbf{x}) + \sum_{g' \neq g}^G \Sigma_{sg \rightarrow g'}(\mathbf{x}) \quad (2.10)$$

De esta forma $\Sigma_{rg}(\mathbf{x}) \cdot \phi_g(\mathbf{x}) d^M \mathbf{x}$ es el ritmo al cual son removidos—ya sea absorbidos que generen o no una fisión, o removidos del grupo g por scattering—los neutrones del grupo g en un diferencial de volumen alrededor del punto \mathbf{x} . Este reemplazo lo hacemos sólo para simplificar la matemática que sigue. En el código milonga usamos la formulación original del problema, aunque dependiendo de los datos del problema—i.e. si tenemos sección eficaz total o sólo secciones eficaces de absorción neta—en el código utilizamos los términos de ritmos de reacción y scattering por separado o combinados como en la ecuación (4.38). Por ser más compacta, en lo que sigue preferimos trabajar con la expresión (4.38).

Para analizar los términos de fuga aplicamos la identidad de Green (apéndice B)

$$\int_U \theta(\mathbf{x}) \cdot \text{div} [\psi(\mathbf{x})] d^M \mathbf{x} = - \int_U \text{grad} [\theta(\mathbf{x})] \cdot \psi(\mathbf{x}) d^M \mathbf{x} + \int_{\partial U} \theta(\mathbf{x}) \cdot [\psi(\mathbf{x}) \cdot \hat{\mathbf{n}}] dS \quad (4.39)$$

poniendo $\theta(\mathbf{x}) = \varphi_g(\mathbf{x})$ y $\psi(\mathbf{x}) = \text{grad} [D_g(\mathbf{x}) \cdot \phi_g(\mathbf{x})]$. Entonces

$$\begin{aligned} - \int_U \varphi_g(\mathbf{x}) \cdot \text{div} [D_g(\mathbf{x}) \cdot \text{grad} \phi_g(\mathbf{x})] dx dy &= \int_U \text{grad} \varphi_g(\mathbf{x}) \cdot D_g \cdot \text{grad} \phi_g(\mathbf{x}) dx dy \\ &\quad - \int_{\partial U} \varphi_g(\mathbf{x}) \cdot D_g(\mathbf{x}) \cdot [\text{grad} \phi_g(\mathbf{x}) \cdot \hat{\mathbf{n}}] dS \end{aligned}$$



La mezcla de productos escalares entre vectores y productos escalar-vector complica el análisis de esta expresión. Sin embargo, si escribimos explícitamente cada uno de los términos de esta expresión

$$- \int_U \varphi_g(\mathbf{x}) \cdot \operatorname{div} \left[D_g(\mathbf{x}) \cdot \operatorname{grad} \phi_g(\mathbf{x}) \right] dx dy = \int_U \left(\frac{\partial \varphi_g}{\partial x} \cdot D_g \cdot \frac{\partial \phi_g}{\partial x} + \frac{\partial \varphi_g}{\partial y} \cdot D_g \cdot \frac{\partial \phi_g}{\partial y} \right) dx dy - \int_{\partial U} \varphi_g \cdot D_g \cdot \left(\frac{\partial \phi_g}{\partial x} \cdot \hat{n}_x + \frac{\partial \phi_g}{\partial y} \cdot \hat{n}_y \right) dS$$

entonces podemos poner los dos primeros términos de la ecuación (4.37) como

$$- \int_U \varphi_1 \cdot \operatorname{div} (D_1 \cdot \phi_1) dx dy - \int_U \varphi_2 \cdot \operatorname{div} (D_2 \cdot \phi_2) dx dy = \int_U \left(\frac{\partial \varphi_1}{\partial x} \cdot D_1 \cdot \frac{\partial \phi_1}{\partial x} + \frac{\partial \varphi_2}{\partial x} \cdot D_2 \cdot \frac{\partial \phi_2}{\partial x} + \frac{\partial \varphi_1}{\partial y} \cdot D_1 \cdot \frac{\partial \phi_1}{\partial y} + \frac{\partial \varphi_2}{\partial y} \cdot D_2 \cdot \frac{\partial \phi_2}{\partial y} \right) dx dy - \int_{\partial U} \left[\varphi_1 \cdot D_1 \cdot \left(\frac{\partial \phi_1}{\partial x} \cdot \hat{n}_x + \frac{\partial \phi_1}{\partial y} \cdot \hat{n}_y \right) + \varphi_2 \cdot D_2 \cdot \left(\frac{\partial \phi_2}{\partial x} \cdot \hat{n}_x + \frac{\partial \phi_2}{\partial y} \cdot \hat{n}_y \right) \right] dS$$

para finalmente escribir el integrando de la integral sobre el dominio en forma matricial como

$$\begin{aligned} & \frac{\partial \varphi_1}{\partial x} \cdot D_1 \cdot \frac{\partial \phi_1}{\partial x} + \frac{\partial \varphi_2}{\partial x} \cdot D_2 \cdot \frac{\partial \phi_2}{\partial x} + \frac{\partial \varphi_1}{\partial y} \cdot D_1 \cdot \frac{\partial \phi_1}{\partial y} + \frac{\partial \varphi_2}{\partial y} \cdot D_2 \cdot \frac{\partial \phi_2}{\partial y} \\ &= \begin{bmatrix} \partial \varphi_1 / \partial x & \partial \varphi_2 / \partial x & \partial \varphi_1 / \partial y & \partial \varphi_2 / \partial y \end{bmatrix} \cdot \begin{bmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ 0 & 0 & D_1 & 0 \\ 0 & 0 & 0 & D_2 \end{bmatrix} \cdot \begin{bmatrix} \partial \phi_1 / \partial x \\ \partial \phi_2 / \partial x \\ \partial \phi_1 / \partial y \\ \partial \phi_2 / \partial y \end{bmatrix} \end{aligned}$$

y el integrando de la integral sobre el borde del dominio ∂U como

$$\begin{aligned} & \varphi_1 \cdot D_1 \cdot \left(\frac{\partial \phi_1}{\partial x} \cdot \hat{n}_x + \frac{\partial \phi_1}{\partial y} \cdot \hat{n}_y \right) \\ & + \varphi_2 \cdot D_2 \cdot \left(\frac{\partial \phi_2}{\partial x} \cdot \hat{n}_x + \frac{\partial \phi_2}{\partial y} \cdot \hat{n}_y \right) = \begin{bmatrix} \varphi_1 & \varphi_2 \end{bmatrix} \cdot \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} \partial \phi_1 / \partial n \\ \partial \phi_2 / \partial n \end{bmatrix} \end{aligned}$$

Con esta notación, la formulación débil para el caso particular de dos grupos de energía $G = 2$ en dos dimensiones $M = 2$ dada por la ecuación (4.37) queda



$$\begin{aligned}
& \int_U \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_2}{\partial x} & \frac{\partial \varphi_1}{\partial y} & \frac{\partial \varphi_2}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ 0 & 0 & D_1 & 0 \\ 0 & 0 & 0 & D_2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \\ \frac{\partial \phi_2}{\partial x} \\ \frac{\partial \phi_1}{\partial y} \\ \frac{\partial \phi_2}{\partial y} \end{bmatrix} dx dy \\
& - \int_{\partial U} [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \phi_1}{\partial n} \\ \frac{\partial \phi_2}{\partial n} \end{bmatrix} dS \\
& + \int_U [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \Sigma_{r1} & -\Sigma_{s2 \rightarrow 1} \\ -\Sigma_{s1 \rightarrow 2} & \Sigma_{r2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} dx dy \\
& = \frac{1}{k_{\text{eff}}} \int_U [\varphi_1 \quad \varphi_2] \cdot \begin{bmatrix} \chi_1 \nu \Sigma_{f1} & \chi_1 \nu \Sigma_{f2} \\ \chi_2 \nu \Sigma_{f1} & \chi_2 \nu \Sigma_{f2} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} dx dy
\end{aligned}$$

Es decir, la ecuación de difusión de neutrones en su formulación débil en dos dimensiones y a dos grupos de energía es

$$\begin{aligned}
& \int_U \boldsymbol{\varphi}'^T(\mathbf{x}) \cdot D'(\mathbf{x}) \cdot \boldsymbol{\phi}'(\mathbf{x}) d^M \mathbf{x} - \int_{\partial U} \boldsymbol{\varphi}^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) dS \\
& + \int_U \boldsymbol{\varphi}^T(\mathbf{x}) \cdot A(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}) d^M \mathbf{x} = \frac{1}{k_{\text{eff}}} \int_U \boldsymbol{\varphi}^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}) d^M \mathbf{x} \quad (4.40)
\end{aligned}$$

donde

$$\begin{aligned}
\boldsymbol{\varphi}(\mathbf{x}) &= \begin{bmatrix} \varphi_1(\mathbf{x}) \\ \varphi_2(\mathbf{x}) \end{bmatrix} \\
\boldsymbol{\phi}(\mathbf{x}) &= \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \end{bmatrix} \\
\boldsymbol{\varphi}'(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} \\ \frac{\partial \varphi_2}{\partial x} \\ \frac{\partial \varphi_1}{\partial y} \\ \frac{\partial \varphi_2}{\partial y} \end{bmatrix} \\
\boldsymbol{\phi}'(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \\ \frac{\partial \phi_2}{\partial x} \\ \frac{\partial \phi_1}{\partial y} \\ \frac{\partial \phi_2}{\partial y} \end{bmatrix} \\
D(\mathbf{x}) &= \begin{bmatrix} D_1(\mathbf{x}) & 0 \\ 0 & D_2(\mathbf{x}) \end{bmatrix} \\
D'(\mathbf{x}) &= \begin{bmatrix} D_1(\mathbf{x}) & 0 & 0 & 0 \\ 0 & D_2(\mathbf{x}) & 0 & 0 \\ 0 & 0 & D_1(\mathbf{x}) & 0 \\ 0 & 0 & 0 & D_2(\mathbf{x}) \end{bmatrix} \\
\mathbf{N}(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \phi_1}{\partial n} \\ \frac{\partial \phi_2}{\partial n} \end{bmatrix}
\end{aligned}$$



$$A(\mathbf{x}) = \begin{bmatrix} \Sigma_{r1}(\mathbf{x}) & -\Sigma_{2 \rightarrow 1}(\mathbf{x}) \\ -\Sigma_{1 \rightarrow 2}(\mathbf{x}) & \Sigma_{r2}(\mathbf{x}) \end{bmatrix}$$

$$X(\mathbf{x}) = \begin{bmatrix} \chi_1 \cdot \nu \Sigma_{f1}(\mathbf{x}) & \chi_1 \cdot \nu \Sigma_{f2}(\mathbf{x}) \\ \chi_2 \cdot \nu \Sigma_{f1}(\mathbf{x}) & \chi_2 \cdot \nu \Sigma_{f2}(\mathbf{x}) \end{bmatrix}$$

Podemos generalizar la formulación débil para una cantidad arbitraria de grupos de energía G y M coordenadas espaciales reemplazando en la ecuación (4.40) las siguientes definiciones:

$$\varphi(\mathbf{x}) = \begin{bmatrix} \varphi_1(\mathbf{x}) \\ \varphi_2(\mathbf{x}) \\ \vdots \\ \varphi_G(\mathbf{x}) \end{bmatrix} \quad \text{vector de tamaño } G$$

$$\phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_G(\mathbf{x}) \end{bmatrix} \quad \text{vector de tamaño } G$$

$$\varphi'(\mathbf{x}) = \begin{bmatrix} \partial\varphi_1/\partial x_1 \\ \partial\varphi_2/\partial x_1 \\ \vdots \\ \partial\varphi_G/\partial x_1 \\ \partial\varphi_1/\partial x_2 \\ \vdots \\ \partial\varphi_G/\partial x_M \end{bmatrix} \quad \text{vector de tamaño } MG$$

$$\phi'(\mathbf{x}) = \begin{bmatrix} \partial\phi_1/\partial x_1 \\ \partial\phi_2/\partial x_1 \\ \vdots \\ \partial\phi_G/\partial x_1 \\ \partial\phi_1/\partial x_2 \\ \vdots \\ \partial\phi_G/\partial x_M \end{bmatrix} \quad \text{vector de tamaño } MG$$

$$D(\mathbf{x}) = \begin{bmatrix} D_1(\mathbf{x}) & 0 & \dots & 0 \\ 0 & D_2(\mathbf{x}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_G(\mathbf{x}) \end{bmatrix} \quad \text{matriz de tamaño } G \times G$$



$$\begin{aligned}
 D'(\mathbf{x}) &= \begin{bmatrix} D_1(\mathbf{x}) & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & D_2(\mathbf{x}) & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \dots & D_G(\mathbf{x}) & 0 & 0 & \dots \\ 0 & 0 & \dots & 0 & D_1(\mathbf{x}) & 0 & \dots \\ 0 & 0 & \dots & 0 & 0 & D_2(\mathbf{x}) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\
 &= \begin{bmatrix} \left[D(\mathbf{x}) \right] & \left[\begin{matrix} 0 \\ D(\mathbf{x}) \end{matrix} \right] & \dots \\ \left[\begin{matrix} 0 \\ D(\mathbf{x}) \end{matrix} \right] & \left[\begin{matrix} 0 \\ D(\mathbf{x}) \end{matrix} \right] & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}
 \end{aligned}$$

tamaño $MG \times MG$

$$\mathbf{N}(\mathbf{x}) = \begin{bmatrix} \partial\phi_1/\partial n \\ \partial\phi_2/\partial n \\ \vdots \\ \partial\phi_G/\partial n \end{bmatrix}$$

vector de tamaño G

$$A(\mathbf{x}) = \begin{bmatrix} \Sigma_{r1}(\mathbf{x}) & -\Sigma_{2 \rightarrow 1}(\mathbf{x}) & \dots & -\Sigma_{G \rightarrow 1}(\mathbf{x}) \\ -\Sigma_{1 \rightarrow 2}(\mathbf{x}) & \Sigma_{r2}(\mathbf{x}) & \dots & -\Sigma_{G \rightarrow 2}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{1 \rightarrow G}(\mathbf{x}) & -\Sigma_{2 \rightarrow G}(\mathbf{x}) & \dots & \Sigma_{rG}(\mathbf{x}) \end{bmatrix}$$

matriz de tamaño $G \times G$

$$X(\mathbf{x}) = \begin{bmatrix} \chi_1 \cdot \nu \Sigma_{f1}(\mathbf{x}) & \chi_1 \cdot \nu \Sigma_{f2}(\mathbf{x}) & \dots & \chi_1 \cdot \nu \Sigma_{fG}(\mathbf{x}) \\ \chi_2 \cdot \nu \Sigma_{f1}(\mathbf{x}) & \chi_2 \cdot \nu \Sigma_{f2}(\mathbf{x}) & \dots & \chi_2 \cdot \nu \Sigma_{fG}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_G \cdot \nu \Sigma_{f1}(\mathbf{x}) & \chi_G \cdot \nu \Sigma_{f2}(\mathbf{x}) & \dots & \chi_G \cdot \nu \Sigma_{fG}(\mathbf{x}) \end{bmatrix}$$

matriz de tamaño $G \times G$

4.4. Esquema basado en elementos

Estamos ahora en condiciones desarrollar una discretización basada en el método de elementos finitos para resolver la ecuación de difusión de neutrones multigrupo en una malla no estructurada. Para ello utilizamos todas las ideas y conceptos introducidos a lo largo de este capítulo. Sin embargo, en lugar de aplicar directamente el método de Galerkin a las ecuaciones diferenciales, discretizamos espacialmente la formulación débil encontrada en la sección anterior para obtener las ecuaciones discretas del esquema en elementos. Esta formulación—que es equivalente al método de Galerkin como mostramos en la sección 4.5—es, desde el punto de vista del autor, más conveniente para el planteo del problema.

Para resolver la ecuación (4.40) utilizando el método de elementos finitos, primero notamos que las tres integrales sobre el dominio U pueden escribirse como suma de integrales elementales sobre I elementos, como demostramos en la ecuación (4.18) de la sección 4.2. A su vez, vamos a buscar una función aproximada $\hat{\phi}(\mathbf{x})$ que podamos escribir, dentro del elemento e_i , como una combinación lineal de las J_i funciones de forma elementales $h_j(\mathbf{x})$ utilizando los J_i valores nodales $\hat{\phi}_g^j$ como coeficientes de peso.



Por otro lado, la formulación débil implica funciones virtuales que en principio son continuas pero arbitrarias por naturaleza, de tal manera que la ecuación (4.40) para cualquier posible valor que puedan tomar estos flujos virtuales. En la discretización en elementos finitos, vamos a restringir las funciones virtuales al mismo espacio vectorial que las funciones incógnita aproximadas. Es decir, vamos a utilizar como flujos virtuales a la función aproximada $\tilde{\varphi}(\mathbf{x})$ que, dentro del elemento e_i también es igual a una combinación lineal de las mismas funciones de forma $h_j(\mathbf{x})$ utilizando un número J_i de ciertos coeficientes arbitrarios $\hat{\varphi}_i^j$. La ecuación exacta (4.40) se transforma entonces en

$$\int_U \tilde{\varphi}'^T(\mathbf{x}) \cdot D'(\mathbf{x}) \cdot \tilde{\varphi}'(\mathbf{x}) d^M \mathbf{x} - \int_{\partial U} \tilde{\varphi}^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) dS + \int_U \tilde{\varphi}^T(\mathbf{x}) \cdot A(\mathbf{x}) \cdot \tilde{\varphi}(\mathbf{x}) d^M \mathbf{x} = \frac{1}{k_{\text{eff}}} \int_U \tilde{\varphi}^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot \tilde{\varphi}(\mathbf{x}) d^M \mathbf{x} \quad (4.41)$$

Para fijar ideas, supongamos que $M = 2$, $G = 2$ y $J_i = 3$. Entonces, proponemos funciones aproximadas

$$\begin{aligned} \tilde{\phi}(x, y) &= \begin{bmatrix} \sum_{j=1}^{J_i} h_j(x, y) \cdot \hat{\phi}_1^j \\ \sum_{j=1}^{J_i} h_j(x, y) \cdot \hat{\phi}_2^j \end{bmatrix} \\ &= \begin{bmatrix} h_1(\mathbf{x}) & 0 & h_2(\mathbf{x}) & 0 & h_3(\mathbf{x}) & 0 \\ 0 & h_1(\mathbf{x}) & 0 & h_2(\mathbf{x}) & 0 & h_3(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_1^2 \\ \hat{\phi}_1^3 \\ \hat{\phi}_2^1 \\ \hat{\phi}_2^2 \\ \hat{\phi}_2^3 \end{bmatrix} \\ &= H(x, y) \cdot \hat{\phi} \end{aligned}$$

$$\begin{aligned} \tilde{\varphi}(x, y) &= \begin{bmatrix} \sum_{j=1}^{J_i} h_j(x, y) \cdot \hat{\varphi}_1^j \\ \sum_{j=1}^{J_i} h_j(x, y) \cdot \hat{\varphi}_2^j \end{bmatrix} \\ &= \begin{bmatrix} h_1(\mathbf{x}) & 0 & h_2(\mathbf{x}) & 0 & h_3(\mathbf{x}) & 0 \\ 0 & h_1(\mathbf{x}) & 0 & h_2(\mathbf{x}) & 0 & h_3(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \hat{\varphi}_1^1 \\ \hat{\varphi}_1^2 \\ \hat{\varphi}_1^3 \\ \hat{\varphi}_2^1 \\ \hat{\varphi}_2^2 \\ \hat{\varphi}_2^3 \end{bmatrix} \\ &= H(x, y) \cdot \hat{\varphi} \end{aligned}$$



4.4.1. Fisiones

La integral del ritmo de fisiones sobre el elemento e_i queda

$$\begin{aligned}
 \int_{e_i} \tilde{\varphi}^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot \tilde{\phi}(\mathbf{x}) d^M \mathbf{x} &= \int_{e_i} [H(\mathbf{x})\hat{\phi}]^T \cdot X(\mathbf{x}) \cdot [H(\mathbf{x})\hat{\phi}] d^M \mathbf{x} \\
 &= \int_{e_i} [\hat{\phi}^T H^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot H(\mathbf{x})\hat{\phi}] d^M \mathbf{x} \\
 &= \hat{\phi}^T \cdot \left[\int_{e_i} H^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot H(\mathbf{x}) d^M \mathbf{x} \right] \cdot \hat{\phi} \\
 &= \hat{\phi}^T \cdot \dot{X}_i \cdot \hat{\phi}
 \end{aligned} \tag{4.42}$$

donde hemos definido la matriz elemental de fisión \dot{X}_i como

$$\dot{X}_i = \int_{e_i} H^T(\mathbf{x}) \cdot X(\mathbf{x}) \cdot H(\mathbf{x}) d^M \mathbf{x}$$

que para $M = 2, G = 2$ y $J_i = 3$ podemos evaluar como

$$\begin{aligned}
 \dot{X}_i &= \int_{-1}^1 \int_{-1}^1 H^T(r, s) \cdot X(r, s) \cdot H(r, s) \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| dr ds \\
 &= \int_{-1}^1 \int_{-1}^1 \left(\begin{bmatrix} h_1 & 0 \\ 0 & h_1 \\ h_2 & 0 \\ 0 & h_2 \\ h_3 & 0 \\ 0 & h_3 \end{bmatrix} \cdot \begin{bmatrix} \chi_1 \nu \Sigma_{f1} & \chi_1 \nu \Sigma_{f2} \\ \chi_2 \nu \Sigma_{f1} & \chi_2 \nu \Sigma_{f2} \end{bmatrix} \cdot \begin{bmatrix} h_1 & 0 & h_2 & 0 & h_3 & 0 \\ 0 & h_1 & 0 & h_2 & 0 & h_3 \end{bmatrix} \right. \\
 &\quad \left. \cdot \det \left[\begin{array}{cc} \sum_{j=1}^3 \frac{\partial h_j}{\partial r} x_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} x_j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial r} y_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} y_j \end{array} \right] \right) dr ds
 \end{aligned}$$

donde x_j e y_j son las coordenadas absolutas de los $J_i = 3$ nodos que definen el elemento e_i y las funciones $h_j(r, s)$ son las tres funciones de forma asociadas a cada uno de los tres nodos del elemento e_i . Las secciones eficaces de fisión son funciones conocidas de las coordenadas globales x, y , mientras que la integral la realizamos sobre las coordenadas locales r, s . Pero utilizando el mapeo correspondiente, cualquier función $f(\mathbf{x})$ puede evaluarse en función del vector local \mathbf{r} como

$$f(x, y) = f\left(x(r, s), y(r, s)\right) = f\left(\sum_{j=1}^{J_i} h_j(r, s) \cdot x_j, \sum_{j=1}^{J_i} h_j(r, s) \cdot y_j\right)$$

4.4.2. Remociones y scattering

Procediendo análogamente con el término de absorciones y de scattering tenemos

$$\int_{e_i} \tilde{\varphi}^T(\mathbf{x}) \cdot A(\mathbf{x}) \cdot \tilde{\phi}(\mathbf{x}) d^M \mathbf{x} = \hat{\varphi}^T \cdot \dot{A}_i \cdot \hat{\phi} \tag{4.43}$$



$$\begin{aligned}
 \dot{A}_i &= \int_{e_i} H^T(\mathbf{x}) \cdot A(\mathbf{x}) \cdot H(\mathbf{x}) d^M \mathbf{x} \\
 &= \int_{-1}^1 \int_{-1}^1 H^T(r, s) \cdot A(r, s) \cdot H(r, s) \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| dr ds \\
 &= \int_{-1}^1 \int_{-1}^1 \left(\begin{array}{cc} h_1 & 0 \\ 0 & h_1 \\ h_2 & 0 \\ 0 & h_2 \\ h_3 & 0 \\ 0 & h_3 \end{array} \cdot \begin{bmatrix} \Sigma_{r1} & -\Sigma_{s2 \rightarrow 1} \\ -\Sigma_{s1 \rightarrow 2} & \Sigma_{r2} \end{bmatrix} \cdot \begin{bmatrix} h_1 & 0 & h_2 & 0 & h_3 & 0 \\ 0 & h_1 & 0 & h_2 & 0 & h_3 \end{bmatrix} \right. \\
 &\quad \left. \cdot \det \left[\begin{array}{cc} \sum_{j=1}^3 \frac{\partial h_j}{\partial r} x_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} x_j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial r} y_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} y_j \end{array} \right] \right) dr ds
 \end{aligned}$$

4.4.3. Fugas

El término de fugas involucra el operador diferencial gradiente, y por lo tanto es ligeramente más complicado. El vector primado de flujos aproximados es

$$\begin{aligned}
 \tilde{\phi}' &= \begin{bmatrix} \frac{\partial \tilde{\phi}_1}{\partial x} \\ \frac{\partial \tilde{\phi}_2}{\partial x} \\ \frac{\partial \tilde{\phi}_1}{\partial y} \\ \frac{\partial \tilde{\phi}_2}{\partial y} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^3 \frac{\partial h_j}{\partial x} \cdot \hat{\phi}_1^j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial x} \cdot \hat{\phi}_2^j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial y} \cdot \hat{\phi}_1^j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial y} \cdot \hat{\phi}_2^j \end{bmatrix} \\
 &= \begin{bmatrix} \partial h_1 / \partial x & 0 & \partial h_2 / \partial x & 0 & \partial h_3 / \partial x & 0 \\ 0 & \partial h_1 / \partial x & 0 & \partial h_2 / \partial x & 0 & \partial h_3 / \partial x \\ \partial h_1 / \partial y & 0 & \partial h_2 / \partial y & 0 & \partial h_3 / \partial y & 0 \\ 0 & \partial h_1 / \partial y & 0 & \partial h_2 / \partial y & 0 & \partial h_3 / \partial y \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_2^1 \\ \hat{\phi}_1^2 \\ \hat{\phi}_2^2 \\ \hat{\phi}_1^3 \\ \hat{\phi}_2^3 \end{bmatrix} \\
 &= B_x(x, y) \cdot \hat{\phi}
 \end{aligned}$$

Análogamente,

$$\hat{\varphi}' = B_x(x, y) \cdot \hat{\varphi}$$

Como discutimos en el ejemplo 8, la matriz B_x involucra la derivadas de las funciones de forma con respecto a las coordenadas globales—que son las coordenadas del operador diferencial del problema que estamos resolviendo. Estas derivadas son más “difíciles” de calcular



que las derivadas con respecto a las coordenadas locales, que podemos evaluar directamente a partir de las expresiones algebraicas sencillas derivadas en la sección 4.2.2, es decir las ecuaciones (4.21), (4.27) o (4.32) según corresponda al tipo de elemento. En el ejemplo 8, encontramos que la matriz B_x contiene las derivadas con respecto a las coordenadas globales y estaba relacionada con la matriz B_r de las derivadas locales a través de la matriz jacobiana inversa $\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right]$. Si bien aún podemos encontrar una relación de este tipo para B_x y B_r , como $G = 2$ la matriz involucrada ya no es el inverso del jacobiano. En efecto,

$$\begin{bmatrix} \frac{\partial h_1}{\partial x} & 0 & \frac{\partial h_2}{\partial x} & 0 & \frac{\partial h_3}{\partial x} & 0 \\ 0 & \frac{\partial h_1}{\partial x} & 0 & \frac{\partial h_2}{\partial x} & 0 & \frac{\partial h_3}{\partial x} \\ \frac{\partial h_1}{\partial y} & 0 & \frac{\partial h_2}{\partial y} & 0 & \frac{\partial h_3}{\partial y} & 0 \\ 0 & \frac{\partial h_1}{\partial y} & 0 & \frac{\partial h_2}{\partial y} & 0 & \frac{\partial h_3}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial x} & 0 & \frac{\partial s}{\partial x} & 0 \\ 0 & \frac{\partial r}{\partial x} & 0 & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & 0 & \frac{\partial s}{\partial y} & 0 \\ 0 & \frac{\partial r}{\partial y} & 0 & \frac{\partial s}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial h_1}{\partial r} & 0 & \frac{\partial h_2}{\partial r} & 0 & \frac{\partial h_3}{\partial r} & 0 \\ 0 & \frac{\partial h_1}{\partial r} & 0 & \frac{\partial h_2}{\partial r} & 0 & \frac{\partial h_3}{\partial r} \\ \frac{\partial h_1}{\partial s} & 0 & \frac{\partial h_2}{\partial s} & 0 & \frac{\partial h_3}{\partial s} & 0 \\ 0 & \frac{\partial h_1}{\partial s} & 0 & \frac{\partial h_2}{\partial s} & 0 & \frac{\partial h_3}{\partial s} \end{bmatrix}$$

Es decir,

$$B_x(r, s) = \left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right]^T \cdot B_r(r, s)$$

donde la matriz que relaciona la matriz de derivadas locales B_r con la de derivadas globales es la transpuesta de

$$\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right] = \begin{bmatrix} \frac{\partial r}{\partial x} & 0 & \frac{\partial r}{\partial y} & 0 \\ 0 & \frac{\partial r}{\partial x} & 0 & \frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial x} & 0 & \frac{\partial s}{\partial y} & 0 \\ 0 & \frac{\partial s}{\partial x} & 0 & \frac{\partial s}{\partial y} \end{bmatrix}$$

que a su vez es una extensión de la matriz jacobiana básica de la transformación entre coordenadas globales y locales rellena con ceros

$$\left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}}\right] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix}$$

cuyos elementos calculamos directamente desde las expresiones (4.21), (4.27) o (4.32), mientras que para calcular los elementos de la matriz $\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right]$ primero invertimos la la matriz $\left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}}\right]$



$$\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right] = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial x} & \frac{\partial s}{\partial y} \end{bmatrix}$$

con algún método sencillo—como por ejemplo la ecuación (4.30)—y luego extraemos uno a uno sus elementos y los ubicamos en la estructura de $\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right]$ o directamente calculamos los elementos uno a uno utilizando las ecuaciones (4.31).

En cualquier caso, el término de fugas correspondiente al elemento e_i de la formulación débil aproximada (4.40) queda

$$\begin{aligned} \int_{e_i} \tilde{\varphi}'^T(\mathbf{x}) \cdot D'(\mathbf{x}) \cdot \tilde{\phi}'(\mathbf{x}) \, d^M \mathbf{x} &= \int_{e_i} [B_x(\mathbf{x}) \cdot \tilde{\varphi}]^T \cdot D'(\mathbf{x}) \cdot [B_x(\mathbf{x}) \cdot \hat{\phi}] \, d^M \mathbf{x} \\ &= \int_{e_i} \left[\hat{\varphi}^T B_x(\mathbf{x})^T \cdot D'(\mathbf{x}) \cdot B_x(\mathbf{x}) \hat{\phi} \right] \, d^M \mathbf{x} \\ &= \hat{\varphi} \cdot \left[\int_{e_i} B_x(\mathbf{x})^T \cdot D'(\mathbf{x}) \cdot B_x(\mathbf{x}) \, d^M \mathbf{x} \right] \cdot \hat{\phi} \\ &= \hat{\varphi} \cdot \mathring{K} \cdot \hat{\phi} \end{aligned} \quad (4.44)$$

con la matriz elemental de fugas que hace las veces de matriz elemental de rigidez

$$\begin{aligned} \mathring{K}_i &= \int_{e_i} B_x^T(\mathbf{x}) \cdot D'(\mathbf{x}) \cdot B_x(\mathbf{x}) \, d^M \mathbf{x} \\ &= \int_{-1}^1 \int_{-1}^1 B_x^T(r, s) \cdot D'(r, s) \cdot B_x(r, s) \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right| \, dr \, ds \\ &= \int_{-1}^1 \int_{-1}^1 \left(\begin{bmatrix} h'_{1x} & 0 & h'_{1y} & 0 \\ 0 & h'_{1x} & 0 & h'_{1y} \\ h'_{2x} & 0 & h'_{2y} & 0 \\ 0 & h'_{2x} & 0 & h'_{2y} \\ h'_{3x} & 0 & h'_{3y} & 0 \\ 0 & h'_{3x} & 0 & h'_{3y} \end{bmatrix} \cdot \begin{bmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ 0 & 0 & D_1 & 0 \\ 0 & 0 & 0 & D_2 \end{bmatrix} \right. \\ &\quad \left. \cdot \begin{bmatrix} h'_{1x} & 0 & h'_{2x} & 0 & h'_{3x} & 0 \\ 0 & h'_{1x} & 0 & h'_{2x} & 0 & h'_{3x} \\ h'_{1y} & 0 & h'_{2y} & 0 & h'_{3y} & 0 \\ 0 & h'_{1y} & 0 & h'_{2y} & 0 & h'_{3y} \end{bmatrix} \cdot \det \begin{bmatrix} \sum_{j=1}^3 \frac{\partial h_j}{\partial r} x_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} x_j \\ \sum_{j=1}^3 \frac{\partial h_j}{\partial r} y_j & \sum_{j=1}^3 \frac{\partial h_j}{\partial s} y_j \end{bmatrix} \right) \, dr \, ds \end{aligned}$$

4.4.4. Condiciones de contorno

Finalmente, prestemos atención al segundo término de la ecuación (4.41) que corresponde al término de superficie del teorema de Green, que es el más *tricky* de todos y el responsable del prolegómeno introducido en las secciones 4.1 y 4.2. En cualquier caso, este término es la suma de una cierta cantidad de términos elementales



$$\begin{aligned}
\int_{\partial U_i} \tilde{\varphi}^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) \, dS &= \int_{\partial U_i} [H(\mathbf{x}) \cdot \hat{\varphi}]^T \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) \, dS \\
&= \hat{\varphi}^T \left[\int_{\partial U_i} H^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) \, dS \right] \quad (4.45)
\end{aligned}$$

correspondientes a los elementos e_i que tienen al menos uno de sus bordes en la frontera ∂U del dominio. Para el caso que estamos utilizando como ejemplo particular de $J = 3$ $M = 2$ y $G = 2$, el integrando es

$$\begin{aligned}
H^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) &= H^T(\mathbf{x}) \cdot \begin{bmatrix} D_1(\mathbf{x}) & 0 \\ 0 & D_2(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \partial\phi_1/\partial n \\ \partial\phi_2/\partial n \end{bmatrix} \\
&= \begin{bmatrix} h_1(\mathbf{x}) & 0 \\ 0 & h_1(\mathbf{x}) \\ h_2(\mathbf{x}) & 0 \\ 0 & h_2(\mathbf{x}) \\ h_3(\mathbf{x}) & 0 \\ 0 & h_3(\mathbf{x}) \end{bmatrix} \begin{bmatrix} D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \end{bmatrix} \\
&= \begin{bmatrix} h_1(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ h_1(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \\ h_2(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ h_2(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \\ h_3(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ h_3(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \end{bmatrix}
\end{aligned}$$

Para fijar ideas una vez más, consideremos el caso ilustrado en la figura 4.12 en el que debemos calcular la contribución de la frontera ∂U_i a las matrices elementales del elemento i . Según discutimos en la sección 2.2.2, tenemos tres casos: condiciones de Dirichlet, de Neumann y de Robin. Por otro lado, debemos recordar los resultados obtenidos en el ejemplo 3 y discutidos en la página 41 concerniente a la aparición del símbolo $\partial\phi/\partial n$ en el término de superficie de la integral de Green:

[...] cuando aparezca $d\tilde{\phi}/dx$ evaluada en una superficie con condiciones de contorno esenciales, podemos ignorar su valor ya que no formará parte de la formulación. Y cuando aparezca $d\tilde{\phi}/dx$ en una superficie con condiciones de contorno naturales, debemos reemplazarla por el valor numérico requerido en la condición de Neumann, y no por $B(x)\hat{\phi}$ como sí debemos hacer cuando la derivada aparece en algún integrando de la formulación débil.

4.4.4.1. Condiciones de Dirichlet

Sobre la superficie ∂U_i de la figura 4.12, la función $h_3(\mathbf{x})$ es idénticamente igual a cero, por lo que por un lado los últimos dos elementos del integrando se anulan

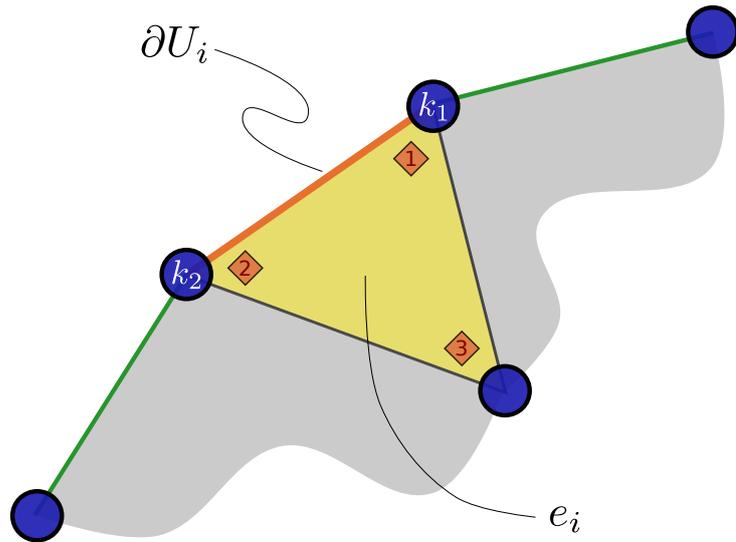


Figura 4.12 – Elemento e_i hipotético con uno de sus bordes sobre la frontera del dominio U del problema.

$$H^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ h_1(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \\ h_2(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ h_2(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \\ 0 \cdot D_1(\mathbf{x}) \cdot \partial\phi_1/\partial n \\ 0 \cdot D_2(\mathbf{x}) \cdot \partial\phi_2/\partial n \end{bmatrix}$$

Por otro lado, no podemos evaluar los primeros cuatro elementos con la información disponible. Sin embargo, dado que luego del proceso de ensamble estos cuatro elementos van a contribuir como término independiente—ya que la ecuación (4.45) no contiene al vector $\hat{\phi}$ como sí aparece en las ecuaciones (4.42), (4.43) y (4.44)—a las ecuaciones correspondientes a los nodos globales k_1 y k_2 , ecuaciones que serán luego del ensamble y antes de la solución del problema, reemplazadas de forma tal de fijar los valores nodales de las incógnitas en k_1 y k_2 . En efecto, el primer elemento aparecerá en la ecuación correspondiente a $\hat{\phi}_1^{k_1}$, el segundo en la ecuación de $\hat{\phi}_2^{k_2}$, el siguiente en $\hat{\phi}_1^{k_2}$ y el último en $\hat{\phi}_2^{k_3}$. Y estas cuatro ecuaciones serán reemplazadas por la condición de flujo nulo en la frontera

$$\begin{aligned} \phi_1(\mathbf{x}_{k_1}) &= \hat{\phi}_1^{k_1} = 0 \\ \phi_2(\mathbf{x}_{k_1}) &= \hat{\phi}_2^{k_1} = 0 \\ \phi_1(\mathbf{x}_{k_2}) &= \hat{\phi}_1^{k_2} = 0 \\ \phi_2(\mathbf{x}_{k_2}) &= \hat{\phi}_2^{k_2} = 0 \end{aligned}$$

Que el valor nodal de los flujos sea cero proviene de la homogeneidad de la ecuación de difusión sin fuente. Pero la idea de que las ecuaciones que no se anulan en el integrando del término de superficie de Green son irrelevantes se mantiene para condiciones no homogéneas. De la misma manera, el resultado es cierto para una, dos o tres dimensiones M y para cualquier tipo de elemento e_i .

Como corolario de esta discusión, tal como aprendimos en el ejemplo 3, si ∂U_i tiene condiciones de contorno de Dirichlet, podemos directamente ignorar el término de superficie del



teorema de Green ya que, o bien sus elementos son cero porque las funciones de forma se anulan en ∂U o su contribución a las ecuaciones globales es sobre-escrita luego del proceso de ensamble.

Notar que el hecho de que podamos ignorar la contribución del término de superficie del teorema de Green cuando tenemos condiciones de Dirichlet no implica que los flujos virtuales deban tener las mismas condiciones de contorno homogéneas que los flujos incógnita. La formulación débil dada en la ecuación (4.34) debe cumplirse para cualesquiera funciones continuas $\varphi_g(\mathbf{x})$, sin poner restricciones sobre los valores que deben tomar estas funciones arbitrarias en el borde del dominio. La posibilidad de ignorar esta contribución proviene de la forma de introducir condiciones de contorno esenciales en el método de elementos finitos y, en forma más general, en el método de residuos ponderados.

Por supuesto, debemos reemplazar las filas de las matrices R y F del problema de forma tal de fijar en cero los valores nodales de aquellos nodos que estén ubicados en un borde con condición de Dirichlet. Para ello, recordando el capítulo 2 (ver también sección 4.6) donde introducimos la formulación del problema autovalores generalizado de la forma

$$R \cdot \hat{\phi} = \frac{1}{k_{\text{eff}}} \cdot F \cdot \hat{\phi} \quad (2.11)$$

entonces es posible colocar la condición de contorno mediante el reemplazo de las filas que corresponden a los nodos que están en la superficie con condición de Dirichlet por valores nulos, excepto por el elemento ubicado en la diagonal de cualquiera de las dos matrices. Supongamos por ejemplo anulamos cero todas las filas correspondientes al grupo g y al nodo global k_1 —cuyo índice es $\ell(k_1, g)$ —tanto en R como en F , excepto en la diagonal de R donde sobre-escribimos el valor anterior por

$$R_{\ell(k_1, g), \ell(k_1, g)} = 1$$

Entonces la fila ℓ de la formulación matricial leerá

$$1 \cdot \phi_g^{k_1} = \frac{1}{k_{\text{eff}}} \cdot 0$$

cuya única solución será $\phi_g^{k_1}$, independientemente del resto de las ecuaciones. Incluso no es necesario que hayamos puesto exactamente uno en la diagonal de R , cualquier valor diferente de cero será suficiente para fijar la condición de flujo nulo en \mathbf{x}_{k_1} .

Notar que si las matrices son simétricas antes de la aplicación de las condiciones de contorno, el procedimiento propuesto rompe esta simetría y el problema numérico asociado requiere un esfuerzo computacional mayor que en el caso original. Sin embargo, en la mayoría de los casos prácticos, las matrices R y F no son simétricas aún antes de la aplicación de las condiciones de Dirichlet, por lo que no ahondamos en este trabajo sobre cómo recuperar la simetría del problema para poder utilizar el bagaje de algoritmos apropiados para problemas generalizados de autovalores con matrices hermíticas.

4.4.4.2. Condiciones de Neumann

Como discutimos en la sección 2.2.2, debido a que la ecuación de difusión sin fuente es homogénea, si el problema tiene condiciones de contorno de Neumann entonces éstas deben ser nulas. Luego, todos los elementos del integrando se anulan



$$H^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot 0 \\ h_1(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot 0 \\ h_2(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot 0 \\ h_2(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot 0 \\ h_3(\mathbf{x}) \cdot D_1(\mathbf{x}) \cdot 0 \\ h_3(\mathbf{x}) \cdot D_2(\mathbf{x}) \cdot 0 \end{bmatrix} = \mathbf{0}$$

ya que las derivadas normales son cero por condición de contorno. Luego, si ∂U_i tiene condiciones de Neumann, también podemos directamente ignorar el término de superficie del teorema de Green. En este caso, como se trata de una condición de contorno natural, no es necesario fijar ningún valor nodal de las funciones incógnita.

4.4.4.3. Condiciones de Robin

Si la frontera ∂U_i del elemento i -ésimo está sujeta a una condición de contorno de Robin entonces, según la sección 2.2.2

$$\frac{\partial \phi_g}{\partial n} = -\frac{a}{D_g(\mathbf{x})} \cdot \phi_g(\mathbf{x}) \quad (4.46)$$

para $\mathbf{x} \in \partial U_i$. Dado que es usualmente el caso, suponemos que el coeficiente a no depende ni del grupo de energía g ni de la posición \mathbf{x} . Entonces,

$$\begin{aligned} \mathbf{N}(\mathbf{x}) &= \begin{bmatrix} a/D_1 \cdot \phi_1(\mathbf{x}) \\ a/D_2 \cdot \phi_2(\mathbf{x}) \end{bmatrix} \\ &= \begin{bmatrix} a/D_1 & 0 \\ 0 & a/D_2 \end{bmatrix} \cdot \phi(\mathbf{x}) \end{aligned}$$

y

$$\begin{aligned} D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) &= \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} a/D_1 & 0 \\ 0 & a/D_2 \end{bmatrix} \cdot \phi(\mathbf{x}) \\ &= a \cdot \phi(\mathbf{x}) \\ &\approx a \cdot H(\mathbf{x}) \cdot \hat{\phi} \end{aligned}$$

De esta manera, el término de superficie de la ecuación (4.45) queda

$$\begin{aligned} \hat{\varphi}^T \left[\int_{\partial U_i} H^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) \, dS \right] &= \hat{\varphi}^T \left[\int_{\partial U_i} H^T(\mathbf{x}) \cdot a \cdot H(\mathbf{x}) \cdot \hat{\phi} \, dS \right] \\ &= \hat{\varphi}^T \cdot \left[a \cdot \int_{\partial U_i} H^T(\mathbf{x}) \cdot H(\mathbf{x}) \, dS \right] \cdot \hat{\phi} \end{aligned}$$

Para el caso de la figura 4.12, la integral resulta



$$\int_{\partial U_i} H^T(\mathbf{x}) \cdot H(\mathbf{x}) dS = \int_{\partial U_i} \begin{bmatrix} h_1 & 0 \\ 0 & h_1 \\ h_2 & 0 \\ 0 & h_2 \\ h_3 & 0 \\ 0 & h_3 \end{bmatrix} \cdot \begin{bmatrix} h_1 & 0 & h_2 & 0 & h_3 & 0 \\ 0 & h_1 & 0 & h_2 & 0 & h_3 \end{bmatrix} dS$$

$$= \int_{\partial U_i} \begin{bmatrix} h_1 h_1 & 0 & h_1 h_2 & 0 & h_1 h_3 & 0 \\ 0 & h_1 h_1 & 0 & h_1 h_2 & 0 & h_1 h_3 \\ h_2 h_1 & 0 & h_2 h_2 & 0 & h_2 h_3 & 0 \\ 0 & h_2 h_1 & 0 & h_2 h_2 & 0 & h_2 h_3 \\ h_3 h_1 & 0 & h_3 h_2 & 0 & h_3 h_3 & 0 \\ 0 & h_3 h_1 & 0 & h_3 h_2 & 0 & h_3 h_3 \end{bmatrix} dS$$

Como $h_3(\mathbf{x}) = 0$, entonces podemos ignorar las últimas dos filas y las últimas dos columnas del integrando, como así también los últimos dos elementos de los vectores $\hat{\phi}_i$ y $\hat{\phi}_i$ para prestar atención sólo a los elementos que luego serán ensamblados a las matrices globales, es decir a los nodos k_1 y k_2 que corresponden a $j = 1$ y a $j = 2$ respectivamente. Entonces la integral sobre ∂U_i se reduce a

$$\left[\int_{\partial U_i} H^T(\mathbf{x}) \cdot H(\mathbf{x}) dS \right] \cdot \hat{\phi}_i = \begin{bmatrix} \int h_1 h_1 dS & 0 & \int h_1 h_2 dS & 0 \\ 0 & \int h_1 h_1 dS & 0 & \int h_1 h_2 dS \\ \int h_2 h_1 dS & 0 & \int h_2 h_2 dS & 0 \\ 0 & \int h_2 h_1 dS & 0 & \int h_2 h_2 dS \end{bmatrix} \cdot \begin{bmatrix} \hat{\phi}_1^1 \\ \hat{\phi}_2^1 \\ \hat{\phi}_1^2 \\ \hat{\phi}_2^2 \end{bmatrix}$$

En dos dimensiones, sobre el segmento ∂U_i —que hace las veces de hipersuperficie—es posible re-parametrizar las funciones de forma $h_1(\mathbf{x})$ y $h_2(\mathbf{x})$ en función de un único parámetro $r \in [-1, 1]$, digamos

$$h_1(\mathbf{x}) = \frac{1-r}{2}$$

$$h_2(\mathbf{x}) = \frac{1+r}{2}$$

con lo que, si S_i es la longitud (área) del segmento (superficie) ∂U_i

$$\int_{\partial U_i} h_1(\mathbf{x}) \cdot h_1(\mathbf{x}) dS = \int_{-1}^1 \left(\frac{1-r}{2} \right)^2 \frac{S_i}{2} dr = \frac{S_i}{3}$$

$$\int_{\partial U_i} h_2(\mathbf{x}) \cdot h_2(\mathbf{x}) dS = \int_{-1}^1 \left(\frac{1+r}{2} \right)^2 \frac{S_i}{2} dr = \frac{S_i}{3}$$

$$\int_{\partial U_i} h_1(\mathbf{x}) \cdot h_2(\mathbf{x}) dS = \int_{-1}^1 \left(\frac{1-r}{2} \right) \left(\frac{1+r}{2} \right) \frac{S_i}{2} dr = \frac{S_i}{6}$$

Finalmente el término de superficie de la ecuación (4.45) queda



$$\int_{\partial U_i} \tilde{\varphi}^T(\mathbf{x}) \cdot D(\mathbf{x}) \cdot \mathbf{N}(\mathbf{x}) \, dS = \hat{\varphi} \cdot \left(a \cdot S_i \begin{bmatrix} 1/3 & 0 & 1/6 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 1/6 & 0 & 0 \\ 1/6 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right) \cdot \hat{\varphi}$$

La generalización para cualquier elemento bidimensional que tenga dos nodos globales k_1 y k_2 sobre una frontera con condición de contorno de Robin caracterizada por un coeficiente a indica que su contribución a la matriz global R de remociones es con un coeficiente $aS_i/3$ en la diagonal de las filas correspondientes a los nodos k_1 y k_2 para cada uno de los grupos g , y con un coeficiente $aS_i/3$ en la fila de k_1 (k_2) en la columna de k_2 (k_1), para cada g .

Para $M = 1$, el término de superficie se reduce a evaluar

$$\hat{\varphi}^T \cdot H^T(x) \cdot a \cdot H(x) \cdot \hat{\varphi}$$

sin realizar ninguna integración, ya que este término proviene de integrar por partes la expresión de las fugas en una dimensión (ver apéndice B). Para $G = 2$,

$$\begin{aligned} H^T(x) \cdot a \cdot H(x) &= \begin{bmatrix} h_1 & 0 \\ 0 & h_1 \\ h_2 & 0 \\ 0 & h_2 \end{bmatrix} \cdot a \cdot \begin{bmatrix} h_1 & 0 & h_2 & 0 \\ 0 & h_1 & 0 & h_2 \end{bmatrix} \\ &= a \begin{bmatrix} h_1 h_1 & 0 & h_1 h_2 & 0 \\ 0 & h_1 h_1 & 0 & h_1 h_2 \\ h_2 h_1 & 0 & h_2 h_2 & 0 \\ 0 & h_2 h_1 & 0 & h_2 h_2 \end{bmatrix} \end{aligned}$$

Si la condición de contorno se aplica en el nodo 1, entonces $h_1(x) = 1$ y $h_2(x) = 0$

$$H^T(x) \cdot a \cdot H(x) = a \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Luego los únicos elementos que no se anulan son los que corresponden a la diagonal de las filas que corresponden a los valores nodales de la función incógnita en el o los nodos que tienen condición de contorno de Robin para cada uno de los grupos de energía. La contribución a la matriz global R de cada una de las condiciones de Robin es entonces un término a sumando en la diagonal de las G filas que corresponden al nodo que tiene la condición.

4.5. Equivalencia con el método de Galerkin

Hemos comenzado este capítulo con la introducción del método de residuos ponderados aplicado primero a una ecuación escalar en una dimensión, luego a una ecuación en varias dimensiones espaciales y finalmente a una ecuación vectorial en varias dimensiones. Mencionamos que el método de elementos finitos se basa principalmente en el método de Galerkin que consiste en usar como funciones de peso las mismas funciones de forma con la que aproximamos la función incógnita. Sin embargo, al aplicar dicho método para resolver la ecuación de difusión



de neutrones multigrupo—que consiste en una ecuación vectorial si $G > 1$ y en varias dimensiones si $M > 1$ —hemos partido de una formulación débil del problema diferencial original diciendo primero que esa formulación era más simple desde el punto de vista del desarrollo matemático propuesto. Luego dijimos que esta formulación era equivalente al método de Galerkin que usamos para introducir las ideas de elementos finitos en ejemplos de complejidad creciente. En esta sección mostramos que efectivamente ambos enfoques son equivalentes.

En efecto, volvamos una vez más a la ecuación vectorial en varias dimensiones del principio del capítulo:

$$\mathcal{A}[\phi(\mathbf{x})] = \begin{bmatrix} \mathcal{A}_1[\phi(\mathbf{x})] \\ \mathcal{A}_2[\phi(\mathbf{x})] \\ \vdots \\ \mathcal{A}_G[\phi(\mathbf{x})] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.1)$$

con las funciones incógnita

$$\phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_G(\mathbf{x}) \end{bmatrix}$$

El método de los residuos ponderados implica escribir cada una de las G incógnitas aproximadas $\tilde{\phi}_g(\mathbf{x})$ en función de K parámetros ajustables $\hat{\phi}_g^k$ con $g = 1, \dots, G$ y $k = 1, \dots, K$, pesar e integrar los G residuos escalares en el dominio usando K funciones de peso independientes para obtener un sistema de KG ecuaciones algebraicas de donde despejar los parámetros ajustables $\hat{\phi}_g^k$. Si en particular elegimos la dependencia funcional de las incógnitas aproximadas $\tilde{\phi}(\mathbf{x})$ como una combinación lineal de K funciones de forma $h_k(\mathbf{x})$ para $k = 1, \dots, K$ siendo los parámetros ajustables los coeficientes de la combinación lineal y, además, usamos como funciones de peso las mismas funciones de forma $h_k(\mathbf{x})$ decimos que estamos usando el método de Galerkin,³ es decir, las KG ecuaciones del método de Galerkin son

$$\begin{aligned} \int_U h_1(\mathbf{x}) \cdot \mathcal{A}_1 [\tilde{\phi}(\mathbf{x})] d^M \mathbf{x} &= 0 \\ \int_U h_1(\mathbf{x}) \cdot \mathcal{A}_2 [\tilde{\phi}(\mathbf{x})] d^M \mathbf{x} &= 0 \\ &\vdots \\ \int_U h_1(\mathbf{x}) \cdot \mathcal{A}_G [\tilde{\phi}(\mathbf{x})] d^M \mathbf{x} &= 0 \\ \int_U h_2(\mathbf{x}) \cdot \mathcal{A}_1 [\tilde{\phi}(\mathbf{x})] d^M \mathbf{x} &= 0 \\ &\vdots \\ \int_U h_K(\mathbf{x}) \cdot \mathcal{A}_G [\tilde{\phi}(\mathbf{x})] d^M \mathbf{x} &= 0 \end{aligned} \quad (4.47)$$

donde recordamos la ecuación (4.10)

³El método de elementos finitos consiste en una receta para encontrar las funciones de forma a partir de la división del dominio continuo original en I elementos definidos por K nodos y suponemos que para esta página del trabajo ya hemos entendido cómo funciona.



$$\tilde{\boldsymbol{\phi}}(\mathbf{x}) = \begin{bmatrix} \tilde{\phi}_1(\mathbf{x}) \\ \tilde{\phi}_2(\mathbf{x}) \\ \vdots \\ \tilde{\phi}_G(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_1^k \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_2^k \\ \vdots \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\phi}_G^k \end{bmatrix} \quad (4.10)$$

Por otro lado, la formulación de la sección 4.3, implica que la igualdad

$$\int_U \tilde{\boldsymbol{\varphi}}(\mathbf{x}) \cdot \mathcal{A}[\tilde{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} = 0 \quad (4.48)$$

debe satisfacerse para cualesquiera valores nodales $\hat{\varphi}_g^k$ de los flujos virtuales

$$\tilde{\boldsymbol{\varphi}}(\mathbf{x}) = \begin{bmatrix} \tilde{\varphi}_1(\mathbf{x}) \\ \tilde{\varphi}_2(\mathbf{x}) \\ \vdots \\ \tilde{\varphi}_G(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\varphi}_1^k \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\varphi}_2^k \\ \vdots \\ \sum_{k=1}^K h_k(\mathbf{x}) \cdot \hat{\varphi}_G^k \end{bmatrix}$$

Ahora bien, la ecuación (4.48) es una igualdad escalar ya que su integrando es el producto interno de dos vectores de tamaño G , que podemos escribir explícitamente como

$$\int_U \left\{ \tilde{\varphi}_1(\mathbf{x}) \cdot \mathcal{A}_1[\tilde{\boldsymbol{\phi}}(\mathbf{x})] + \tilde{\varphi}_2(\mathbf{x}) \cdot \mathcal{A}_2[\tilde{\boldsymbol{\phi}}(\mathbf{x})] + \cdots + \tilde{\varphi}_G(\mathbf{x}) \cdot \mathcal{A}_G[\tilde{\boldsymbol{\phi}}(\mathbf{x})] \right\} d^M \mathbf{x} = 0$$

Si a su vez expandimos cada uno de los flujos virtuales como combinación lineal de los valores nodales $\hat{\varphi}_g^k$ tenemos

$$\begin{aligned} \int_U \left\{ (\hat{\varphi}_1^1 \cdot h_1(\mathbf{x}) + \hat{\varphi}_1^2 \cdot h_2(\mathbf{x}) + \cdots + \hat{\varphi}_1^K \cdot h_K(\mathbf{x})) \cdot \mathcal{A}_1[\hat{\boldsymbol{\phi}}(\mathbf{x})] + \right. \\ \left. (\hat{\varphi}_2^1 \cdot h_1(\mathbf{x}) + \hat{\varphi}_2^2 \cdot h_2(\mathbf{x}) + \cdots + \hat{\varphi}_2^K \cdot h_K(\mathbf{x})) \cdot \mathcal{A}_2[\hat{\boldsymbol{\phi}}(\mathbf{x})] + \right. \\ \vdots \\ \left. + (\hat{\varphi}_G^1 \cdot h_1(\mathbf{x}) + \hat{\varphi}_G^2 \cdot h_2(\mathbf{x}) + \cdots + \hat{\varphi}_G^K \cdot h_K(\mathbf{x})) \cdot \mathcal{A}_G[\hat{\boldsymbol{\phi}}(\mathbf{x})] \right\} d^M \mathbf{x} = 0 \end{aligned}$$

Distribuyendo los productos y las integrales, escribimos

$$\begin{aligned} & \hat{\varphi}_1^1 \cdot \int_U h_1(\mathbf{x}) \cdot \mathcal{A}_1[\hat{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} + \\ & \hat{\varphi}_1^2 \cdot \int_U h_2(\mathbf{x}) \cdot \mathcal{A}_1[\hat{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} + \\ & \vdots \\ & \hat{\varphi}_1^K \cdot \int_U h_K(\mathbf{x}) \cdot \mathcal{A}_1[\hat{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} + \\ & \hat{\varphi}_2^1 \cdot \int_U h_1(\mathbf{x}) \cdot \mathcal{A}_2[\hat{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} + \\ & \vdots \\ & \hat{\varphi}_G^K \cdot \int_U h_K(\mathbf{x}) \cdot \mathcal{A}_G[\hat{\boldsymbol{\phi}}(\mathbf{x})] d^M \mathbf{x} = 0 \end{aligned}$$



Como esta igualdad escalar debe satisfacerse para cualesquiera valores $\hat{\varphi}_g^k$ entonces necesariamente deben ser cero cada una de las integrales, obteniendo nuevamente las ecuaciones (4.47) que asignamos directamente al método de Galerkin. Luego ambas formulaciones son matemáticamente equivalentes, que es lo que queríamos demostrar.

4.6. Formulación matricial

El desarrollo de las ecuaciones que resultan discretizar la ecuación de difusión de neutrones con el esquema basado en elementos finitos propuesto en este capítulo es sensiblemente más complicado que el equivalente basado en volúmenes finitos introducido en el capítulo 3. Es por eso que mostramos en esta sección directamente un posible algoritmo para construir las matrices R y F que formulan el problema de autovalores discretizado a partir del cual obtener el factor de multiplicación k_{eff} y los valores nodales de la distribución de flujo $\hat{\varphi}_g^k$ sin poder escribir un resumen de todas las ecuaciones desarrolladas en el capítulo.

Dado un problema de $M = 1$ o $M = 2$ dimensiones con G grados de libertad (i.e. grupos de energía), sea I la cantidad de elementos en los que fue discretizado el dominio y sea K la cantidad de nodos. Para cada elemento $i = 1, \dots, I$ sea J_i la cantidad de nodos del elemento, que puede cambiar con i si el problema mezcla digamos elementos triangulares con cuadrangulares. Sea $N_i = J_i G$ el tamaño característico de las matrices elementales. Para cada elemento, definimos los siguientes objetos elementales:

\mathbf{r}	vector de tamaño M
\mathbf{x}	vector de tamaño M
\mathbf{h}	vector de tamaño J
$\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right]$	matriz de $M \times M$
$\left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right]$	matriz de $M \times M$
$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{r}} \right]$	matriz de $J \times M$
$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]$	matriz de $J \times M$
$H(\mathbf{x})$	matriz de $G \times N$
$B(\mathbf{x})$	matriz de $MG \times N$
$D'(\mathbf{x})$	matriz de $MG \times MG$
$A(\mathbf{x})$	matriz de $G \times G$
$X(\mathbf{x})$	matriz de $G \times G$
\hat{K}_i	matriz de $N \times N$
\hat{A}_i	matriz de $N \times N$
\hat{X}_i	matriz de $N \times N$

En la figura 4.13 mostramos un posible algoritmo para construir las matrices elementales \hat{K}_i , \hat{A}_i y \hat{X}_i , que luego podemos ensamblar con un algoritmo similar al de la figura 4.14—que puede ser incorporado en el mismo lazo del algoritmo de la figura 4.13—para obtener las



matrices globales R y F . A diferencia del algoritmo de volúmenes finitos de la figura 3.7, esta vez explícitamente incluimos la integración numérica utilizando la cuadratura de Gauss multi-dimensional de puntos fijos discutida en el apéndice C ya que si bien podríamos utilizar algún método adaptivo o utilizar alguna biblioteca científica, la integración sobre las coordenadas locales $\mathbf{r} \in \mathbb{R}^M$ está intrínsecamente ligada al método de elementos finitos. Probablemente en futuras versiones implementamos algún esquema de integración adaptivo de forma tal de optimizar la cantidad de evaluaciones de las funciones de forma y sus jacobianos. Por defecto el código milonga toma dos puntos de Gauss en cada dimensión m , y aunque el usuario puede elegir otro valor en el archivo de entrada, es esta cantidad de puntos la que utilizamos para resolver los problemas del capítulo 5.

Los algoritmos mostrados reflejan en forma aproximada lo que hace el código milonga para resolver la ecuación de difusión de neutrones multigrupo con el esquema de elementos finitos desarrollado en este capítulo, aunque no muestra todos los detalles.⁴ En el algoritmo de la figura 4.13, asumimos que tanto las funciones de forma $h_j^i(\mathbf{r})$ como sus derivadas con respecto a las coordenadas locales $\partial h_j^i / \partial r_m$ están disponibles para ser evaluadas para cualquier coordenada local $-1 < r_m < 1$. Con $(\hat{x}_m)_j^i$ denotamos a la coordenada correspondiente a la dimensión m (i.e. $x_1 = x$, $x_2 = y$ y eventualmente $x_3 = z$ en un futuro) del nodo local j del elemento i . Si bien la forma de ensamblar las matrices elementales para obtener las matrices globales R y F mostrado en la figura 4.14 puede ser utilizado, por razones de eficiencia utilizamos otro de más bajo nivel—y que se apoya en ciertas funciones provistas por la biblioteca PETSc [8] justamente para realizar ensambles en forma eficiente—que no mostramos para no complicar aún más el capítulo.

Una vez construidas y ensambladas las matrices elementales según las figuras 4.13 y 4.14 debemos colocar las condiciones de contorno. Como discutimos en la sección 2.2.2, éstas deben ser homogéneas. Llamemos ∂U_D a la frontera del dominio U que tiene condiciones de Dirichlet $\phi_g(\mathbf{x}) = 0$, ∂U_N a la que tiene condiciones tipo Neumann $\partial \phi_g / \partial n = 0$ y ∂U_R a la que tiene condiciones tipo Robin $\partial \phi_g / \partial n = a_g(\mathbf{x}) \cdot \phi_g(\mathbf{x})$. En la figura 4.15 mostramos un algoritmo para poner las condiciones de contorno de flujo nulo. Como discutimos en la sección 4.4.4.1 (y en los ejemplos de la sección 4.2), para fijar que el flujo en un cierto nodo sea cero anulamos las filas correspondiente a dicho nodo en las matrices R y F , excepto en la diagonal de R donde ponemos cualquier valor diferente de cero. En el algoritmo de la figura 4.15 (y en el código milonga) elegimos arbitrariamente el valor uno. Sin embargo, deberíamos estudiar el radio espectral de las matrices R y F antes de sobre-escribir sus filas para elegir el valor de la diagonal de forma tal de no modificar sensiblemente el esfuerzo computacional necesario para resolver el problema de autovalores, tarea que dejamos para futuras versiones. Nuevamente, el algoritmo que utiliza milonga para sobre-escribir las filas de R y de F excepto en la diagonal de R no es el de la figura 4.15 sino uno de más bajo nivel provisto por la biblioteca PETSc [8].

Como el único tipo de condición de contorno de tipo Neumann posible es de derivada

⁴Tampoco es su objetivo hacerlo. Es interesante remarcar la siguiente cita que hace Donald Knuth en su adenda del año 2002 a un paper suyo publicado en 1959 y recopilado en la referencia [31], donde comenta que Christopher Strachey dijo en 1967 [32] lo siguiente:

It is important to realise that diagrams are no suitable for conveying complicated information. They are only suitable for conveying very simple information and we use block diagrams, flow diagrams and so on in order to simplify what we are doing. We do not try to put an entire problem on to one diagram. You cannot see it all at the one time. I think there is a very important way in which we think about complicated things and that is the hierarchical way. If we have got a shoe system, we like to think of the overall system to start with and the break that down into smaller sections and consider those one at a time. [...] It is a great mistake to think that you have got to have access to everything from everything. If you do this you immediately get into a very complicated state and you cannot structure the problem in the way which is necessary to help human thought.



inicializar $R \leftarrow 0, F \leftarrow 0$
para cada elemento $i = 1, \dots, I$
 inicializar $\dot{K}_i \leftarrow 0, \dot{A}_i \leftarrow 0, \dot{X}_i \leftarrow 0$
 para cada punto de Gauss \mathbf{r}
 $\alpha \leftarrow \prod_{m=1}^M \alpha_m$
 para cada grupo $g = 1, \dots, G$
 para cada nodo $j = 1, \dots, J_i$
 $[H]_{g,g+G(j-1)} \leftarrow h_j^i(\mathbf{r})$
 para cada dimensión $m = 1, \dots, M$
 para cada dimensión $m' = 1, \dots, M$
 $\left[\begin{array}{c} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \end{array} \right]_{m,m'} \leftarrow \sum_{j=1}^{J_i} \frac{\partial h_j^i}{\partial r_{m'}} \cdot (\hat{x}_m)_{j'}^i$
 $w \leftarrow \alpha \cdot \left| \det \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right] \right|$
 $\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right] \leftarrow \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right]^{-1}$
 para cada nodo $j = 1, \dots, J_i$
 para cada dimensión $m = 1, \dots, M$
 $\left[\begin{array}{c} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \end{array} \right]_{j,m} \leftarrow \sum_{m'=1}^M \frac{\partial h_j^i}{\partial r_{m'}} \cdot \left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right]_{m',m}$
 para cada dimensión $m = 1, \dots, M$
 para cada grupo $g = 1, \dots, G$
 para cada nodo $j = 1, \dots, J_i$
 $[B]_{g+G(m-1),g+G(j-1)} \leftarrow \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]_{j,m}$
 para cada dimensión $m = 1, \dots, M$
 $[\mathbf{x}]_m \leftarrow \sum_{j=1}^{J_i} h_j^i(\mathbf{r}) \cdot [\mathbf{r}]_m$
 para cada grupo $g = 1, \dots, G$
 para cada grupo $g' = 1, \dots, G$
 $[A]_{g,g'} \leftarrow -\Sigma_{s,g \rightarrow g'}(\mathbf{x})$
 $[X]_{g,g'} \leftarrow \chi_g \cdot \nu \Sigma_{f g'}(\mathbf{x})$
 $[A]_{g,g} \leftarrow A_{g,g} + \Sigma_t(\mathbf{x}, g)$
 para cada dimensión $m = 1, \dots, M$
 $D'_{g+G(m-1),g+G(m-1)} \leftarrow D_g(\mathbf{x})$
 $\dot{K}_i \leftarrow \dot{K}_i + w \cdot (B^T \cdot D' \cdot B)$
 $\dot{A}_i \leftarrow \dot{A}_i + w \cdot (H^T \cdot A \cdot H)$
 $\dot{X}_i \leftarrow \dot{X}_i + w \cdot (H^T \cdot X \cdot H)$

Figura 4.13 – Construcción de las matrices elementales



para cada elemento $i = 1, \dots, I$
 para cada nodo $j = 1, \dots, J_i$
 para cada grupo $g = 1, \dots, G$
 $n = g + G(j - 1)$
 para cada nodo $j' = 1, \dots, J_i$
 para cada grupo $g' = 1, \dots, G$
 $n' = g' + G(j' - 1)$
 $R_{\ell(n,i),\ell(n',i)} \leftarrow R_{\ell(n,i),\ell(n',i)} + \overset{\circ}{K}_{i,n,n'} + \overset{\circ}{A}_{i,n,n'}$
 $F_{\ell(n,i),\ell(n',i)} \leftarrow F_{\ell(n,i),\ell(n',i)} + \overset{\circ}{X}_{i,n,n'}$

Figura 4.14 – Ensamble de las matrices R y F sin condiciones de contorno

para cada nodo $k_1/\mathbf{x}_{k_1} \in \partial U_D$
 para cada grupo $g = 1, \dots, G$
 para $p = 1, \dots, KG$
 $R_{\ell(k_1,g),p} \leftarrow 0$
 $F_{\ell(k_1,g),p} \leftarrow 0$
 $R_{\ell(k_1,g),\ell(k_1,g)} \leftarrow 1$

Figura 4.15 – Colocación de condiciones de contorno tipo Dirichlet

para cada nodo $k_1/\mathbf{x}_{k_1} \in \partial U_R$
 para cada grupo $g = 1, \dots, G$
 $R_{\ell(k_1,g),\ell(k_1,g)} \leftarrow R_{\ell(k_1,g),\ell(k_1,g)} + a_g(\mathbf{x})|_{\mathbf{x}=\partial U_R}$

Figura 4.16 – Colocación de condiciones de contorno tipo Robin para $M = 1$

para cada par de nodos $k_1 \wedge k_2/\mathbf{x}_{k_1} - \mathbf{x}_{k_2} \in \partial U_R$
 para cada grupo $g = 1, \dots, G$
 $S \leftarrow \|\mathbf{x}_{k_1} - \mathbf{x}_{k_2}\|$
 $R_{\ell(k_1,g),\ell(k_1,g)} \leftarrow R_{\ell(k_1,g),\ell(k_1,g)} + \frac{S}{3} \cdot a_g(\mathbf{x}) \Big|_{\mathbf{x}=\partial U_R}$
 $R_{\ell(k_2,g),\ell(k_2,g)} \leftarrow R_{\ell(k_2,g),\ell(k_2,g)} + \frac{S}{3} \cdot a_g(\mathbf{x}) \Big|_{\mathbf{x}=\partial U_R}$
 $R_{\ell(k_1,g),\ell(k_2,g)} \leftarrow R_{\ell(k_1,g),\ell(k_2,g)} + \frac{S}{6} \cdot a_g(\mathbf{x}) \Big|_{\mathbf{x}=\partial U_R}$
 $R_{\ell(k_2,g),\ell(k_1,g)} \leftarrow R_{\ell(k_2,g),\ell(k_1,g)} + \frac{S}{6} \cdot a_g(\mathbf{x}) \Big|_{\mathbf{x}=\partial U_R}$

Figura 4.17 – Colocación de condiciones de contorno tipo Robin para $M = 2$



normal nula, como mostramos en la sección (4.4.4.2), la contribución del término de superficie en ∂U_N es idénticamente nulo. Por lo tanto, no es necesario tomar ninguna acción para incorporar las condiciones de contorno de simetría en el problema. Finalmente, las figuras 4.16 y 4.17 muestran cómo colocamos las condiciones de contorno de tipo Robin (sección 4.4.4.3) sobre la frontera ∂U_R caracterizadas por un coeficiente $a_g(\mathbf{x})$, para una y para dos dimensiones respectivamente. Como ya dijimos, el manejo de problemas tridimensionales está en el TODO-list y será implementado en futuras versiones.

Resultados

But for me, the pursuit of money has always played second fiddle to the pursuit of beauty. Thank goodness for mathematics and computer science, whose methods offer many ways by which our left brains and our right brains can both be happy, simultaneously.

Donald. E. Knuth, Selected Papers on Fun & Games, 2011

Este es el capítulo en el cual realizamos la prometida comparación entre las soluciones numéricas de la ecuación de difusión de neutrones multigrupo utilizando, por un lado, el método de volúmenes finitos y por el otro, el método de elementos finitos. Para ello, resolvemos el mismo problema y obtenemos los mismos tipos de resultados—es decir, factor de multiplicación efectivo, distribución espacial de flujo neutrónico, factores de pico, potencias en ensambles, etc.—utilizando los esquemas desarrollados en los capítulos 3 y 4, respectivamente.

Comenzamos resolviendo problemas triviales unidimensionales que tienen solución analítica para introducir la metodología de análisis en casos sencillos y, además, para comparar las soluciones numéricas aproximadas de las ecuaciones diferenciales involucradas con las soluciones teóricas exactas. Luego introducimos en forma creciente complejidades en los casos a resolver, hasta llegar a benchmarks estándar y problemas que pueden llegar a tener algún nivel de interés, pero siempre teniendo en cuenta que esta monografía trata solamente sobre matemática. Las aplicaciones ingenieriles necesitan de un paso extra de juicio de experto que está fuera del alcance de este trabajo.

Básicamente, la solución de la ecuación de difusión multigrupo está a cargo del código milonga [7], desarrollado desde cero por el autor de esta monografía en el marco de su tesis doctoral, el cual a partir de su versión 0.2 ha incorporado los métodos y esquemas discutidos en los capítulos anteriores. Sin embargo, actualmente el código es capaz de resolver solamente problemas en una y en dos dimensiones espaciales. El soporte para problemas tridimensionales y otras características deseadas—como paralelización e implementación en GPUs—está planeado para futuras versiones. Este programa se distribuye en forma libre bajo la licencia General Pública GNU, por lo que comentarios, sugerencias, reportes de errores y, mejor aún, correcciones son siempre bienvenidas.

Para definir la geometría y generar las mallas no estructuradas, el código milonga lee los archivos generados por el programa gmsh [26], también disponible bajo licencia GNU GPL. Las salidas de milonga, bien pueden ser archivos de texto ASCII—completamente definidos por el usuario en el archivo de entrada de milonga, según una de las decisiones fundamentales de base de diseño del código [6]—que pueden ser interpretados y graficados con herramientas



estándar libres como gnuplot¹ o pyxplot, o archivos en formato de post-procesamiento para gmsh.

Tanto el planteo como la solución de todos los problemas presentados a lo largo de este capítulo son generados a partir de scripts que llaman a los códigos involucrados—mallador, solver y post-procesador—de forma tal que todos los resultados expuestos pueden ser recalculados y reproducidos por el lector. Para ello hemos preparado un paquete que contiene los inputs de gmsh y de milonga, junto con los scripts que ejecutan los códigos en el orden correcto y utilizando los argumentos apropiados para obtener los resultados que a continuación discutimos. Este paquete será provisto por el autor accediendo a gentiles pedidos por correo electrónico. Es por eso que cada uno de los casos resueltos sigue la misma estructura: primero damos una breve introducción y explicación sobre el problema y cómo se resuelve con milonga. Luego mostramos el input de milonga, seguido de una mímica de una terminal que muestra los comandos utilizados para llamar a los códigos apropiados junto con las salidas intermedias de los programas involucrados, para finalmente mostrar una serie de figuras con resultados gráficos e introducir las conclusiones.

La forma de utilizar tanto gmsh como milonga, y las características propias de cada programa son ilustradas mediante el uso de comentarios en los archivos de entrada. Cuando utilizamos por primera vez alguna palabra clave o construcción específica, explicamos en el input por qué y cómo solicitamos a los códigos cierta funcionalidad. Una vez introducida una característica, la utilizamos ulteriormente sin dar mayor detalles. Se refiere al lector a la documentación propia de cada código para más detalles.

Los gráficos elegidos como resultados para discutir se refieren a la comparación entre las soluciones obtenidas con el método de volúmenes finitos y el método de elementos finitos. En general, los colores de la gama del azul y los bullets con forma de polígonos cerrados indican volúmenes finitos mientras que colores basados en verde y bullets conformados por trazos abiertos denotan elementos finitos.

Para dejar sentado la versión y revisión del código milonga usado, comenzamos la resolución de los problemas de esta monografía con una invocación de milonga indicando que reporte su versión, bibliotecas asociadas y copyright (modificador -v).



Figura 5.1 – <http://www.talador.com.ar/jeremy/wasora/milonga>

¹A pesar de su nombre, este programa no es parte del proyecto GNU y más aún, según algunos puristas del free software, gnuplot no es libre ya que no permite redistribución de modificaciones sin el previo consentimiento del autor original. Debian ha decidido clasificarlo como libre, así que confiamos en que podemos utilizarlo para preparar algunas figuras sin ir en contra de los principios en los que nos basamos al escribir milonga.



```
$ milonga -v
milonga 0.2.1 trunk (2013-12-14 23:41:28 -0300 clean)
free nuclear reactor core analysis code

branch jeremy@tom-20131215024128-6m8y72lm73ok3f0c
last commit on 2013-12-14 23:41:28 -0300 (rev 1 clean)
last build on 2013-12-14 23:59:11 -0300

compiled on 2013-12-14 23:59:21 by jeremy@tom (x86_64)
with gcc (Debian 4.8.2-1) 4.8.2 using -O2 linked against
SLEPc Release Version 3.4.3, oct 14, 2013
Petsc Release Version 3.4.3, Oct, 15, 2013 linux-c-opt
running on Linux 3.2.0-4-amd64 #1 SMP Debian 3.2.51-1 x86_64
8 Intel(R) Core(TM) i7 CPU          920 @ 2.67GHz

milonga is copyright (c) 2010-2013 jeremy theler
licensed under GNU GPL version 3 or later.
milonga is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

-----
wasora 0.2.27 trunk (2013-12-14 23:18:35 -0300 clean)
wasora's an advanced suite for optimization & reactor analysis

branch jeremy@tom-20131215021835-e3becvxgbq23p0e0
last commit on 2013-12-14 23:18:35 -0300 (rev 27 clean)
last build on 2013-12-14 23:58:40 -0300

compiled on 2013-12-14 23:58:46 by jeremy@tom (x86_64)
with gcc (Debian 4.8.2-1) 4.8.2 using -O2 and linked against
GNU Scientific Library version 1.16
GNU Readline version 6.2
SUNDIALS Library version 2.5.0

wasora is copyright (C) 2009-2013 jeremy theler
licensed under GNU GPL version 3 or later.
wasora is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```



5.1. Slabs unidimensionales

Los problemas de difusión de neutrones en una dimensión tienen un interés más que nada pedagógico-académico, principalmente debido a que resultan éstos los casos más sencillos que presentan algunos de los fenómenos que se observan en geometrías y condiciones más complejas. Además, algunos problemas triviales pueden ser resueltos en forma analítica, por lo que podemos entonces entender y analizar las particularidades inherentes a las soluciones numéricas. Más aún, en algunos casos es posible incluso arribar a expresiones analíticas que involucran a las soluciones numéricas, como es el caso de la notable ecuación (5.7) deducida en la sección 5.1.1.5. En particular, la comparación entre volúmenes y elementos finitos es más ilustrativa si además podemos comparar las soluciones numéricas con las analíticas. Es por esto que comenzamos estudiando este tipo de problemas, para ir luego resolviendo problemas de complejidad creciente e introduciendo ideas y conceptos en forma paulatina, llegando a slabs compuestos de varios materiales que a su vez presentan secciones eficaces que dependen de la posición a través de distribuciones intermedias de propiedades como quemado y temperatura de combustible.

5.1.1. Slabs desnudos con propiedades homogéneas

Los casos estudiados en esta sección están basados en un una geometría tipo slab desnudo unidimensional de ancho a cuyas superficies externas son normales al eje x . La superficie izquierda está ubicada en $x = 0$ y la derecha en $x = a$ (figura 5.2). Consideramos un sólo grupo de energía ($G = 1$) y que las secciones eficaces $D_1(x)$, $\Sigma_{a1}(x)$ y $\nu\Sigma_{f1}(x)$ son uniformes en el slab.

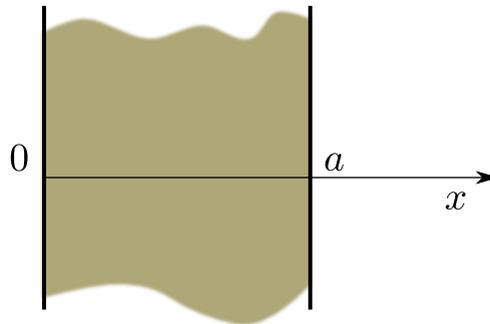


Figura 5.2 – Slab unidimensional desnudo con propiedades homogéneas

Un hipotético reactor de estas características, analizado con la teoría de difusión continua con condiciones de contorno de flujo nulo en los extremos, tiene solución analítica. El factor de multiplicación k_{eff} es

$$k_{\text{eff}} = \frac{\nu\Sigma_{f1}}{\Sigma_{a1} + D_1 \cdot \left(\frac{\pi}{a}\right)^2}$$

y la distribución de flujo resulta

$$\phi_1(x) = \frac{\pi}{2} \sin\left(\frac{x}{a} \cdot \pi\right)$$

donde el factor $\pi/2$ es tal que el valor medio del flujo sea igual a la unidad.



Para mallar los slabs objeto de estudio—y de hecho todos los problemas atacados en esta monografía—utilizamos el software libre gmsht, que requiere o bien definir la geometría utilizando una interfaz gráfica con herramientas de tipo CAD, o bien escribiendo un archivo de texto conteniendo la definición de la geometría y de las entidades físicas utilizando un tipo de language de programación, o una combinación de ambas. La geometría puede estar definida en función de variables o parámetros, y el archivo de texto de definición de geometría admite la introducción de expresiones algebraicas y algoritmos sencillos de programación, dando lugar a una gran versatilidad. Además, es posible generar la malla a partir del archivo de geometría directamente desde la línea de comandos sin tener que llamar a la interfaz gráfica, lo que posibilita que milonga llame a gmsht al momento de resolver el problema o llegado el caso, incluso re-mallar en cada iteración.

Por ejemplo, al abrir el siguiente input—que tiene sólo ocho instrucciones—con la interfaz gráfica de gmsht se observa la geometría mostrada en la figura 5.3:

slab100.geo

```
// slab unidimensional de ancho a=100

lc = 10; // longitud característica del elemento
a = 100; // ancho del slab

// definimos dos puntos, uno en x=0 y otro en x=a
// ambos con la misma longitud característica
Point(1) = {0, 0, 0, lc};
Point(2) = {a, 0, 0, lc};

// definimos una línea que une ambos puntos
Line(1) = {1, 2};

// definimos tres entidades físicas con nombre:
// los dos puntos los llamamos left y right para
// hacer referencia a ellos si queremos setear
// condiciones de contorno que no sean null
Physical Point("left") = {1};
Physical Point("right") = {2};

// la línea la llamamos "fuel" para que milonga la
// asocie con el material llamado de la misma manera
// sin tener que definir explícitamente en el input
// de milonga que juego de XS tiene que asociarle
Physical Line("fuel") = {1};
```

Al solicitarle al programa que malle esta geometría se obtiene un archivo de texto como el que sigue:

slab100.msh

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$PhysicalNames
3
0 1 "left"
0 2 "right"
1 3 "fuel"
$EndPhysicalNames
```

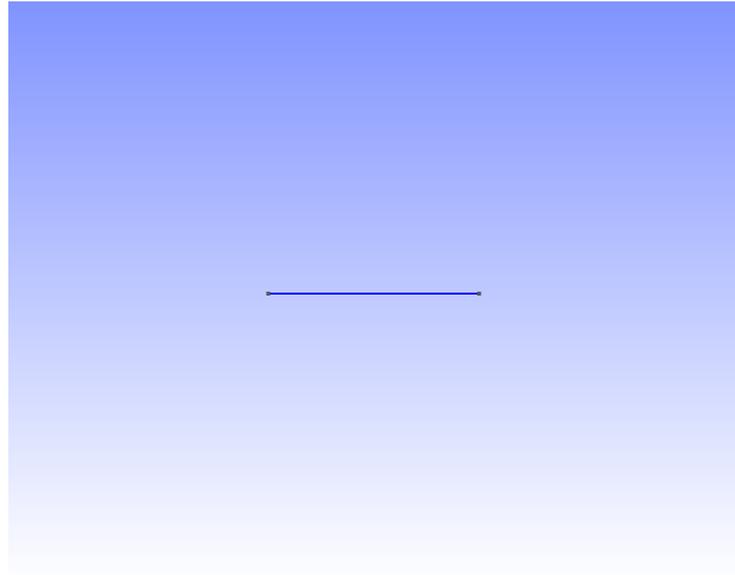


Figura 5.3 – Representación gráfica en gmsh del archivo de geometría slab100.geo

```
$Nodes
11
1 0 0 0
2 100 0 0
3 9.99999999999961581 0 0
4 19.9999999999991413 0 0
5 29.9999999999986224 0 0
6 39.9999999999984005 0 0
7 49.9999999999982369 0 0
8 59.9999999999980733 0 0
9 69.9999999999983592 0 0
10 79.9999999999989062 0 0
11 89.999999999999453 0 0
$EndNodes
$Elements
12
1 15 2 1 1 1
2 15 2 2 2 2
3 1 2 3 1 1 3
4 1 2 3 1 3 4
5 1 2 3 1 4 5
6 1 2 3 1 5 6
7 1 2 3 1 6 7
8 1 2 3 1 7 8
9 1 2 3 1 8 9
10 1 2 3 1 9 10
11 1 2 3 1 10 11
12 1 2 3 1 11 2
$EndElements
```

Es éste el formato de archivo milonga lee para definir los volúmenes/elementos necesarios para resolver el problema de difusión de neutrones en mallas no estructuradas. Utilizando este esquema de escribir un archivo de geometría y generar un archivo de malla con gmsh para luego resolver la ecuación de difusión con milonga, comparamos la solución analítica del slab



desnudo con las soluciones obtenidas utilizando volúmenes y elementos finitos.

Notar que al darle la misma longitud característica $\ell_c = 10$ tanto al punto uno ubicado en $x = 0$ como al punto dos en $x = 100$, los nodos resultantes están equiespaciados en el intervalo $[0, 100]$ todos los elementos resultan iguales con una longitud $\Delta x = \ell_c = 10$. Debemos tener en cuenta que este no siempre es el caso, y el mallador puede llegar a generar nodos que no estén uniformemente distribuidos aún cuando las longitudes características asociadas a todos los puntos sean las mismas.

5.1.1.1. Factor de multiplicación con volúmenes

El primer problema consiste en calcular el factor de multiplicación efectivo k_{eff} de un slab de ancho $a = 100$ utilizando el método de volúmenes finitos con una discretización espacial tal que la longitud característica de los volúmenes sea $\ell_c = 10$, es decir, sobre la malla generada por la geometría dada por el archivo slab100.geo mostrado arriba. Para ello, preparamos el input de milonga llamado keff-volumes.was, que listamos más abajo. Allí definimos que el problema a resolver será unidimensional a un sólo grupo de energías y que los datos de la geometría y de la malla deben ser buscados en el archivo slab100.msh. A continuación le pedimos a milonga que utilice el esquema de volúmenes finitos. Luego definimos las secciones eficaces del material llamado “fuel”—que coincide con la entidad física llamada “fuel” definida en el archivo de geometría—como homogéneas e idénticamente iguales a

$$\begin{aligned} D_1(x) &= 1 \\ \Sigma_{a1}(x) &= 0,1 \\ \nu\Sigma_{f1}(x) &= 0,101 \end{aligned}$$

Las unidades de estas propiedades deben ser consistentes con las unidades de longitud en las cuales está definida la geometría, es decir, si a —y por ende todas las longitudes—está en centímetros, el coeficiente de difusión debe estar también en centímetros y las secciones eficaces deben estar dadas en cm^{-1} . Una vez que todas las entidades físicas cuya dimensión es igual a la del problema—i.e. todos los segmentos—han sido asignados a un material, estamos en condiciones de solicitarle a milonga que resuelva el problema. Como no hemos definido ninguna condición de contorno, el código toma por defecto condiciones de flujo nulo en los extremos. Finalmente, la última línea del input indica a milonga que imprima en la salida estándar el valor de la variable k_{eff} —que acaba de ser calculada—con una precisión de diez cifras decimales. Este input es muy sencillo y consiste en cinco líneas más una definición de geometría sencilla, que estrictamente puede ser reducido incluso aún más. Esta característica de que problemas sencillos requieran inputs sencillos—y de que problemas complejos requieran inputs complejos—es también una decisión de base de diseño [7, 6].

Como también mostramos en el mímico de la terminal más abajo, para ejecutar los códigos y resolver el problema, primero debemos llamar a gmesh con la instrucción que realice un mallado sobre los objetos unidimensionales del archivo de geometría slab100.geo, con lo que generamos el archivo slab100.msh que necesita milonga para correr. Finalmente, ejecutamos milonga dando como único argumento el nombre del input, keff-volumes.was.

keff-volumes.was

```
# problema unidimensional a un grupo de energias
# sobre la malla no estructurada definida en slab100.msh
MESH FILE_PATH slab100.msh DIMENSIONS 1 DEGREES 1
SCHEME VOLUMES # volumenes finitos
```



```
# definimos un material llamado fuel que abarca la
# entidad fisica definida en slab100.geo llamada
# tambien fuel con propiedades uniformes
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101

# pedimos a milonga que resuelva el problema
MILONGA_SOLVE

# y una vez que fue resuelto , imprimimos el resultado
PRINT %.10f keff
```

```
$ gmsh -1 slab100.geo
Info : Running 'gmsh -1 slab100.geo' [Gmsh 2.8.3, 1 node, max. 1 thread]
Info : Started on Sun Dec 15 00:02:19 2013
Info : Reading 'slab100.geo'...
Info : Done reading 'slab100.geo'
Info : Meshing 1D...
Info : Meshing curve 1 (Line)
Info : Done meshing 1D (0 s)
Info : 11 vertices 12 elements
Info : Writing 'slab100.msh'...
Info : Done writing 'slab100.msh'
Info : Stopped on Sun Dec 15 00:02:19 2013
$ milonga keff-volumes.was
1.0002092549
$
```

5.1.1.2. Factor de multiplicación con elementos

Para resolver el mismo problema pero utilizando el método de elementos finitos, lo único que debemos hacer es reemplazar la palabra VOLUMES por ELEMENTS en el input anterior y volver a ejecutar milonga. Como ya generamos la malla, no es necesario volver a ejecutar gmsh.

keff-elements.was

```
MESH_FILE_PATH slab100.msh DIMENSIONS 1 DEGREES 1
SCHEME ELEMENTS
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101
MILONGA_SOLVE
PRINT %.10f keff
```

```
$ milonga keff-elements.was
1.0000484746
$
```

Resulta interesante escribir las dos soluciones encontradas, una abajo de otra

$k_{\text{eff}} = 1,0002092549$ con volúmenes finitos

$k_{\text{eff}} = 1,0000484746$ con elementos finitos



para apreciar que existe una diferencia de un orden de magnitud entre ambas soluciones, atribuible a la gruesa discretización espacial. Es de esperar que esta diferencia se reduzca, y las soluciones numéricas tiendan a la solución analítica si tomamos mallas cada vez más finas.

5.1.1.3. Dependencia con la malla

Para estudiar entonces cómo cambian las soluciones numéricas con la malla, ejecutamos milonga en forma paramétrica sobre la variable n , que usamos para definir la longitud característica como $\ell_c = a/n$. Esto lo logramos teniendo un archivo de geometría template llamado slab.tpl, en todo igual al archivo slab100.geo mostrado antes, pero que en lugar de definir tanto a como ℓ_c a valores fijos, utilizando los comodines \$1 y \$2:

slab.tpl

```
a = $1;
lc = $2;
Point(1) = {0, 0, 0, lc};
Point(2) = {a, 0, 0, lc};

Line(1) = {1, 2};
Physical Line("fuel") = {1};
Physical Point("left") = {1};
Physical Point("right") = {2};
```

Este archivo de template es “rellenado” por milonga, a través del keyword TEMPLATE que reemplaza \$1 por el tercer argumento (en este caso la variable a) y \$2 por el argumento siguiente (en este caso ℓ_c), dejando un archivo de geometría llamado slab.geo (el segundo argumento) que gmsh puede entender. Una vez generada la geometría correspondiente al valor de n deseado, milonga invoca a gmsh para generar el archivo de malla slab.msh con el keyword SHELL. En este momento, estamos en condiciones primero de leer la malla que acabamos de generar con el keyword MESH y luego de resolver el problema explícitamente mediante el keyword PROBLEM_SOLVE.

Para comparar el k_{eff} numérico con la solución analítica, calculamos una variable k_{real} como

$$k_{\text{real}} = \frac{\nu \Sigma_{f1}(a/2)}{\Sigma_{a1}(a/2) + D_1 \cdot \left(\frac{\pi}{a}\right)^2}$$

Una vez parseado el archivo de malla y asignados los materiales, milonga deja las distribuciones de secciones eficaces macroscópicas en función del espacio (en este caso de la única coordenada x) para que el usuario pueda evaluarlas, integrarlas, etc. En este caso, para $0 < x < a$ las secciones eficaces se evalúan a la constante entrada en la definición del material “fuel”, con lo que el argumento de dichas funciones no es relevante, mientras que se encuentre en el intervalo $0 < x < a$. De esta manera, podemos calcular el k_{eff} y la diferencia entre el k_{real} e imprimirla en la salida estándar con la instrucción PRINT.

En el transcurso de este capítulo vamos a estar comparando soluciones en volúmenes y en elementos. Para poder resolver el mismo problema definido en el archivo de entrada con ambos métodos, podemos usar los modificadores “- -volumes” y “- -elements” en la línea de comandos a continuación del nombre del archivo de entrada. Estos modificadores tienen precedencia sobre la palabra clave SCHEME, si es que ésta aparece en el input.

Las variables especiales cells y nodes contienen la cantidad de celdas y de nodos respectivamente. En problemas resueltos con el método de elementos finitos, el número de celdas es



ceros. En cada paso del cálculo paramétrico, escribimos en la salida estándar una fila de datos que contiene el número de nodos y el número de celdas (que puede llegar a ser cero), el k_{eff} numérico, el k_{eff} analítico y el valor absoluto de su diferencia. Al redireccionar la salida estándar a un archivo, obtenemos un conjunto de datos que puede ser fácilmente graficado.

La terminal muestra, además de la invocación a milonga, cómo se ven los datos de salida en cada uno de los dos casos. Como este caso es el primero en el cual graficamos la solución, mostramos el script de entrada para el programa pyxplot (distribuido libremente bajo licencia GPL) que utilizamos para generar las figuras en formato PDF.

keff-vs-mesh.was

```
# calcula el keff vs. la cantidad de incongnitas
# para un slab de ancho a constante

# hacemos variar parametricamente (i.e. resolvemos el problema para
# cada valor del parametro) la variable n entre 10 y 200 (inclusive)
# con incrementos de 10
# si descomentamos la parte de max_daughters podemos hacer el calculo
# en paralelo (-1 indica utilizar todos los nucleos disponibles)
PARAMETRIC n 10 200 10 # MAX_DAUGHTERS -1

# la variable a contiene el ancho del slab, y la fijamos en 100
# la necesitamos para generar el archivo de geometria y para
# calcular el keff analitico del slab
a = 100

# la longitud caracteristica de los elementos es a/n
# mientras mas grande sea n (con a fijo), mas chico es lc
lc = a/n

# generamos un archivo slab.geo a partir de un template slab.tpl
# donde reemplazamos $1 por el valor de a y $2 por el valor de lc
TEMPLATE slab.tpl slab.geo a lc

# llamamos a gmsh con la geometria recién creada
SHELL "gmsh_v0_2_slab.geo"

# leemos la malla (y definimos las dimensiones del problema)
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1

# el mismo material del ejemplo anterior (debe venir despues de MESH)
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101

# resolvemos el problema
MILONGA_SOLVE

# calculamos el keff analitico segun la formula a un grupo,
# las XS son uniformes asi que da lo mismo donde las evaluamos
# usamos el punto medio del slab x = a/2 por elegancia
kreal = nuSigmaF(a/2)/(SigmaA(a/2) + D(a/2)*(pi/a)^2)

# imprimimos en la salida estandar keff, kreal y el valor
# absoluto del error cometido en funcion de n, de la cantidad
# de nodos (fem) y de la cantidad de celdas (fom)
# notar que si el calculo se hace en paralelo es posible
# que las lineas no salgan ordenadas con n ascendente
PRINT %.f n nodes cells %.12f keff kreal %e abs(keff-kreal)
```



```
$ milonga keff-vs-mesh.was --volumes > keff-vs-mesh-volumes.dat
$ cat keff-vs-mesh-volumes.dat
10    11    10    1.000209254926  1.000129121224  8.013370e-05
20    21    20    1.000149202928  1.000129121224  2.008170e-05
30    31    30    1.000138050405  1.000129121224  8.929181e-06
40    41    40    1.000134144672  1.000129121224  5.023448e-06
50    51    50    1.000132336462  1.000129121224  3.215239e-06
60    61    60    1.000131354116  1.000129121224  2.232892e-06
70    71    70    1.000130761755  1.000129121224  1.640531e-06
80    81    80    1.000130377274  1.000129121224  1.256051e-06
90    91    90    1.000130113669  1.000129121224  9.924458e-07
100   101   100   1.000129925111  1.000129121224  8.038871e-07
110   111   110   1.000129785597  1.000129121224  6.643732e-07
120   121   120   1.000129679484  1.000129121224  5.582604e-07
130   131   130   1.000129596903  1.000129121224  4.756791e-07
140   141   140   1.000129531376  1.000129121224  4.101529e-07
150   151   150   1.000129478513  1.000129121224  3.572895e-07
160   161   160   1.000129435248  1.000129121224  3.140246e-07
170   171   170   1.000129399391  1.000129121224  2.781675e-07
180   181   180   1.000129369342  1.000129121224  2.481188e-07
190   191   190   1.000129343912  1.000129121224  2.226886e-07
200   201   200   1.000129322200  1.000129121224  2.009766e-07
$ milonga keff-vs-mesh.was --elements > keff-vs-mesh-elements.dat
$ cat keff-vs-mesh-elements.dat
10    11    0    1.000048474649  1.000129121224  8.064657e-05
20    21    0    1.000109007316  1.000129121224  2.011391e-05
30    31    0    1.000120185676  1.000129121224  8.935548e-06
40    41    0    1.000124095760  1.000129121224  5.025463e-06
50    51    0    1.000125905159  1.000129121224  3.216064e-06
60    61    0    1.000126887933  1.000129121224  2.233290e-06
70    71    0    1.000127480477  1.000129121224  1.640747e-06
80    81    0    1.000127865046  1.000129121224  1.256177e-06
90    91    0    1.000128128699  1.000129121224  9.925248e-07
100   101   0    1.000128317284  1.000129121224  8.039391e-07
110   111   0    1.000128456815  1.000129121224  6.644087e-07
120   121   0    1.000128562938  1.000129121224  5.582855e-07
130   131   0    1.000128645526  1.000129121224  4.756973e-07
140   141   0    1.000128711057  1.000129121224  4.101665e-07
150   151   0    1.000128763924  1.000129121224  3.572998e-07
160   161   0    1.000128807191  1.000129121224  3.140325e-07
170   171   0    1.000128843050  1.000129121224  2.781738e-07
180   181   0    1.000128873100  1.000129121224  2.481238e-07
190   191   0    1.000128898531  1.000129121224  2.226926e-07
200   201   0    1.000128920244  1.000129121224  2.009799e-07
$ pyxplot keff-vs-mesh.ppl
$ pyxplot error-vs-mesh.ppl
$
```

En la figura 5.4 vemos que volúmenes finitos sobrestima el factor de multiplicación y elementos finitos lo subestima. Si bien en la figura el eje de las abscisas es el parámetro n , la cantidad de incógnitas—y por ende el tamaño de las matrices R y F —no es igual en volúmenes que en elementos. En el primer caso, la cantidad de incógnitas es igual a la cantidad N de celdas, mientras que en el segundo es igual a la cantidad M de nodos. En una dimensión, $M = N + 1$.

Observamos en la figura 5.4—y en los datos contenidos en los archivos de texto de salida en la terminal—que en función de n , los errores son sorprendentemente simétricos con respecto a la solución analítica. Discutimos esta particularidad más adelante.

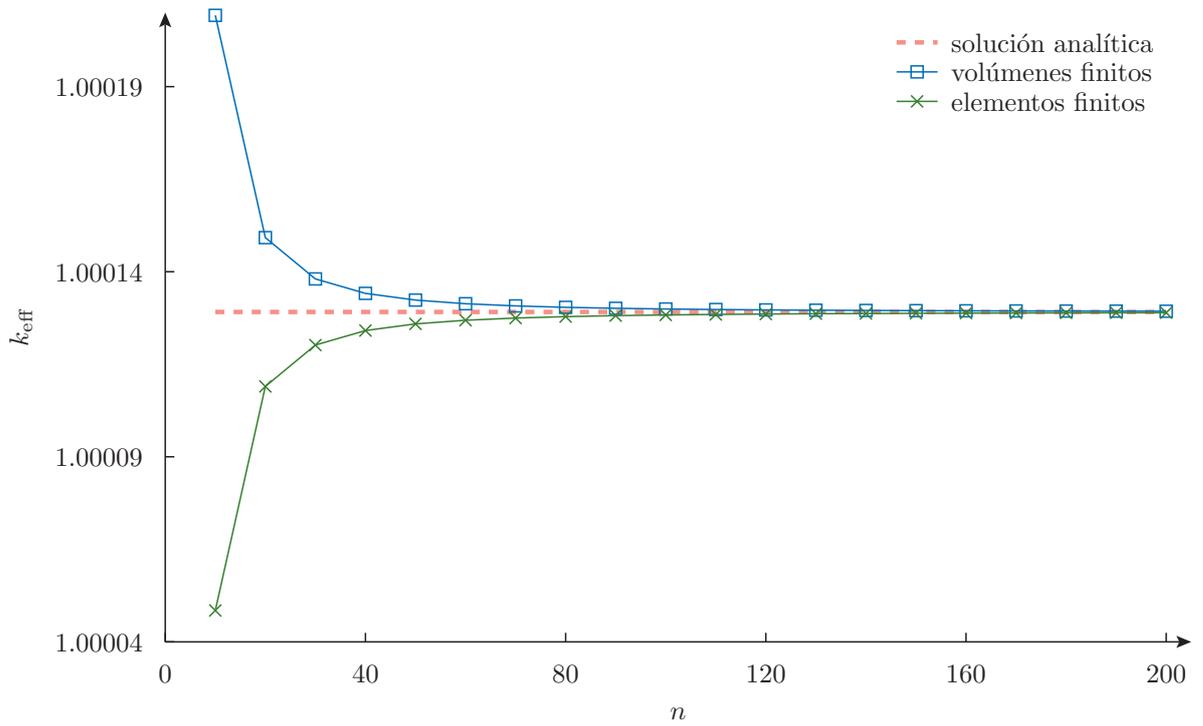


Figura 5.4 – k_{eff} calculado numéricamente con volúmenes y con elementos para diferentes valores del parámetro n que indica la refinación de la malla.

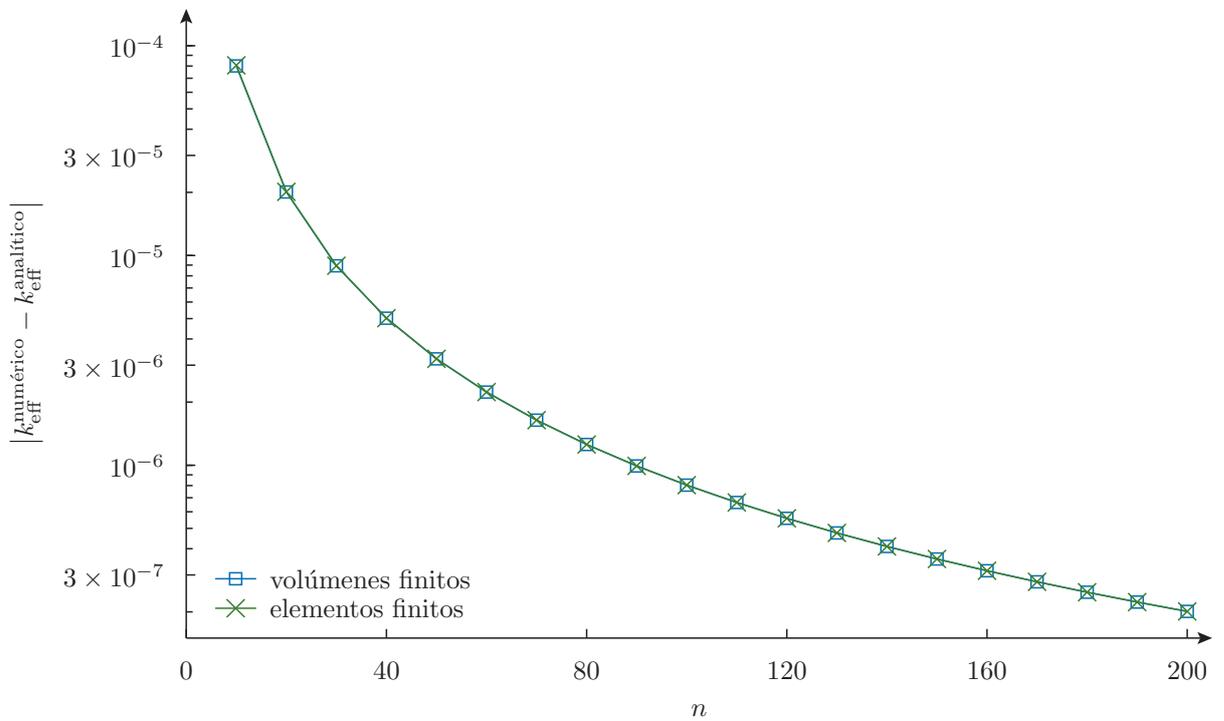


Figura 5.5 – Error absoluto cometido al calcular numéricamente k_{eff} en función del parámetro n .



El primer punto de la figura corresponde a $n = 10$, que es el caso resuelto en las secciones 5.1.1.1 y 5.1.1.2. Vemos allí que, tal como sospechábamos, los errores cometidos son efectivamente significativos debido a que la malla es demasiado gruesa. Esta diferencia disminuye a medida que aumenta n y a partir de aproximadamente $n \gtrsim 100$, los valores de k_{eff} son prácticamente indistinguibles en la escala de la figura 5.4. Es por eso que en la figura 5.5 graficamos el valor absoluto del error en escala logarítmica para ambos casos. Afortunadamente, el error tiende a cero cuando $\ell_c \rightarrow 0$. Esto indica—no demuestra—que ambos esquemas son convergentes, consistentes y estables, al menos para la configuración unidimensional utilizada para resolver este problema en particular. Aún en escala logarítmica, los valores absolutos de los errores cometidos por ambos métodos en función de n son indistinguibles—aunque sabemos por la figura 5.4 que tienen signo contrario.

5.1.1.4. Dependencia con el ancho del slab

En esta sección estudiamos, para una cantidad fija n de volúmenes/elementos, cómo varía el error cometido al calcular factor de multiplicación numéricamente con respecto al teórico analítico con ambos esquemas. Esta vez, fijando $n = 50$, hacemos variar paramétricamente el valor del ancho a entre 40 y 160 con incrementos $\Delta a = 5$ en unidades de longitud consistentes con las de las secciones eficaces.

Esta vez, en lugar de calcular k_{real} como una variable, la definimos como una función del parámetro a . De esta manera, podemos obtener la curva continua de $k_{\text{real}}(a)$ contra la cual comparar las soluciones numéricas gráficamente, en lugar de tenerla evaluada sólo para los valores de a utilizados en la solución numérica. Por un lado escribimos esta función cuasi-continua en el archivo `kreal-vs-a.dat` utilizando pequeños intervalos $\Delta a = 0,1$ y por otro escribimos en la salida estándar, cómo depende el k_{eff} numérico con el ancho a para cada paso del cálculo paramétrico.

keff-vs-a.was

```
# calcula el keff vs. el ancho a de un slab manteniendo
# la cantidad de elementos/volumenes constante

# calculo parametrico sobre el ancho a
PARAMETRIC a 40 160 5

n = 50      # cantidad de elementos
lc = a/n    # longitud caracteristica

TEMPLATE slab.tpl slab.geo a lc
SHELL "gmesh_v_0_2_slab.geo"
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1

MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101

MILONGA_SOLVE

# definimos una funcion que da el keff analitico en funcion del
# ancho a con las XS del material definido arriba esta es una
# funcion de a (en lugar de una variable como en el caso anterior)
# porque la necesitamos asi para poder escribirla con un step mas
# chico con la instruccion PRINT_FUNCTION
kreal(a) := nuSigmaF(0)/(SigmaA(0) + D(0)*(pi/a)^2)

# escribimos en un archivo de texto la funcion kreal en
# forma casi-continua (delta a = 0.1)
```



```

OUTPUT_FILE analitico kreal-vs-a-%g.dat a
PRINT_FUNCTION_FILE analitico kreal MIN 40 MAX 160 STEP 0.1

# en cada corrida parametrica , escribimos en la salida estandar
# el keff calculado numericamente y el error vs el ancho a
PRINT %.f a %.12f keff %e abs(keff-kreal(a))

```

```

$ milonga keff-vs-a.was --elements > keff-vs-a-elements.dat
$ milonga keff-vs-a.was --volumes > keff-vs-a-volumes.dat
$ pyxplot keff-vs-a.ppl
$ pyxplot error-vs-a.ppl
$

```

En la figura 5.6, las soluciones numéricas parecen estar sobre la curva que da la solución analítica del k_{eff} del problema. Sin embargo, un gráfico en escala logarítmica del valor absoluto del error cometido indica que a medida que aumenta a , el error disminuye (figura 5.7). Esto se explica de la siguiente manera. Para anchos a pequeños la curvatura resultante del flujo es más grande que para valores de a mayores, y por lo tanto la importancia relativa del término de difusión—que es la divergencia del gradiente del flujo, calculada numéricamente—con respecto a los términos de fisiones y absorciones también lo es. Como este término depende mucho más fuertemente de la discretización espacial que los otros, entonces para la misma cantidad de volúmenes/elementos, es de esperar que el error cometido para a pequeños sea mayor. De hecho, si $a \rightarrow \infty$ entonces estaríamos en el caso de un reactor infinito en el cual el flujo es independiente de la posición x y por lo tanto la curvatura de la solución es cero. En este caso, cualquier esquema numérico debería arrojar la solución analítica, a menos de errores de redondeo por representación de números reales en coma flotante y de las tolerancias en los métodos numéricos de resolución del problema de autovalores asociado. Nuevamente, volúmenes finitos sobreestima k_{eff} y elementos finitos lo subestima, y los valores absolutos de los errores cometidos en ambos casos son prácticamente iguales.

5.1.1.5. Distribución de flujo

Además del autovalor k_{eff} , resulta interesante comparar el autovector asociado ϕ que representa la distribución de flujo del reactor crítico. Para el mismo slab de ancho a , volúmenes finitos da como resultado el valor medio del flujo $\phi(x)$ en cada una de las celdas y que suponemos es uniforme dentro de cada una de ellas. Por otro lado, elementos finitos calcula los valores que toma $\phi(x)$ en las posiciones donde se encuentran los nodos y, más aún, indica explícitamente cómo interpolar mediante las funciones de forma utilizadas para discretizar el problema y poder evaluar $\phi(x)$ para cualquier valor de $0 < x < a$.

Para comparar las distribuciones de flujo resultantes, volvemos a resolver el mismo slab de la figura 5.2. Esta vez, utilizamos la capacidad que tiene milonga de interpretar argumentos en la línea de comandos y reemplazarlos en el archivo de entrada, que funciona de la siguiente manera: si en el input aparece una construcción de la forma $\$i$ siendo i algún número entero positivo, dicha construcción es reemplazada literalmente por el i -ésimo argumento que aparece luego del nombre del archivo de entrada en la línea de comandos. De esta manera, el esquema numérico (volúmenes o elementos) lo seleccionamos dando o bien la palabra “volumes” o “elements” sin los guiones a modo de prefijo, y luego pasando como argumento del keyword SCHEME la construcción $\$1$. De esta manera, además de elegir el esquema numérico, podemos definir nombres de archivos de salida diferentes para cada uno de los métodos en lugar de tener que redireccionar la salida estándar a un archivo con un nombre apropiado.

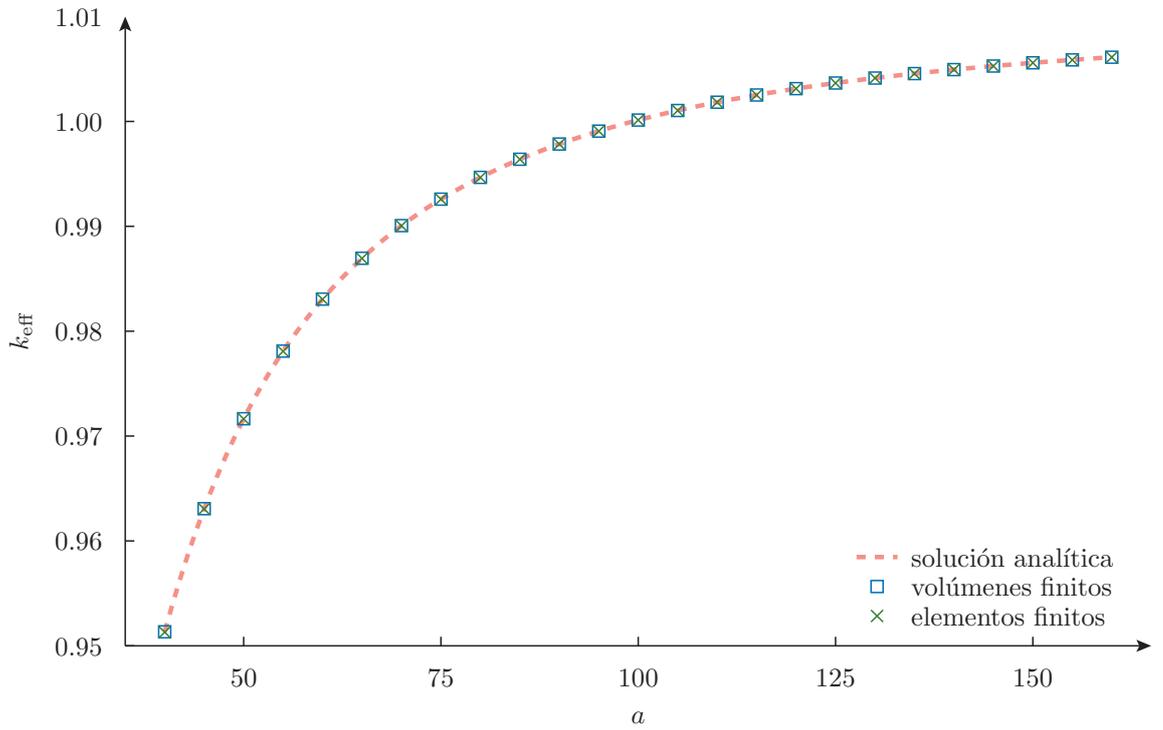


Figura 5.6 – k_{eff} calculado numéricamente con volúmenes y con elementos para diferentes valores del ancho a del slab.

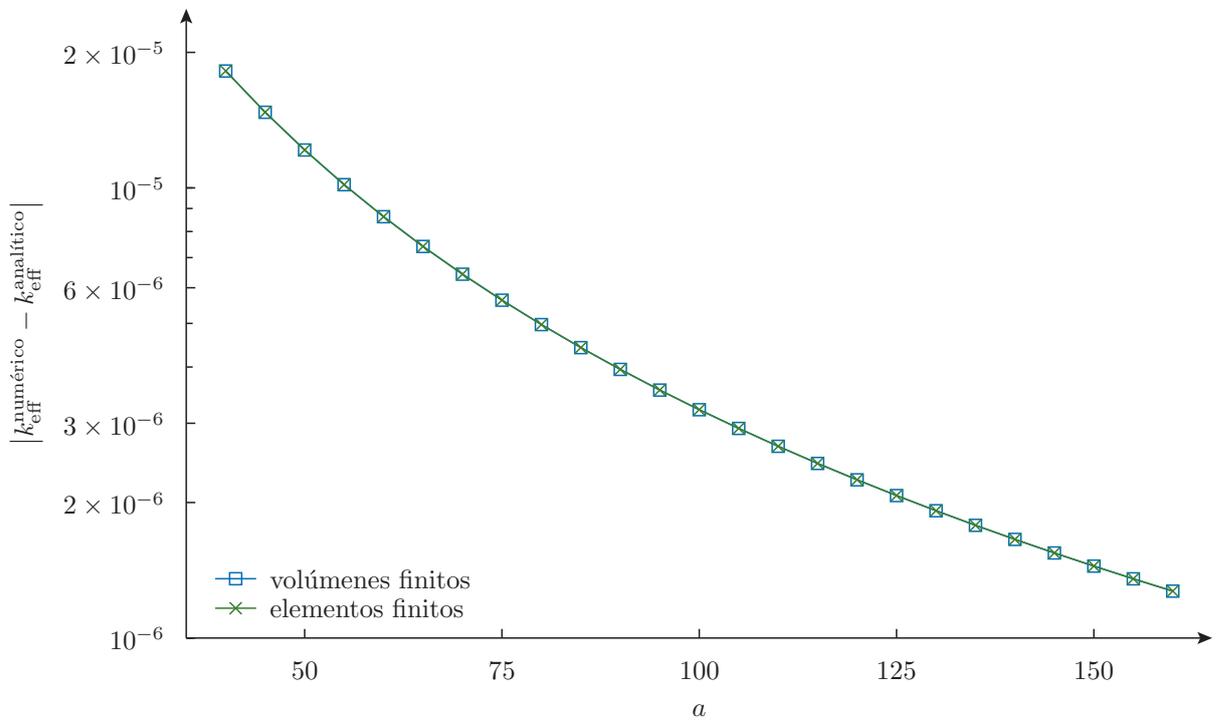


Figura 5.7 – Error absoluto cometido al calcular numéricamente k_{eff} en función del ancho a .



De la misma forma, el parámetro n lo seleccionamos desde la línea de comandos y así, utilizando el mismo archivo de input, podemos resolver el mismo slab para una malla gruesa y para una malla fina simplemente invocando a milonga con diferentes valores de n en la línea de comandos. En este caso, hacemos que la longitud característica l_c sea igual a a/n y es éste valor el que utilizamos para generar el archivo de geometría a partir del template. Una vez resuelto el problema, además de dejar en la variable keff el valor del factor de multiplicación efectivo calculado numéricamente, milonga define G funciones $\phi_g(\mathbf{x})$ continuas, que pueden ser evaluadas, integradas, derivadas, etc. Si el argumento \mathbf{x} coincide con alguno de los puntos correspondiente a la ubicación geométrica de las incógnitas—i.e el baricentro de una celda en volúmenes o la ubicación de un nodo en elementos—entonces $\phi_g(\mathbf{x})$ se evalúa al valor correspondiente. De lo contrario, $\phi_g(\mathbf{x})$ devuelve la interpolación que corresponda según el método utilizado para el cálculo.

Para comparar no sólo las soluciones numéricas entre sí sino también con respecto a la solución analítica

$$\phi_{\text{real}}(x) = \frac{\pi}{2} \cdot \sin\left(\pi \cdot \frac{x}{a}\right) \quad (5.1)$$

entonces definimos, además de una función llamada $\text{flujoreal}(x)$, otra llamada $\text{error}(x)$ que es igual a la diferencia entre la función definida por puntos—pero que se puede evaluar en cualquier punto del dominio— $\phi_1(x)$ provista por milonga y la función definida algebraicamente $\phi_{\text{real}}(x)$. Para entender mejor cómo son las interpolaciones propuestas, generados dos archivos de salida en este problema: uno conteniendo sólo los puntos de definición de $\phi_1(x)$ y otro barriendo el intervalo $[0, a]$ con un incremento $\Delta x = 0,02$. Como ilustramos en la terminal, resolvemos el mismo problema tanto para volúmenes como para elementos sobre una malla gruesa con $n = 20$ y sobre una malla fina $n = 120$.

flux.was

```
# calcula la distribucion de flujo de un slab de ancho a
# da como salida el flujo numerico, el flujo analitico y
# la diferencia en funcion de la posicion x:
# 1. evaluada solamente en los puntos calculados
# 2. interpolada en todo el ancho del slab

# usamos como esquema numerico lo que nos hayan pedido en
# la linea de comandos a continuacion del input
SCHEME $1

a = 100      # ancho a fijo
n = $2      # cantidad de elementos dado en la linea de comandos
lc = a/n

TEMPLATE slab.tpl slab.geo  a  lc
SHELL "gmsht-v_0_-2_slab.geo"
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101
MILONGA SOLVE

# definimos una funcion de x con la solucion analitica
flujoreal(x) := pi/2*sin(pi*x/a)
# y otra con la diferencia entre el flujo numerico calculado
# por milonga y dado en la funcion phi_1(x)
error(x) := phi_1(x) - flujoreal(x)

# generamos el nombre del archivo de salida con los puntos
```



```

# evaluados en la malla incluyendo el esquema y la cantidad
# de elementos/volumenes
OUTPUT_FILE grid flux-$1-$2-grid.dat
# escribimos en el archivo grid las tres funciones evaluadas
# solo en los puntos incognita
PRINT_FUNCTION phi_1 flujoreal error FILE grid

# lo mismo pero para las funciones interpoladas
OUTPUT_FILE cont flux-$1-$2-cont.dat
# escribimos en el archivo cont las tres funciones interpoladas
# como cada delta x = 0.1 en el rango [0:a]
PRINT_FUNCTION phi_1 flujoreal error FILE cont MIN 0 MAX a STEP a/2000

```

```

$ milonga flux.was volumes 20
$ milonga flux.was elements 20
$ pyxplot flux-coarse.ppl
$ milonga flux.was volumes 120
$ milonga flux.was elements 120
$ pyxplot flux-fine.ppl
$ pyxplot error-coarse.ppl
$

```

La figura 5.8 muestra los perfiles de flujo calculados con ambos métodos para una malla gruesa con $I = 20$ y la figura 5.9 la distribución para $I = 120$. Ambas figuras sugieren que existe una relación entre las $K = I + 1$ soluciones nodales ϕ_i^\times obtenidas por elementos finitos y los valores medios de las I celdas ϕ_i^\square de tal forma que

$$\phi_i^\square = \frac{\phi_i^\times + \phi_{i+1}^\times}{2} \quad (5.2)$$

En la figura 5.10 mostramos la diferencia entre la solución lineal a trozos obtenida por el método de elementos finitos y la solución analítica dada por la ecuación (5.1). Para volúmenes finitos solamente mostramos el error cometido en los centros de las celdas ya que si incluyéramos el error cometido por la solución constante a trozos en todos los puntos deberíamos cambiar la escala del eje vertical. Vemos allí que en los nodos—indicados por cruces—el error es máximo por exceso y en los centros de las celdas—indicados por cuadrados—el error es máximo por defecto. Por supuesto, a medida que aumenta I , los errores con respecto a la solución analítica disminuyen. Sin embargo, esta figura realizada para una malla gruesa desliza, tal como las dos anteriores, que la ecuación (5.2) parece ser exacta para un slab uniforme a un grupo con condiciones de contorno de flujo nulo independientemente de la nodalización.

En efecto, demosremos que el operador diferencia d^2/dx^2 con condiciones de contorno Dirichlet nulas tiene expresiones similares tanto en elementos como en volúmenes. Partamos de la ecuación de Laplace en un slab de ancho a

$$\frac{\partial^2 \phi}{\partial x^2} = 0 \quad (5.3)$$

Para volver a derivar las ecuaciones del método de volúmenes finitos del capítulo 3 dividamos el slab en I celdas uniformes de ancho Δx , donde la celda i -ésima está definida por las coordenadas $x_i < x < x_{i+1}$ (figura 5.11), e integremos la ecuación (5.3) entre x_i y x_{i+1} con $1 < i < N$.

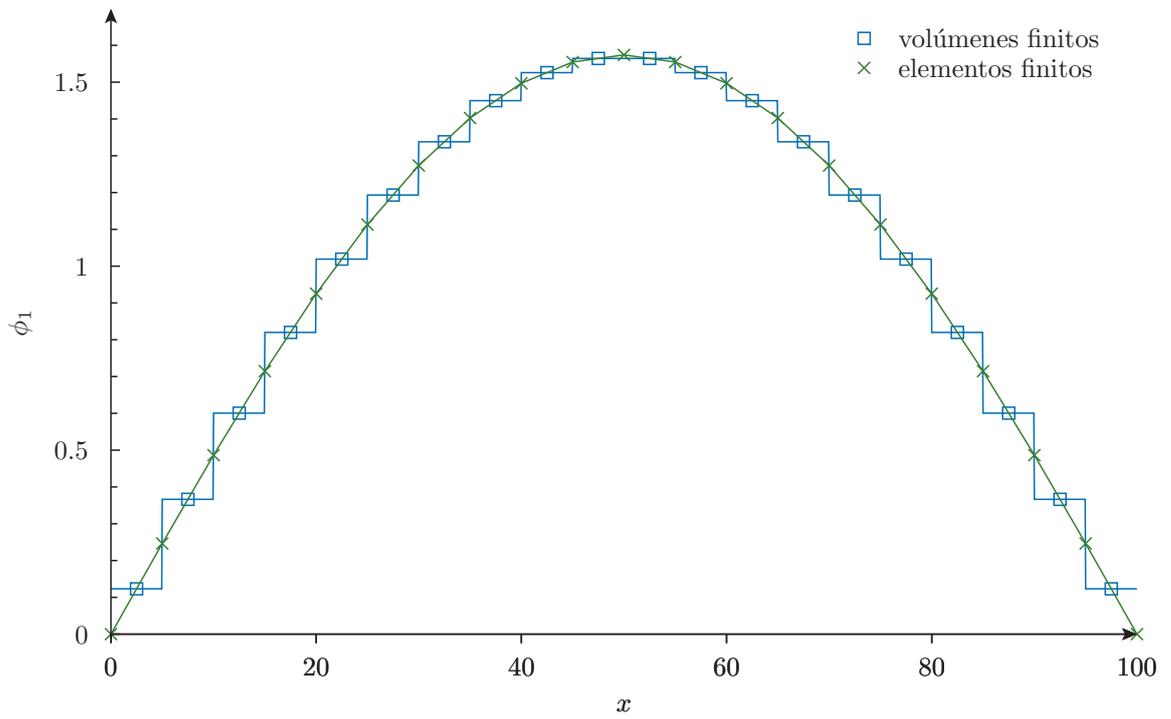


Figura 5.8 – Perfil de flujo para un slab uniforme con condiciones de contorno de flujo nulo en ambos extremos a un grupo de energía para $I = 20$ elementos.

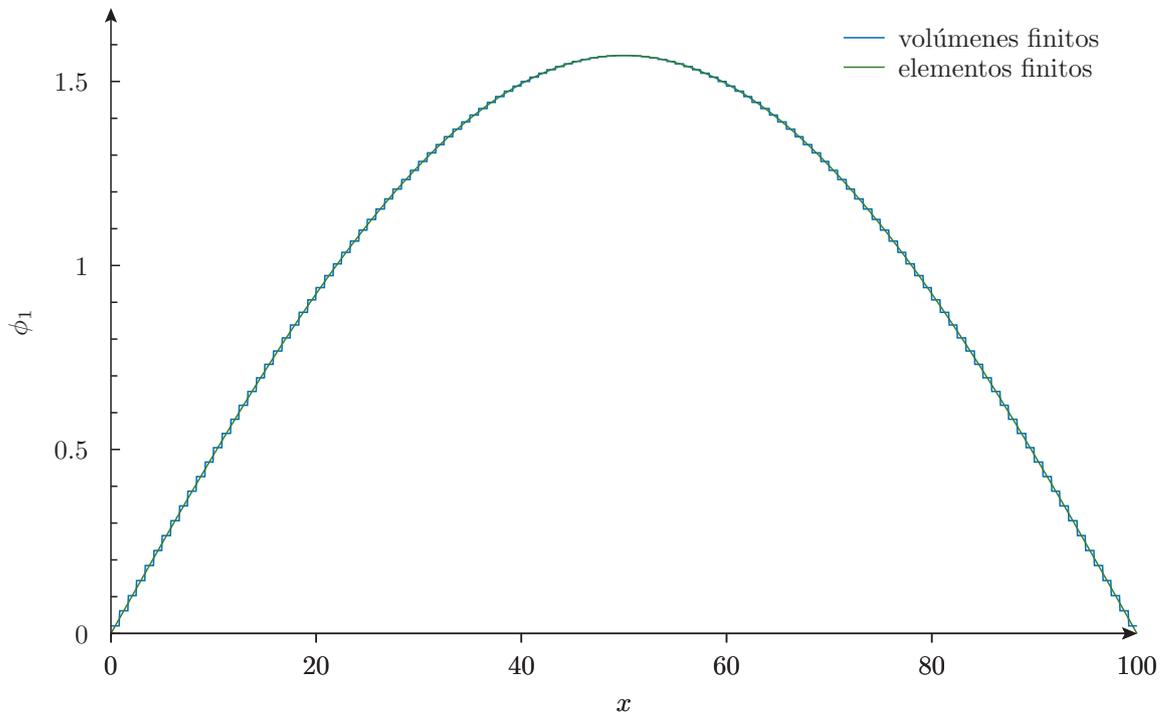


Figura 5.9 – Perfil de flujo para un slab uniforme con condiciones de contorno de flujo nulo en ambos extremos a un grupo de energía para $I = 120$ elementos.

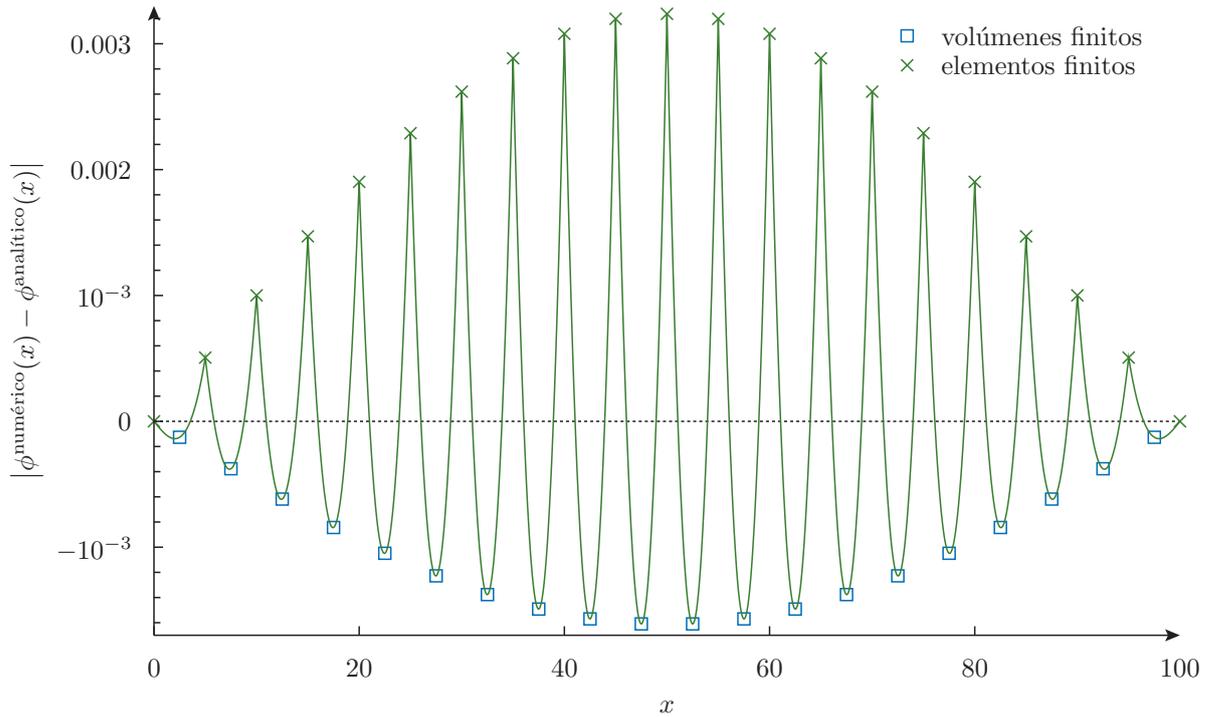


Figura 5.10 – Diferencia entre la solución continua obtenida por el método de elementos finitos y la solución analítica para $I = 20$. Se muestra también el error cometido por volúmenes finitos en los centros de las celdas.

$$\begin{aligned}
 0 &= \int_{x_i}^{x_{i+1}} \frac{\partial^2 \phi}{\partial x^2} dx \\
 0 &= \left[\frac{\partial \phi}{\partial x} \right]_{x_i}^{x_{i+1}} \\
 0 &= \frac{\phi_{i+1}^{\square} - \phi_i^{\square}}{\Delta x} - \frac{\phi_i^{\square} - \phi_{i-1}^{\square}}{\Delta x}
 \end{aligned}$$

donde hemos definido ϕ_i^{\square} como el valor medio del flujo en la celda i . Para $i = 1$, debemos incluir la condición de contorno en el término de fugas:

$$\begin{aligned}
 0 &= \frac{0 - \phi_1^{\square}}{\frac{\Delta x}{2}} - \frac{\phi_2^{\square} - \phi_1^{\square}}{\Delta x} \\
 0 &= -3\phi_1^{\square} + \phi_2^{\square}
 \end{aligned}$$

Análogamente para $i = I$

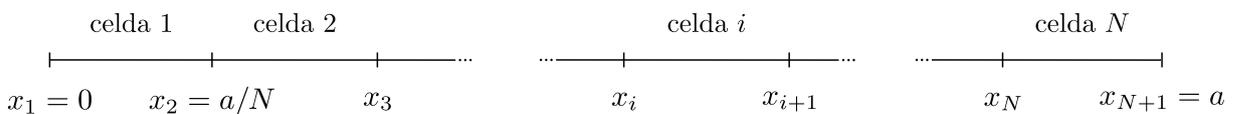


Figura 5.11 – División del dominio $x = [0, a]$ en I celdas iguales.



$$\begin{aligned}
 \hat{\varphi}^T \left(\int_{-1}^1 B_x^T \cdot B_x \frac{\partial x}{\partial r} dr \right) \hat{\phi} &= \hat{\varphi}^T \left(\int_{-1}^1 \frac{\partial r}{\partial x} \cdot \begin{bmatrix} \partial h_1 / \partial r \\ \partial h_2 / \partial r \end{bmatrix} \begin{bmatrix} \partial h_1 / \partial r & \partial h_2 / \partial r \end{bmatrix} \frac{\partial r}{\partial x} \cdot \frac{\partial x}{\partial r} dr \right) \hat{\phi} \\
 &= \hat{\varphi}^T \left(\int_{-1}^1 \frac{2}{\Delta x} \cdot \begin{bmatrix} -1/2 \\ 1/2 \end{bmatrix} \begin{bmatrix} -1/2 & 1/2 \end{bmatrix} \frac{2}{\Delta x} \cdot \frac{\Delta x}{2} dr \right) \hat{\phi} \\
 &= \hat{\varphi}^T \left(\frac{D}{\Delta x} \cdot \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \right) \hat{\phi}
 \end{aligned}$$

Como las condiciones de contorno son de Dirichlet, entonces no tenemos que prestar atención al término de superficie de la formulación débil. Fijando $K = 6$, ensamblando las contribuciones elementales y poniendo las condiciones de contorno $\phi_1^\times = \phi_K^\times = 0$ obtenemos

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_1^\times \\ \phi_2^\times \\ \phi_3^\times \\ \phi_4^\times \\ \phi_5^\times \\ \phi_6^\times \end{bmatrix} = 0$$

Supongamos que vale la suposición (5.2)

$$\phi_i^\square = \frac{\phi_i^\times + \phi_{i+1}^\times}{2} \quad (5.2)$$

y la reemplazamos en la expresión para la celda i en el esquema de volúmenes finitos

$$\begin{aligned}
 0 &= \phi_{i-1}^\square - 2\phi_i^\square + \phi_{i+1}^\square \\
 &= \frac{\phi_{i-1}^\times + \phi_i^\times}{2} - (\phi_i^\times + \phi_{i+1}^\times) + \frac{\phi_{i+1}^\times + \phi_{i+2}^\times}{2} \\
 &= (\phi_{i-1}^\times - 2\phi_i^\times + \phi_{i+1}^\times) + (\phi_i^\times - 2\phi_{i+1}^\times + \phi_{i+2}^\times)
 \end{aligned}$$

con lo que obtenemos la suma de las expresiones para los nodos $i > 1$ e $i + 1$ de elementos finitos. En el nodo $i = 1$

$$\begin{aligned}
 0 &= -3\phi_1^\square + \phi_2^\square \\
 &= -3 \cdot \frac{\phi_1^\times + \phi_2^\times}{2} + \frac{\phi_2^\times + \phi_3^\times}{2}
 \end{aligned}$$

Como $\phi_1^\times = 0$ por condición de contorno, resulta

$$0 = (-2\phi_2^\times + \phi_3^\times)$$

que es la ecuación del método de elementos finitos para el nodo $i = 2$. Para $i = K$ obtendríamos una ecuación similar, con lo que completamos la pseudo-demostración de que la ecuación (5.2) vale para un slab unidimensional con propiedades homogéneas.

Poniendo $\Delta x = a/I$, podemos escribir la ecuación de difusión a un grupo usando el esquema de volúmenes finitos del capítulo 3 como



$$\begin{aligned}
D \frac{I^2}{a^2} [3\phi_1^\square - \phi_2^\square] &= \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\square} - \Sigma_a \right) \cdot \phi_1^\square \\
D \frac{I^2}{a^2} [2\phi_i^\square - \phi_{i-1}^\square - \phi_{i+1}^\square] &= \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\square} - \Sigma_a \right) \cdot \phi_i^\square \quad \text{para } i = 2, \dots, I-1 \\
D \frac{I^2}{a^2} [3\phi_n^\square - \phi_{n-1}^\square] &= \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\square} - \Sigma_a \right) \cdot \phi_n^\square
\end{aligned}$$

Análogamente, las $K = I + 1$ ecuaciones correspondientes al método basado en elementos finitos discutido en el capítulo 4 son

$$\begin{aligned}
\phi_1^\times &= 0 \\
D \frac{I^2}{a^2} [2\phi_i^\times - \phi_{i-1}^\times - \phi_{i+1}^\times] &= \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\times} - \Sigma_a \right) \left[\frac{2}{3}\phi_i^\times + \frac{1}{6}\phi_{i-1}^\times + \frac{1}{6}\phi_{i+1}^\times \right] \quad \text{para } i = 2, \dots, K-1 \\
\phi_K^\times &= 0
\end{aligned}$$

Por un lado, sumamos las I ecuaciones de volúmenes finitos para obtener

$$D \frac{I^2}{a^2} [2\phi_1^\square + 2\phi_I^\square] = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\square} - \Sigma_a \right) \cdot \sum_{i=1}^I \phi_i^\square$$

que por la simetría del problema podemos simplificar a

$$D \frac{I^2}{a^2} \cdot 4\phi_1^\square = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\square} - \Sigma_a \right) \cdot \sum_{i=1}^I \phi_i^\square \quad (5.4)$$

Por otro lado, sumando las K ecuaciones de elementos finitos tenemos

$$D \frac{I^2}{a^2} [\phi_1^\times + \phi_2^\times + \phi_I^\times + \phi_K^\times] = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\times} - \Sigma_a \right) \cdot \left[\frac{1}{6}\phi_1^\times + \frac{5}{6}\phi_2^\times + \sum_{i=3}^{I-1} \phi_i^\times + \frac{5}{6}\phi_I^\times + \frac{1}{6}\phi_K^\times \right]$$

que con las condiciones de contorno de flujo nulo y la simetría re-escribimos como

$$D \frac{I^2}{a^2} \cdot 2\phi_2^\times = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^\times} - \Sigma_a \right) \cdot \left[\sum_{i=2}^I \phi_i^\times - \frac{1}{3}\phi_2^\times \right] \quad (5.5)$$

Como el autovector solución del problema está determinado a menos de una constante multiplicativa, para poder comparar dos soluciones al problema continua utilizando diferente cantidad de incógnitas debemos tener una condición de normalización. En particular, por defecto milonga normaliza los resultados de forma tal que el valor medio del flujo sea igual a la unidad, es decir



$$\frac{\int_0^a \phi(x) dx}{\int_0^a dx} = 1$$

que es la condición que arroja como resultado el flujo normalizado $\phi_{\text{real}}(x)$ dado por la ecuación (5.1). Para volúmenes finitos, podemos escribir la integral del numerador como

$$\int_0^a \phi(x) dx = \sum_{i=1}^I \left(\phi_i^{\square} \cdot \frac{a}{I} \right) = a$$

por lo que

$$\sum_{i=1}^I \phi_i^{\square} = I$$

Para elementos, podemos aplicar el método de los trapecios para calcular exactamente la integral

$$\int_0^a \phi(x) dx = \frac{1}{2} \phi_1^{\times} \cdot \frac{a}{I} + \sum_{i=2}^I \left(\phi_i^{\times} \cdot \frac{a}{I} \right) + \frac{1}{2} \phi_K^{\times} \cdot \frac{a}{I} = a$$

Como $\phi_1^{\times} = \phi_K^{\times} = 0$ entonces

$$\sum_{i=2}^I \phi_i^{\times} = I$$

y las ecuaciones (5.4) y (5.5) resultan

$$D \frac{I^2}{a^2} \cdot 4\phi_1^{\square} = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^{\square}} - \Sigma_a \right) \cdot I$$

$$D \frac{I^2}{a^2} \cdot 2\phi_2^{\times} = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^{\times}} - \Sigma_a \right) \cdot \left[I - \frac{1}{3} \phi_2^{\times} \right]$$

que podemos restar miembro a miembro

$$D \frac{I^2}{a^2} \cdot \left[4\phi_1^{\square} - 2\phi_2^{\times} \right] = \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^{\square}} - \Sigma_a \right) \cdot I - \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^{\times}} - \Sigma_a \right) \cdot \left[I - \frac{1}{3} \phi_2^{\times} \right]$$

El miembro izquierdo se anula en virtud de la ecuación (5.2). Luego la relación

$$I \cdot \nu \Sigma_f \left(\frac{1}{k_{\text{eff}}^{\times}} - \frac{1}{k_{\text{eff}}^{\square}} \right) = \frac{1}{3} \left(\frac{\nu \Sigma_f}{k_{\text{eff}}^{\times}} - \Sigma_a \right) \cdot \phi_2^{\times}$$

involucra a los factores de multiplicación k_{eff}^{\square} y k_{eff}^{\times} obtenidos en las soluciones numéricas realizadas con volúmenes y elementos finitos respectivamente. Esta relación es exacta, con la salvedad de que involucra a la solución del flujo calculada con elementos finitos ϕ_2^{\times} en el segundo nodo. Es de esperar que este valor sea parecido a la solución analítica $\phi(x)$ evaluada en $x = a/n \approx 0$

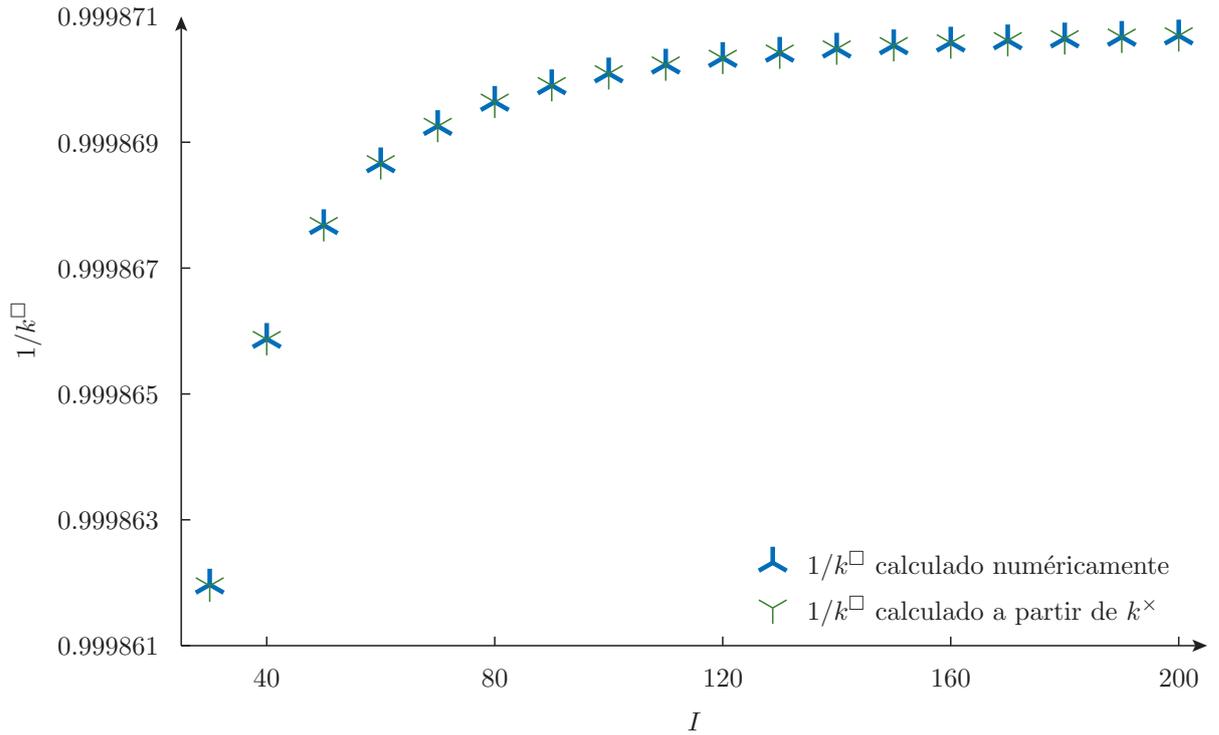


Figura 5.13 – Comparación entre la inversa del factor de multiplicación k_{eff}^{\square} calculada a partir de la solución por volúmenes finitos y la calculada a partir de la solución por elementos finitos con la ecuación (5.7).

$$\phi(x) = \frac{\pi}{2} \sin\left(\pi \cdot \frac{x}{a}\right) \approx \frac{\pi}{2} \cdot \frac{\pi}{a} \cdot x \quad (5.6)$$

entonces

$$\phi_2^{\times} \approx \phi(a/n) \approx \frac{\pi^2}{2I}$$

y podemos simplificar la relación entre los factores de multiplicación numéricos a

$$\frac{1}{k_{\text{eff}}^{\square}} \approx \frac{1}{k_{\text{eff}}^{\times}} + \left[\frac{1}{k_{\infty}} - \frac{1}{k_{\text{eff}}^{\times}} \right] \cdot \frac{\pi^2}{6I^2} \quad (5.7)$$

donde introducimos el factor de multiplicación infinito $k_{\infty} = \nu \Sigma_f / \Sigma_a$. Como el factor de multiplicación de un reactor finito es siempre menor que k_{∞} , entonces la ecuación (5.7) muestra que $k_{\text{eff}}^{\square} > k_{\text{eff}}^{\times}$ para cualquier número de celdas I , que es el comportamiento que observamos en la figura 5.4. Además, la ecuación (5.7) indica que la diferencia entre ambos autovalores disminuye con el cuadrado de I .

En efecto, la figura 5.13 muestra $1/k_{\text{eff}}^{\square}$ calculada por un lado invirtiendo el resultado obtenido al resolver el problema con volúmenes finitos y, por el otro, usando el valor de k_{eff}^{\times} calculado por elementos finitos en la ecuación (5.7). Vemos que si bien los puntos no son exactamente iguales—ya que usamos la aproximación de la ecuación (5.6)—su coincidencia es notable, considerando la forma simple de la ecuación (5.7) que relaciona analíticamente dos resultados numéricos.

Un slab unidimensional a un grupo de energías con propiedades uniformes y condiciones de flujo nulo en los extremos maldado con celdas/elementos uniformemente distribuidos no presenta un gran desafío para ser atacado con métodos numéricos que requieren alguna clase



de complejidad. Sin embargo, el análisis de este caso nos dejó algunos resultados que si bien no pueden ser directamente aplicados en condiciones más generales—donde la aplicación de métodos numéricos es más beneficiosa y de hecho más divertida—es de esperar que puedan ser extrapolados de forma aproximada a dichos casos. Es decir, en dos dimensiones tal vez la convergencia no disminuya exactamente en forma cuadrática con el tamaño de la malla, pero probablemente su comportamiento sea cercano a una ley tipo $1/I^2$. Si las propiedades no son homogéneas sino que dependen de la posición x probablemente el valor medio en una celda no sea igual al promedio aritmético de los valores nodales, pero dichos valores serán similares. Y, en general, los métodos basados en volúmenes sobreestiman el valor real de k_{eff} y los métodos basados en elementos lo subestiman.

5.1.1.6. Medio slab con condición de contorno espejo

Una de las particularidades de los cálculos de núcleo es que en los casos en los cuales existe algún grado de simetría, es posible reducir la cantidad de incógnitas—y por lo tanto el esfuerzo computacional—resolviendo solamente una parte del reactor y poniendo condiciones de contorno tipo espejo en las superficies adecuadas. De hecho era esta la única manera de resolver algunos núcleos en épocas donde la capacidad de almacenamiento se medía en kilobytes. Incluso tal vez algunos reactores que hoy están operando se hayan diseñado utilizando cierta simetría sólo debido a las limitaciones de las herramientas de diseño y cálculo disponibles a la hora de hacer ingeniería de centrales nucleares. Sea cual fuere el caso, en esta sección resolvemos el mismo slab desnudo uniforme pero aprovechando la simetría, generamos una geometría de ancho $a = 100/2$ y ponemos condición de flujo nulo $\phi(x) = 0$ en $x = 0$ y de simetría $\partial\phi/\partial x = 0$ en $x = a$.

El input que sigue calcula el factor de multiplicación k_{eff} del slab de ancho $a = 100$ estudiado en las secciones anteriores en función de la cantidad de celdas/elementos pero resolviendo en realidad un problema que tiene la mitad de incógnitas. El método de resolución se selecciona desde la línea de comando mediante las opciones `--volumes` o `--elements`.

mirror-keff-vs-mesh.was

```
# calcula el keff vs. la cantidad de incongnitas
# para un slab de semiancho a/2 con CC espejo

PARAMETRIC n 10 200 10

# el archivo de geometria llama "left" al punto en x=0
# y "right" al punto en x=a, asi que usamos estos nombres
# para definir que tipo de condiciones de contorno queremos
# por default, milonga asume condiciones de contorno dirichlet
# en los extremos, asi que la definicion sobre "left" no
# es estrictamente necesaria, pero la dejamos por completitud
PHYSICAL_ENTITY NAME left BC dirichlet
PHYSICAL_ENTITY NAME right BC neumann

a = 100/2 # ahora hacemos que a sea la mitad que antes
lc = a/n
TEMPLATE slab.tpl slab.geo a lc
SHELL "gmsH_v_0_2_slab.geo"
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101
MILONGA SOLVE

# el buckling hay que calcularlo con 2a en lugar de a
```

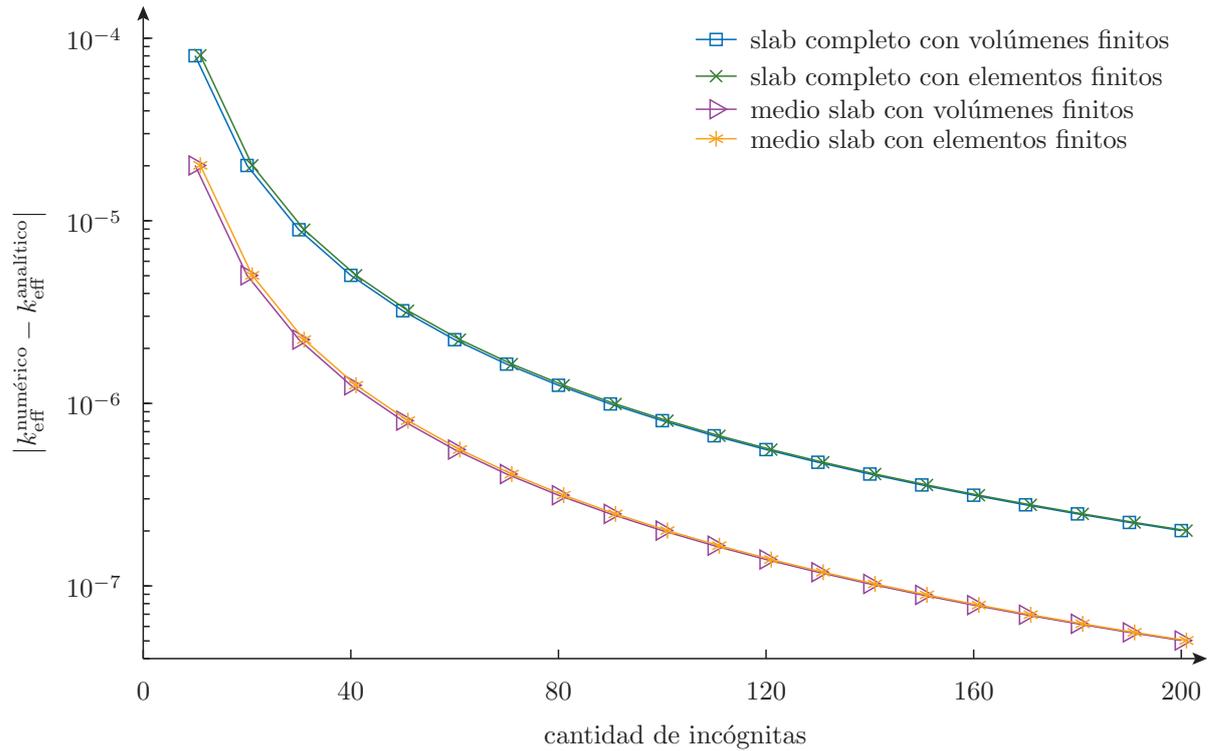


Figura 5.14

```
kreal = nuSigmaF(a/2)/(SigmaA(a/2) + D(a/2)*(pi/(2*a))^2)
PRINT %.f n nodes cells %.12f keff kreal %e abs(keff-kreal)
```

```
$ milonga mirror-keff-vs-mesh.was --volumes > mirror-keff-vs-mesh-volumes.dat
$ milonga mirror-keff-vs-mesh.was --elements > mirror-keff-vs-mesh-elements.dat
$ pyxplot mirror-error-vs-mesh.ppl
$
```

En la figura 5.14 comparamos los errores cometidos tanto con volúmenes como con elementos finitos con respecto a la solución analítica, tanto entre sí como con respecto a los cometidos al calcular el slab completo de ancho $a = 100$ como hicimos en la sección 5.1.1.3 (figura 5.5). Vemos allí que para el mismo tamaño de problema, con el caso simétrico obtenemos menores errores. O, en forma equivalente, para lograr el mismo error se necesitan menos incógnitas. En este caso es aproximadamente la mitad, mientras que en dos y tres dimensiones esta reducción puede ser aún más significativa.

5.1.1.7. Condiciones de contorno de Robin

En la sección anterior resolvimos un slab con condiciones de Dirichlet en $x = 0$ y con condiciones de Neumann en $x = a$. En esta sección introducimos condiciones de contorno de Robin y comparamos los resultados con el caso de condiciones de contorno de flujo nulo. Para hacer más evidentes las diferencias, resolvemos slabs más delgados que en el caso anterior. Esta vez tomamos $a = 10$, fijamos $n = 50$ y le pedimos a milonga que escriba en la salida estándar la distribución de flujo en el intervalo $[0, a]$ con incrementos $\Delta x = a/400$.



El slab con condiciones de contorno nulas lo resolvemos con este input:

thin-null.was

```
# explicitamente ponemos condiciones de contorno
# nulas en ambos extremos
PHYSICAL_ENTITY NAME left   BC dirichlet
PHYSICAL_ENTITY NAME right BC dirichlet

a = 10
n = 50
lc = a/n

TEMPLATE slab.tpl slab.geo   a   lc
SHELL "gmsv_0_2_slab.geo"
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101
MILONGA SOLVE

PRINT_FUNCTION phi_1 MIN 0 MAX a STEP a/400
```

y el slab con condiciones de contorno de Robin con el que sigue, cuya única diferencia radica en las dos líneas que definen justamente las condiciones de contorno sobre las entidades físicas “left” y “right”. En $x = 0$ la condición es

$$\frac{\partial \phi}{\partial x} \cdot (-1) = -\frac{1}{2D_1} \cdot \phi(0)$$

y en $x = a$

$$\frac{\partial \phi}{\partial x} \cdot (+1) = -\frac{1}{3D_1} \cdot \phi(a)$$

donde (-1) y $(+1)$ indican los vectores unitarios en la dirección normal externa a la superficie que contiene la condición de contorno en una dimensión. Debemos recordar que siendo el problema matemáticamente homogéneo, las condiciones de contorno también deben serlo, como discutimos en la sección 2.2.2. El código milonga lee a continuación de la palabra clave robin el coeficiente que multiplica a $\phi(x)/D_g$ en la expresión de la derivada normal del flujo.

thin-robin.was

```
# ponemos condiciones de contorno de robin en ambos
# extremos con coeficientes diferentes coeficientes
PHYSICAL_ENTITY NAME left   BC robin -0.5
PHYSICAL_ENTITY NAME right BC robin -1/3

a = 10
n = 50
lc = a/n

TEMPLATE slab.tpl slab.geo   a   lc
SHELL "gmsv_0_2_slab.geo"
MESH FILE_PATH slab.msh DIMENSIONS 1 DEGREES 1
MATERIAL fuel D 1 SigmaA 0.1 nuSigmaF 0.101
MILONGA SOLVE

PRINT_FUNCTION phi_1 MIN 0 MAX a STEP a/400
```

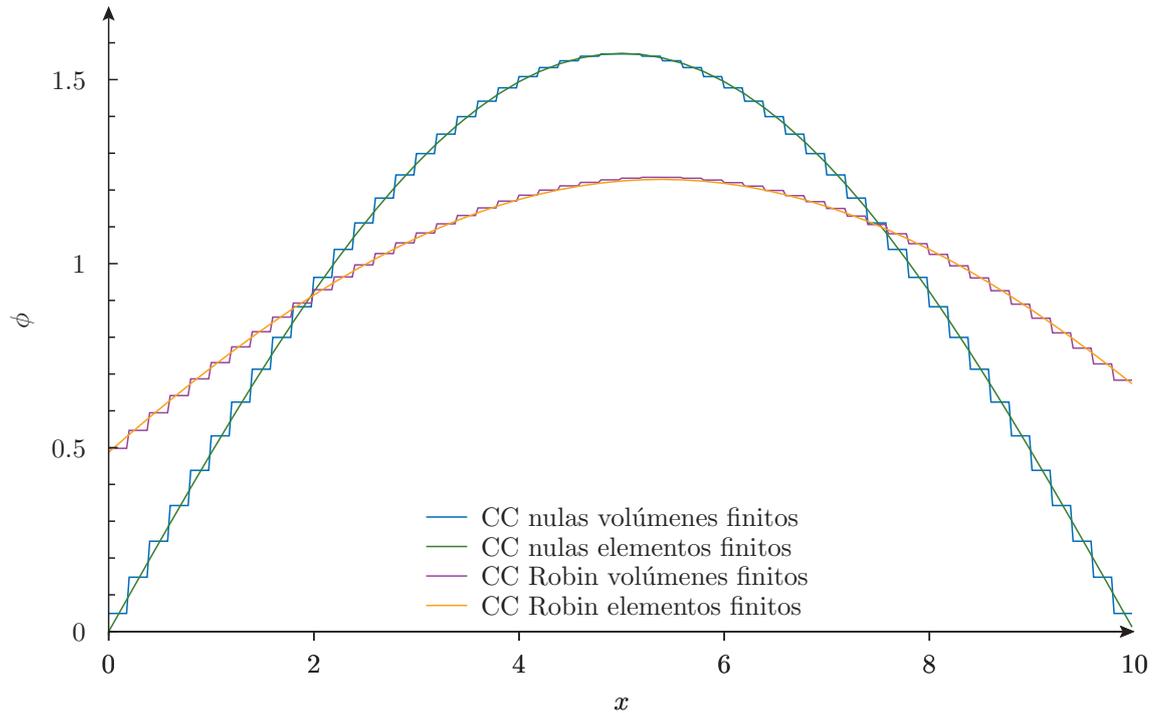


Figura 5.15 – Distribución de flujo en un slab de ancho $a = 10$ calculada con volúmenes y elementos finitos para un problema con (a) condiciones de contorno de flujo nulo en ambas superficies externas (b) condiciones de contorno de tipo Robin con coeficientes diferentes en cada superficie externa

```
$ milonga thin-null.was --volumes > thin-null-volumes.dat  
$ milonga thin-null.was --elements > thin-null-elements.dat  
$ milonga thin-robin.was --volumes > thin-robin-volumes.dat  
$ milonga thin-robin.was --elements > thin-robin-elements.dat  
$ pyxplot null-robin.ppl  
$
```

En la figura 5.15 vemos la comparación entre la solución conocida para el problema con condición de flujo cero en los extremos y con condiciones de Robin en ambos extremos. Como este tipo de condiciones se puede asimilar a una longitud de extrapolación en la cual el flujo se anula la diferencia es apreciable ya que estas distancias—que en este caso sería $2D_1 = 2$ en la superficie izquierda y $3D_1 = 3$ en la superficie externa, medidas en las mismas unidades que el ancho a —son comparables al ancho $a = 10$ del slab. Además, podemos apreciar que en este caso ya no es cierto que el valor medio de una celda en volúmenes es el promedio aritmético de los valores nodales en elementos.

5.1.2. Slab a dos zonas

El problema que resolvemos en esta sección es un slab unidimensional pero compuesto de dos materiales diferentes, cada uno de ellos con propiedades uniformes. Esta configuración es sólo ligeramente diferente a los slabs desnudos estudiados en la sección anterior. Sin embargo, aunque la distribución de flujo tiene solución analítica, su expresión es significativamente más compleja que el simple seno del slab desnudo. Incluso la condición crítica para calcular el factor de multiplicación k_{eff} debe ser resuelta numéricamente. De cualquier manera, dicha solución analítica permite tomar a este problema como benchmark para métodos nodales ya que per-

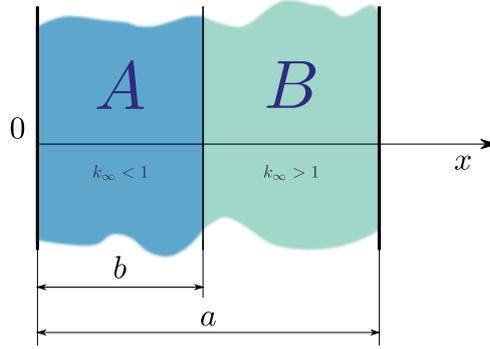


Figura 5.16 – Slab unidimensional compuesto por dos zonas homogéneas, una de material absorbente ($k_{\infty} < 1$) y otra por material físil ($k_{\infty} > 1$).

mite evaluar qué tan bien se conserva la corriente en interfaces materiales en cada esquema numérico. En particular, la comparación de los resultados obtenidos numéricamente con la solución analítica arroja cierta luz sobre lo que sucede en casos más generales de interés en la industria nuclear tales como el efecto de dilución de secciones eficaces cuando las interfaces no coinciden con la malla. Volvemos a este punto en la sección 5.1.2.3.

Mostramos la geometría del problema en la figura 5.16. Consta de un slab unidimensional de ancho total a compuesto por dos materiales, cada uno con propiedades homogéneas a un grupo de energía. El material A se extiende desde $x = 0$ hasta $x = b$ y suponemos que sus propiedades nucleares son tales que $k_{\infty} = \nu\Sigma_{fA}/\Sigma_{aA} < 1$, por lo que se comporta esencialmente como un absorbente. El material B ocupa el intervalo $[b, a]$ y tiene un $k_{\infty} = \nu\Sigma_{fB}/\Sigma_{aB} > 1$. Con condiciones de contorno de flujo nulo en los extremos, la distribución de flujo es

$$\phi(x) = \begin{cases} f \cdot \sinh(B_A \cdot x) & \text{para } 0 < x < b \\ f \cdot \frac{\sinh(B_A \cdot b)}{\sin[B_B \cdot (a - b)]} \cdot \sin[B_B \cdot (a - x)] & \text{para } b < x < a \end{cases} \quad (5.8)$$

donde las raíces cuadradas de los bucklings B_A^2 y B_B^2 de cada material son

$$B_A = \sqrt{\frac{\Sigma_{aA} - \frac{\nu\Sigma_{fA}}{k_{\text{eff}}}}{D_A}} \quad B_B = \sqrt{\frac{\frac{\nu\Sigma_{fB}}{k_{\text{eff}}} - \Sigma_{aB}}{D_B}}$$

El factor de multiplicación k_{eff} debe cumplir la condición crítica

$$\sqrt{D_A \cdot \left(\Sigma_{aA} - \frac{\nu\Sigma_{fA}}{k_{\text{eff}}} \right)} \cdot \tan \left[\sqrt{\frac{1}{D_B} \cdot \left(\frac{\nu\Sigma_{fB}}{k_{\text{eff}}} - \Sigma_{aB} \right)} \cdot (a - b) \right] + \sqrt{D_B \cdot \left(\frac{\nu\Sigma_{fB}}{k_{\text{eff}}} - \Sigma_{aB} \right)} \cdot \tanh \left[\sqrt{\frac{1}{D_A} \cdot \left(\Sigma_{aA} - \frac{\nu\Sigma_{fA}}{k_{\text{eff}}} \right)} \cdot b \right] = 0 \quad (5.9)$$

que no sólo es una expresión trascendente sino que tiene pasajes por infinito y cuya solución para k_{eff} implica cierto cuidado. Finalmente, el coeficiente f de la ecuación (5.8) debe ser tal que se satisfaga la condición de normalización, que tomamos como



Figura 5.17 – Representación gráfica en gmsch del archivo de geometría two-zone-slab.geo

$$\frac{\int_0^a \phi(x) dx}{\int_0^a dx} = 1$$

indicando que el valor medio del flujo debe ser igual a la unidad.

Estas expresiones completan la solución analítica del problema. Debemos notar que si los coeficientes de difusión de ambos materiales son diferentes, entonces el flujo $\phi(x)$ tiene derivada discontinua en $x = b$ de forma tal que la corriente de neutrones se mantenga. De hecho esta es una condición que debemos satisfacer al deducir la distribución de flujo dada por la ecuación (5.8).

La solución numérica implica la generación de un archivo de geometría en función de los parámetros a y b , además de la longitud característica ℓ_c asociada a los puntos. Por ejemplo, el archivo de geometría two-zone-slab.geo define la geometría mostrada por gmsch en la figura 5.17 con $a = 100$ y $b = 40$ unidades de longitud.

two-zone-slab.geo

```
a = 100;
b = 40;
lc = 1;

Point(1) = {0, 0, 0, lc};
Point(2) = {b, 0, 0, lc};
Point(3) = {a, 0, 0, lc};

Line(1) = {1, 2};
Line(2) = {2, 3};
Physical Line("A") = {1};
Physical Line("B") = {2};

Physical Point("left") = {1};
Physical Point("right") = {3};
```

En este archivo asignamos como longitud característica de elementos $\ell_c = 1$ a cada uno de los tres puntos. Además, definimos dos entidades físicas de dimensión uno llamadas A



y B que corresponden a los materiales y dos entidades físicas de dimensión cero llamadas *left* y *right* que corresponden a los puntos donde se deben aplicar las condiciones de contorno. Al momento de resolver los problemas de esta sección, en verdad utilizamos un archivo de template en el que reemplazamos las tres primeras líneas

```
a = 100;
b = 40;
lc = 1;
```

por

```
a = $1;
b = $2;
lc = $3;
```

de forma tal de poder seleccionar el ancho total a , la ubicación de la interfaz b y la longitud característica ℓ_c de los elementos desde el input de milonga.

Para evitar tener que escribir las secciones eficaces en cada uno de los inputs que usamos para resolver cada uno de los problemas propuestos relacionados al slab de dos zonas de la figura 5.16 y a la vez asegurarnos de que todos ellos utilizan las mismas propiedades para los materiales A y B , creamos un archivo llamado `materiales.was` que luego incluimos desde el input principal de milonga:

`materiales.was`

```
MATERIAL A   D 0.5   SigmaA 0.014   nuSigmaF 0.010
MATERIAL B   D 1.5   SigmaA 0.010   nuSigmaF 0.014
```

Como los nombres de los materiales coinciden con los nombres de las entidades físicas del archivo de geometría, el código las vincula automáticamente y aplica las propiedades de los materiales A y B a las celdas/elementos que tengan asignada la entidad física correspondiente. Estas propiedades están en unidades consistentes con las unidades de longitud que definen la geometría. Es decir, si a está en centímetros, entonces Σ_a debe estar expresada en cm^{-1} . Debemos notar que las secciones eficaces deben ser tales que $k_{\infty A} = \nu \Sigma_{fA} / \Sigma_{aA} < 1$ y que $k_{\infty B} = \nu \Sigma_{fB} / \Sigma_{aB} > 1$ para que la solución analítica que discutimos arriba sea aplicable. Además, elegimos los coeficientes de difusión extremadamente diferentes para resaltar la importancia de tener en cuenta las interfaces materiales, que es el objetivo del benchmark propuesto.

5.1.2.1. Factores de multiplicación y flujos

Lo primero que hacemos es resolver tanto analíticamente como numéricamente un slab de ancho total $a = 100$ compuesto por una mitad ($b = 50$) construida en material A principalmente absorbente y por otra mitad de material B mayormente físil. La selección del esquema numérico y la cantidad n de celdas/elementos la realizamos desde la línea de comandos. Generamos la malla apropiada primero reemplazando los valores de a , b y ℓ_c en el archivo de template `two-zone-slab.tpl` para generar el archivo de geometría y, a partir de este, el archivo de malla. Una vez que resolvemos el problema numéricamente, utilizamos el valor del k_{eff} que acabamos de obtener para definir un intervalo $[k_{\text{eff}} - \epsilon, k_{\text{eff}} + \epsilon]$ sobre el cual buscamos el valor analítico del factor de multiplicación k que satisface la condición crítica dada por la ecuación (5.9)



mediante algún algoritmo de bracketing, y que tomamos como la solución analítica del problema continuo. Este paso es importante, ya que la condición crítica tiene pasajes por infinito que hacen complicado encontrar el factor de multiplicación si no disponemos de una buena estimación del intervalo donde se encuentra. Con dicho k calculamos los bucklings B_A y B_B para con ellos definir dos funciones, flujo $_A(x)$ y flujo $_B(x)$ que luego integramos para calcular el factor f de forma tal de normalizar el flujo analítico flujo (x) —que es igual a flujo $_A(x)$ si $x < b$ y a flujo $_B(x)$ de otra manera—de la misma forma en la que milonga normaliza el flujo numérico $\phi_1(x)$ y poder entonces compararlos. A continuación mostramos en la pantalla el esquema utilizado, los valores de a , b , n y ℓ_c seleccionados, junto con el k_{eff} analítico (que no depende de n), el k_{eff} numérico y la diferencia entre ambos. Finalmente escribimos en el archivo analitico.dat el flujo $\phi(x)$ analítico (que tampoco depende de n) en el intervalo $[0, a]$ con incrementos $\Delta x = a/1000$, y en un archivo que indica el esquema y el valor de n seleccionado el flujo numérico en los puntos incógnita de acuerdo al método seleccionado.

fluxes.was

```
# calcula el factor de multiplicacion y la distribucion de flujo
# en un slab de dos zonas tanto numerica como analiticamente

SCHEME $1

b = 50          # ancho de la zona A-comienzo de la zona B
a = 100         # ancho total del slab
n = $2         # cantidad de celdas/elementos
lc = a/n       # longitud caracteristica del elemento

# generamos la geometria con la longitud caracteristica
TEMPLATE two-zone-slab.tpl two-zone-slab.geo  a b lc
# mallamos esa geometria
SHELL "gmsht_v_0_1_two-zone-slab.geo"
# y definimos el problema unidimensional a un grupo
MESH DIMENSIONS 1 DEGREES 1 FILE_PATH two-zone-slab.msh

# tenemos las secciones eficaces en un archivo separado
# que incluimos cada vez que necesitamos resolver un problema
INCLUDE materiales.was

# resolvemos el problema de autovalores numerico
MILONGA SOLVE

# calculamos el k analitico a partir de resolver numericamente
# la condicion critica del problema continuo usando como guess
# inicial el keff calculado numericamente
eps = 0.05
k = { root(
  sqrt(D(0) * (SigmaA(0) - nuSigmaF(0)/k)) *
    tan(sqrt((1/D(a)) * (nuSigmaF(a)/k - SigmaA(a))) * (a-b))
  + sqrt(D(a) * (nuSigmaF(a)/k - SigmaA(a))) *
    tanh(sqrt((1/D(0)) * (SigmaA(0) - nuSigmaF(0)/k)) * b)
  , k, keff-eps, keff+eps) }

# y con este k calculamos la distribucion de flujo analitica
# primero definimos los flujos en cada zona
B_A = sqrt((SigmaA(0) - nuSigmaF(0)/k)/D(0))
flujoA(x) := sinh(B_A*x)

B_B = sqrt((nuSigmaF(a)/k - SigmaA(a))/D(a))
```

```

flujoB(x) := sinh(B_A*b)/sin(B_B*(a-b)) * sin(B_B*(a-x))

# calculamos el factor de normalizacion para que
# el valor medio del flujo sea igual a uno
f = a/(integral(flujoA(x), x, 0, b) + integral(flujoB(x), x, b, a))

# y definimos una funcion por trozos valida para todo el intervalo
flujo(x) := f * if(x < b, flujoA(x), flujoB(x))

# en la pantalla mostramos los keff
PRINT "usando_finite_$1_con"
PRINT "% TEXT "a_=" a TEXT "b_=" b TEXT "n_=" n TEXT "lc_=" lc
PRINT "keff_analitico_=" %.8f k
PRINT "keff_numerico_=" %.8f keff
PRINT "diferencia_=" %e keff-k

# el flujo analitico
OUTPUT_FILE analitico analitico.dat
PRINT_FUNCTION flujo MIN 0 MAX a STEP a/1000 FILE analitico

# el flujo numerico (en los puntos calculados) y el error
OUTPUT_FILE numerico $1-$2.dat
PRINT_FUNCTION phi_1 phi_1(x)-flujo(x) FILE numerico

```

```

$ milonga fluxes.was volumes 20
usando finite volumes con
a = 100 b = 50 n = 20 lc = 5
keff analitico = 1.08349094
keff numerico = 1.08707215
diferencia = 3.581214e-03
$ milonga fluxes.was elements 20
usando finite elements con
a = 100 b = 50 n = 20 lc = 5
keff analitico = 1.08349094
keff numerico = 1.08176970
diferencia = -1.721240e-03
$ milonga fluxes.was volumes 50
usando finite volumes con
a = 100 b = 50 n = 50 lc = 2
keff analitico = 1.08349094
keff numerico = 1.08407471
diferencia = 5.837725e-04
$ milonga fluxes.was elements 50
usando finite elements con
a = 100 b = 50 n = 50 lc = 2
keff analitico = 1.08349094
keff numerico = 1.08321499
diferencia = -2.759441e-04
$ milonga fluxes.was volumes 100
usando finite volumes con
a = 100 b = 50 n = 100 lc = 1
keff analitico = 1.08349094
keff numerico = 1.08363727
diferencia = 1.463377e-04
$ milonga fluxes.was elements 100
usando finite elements con
a = 100 b = 50 n = 100 lc = 1
keff analitico = 1.08349094
keff numerico = 1.08342193

```



```
diferencia = -6.900593e-05
$ pyxplot flux-20.ppl
$ pyxplot error-50.ppl
$ pyxplot error-100.ppl
$
```

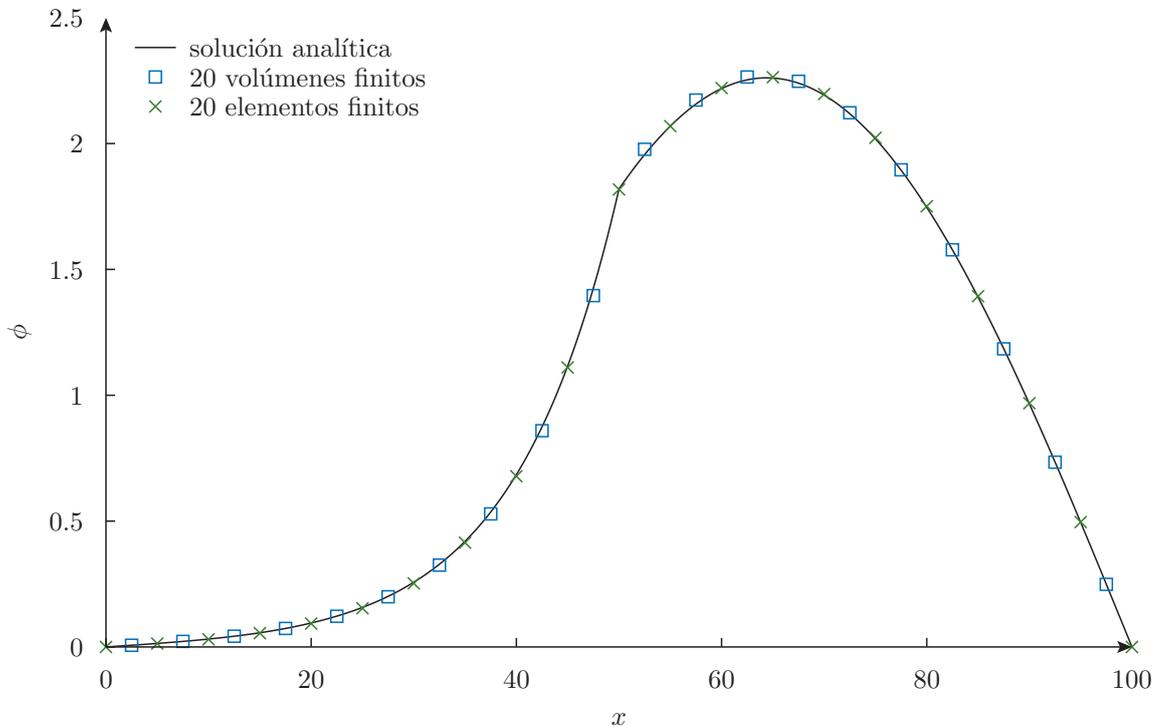


Figura 5.18 – Distribución de flujo de un slab a dos zonas calculada analíticamente, con volúmenes finitos y con elementos finitos.

En la salida de la terminal observamos que sigue siendo válido el resultado de la sección anterior en la que el k_{eff} calculado con volúmenes es mayor que el calculado con elementos. Además, en los tres casos mostrados, el error cometido por volúmenes finitos es, en valor absoluto, es el doble del error cometido por elementos finitos. De cualquier manera, en la figura 5.18 muestra que aún para una nodalización bastante gruesa como $n = 20$, ambos métodos son capaces de reproducir bastante bien la solución analítica aún cuando la derivada del flujo continuo presente una marcada discontinuidad en $x = b$. De todas maneras, parecería incluso que la solución de elementos es mejor que la de volúmenes, especialmente alrededor de la interfaz. En efecto, las figuras 5.19 y 5.20 muestran la diferencia entre las soluciones numéricas y la solución analítica evaluadas en los centros de las celdas para el caso de volúmenes y en los nodos para elementos, para $n = 50$ y para $n = 100$. Vemos que si bien la solución en volúmenes presenta un pequeño error cerca de los extremos, la discrepancia crece bastante cerca de la interfaz y los errores cambian de signo en forma abrupta a un lado y a otro de $x = b$. Por el contrario, si bien el error tiene el mismo signo que la solución de volúmenes en cada material, este error pasa por cero en el punto de la discontinuidad. Ambas figuras tienen la misma escala vertical, por lo que podemos observar cómo se reduce la discrepancia entre el flujo analítico y los resultados numéricos al duplicar la cantidad de celdas/elementos.

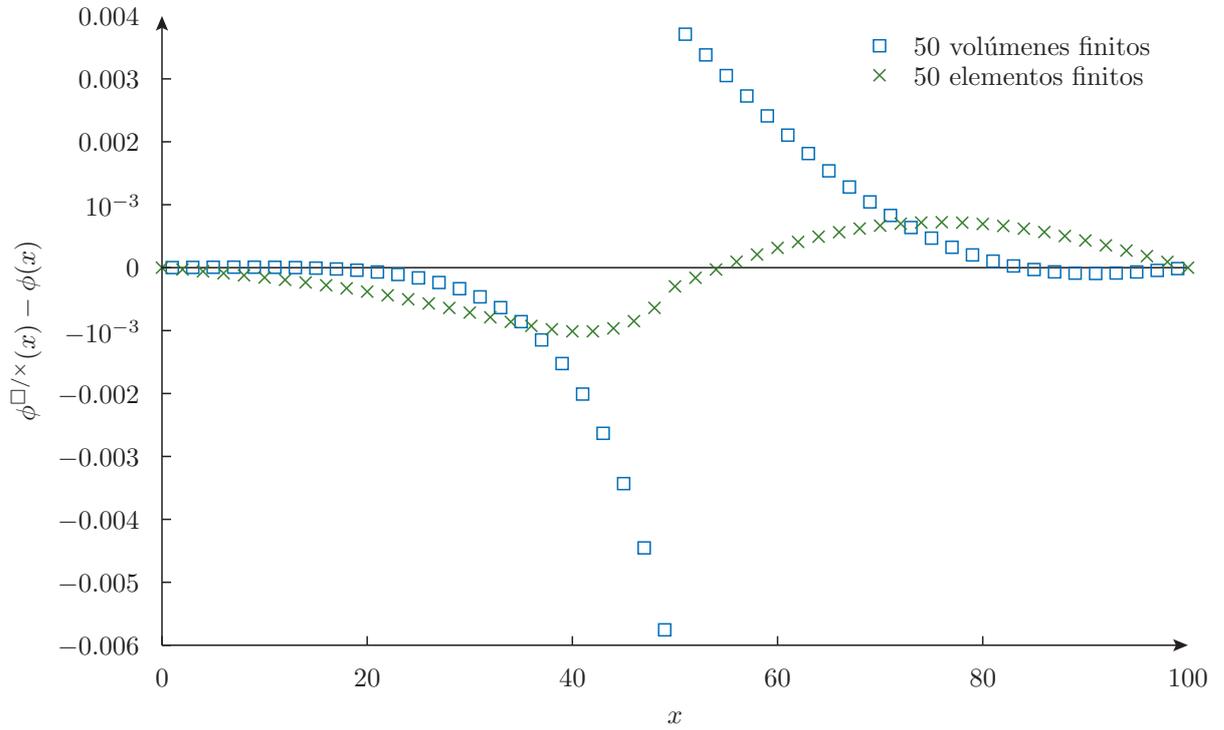


Figura 5.19 – Error cometido en la distribución de flujo de un slab a dos zonas para $n = 50$.

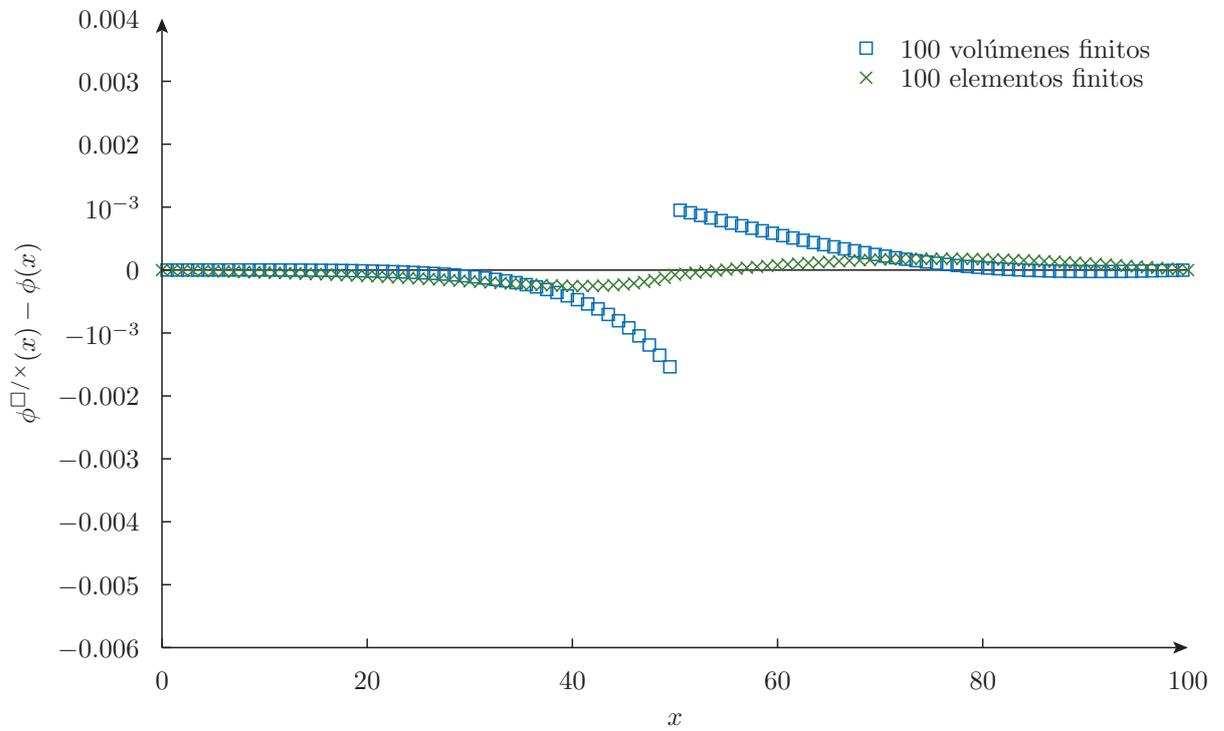


Figura 5.20 – Error cometido en la distribución de flujo de un slab a dos zonas para $n = 100$.



5.1.2.2. Volúmenes finitos con celdas homogéneas

Al emplear el esquema basado en volúmenes desarrollado en el capítulo 3, el código milonga emplea las ecuaciones (3.12) ó (3.17) dependiendo de si las celdas vecinas pertenecen a la misma entidad física (i.e. material) o no. En el problema del slab a dos zonas, solamente hay dos filas de la matriz R que contienen ecuaciones de la forma (3.17) mientras que el resto de las ecuaciones siguen las reglas para celdas homogéneas dadas por la ecuación (3.12). Con el fin de ilustrar la importancia de estas dos ecuaciones, a continuación resolvemos el problema del slab a dos zonas con $a = 100$, $b = 50$ y $n = 50$ pero usando una versión especialmente modificada de milonga que siempre aplica las ecuaciones correspondientes a celdas homogéneas para construir la matriz R .

homogeneo.was

```
# no tenemos en cuenta discontinuidades heterogeneas en
# volúmenes que no pertenecen al mismo material
SCHEME VOLUMES VOLHOM

b = 50
a = 100
lc = a/50

TEMPLATE two-zone-slab.tpl two-zone-slab.geo a b lc
SHELL "gmsht_v_0_1_two-zone-slab.geo"
MESH DIMENSIONS 1 DEGREES 1 FILE_PATH two-zone-slab.msh
INCLUDE materiales.was
MILONGA SOLVE

PRINT %.8f keff
OUTPUT_FILE numerico homogeneo.dat
PRINT_FUNCTION phi_1 FILE numerico
```

```
$ milonga homogeneo.was
0.99440269
$ pyxplot flux-hom.ppl
$
```

En la figura 5.21 mostramos el resultado que arroja no tener en cuenta explícitamente la conservación de la corriente en la interfaz entre dos materiales que tienen una discontinuidad en el coeficiente de difusión. No sólo es ciertamente diferente el perfil de flujo calculado—donde no se observa la discontinuidad en la derivada del flujo que requiere la conservación de neutrones—sino que además el factor de multiplicación k_{eff} calculado es muy diferente del analítico como vemos en la terminal, incluso indicando subcriticidad cuando en realidad el slab es supercrítico.

Luego de resolver este sencillo problema concluimos que aunque la ecuación (3.12) tenga una precisión superior que la ecuación (3.17) para discretizar el operador gradiente, su aplicación a vecinos que presentan una discontinuidad en el coeficiente de difusión conduce a errores que pueden llegar a ser inaceptables. En el esquema basado en elementos esta distinción no es necesaria ya que la formulación hace que la corriente de neutrones se conserve automáticamente al escribir la integral de la formulación débil como una suma contribuciones elementales de forma tal que se satisfaga la ecuación diferencial fuerte en todo el dominio.

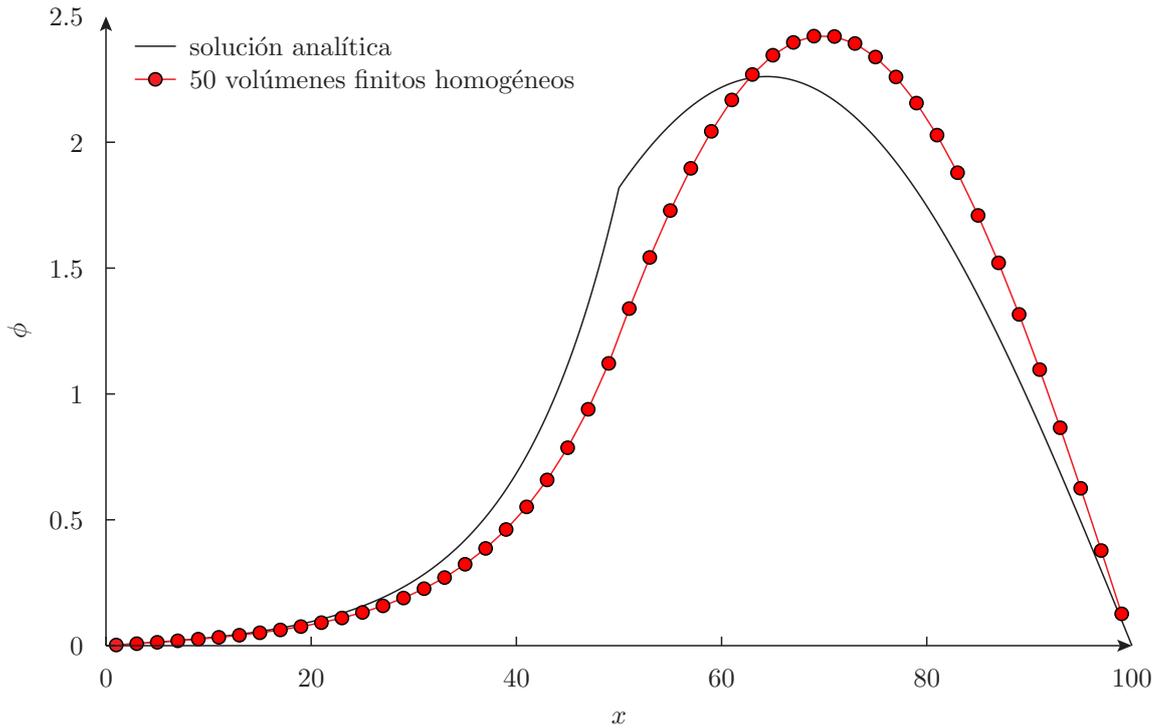
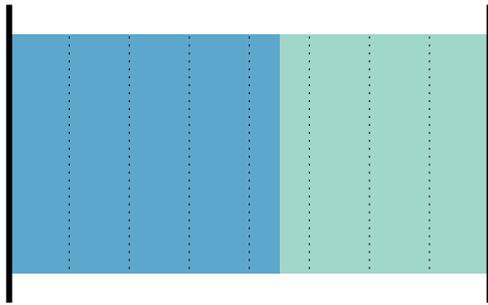


Figura 5.21 – Perfil de flujo analítico y calculado con el método de volúmenes finitos sin tener en cuenta explícitamente que en $x = 50$ el coeficiente de difusión $D(x)$ presenta una discontinuidad. La solución numérica es errónea ya que no conserva neutrones.

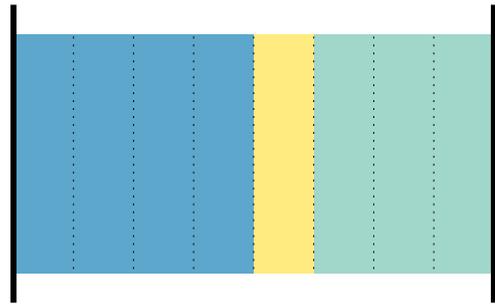
5.1.2.3. Estudio paramétrico

El problema del slab uniforme a dos zonas no tiene una aplicación directa en la industria nuclear ya que en ningún caso se modela un reactor real de la forma propuesta en la figura 5.16. Sin embargo, este problema presenta por un lado una situación simplificada de lo que sucede en la interfaz entre un material principalmente absorbente y otro material principalmente fisible en un modelo basado en difusión de neutrones con secciones eficaces homogeneizadas. Y, por otro, tiene una solución analítica que si bien tiene sus complicaciones algebraicas, es posible evaluarla para cualquier combinación de parámetros geométricos a y b y de secciones eficaces macroscópicas D , $\nu\Sigma_f$ y Σ_a de cada uno de los dos materiales. Si bien no es fácil de observar en la ecuación (5.9), el valor de k_{eff} arrojado por la solución analítica es monótonamente decreciente para el parámetro b , dejando el resto de los parámetros fijos. Este resultado es consistente con la física, que indica que mientras mayor sea la proporción de material absorbente con respecto a un ancho fijo a , la reactividad estática del sistema debe ser menor.

Cuando el problema se resuelve numéricamente utilizando algún método basado en volúmenes o similares tales como CMFD o diferencias con compensación de coeficientes de difusión sobre discretización espacial uniforme aparece un efecto conocido como *rod cusp* o dilución de secciones eficaces que hace que el k_{eff} deje de depender en forma monótona del parámetro b . En efecto, supongamos que dividimos el intervalo $[0, a]$ en N celdas iguales de longitud $\Delta x = a/N$, que es el esquema que utiliza milonga para mallas no estructuradas. Si b es múltiplo de Δx , entonces la interfaz va a coincidir con la superficie que separa dos celdas y podemos usar las ecuaciones que desarrollamos en el capítulo 3 sin problemas. Ahora, en el caso general en el que b no es múltiplo de Δx entonces habrá una celda que contendrá dos materiales. Los términos volumétricos podemos tratarlos de la misma manera que en la sección 3.1.1. Sin embargo, debemos tratar las fugas en forma diferente a la propuesta en la sección 3.1.4. Sin dar deta-



(a) Evaluación de fugas como dos interfaces homogéneas



(b) Evaluación de fugas como dos interfaces heterogéneas introduciendo un pseudo-material

Figura 5.22 – Discretización uniforme para un valor de b que no es múltiplo de Δx . Los términos volumétricos de todas las celdas se tratan de la manera discutida en la sección 3.1.1. Las fugas de las celdas que contienen un único material se tratan de la manera discutida en la sección 3.1.4. Las fugas de la celda que contiene dos materiales pueden calcularse (a) como dos juntas homogéneas, una de cada material (b) como dos juntas heterogéneas entre ambos materiales y un nuevo pseudo-material cuyas propiedades son un promedio pesado de ambos materiales.

lles que escapan a este trabajo, una alternativa es evaluar las corrientes directamente en ambas superficies de la celda mixta, suponiendo que cada una de ellas es homogénea asignando el material A a la junta de la izquierda y el material B a la junta de la derecha, como ilustramos en la figura 5.22a. Otra alternativa es generar un pseudo-material cuyas propiedades son un promedio pesado de las propiedades de los materiales I y II y calcular las fugas en ambas superficies como celdas heterogéneas (figura 5.22b).

Utilizando la versión anterior de milonga 0.1 que trabaja en modo de cálculo estructurado—es decir, sin dar una malla generada por gms—elegimos cómo tratar las fugas con la palabra clave secundaria `D_EPSILON` o `D_MEAN` sobre la palabra clave primaria `SCHEME`. Barremos el parámetro b en intervalo $[25 : 75]$ con incrementos $\Delta b = 0,1$ y escribimos el k_{eff} analítico, el k_{eff} numérico y el error en función de b .

keff-vs-b-struct.was

```

# problema estructurado, sin mesh
# este input debe correrse con milonga v0.1
PROBLEM DIMENSIONS 1 GROUPS 1
# D_EPSILON o D_MEAN controla como se evaluan las fugas
SCHEME VOLUMES $2

# estudio parametrico sobre b entre [25:75]
PARAMETRIC b 25 75 0.1

a = 100
x_bare_length = a
x_cells = $1

# definicion de materiales
MATERIAL A D_1 0.5 SigmaA_1 0.014 nuSigmaF_1 0.010
MATERIAL B D_1 1.5 SigmaA_1 0.010 nuSigmaF_1 0.014

# definicion de zonas
ZONE A MATERIAL A X_MIN 0 X_MAX b
ZONE B MATERIAL B X_MIN b X_MAX a

```



```

SOLVE_PROBLEM

eps = 0.05
k = { root(
  sqrt(D_1(0)*abs(nuSigmaF_1(0)/k-SigmaA_1(0))) *
  tan(sqrt((1/D_1(a))*abs(SigmaA_1(a)-nuSigmaF_1(a)/k))*(a-b))
+ sqrt(D_1(a)*abs(SigmaA_1(a)-nuSigmaF_1(a)/k)) *
  tanh(sqrt((1/D_1(0))*abs(nuSigmaF_1(0)/k-SigmaA_1(0)))*b)
, k, keff-eps, keff+eps, 1e-6, 0, 1) }

PRINT b k keff keff-k

```

Si en lugar de resolver el problema con una malla uniforme utilizamos un mallador para generar celdas/elementos de forma tal sus superficies externas siempre coincidan con las interfaces de la geometría continua original, entonces podemos utilizar los esquemas desarrollados en los capítulos 3 y 4, que son el eje de este trabajo. En este caso, el código gmsh elije el tamaño de las celdas/elementos tomando como base la longitud característica ℓ_c pero ajustando la cantidad y tamaño de elementos de forma tal de que siempre haya un nodo en $x = b$.

El input de milonga para resolver paramétricamente el slab a dos zonas con malla no uniforme es ligeramente diferente al del caso de malla uniforme. Ejecutamos ambos inputs para comparar el k_{eff} en función de b analítico, con volúmenes finitos sobre malla estructurada evaluando las fugas con y sin pseudo-material, con volúmenes finitos sobre malla no estructurada y con elementos finitos sobre malla no estructurada utilizando el desarrollo de los capítulos 3 y 4. Para ilustrar además cómo cambian los resultados con el refinamiento de la malla, tomamos un caso con malla gruesa con $\ell_c = a/10$ y otro con una malla fina con $\ell_c = a/100$.

keff-vs-b.was

```

# estudio parametrico sobre b entre [25:75]
PARAMETRIC b 25 75 0.1

a = 100
lc = a/$1

TEMPLATE two-zone-slab.tpl two-zone-slab.geo a b lc
SHELL "gmsh_v_0_1_two-zone-slab.geo"
MESH DIMENSIONS 1 DEGREES 1 FILE_PATH two-zone-slab.msh
INCLUDE materiales.was
MILONGA_SOLVE

eps = 0.05
k = { root(
  sqrt(D(0)*abs(nuSigmaF(0)/k-SigmaA(0))) *
  tan(sqrt((1/D(a))*abs(SigmaA(a)-nuSigmaF(a)/k))*(a-b))
+ sqrt(D(a)*abs(SigmaA(a)-nuSigmaF(a)/k)) *
  tanh(sqrt((1/D(0))*abs(nuSigmaF(0)/k-SigmaA(0)))*b)
, k, keff-eps, keff+eps, 1e-6, 0, 1) }

PRINT b k keff keff-k

```

```

$ milonga keff-vs-b.was 10 --volumes > keff-vs-b-vol-10.dat
$ milonga keff-vs-b.was 10 --elements > keff-vs-b-ele-10.dat

```



```
$ milonga01 keff-vs-b-struct.mil 10 D_EPSILON > keff-vs-b-struct-epsilon-10.dat
$ milonga01 keff-vs-b-struct.mil 10 D_MEAN > keff-vs-b-struct-mean-10.dat
$ pyxplot keff-vs-b.ppl
$ pyxplot error-vs-b-10.ppl
$ milonga keff-vs-b.was 50 --volumes > keff-vs-b-vol-50.dat
$ milonga keff-vs-b.was 50 --elements > keff-vs-b-ele-50.dat
$ milonga01 keff-vs-b-struct.mil 50 D_EPSILON > keff-vs-b-struct-epsilon-50.dat
$ milonga01 keff-vs-b-struct.mil 50 D_MEAN > keff-vs-b-struct-mean-50.dat
$ pyxplot error-vs-b-50.ppl
$
```

La figura 5.23 muestra la dependencia de k_{eff} en función de b para los diferentes métodos de solución propuestos. Como dijimos, la solución analítica arroja una variación monótonamente decreciente del factor de multiplicación con el ancho del material absorbente.

Los resultados obtenidos al resolver el slab a dos zonas con malla estructurada, tanto con el pseudo-material y son él, arrojan una variación de k_{eff} que no es monótona con b . Calcular las fugas en la celda mixta como dos interfaces homogéneas da un gran cambio en el k_{eff} cada vez que la interfaz $x = b$ pasa por algún múltiplo de Δx . Esto es esperable ya que cuando esto sucede, la corriente en alguna de las celdas pasa de calcularse con el coeficiente de difusión del material A a calcularse con el del material B en forma discontinua.

Si introducimos un pseudo-material y evaluamos las fugas como dos interfaces heterogéneas, la discontinuidad desaparece. En efecto, las propiedades del pseudo-material son promedios pesados de los dos materiales por lo que si la interfaz se encuentra un diferencial dx a la derecha (izquierda) de la interfaz izquierda (derecha) entonces el coeficiente de difusión utilizado para evaluar la corriente es igual al coeficiente de difusión del material A (B) más un diferencial. Luego el k_{eff} debe variar en forma continua con b . Sin embargo, es evidente que su derivada no es continua ya que observamos picos de k_{eff} en múltiplos de Δx . Este efectos se conoce como rod cusp ya que aparece en cálculos tridimensionales de núcleo cuando las barras de control se mueven en forma continua y sus extremos no coinciden con la malla de la discretización espacial. Tal como vemos, puede darse el caso de que la reactividad reportada para una introducción z de una barra de control sea menor que la reportada para una introducción $z + \epsilon$, lo que puede llegar a introducir no sólo errores groseros sino también inestabilidades físicas. En efecto, si en un cálculo transitorio acoplado el sistema de control decide introducir las barras de control para bajar la potencia y el cálculo neutrónico indica que la potencia aumenta, entonces el inconveniente es mayor. El análisis de las causas de este efecto y de las alternativas para mitigar sus consecuencias en mallas estructuradas está fuera del alcance de este trabajo. Sin embargo, una forma de evitar la aparición de este efecto es utilizar una malla no estructurada.

En efecto, al resolver el problema generando una malla no estructurada de forma tal de que siempre haya una interfaz entre celdas/elementos en $x = b$, entonces los picos desaparecen y la dependencia de k_{eff} con b es mucho más suave y se parece a la solución analítica, manteniendo el resultado de que volúmenes finitos sobre-estima el factor de multiplicación y elementos finitos lo sub-estima. Además, en valor absoluto, elementos está más cerca de la solución analítica.

De todas maneras, la malla no estructurada no elimina completamente los picos y discontinuidades. La figura 5.24 muestra el error cometido en cada caso con respecto a la solución analítica. Los pequeños picos en las curvas correspondientes a los cálculos no estructurados provienen de una pequeña discontinuidad geométrica que podemos explicar como sigue. Supongamos que $b \ll a$. Entonces gmsh decidirá asignar un elemento al material A y $n - 1$ al material B . A medida que b aumenta, en algún punto decidirá asignar dos elementos a A y $n - 2$ a B . En este momento, la discretización espacial habrá cambiado discontinuamente con b , por lo que es de esperar que el k_{eff} también tenga una pequeña discontinuidad con b . Manteniendo la escala vertical, la figura 5.25 muestra el error cometido por las soluciones numéricas pero

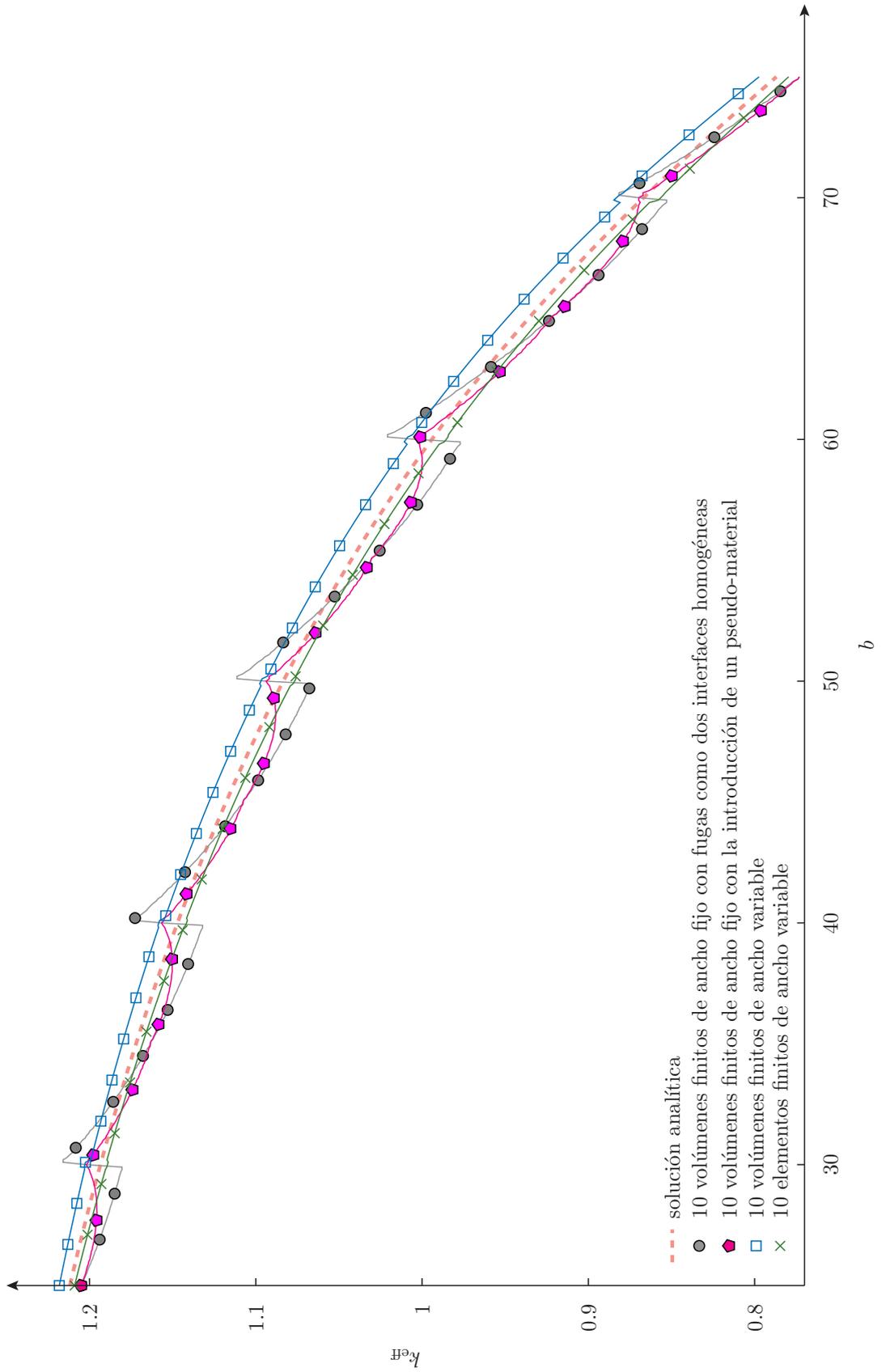


Figura 5.23 – k_{eff} vs. distancia b para $a = 100$ fijo y $n = 10$.

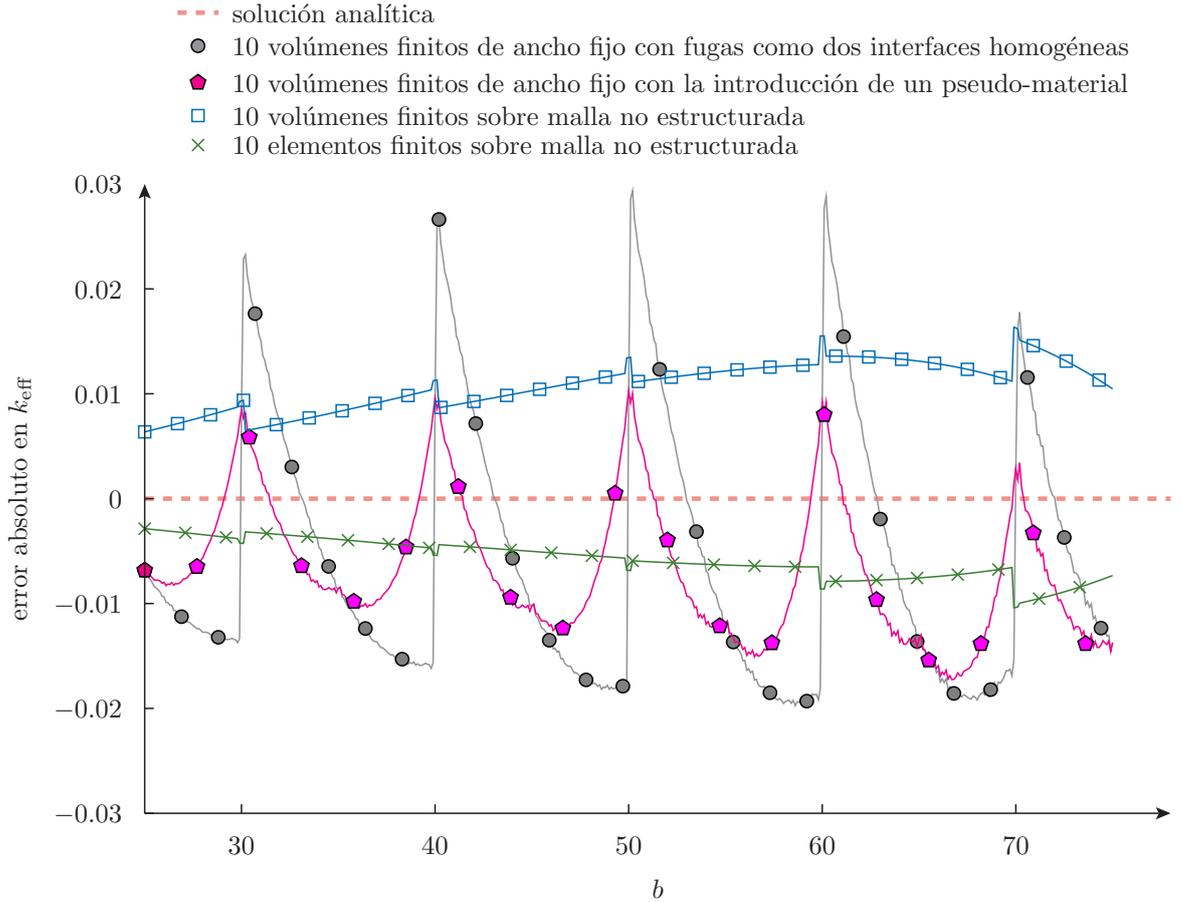


Figura 5.24 – Error en el cálculo de k_{eff} con respecto al resultado analítico vs. distancia b para $a = 100$ fijo y $n = 10$.

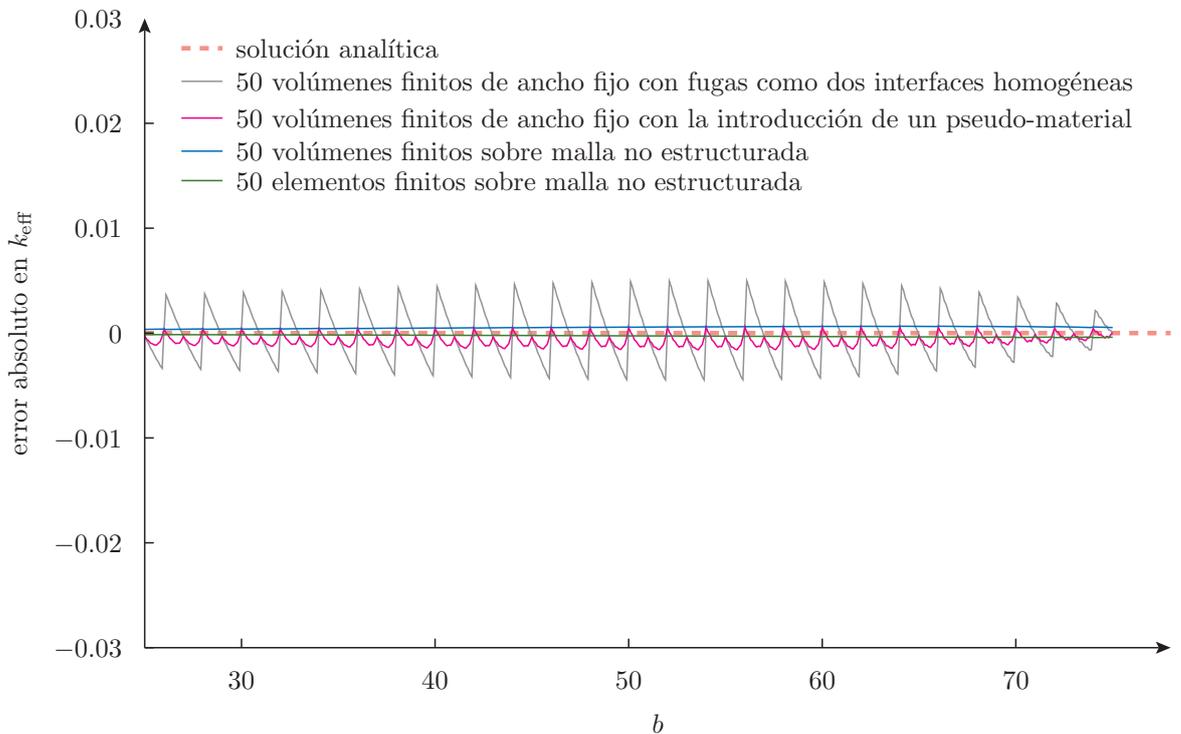


Figura 5.25 – Error en el cálculo de k_{eff} con respecto al resultado analítico vs. distancia b para $a = 100$ fijo y $n = 50$.



para malla fina con $\ell_c = a/50$, que da $n = 50$.

5.1.2.4. Malla refinada en la discontinuidad

Si bien sabemos que una malla fina reduce significativamente los errores cometidos por los esquemas numéricos, no podemos—al menos por ahora—refinar en forma indiscriminada sin tener que pagar los costos computacionales asociados: potencia de cálculo y capacidad de almacenamiento. Pero también sabemos, por lo menos en este problema, que la mayoría de los errores numéricos se pueden asociar la zona de la interfaz entre los materiales A y B . Por lo tanto, podemos pedirle a gmsht que refine selectivamente la malla alrededor del punto $x = b$. Para eso podemos utilizar el siguiente archivo de template:

two-zone-slab-ref.tpl

```
a = $1;
b = $2;
lc = $3;

Point(1) = {0, 0, 0, lc};
Point(2) = {b, 0, 0, 0.15*lc};
Point(3) = {a, 0, 0, lc};

Line(1) = {1, 2};
Line(2) = {2, 3};
Physical Line("A") = {1};
Physical Line("B") = {2};

Physical Point("left") = {1};
Physical Point("right") = {3};
```

El punto número dos, que define la posición de la interfaz entre los dos materiales tiene una longitud característica igual al quince por ciento del parámetros ℓ_c , que se le asigna a los dos puntos en $x = 0$ y en $x = a$.

flux-refinado.was

```
SCHEME $1

b = 50      # ancho de la zona A-comienzo de la zona B
a = 100     # ancho total del slab
n = $2      # cantidad de elementos
lc = a/n    # longitud característica del elemento

TEMPLATE two-zone-slab-ref.tpl two-zone-slab-ref.geo  a b lc
SHELL "gmsht_v0_1_two-zone-slab-ref.geo"
MESH DIMENSIONS 1 DEGREES 1 FILE_PATH two-zone-slab-ref.msh
INCLUDE materiales.was
MILONGA SOLVE

# leemos el keff analitico en funcion de b que calculamos antes
FUNCTION k(b) FILE_PATH keff-vs-b-ele-50.dat COLUMNS 1 2

# imprimimos informacion en la pantalla
IF cells
  PRINT TEXT "celdas=" %g cells
ELSE
```

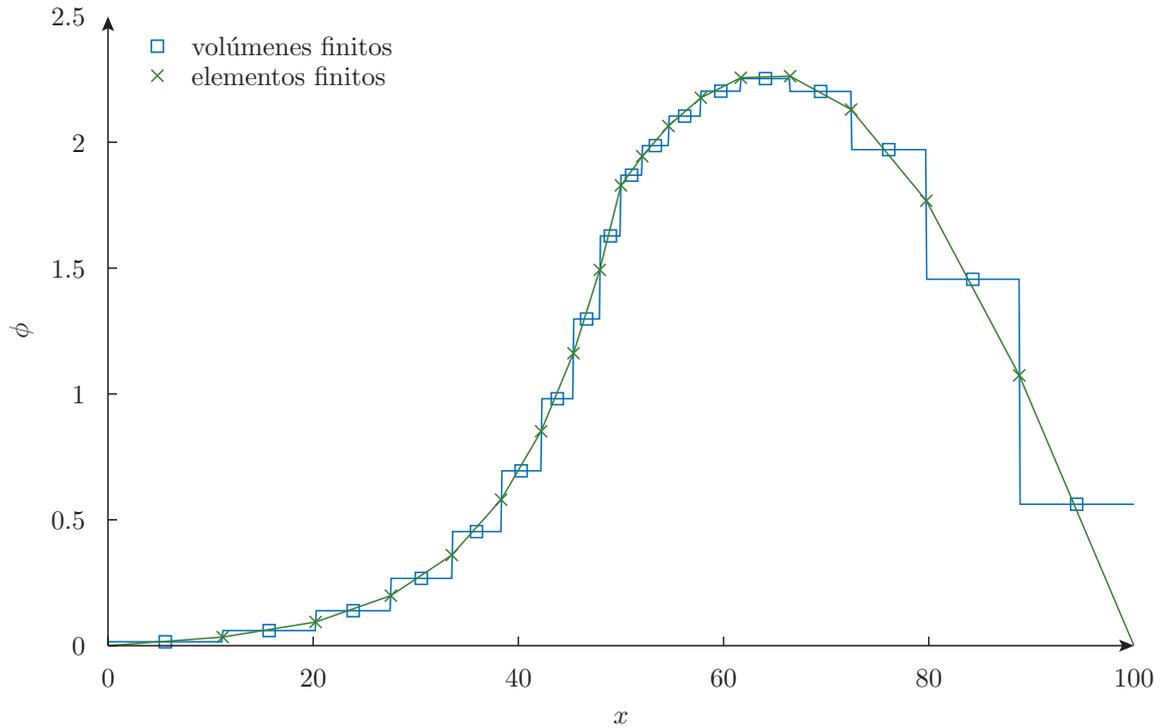


Figura 5.26 – Distribución de flujo calculado con una malla refinada en la discontinuidad.

```

PRINT TEXT "nodos_=" %g nodes
ENDIF
PRINT TEXT "keff_=" %8f keff TEXT "error_=" %e keff-k(b)

# y la distribucion de flujo en archivos
OUTPUT_FILE numerico $1-%0f-ref.dat n
PRINT_FUNCTION phi_1 FILE numerico

OUTPUT_FILE numerico-cont $1-%0f-ref-cont.dat n
PRINT_FUNCTION phi_1 FILE numerico-cont MIN 0 MAX a STEP 0.1

```

```

$ milonga flux-refinado.was volumes 8
celdas= 18
keff = 1.08486871      error = 1.377708e-03
$ milonga flux-refinado.was elements 8
nodos = 19
keff = 1.08184336     error = -1.647640e-03
$ pyxplot flux-ref.ppl
$

```

Elijiendo un valor de $\ell_c = 8$, gmsh genera una malla de 18 elementos por lo que el tamaño del problema es similar al que resolvimos en la sección 5.1.2.1 para $n = 20$. Podemos ver que efectivamente el error cometido al calcular k_{eff} es menor si refinamos selectivamente la malla alrededor de la discontinuidad para obtener matrices de tamaño similar que para el caso de una malla uniforme. En la figura 5.26 mostramos la distribución de flujo resultante al utilizar una malla refinada. La reducción del error es mayor para volúmenes (pasa de $3,58 \times 10^{-6}$ a $1,38 \times 10^{-6}$) que para elementos (pasa de $-1,72 \times 10^{-6}$ a $-1,65 \times 10^{-6}$).



5.1.3. Slabs reflejados a dos grupos

En esta sección estudiamos ahora slabs con una zona físil de combustible y material no físil actuando como reflector en los extremos. Otra vez, estos casos no tienen aplicación directa pero su estudio puede proveer ciertas conclusiones que con un poco de cuidado pueden aplicarse a problemas de interés industrial. Pero además, el hecho de que el flujo siga siendo unidimensional facilita la comparación entre esquemas basados en volúmenes y en elementos que, recordemos, es el tema central del trabajo.

En todos los inputs de milonga incluimos el siguiente archivo que define las propiedades a dos grupos de energías de los materiales que vamos a utilizar en esta sección:

materials.was

```
MATERIAL fuel {
  D_1      1.5
  D_2      0.4
  SigmaS_1->2 0.02
  SigmaA_1  0.01
  SigmaA_2  0.08
  nuSigmaF_2 0.135
}

MATERIAL left_refl {
  D_1      2.0
  D_2      0.3
  SigmaS_1->2 0.04
  SigmaA_1  0.00
  SigmaA_2  0.01
}

MATERIAL right_refl {
  D_1      2.0
  D_2      0.3
  SigmaS_1->2 0.05
  SigmaA_1  0.00
  SigmaA_2  0.005
}

MATERIAL abs1 {
  D_1      1.2
  D_2      0.3
  SigmaS_1->2 0.01
  SigmaA_1  0.10
  SigmaA_2  0.65
}

MATERIAL abs2 {
  D_1      1.35
  D_2      0.45
  SigmaS_1->2 0.005
  SigmaA_1  0.010
  SigmaA_2  0.020
}
```

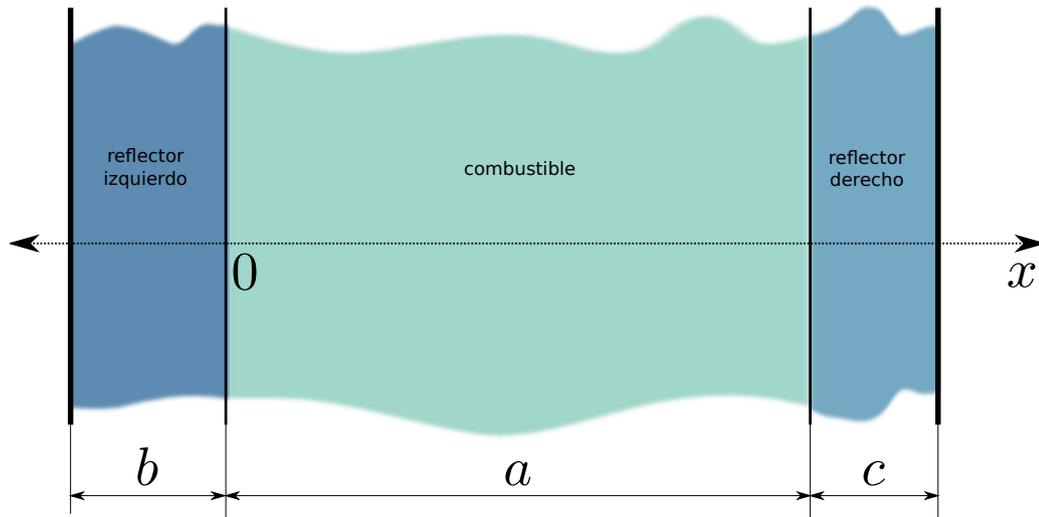


Figura 5.27 – Slab reflejado

5.1.3.1. Distribución de flujo

Resolvemos ahora el slab reflejado que ilustramos en la figura 5.27. Consiste en una zona de material fisible de ancho a , reflejada a la izquierda por un reflector de longitud b y por la derecha por uno de longitud c . Podemos implementar esta geometría a través de un archivo template—que luego milonga transformará en un archivo de geometría para gmsh según los valores que tomen las variables a , b , c y ℓ_c —como sigue:

reflected.tpl

```

a = $1;
b = $2;
c = $3;
lc = $4;

Point(1) = { -b, 0, 0, lc };
Point(2) = { 0, 0, 0, lc };
Point(3) = { a, 0, 0, lc };
Point(4) = { a+c, 0, 0, lc };

Line(1) = { 1, 2 };
Line(2) = { 2, 3 };
Line(3) = { 3, 4 };

Physical Line("left_refl") = { 1 };
Physical Line("fuel") = { 2 };
Physical Line("right_refl") = { 3 };

Physical Point("left") = { 1 };
Physical Point("right") = { 4 };

```

En forma similar a la solución del problema de la sección 5.1.2.4, calculamos tanto el k_{eff} como la distribución de flujos—rápido y térmico en este caso—utilizando volúmenes y elementos sobre una malla uniforme con veinte y con cien celdas/elementos. Tomamos un ancho de combustible $a = 100$ y reflectores desiguales con $b = 20$ y $c = 15$.

reflected-flux.was

```

SCHEME $1

a = 100      # ancho del fuel
b = 20      # ancho del reflector izquierdo
c = 15      # ancho del reflector derecho
n = $2      # cantidad de elementos dado en la linea de comandos
lc = a/n    # longitud característica

TEMPLATE reflected.tpl reflected.geo a b c lc
SHELL "gmsht_v_0_2_reflected.geo"
MESH DIMENSIONS 1 DEGREES 2 FILE_PATH reflected.msh
INCLUDE materials.was
MILONGA SOLVE

PRINT %.8f keff

OUTPUT_FILE grid flux-$1-%.0f-grid.dat n
PRINT_FUNCTION phi_1 phi_2 FILE grid

OUTPUT_FILE cont flux-$1-%.0f-cont.dat n
PRINT_FUNCTION phi_1 phi_2 FILE cont MIN -b MAX a+c STEP 0.1

```

```

$ milonga reflected-flux.was volumes 20
1.08619893
$ milonga reflected-flux.was volumes 100
1.08775910
$ milonga reflected-flux.was elements 20
1.08879051
$ milonga reflected-flux.was elements 100
1.08790118
$ pyxplot flux-20-vol.ppl
$ pyxplot flux-20-ele.ppl
$ pyxplot flux-100-vol.ppl
$ pyxplot flux-100-ele.ppl
$ pyxplot flux-1-vol.ppl
$ pyxplot flux-1-ele.ppl
$ pyxplot flux-2-vol.ppl
$ pyxplot flux-2-ele.ppl
$ pyxplot flux-1-20.ppl
$ pyxplot flux-1-100.ppl
$ pyxplot flux-2-20.ppl
$ pyxplot flux-2-100.ppl
$

```

Como sucede cuando uno va incrementando la complejidad de un cierto problema, la cantidad de resultados interesantes también aumenta tanto en cantidad como en complejidad. Tenemos, además de los cuatro factores de multiplicación efectivos, dos distribuciones de flujo para cada uno de los cuatro casos numéricos que queremos comparar entre sí. Las figuras 5.28–5.31 muestran los dos flujos rápido y térmico en el mismo gráfico para los cuatro casos. Las figuras 5.32–5.35 comparan en el mismo gráfico el cambio de malla fina a malla gruesa para cada flujo y para cada esquema de discretización espacial. Finalmente, las figuras 5.36–5.39 comparan en cada gráfico los flujos calculados con volúmenes y con elementos con la misma malla.

Al utilizar dos grupos de energías podemos ver la aparición del conocido “hombro térmico” en el reflector, cuya existencia nos prometen pero casi nunca nos muestran en cursos de física

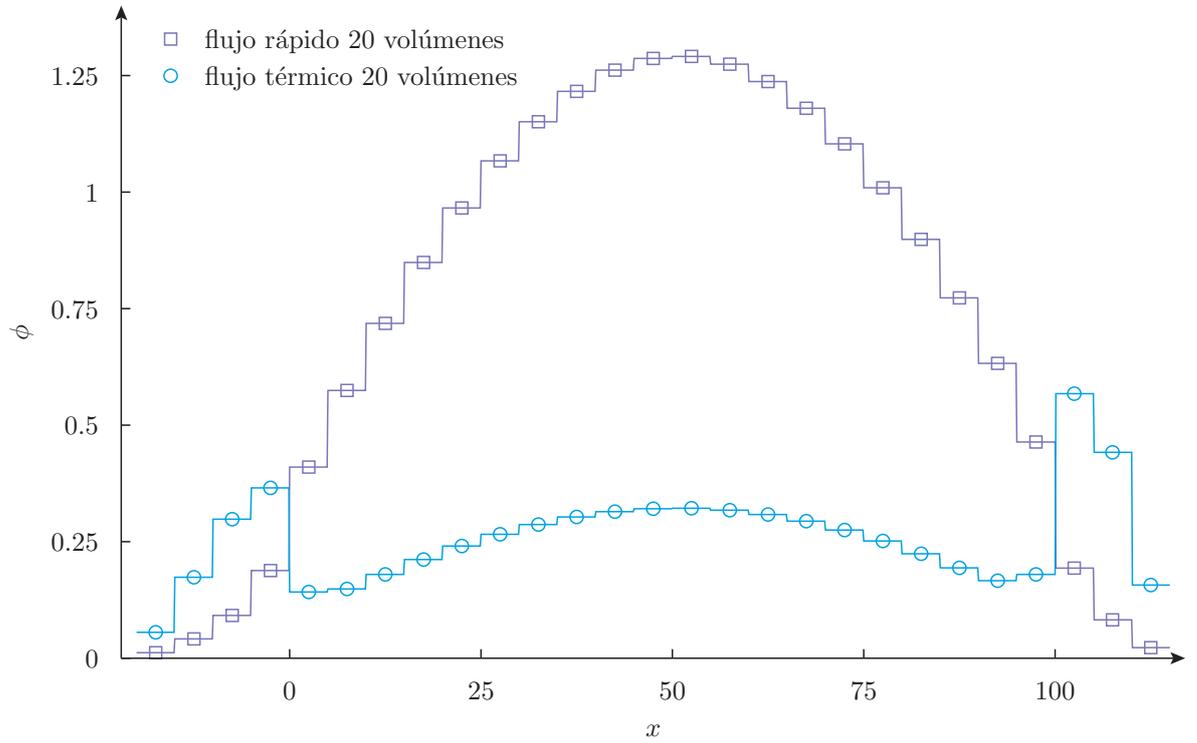


Figura 5.28 – Comparación de flujo rápido y flujo térmico para $n = 20$ con volúmenes finitos

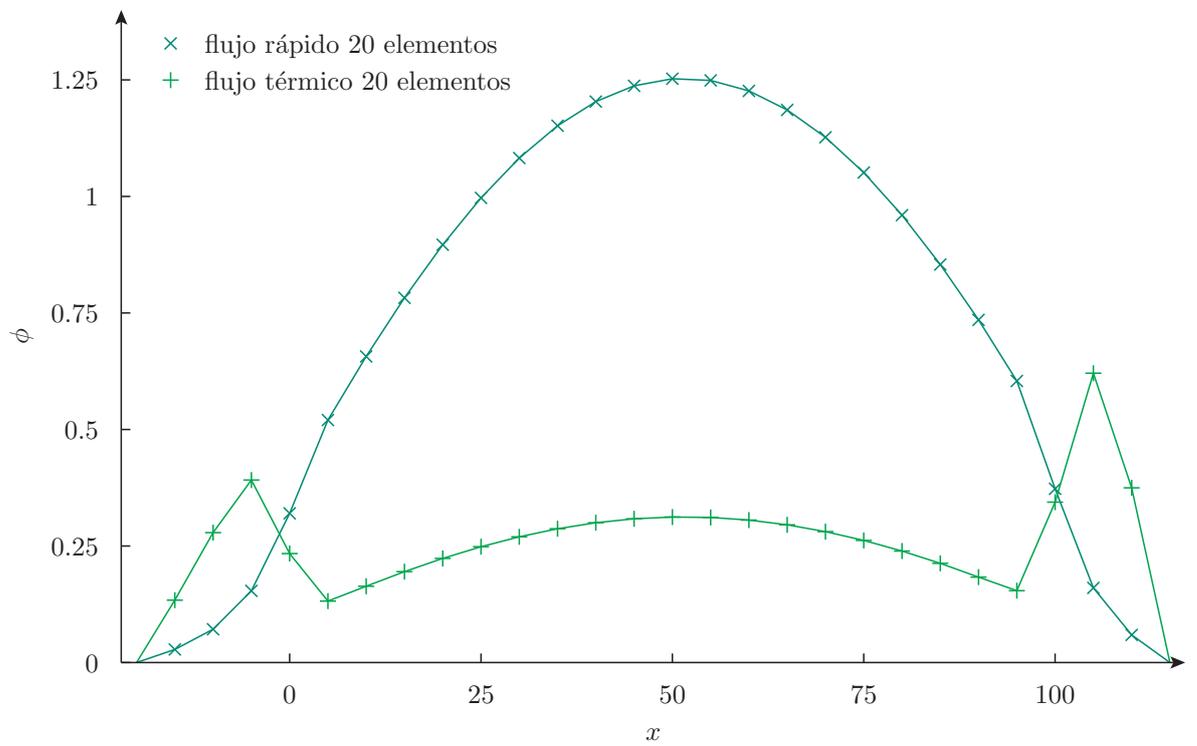


Figura 5.29 – Comparación de flujo rápido y flujo térmico para $n = 20$ con elementos finitos

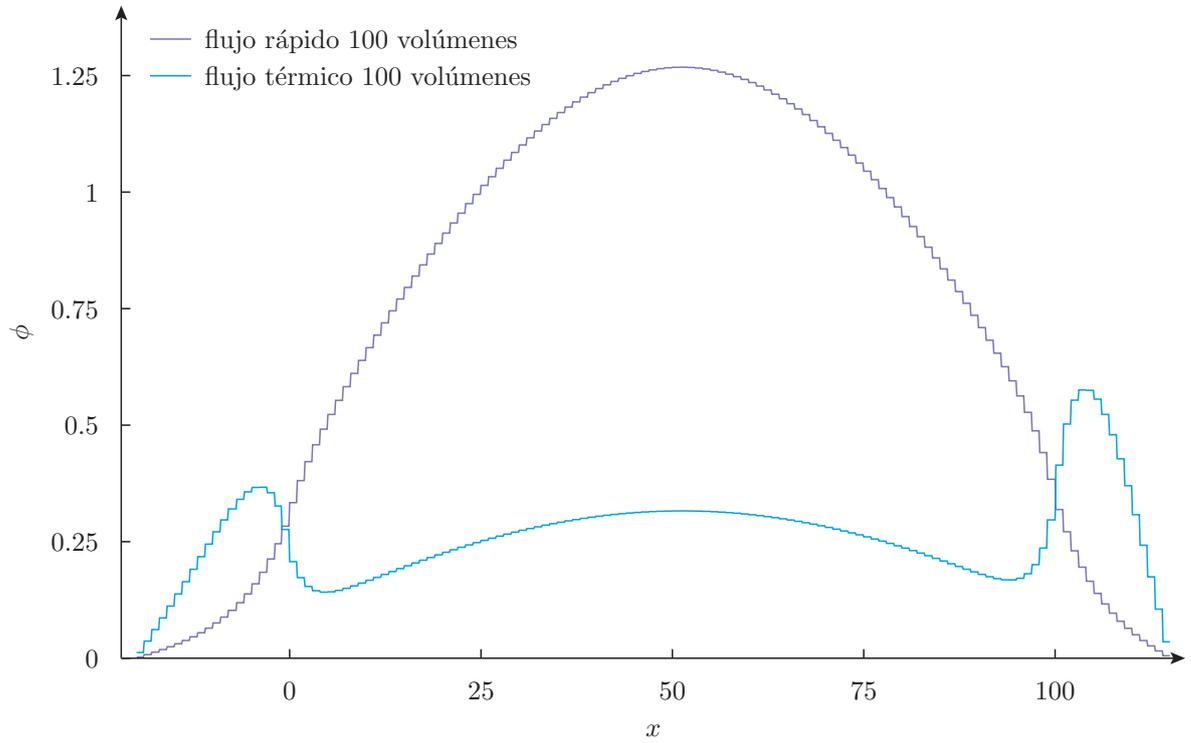


Figura 5.30 – Comparación de flujo rápido y flujo térmico para $n = 100$ con volúmenes finitos

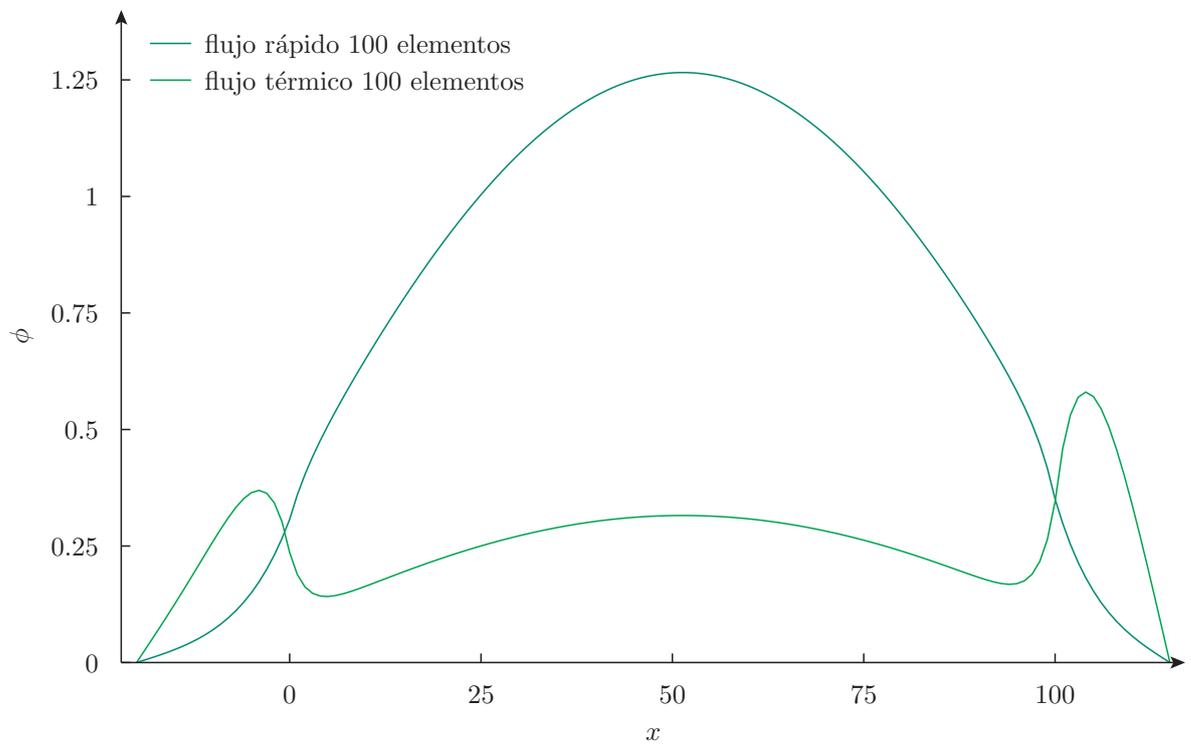


Figura 5.31 – Comparación de flujo rápido y flujo térmico para $n = 100$ con elementos finitos



de reactores basados casi exclusivamente en soluciones analíticas de problemas canónicos. Para la malla gruesa con $n = 20$ es evidente de las figuras que ya no es cierto que las soluciones provistas por volúmenes finitos sean el promedio de las soluciones dadas por elementos en los nodos que definen la celda, aunque si refinamos la malla podemos asumir que este resultado sigue siendo cierto.

5.1.3.2. Efectividad de los reflectores

Es un resultado conocido que la incorporación de material reflector fuera del núcleo hace que la reactividad neta aumente. Sin embargo, no existen en general resultados analíticos que indiquen qué tan efectivo es un reflector de un cierto ancho. Dado que milonga fue diseñado principalmente para resolver problemas de optimización de parámetros de diseño, el estudio de la efectividad de un reflector es particularmente adecuado para ser estudiado utilizando este código.

Para ello, resolvemos el slab de la sección anterior para un ancho de material físil $a = 100$ fijo variando paramétricamente los anchos b y c de los reflectores, cada uno en el intervalo $[1, 40]$ con incrementos $\Delta b = \Delta c = 2$. Fijamos además la cantidad de celdas/elementos $n = 100$ y graficamos el k_{eff} resultante en función de b y de c , tanto para volúmenes como para elementos finitos. Debido a una particularidad del programa gnuplot que usamos para graficar la superficie escalar $k_{\text{eff}}(b, c)$, debemos agregar una línea en blanco cada vez que se incrementa el primer parámetro. Esto lo logramos fácilmente filtrando con sed el archivo generado por milonga en modo paramétrico. Como además también nos interesa calcular la diferencia entre los factores de multiplicación calculados con volúmenes y con elementos, usamos el comando paste para mezclar las dos salidas de milonga en un sólo archivo y poder restar una columna de otra desde gnuplot.

keff-vs-refl.was

```
# estudio parametrico sobre b y sobre c
PARAMETRIC b 1 40 2
PARAMETRIC c 1 40 2

a = 100      # ancho del fuel
n = 100     # cantidad de celdas/elementos
lc = a/n+1  # longitud caracteristica

TEMPLATE reflected.tpl reflected.geo a b c lc
SHELL "gmsh_v_0_2_reflected.geo"
MESH DIMENSIONS 1 DEGREES 2 FILE_PATH reflected.msh
INCLUDE materials.was
MILONGA SOLVE

# imprimimos una linea en blanco cuando c=1 para gnuplot
IF c=1 PRINT

PRINT b c keff
```

```
$ milonga keff-vs-refl.was --volumes > keff-vs-refl-volumes.dat
$ milonga keff-vs-refl.was --elements > keff-vs-refl-elements.dat
$ paste keff-vs-refl-volumes.dat keff-vs-refl-elements.dat > keff-vs-refl-diff. ←
  dat
$ gnuplot keff-vs-refl-volumes.gnuplot
$ gnuplot keff-vs-refl-elements.gnuplot
```

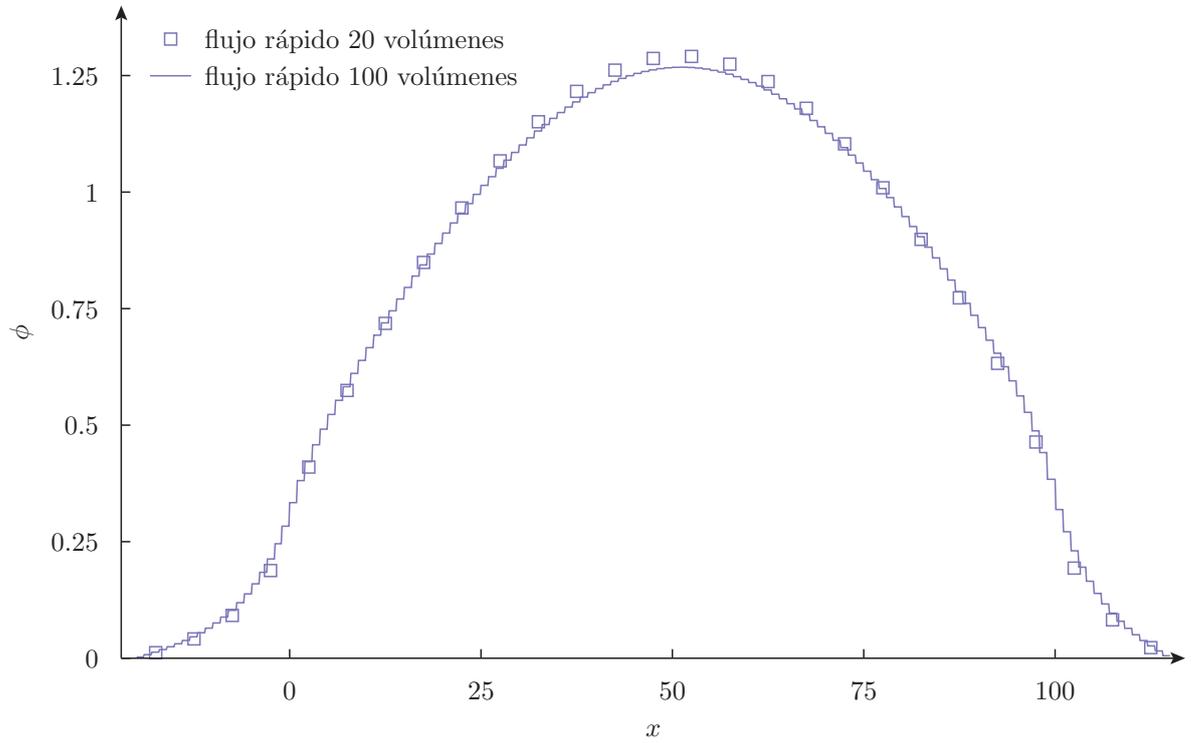


Figura 5.32 – Comparación para diferentes mallas de flujos rápidos con volúmenes finitos

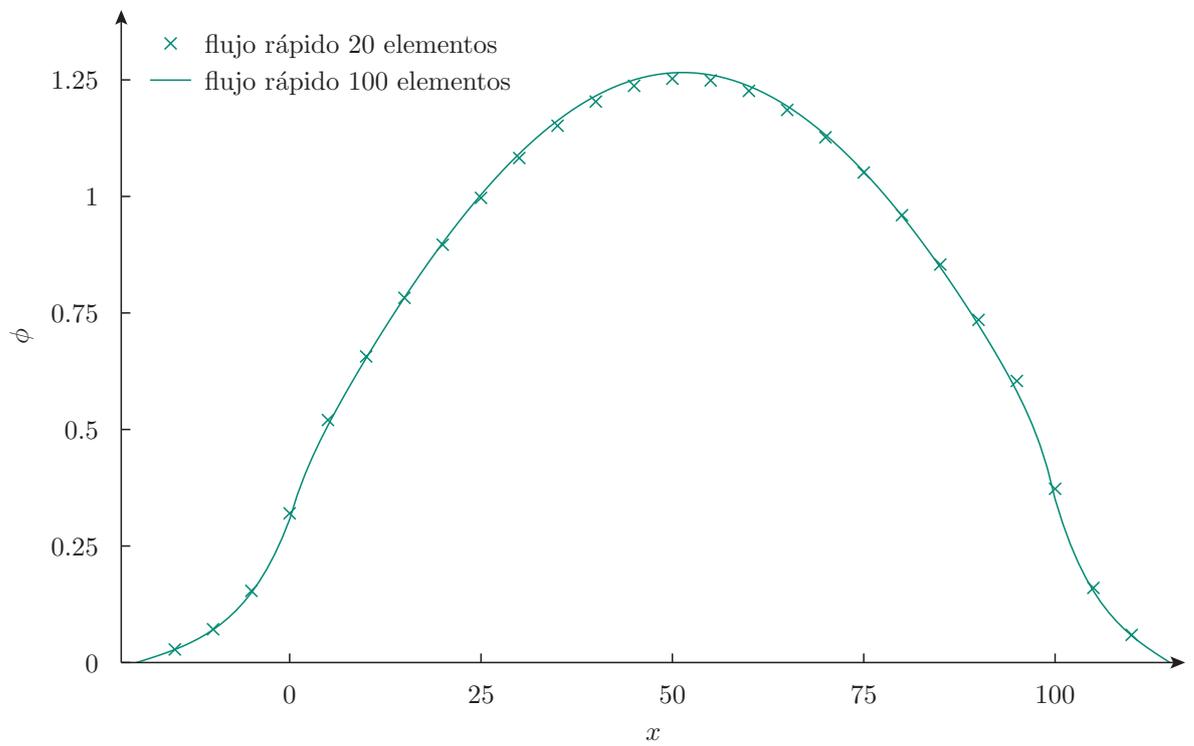


Figura 5.33 – Comparación para diferentes mallas de flujos rápidos con elementos finitos

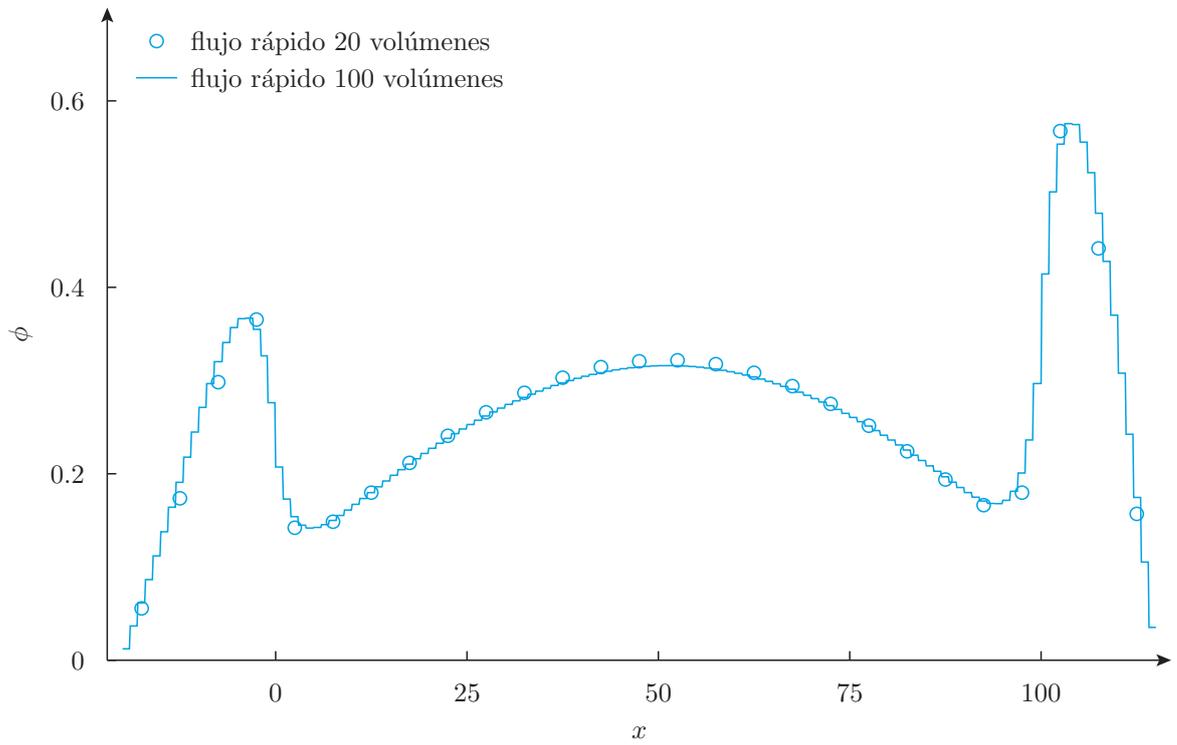


Figura 5.34 – Comparación para diferentes mallas de flujos térmicos con volúmenes finitos

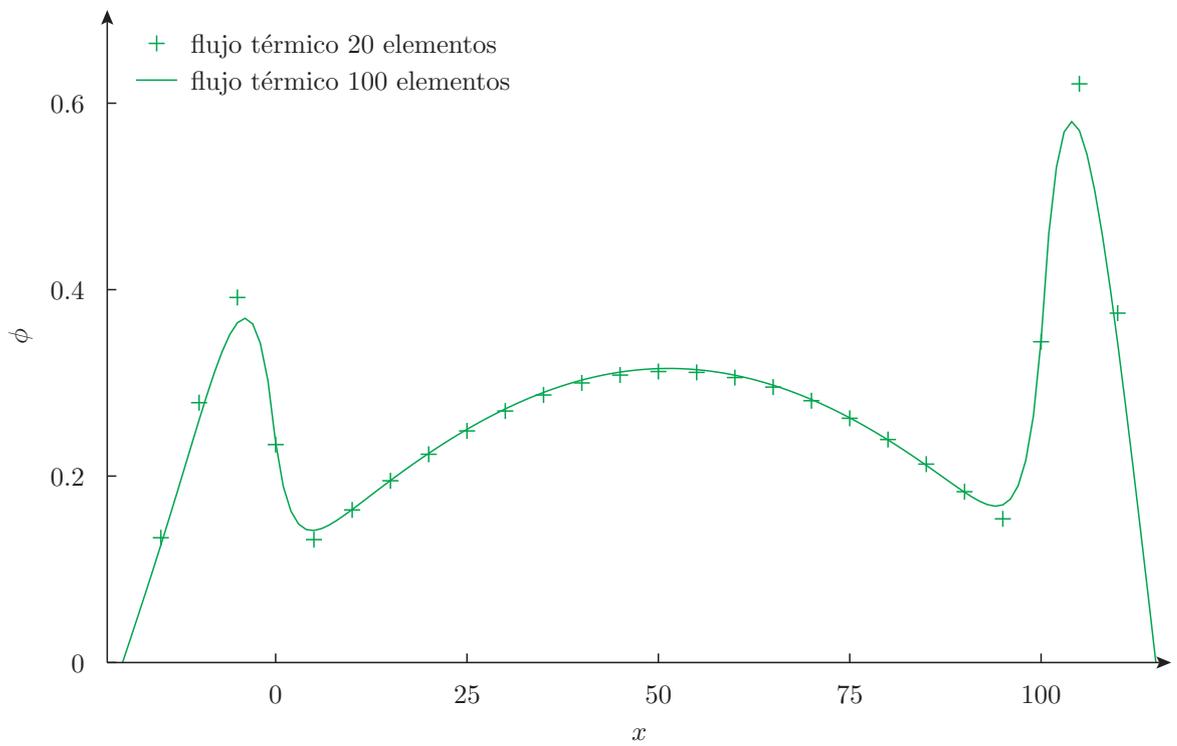


Figura 5.35 – Comparación para diferentes mallas de flujos térmicos con elementos finitos

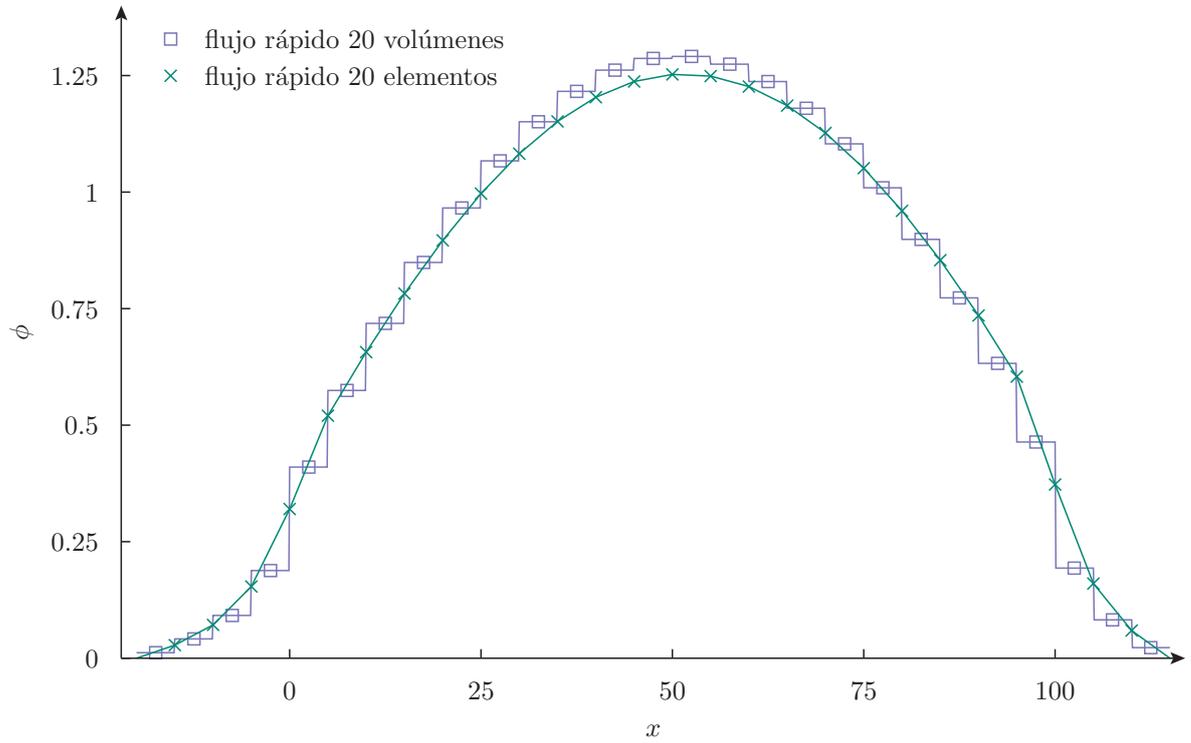


Figura 5.36 – Comparación entre volúmenes y elementos de flujos rápidos con $n = 20$

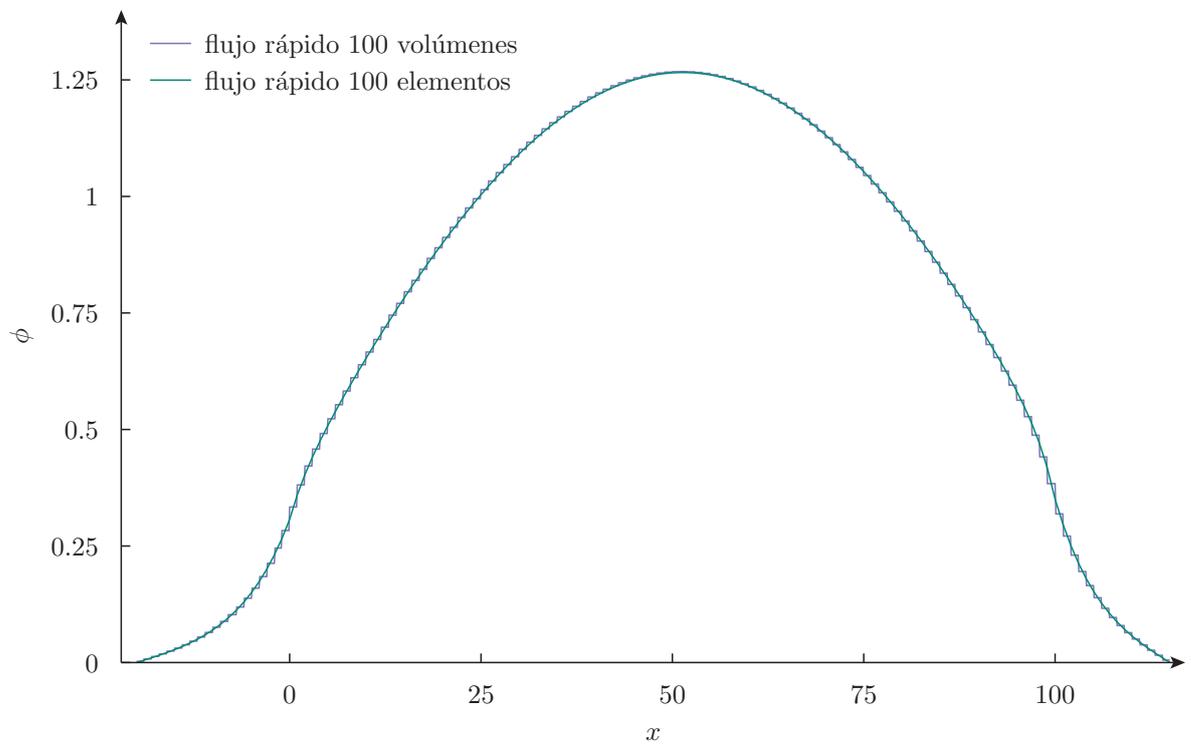


Figura 5.37 – Comparación entre volúmenes y elementos de flujos rápidos con $n = 100$

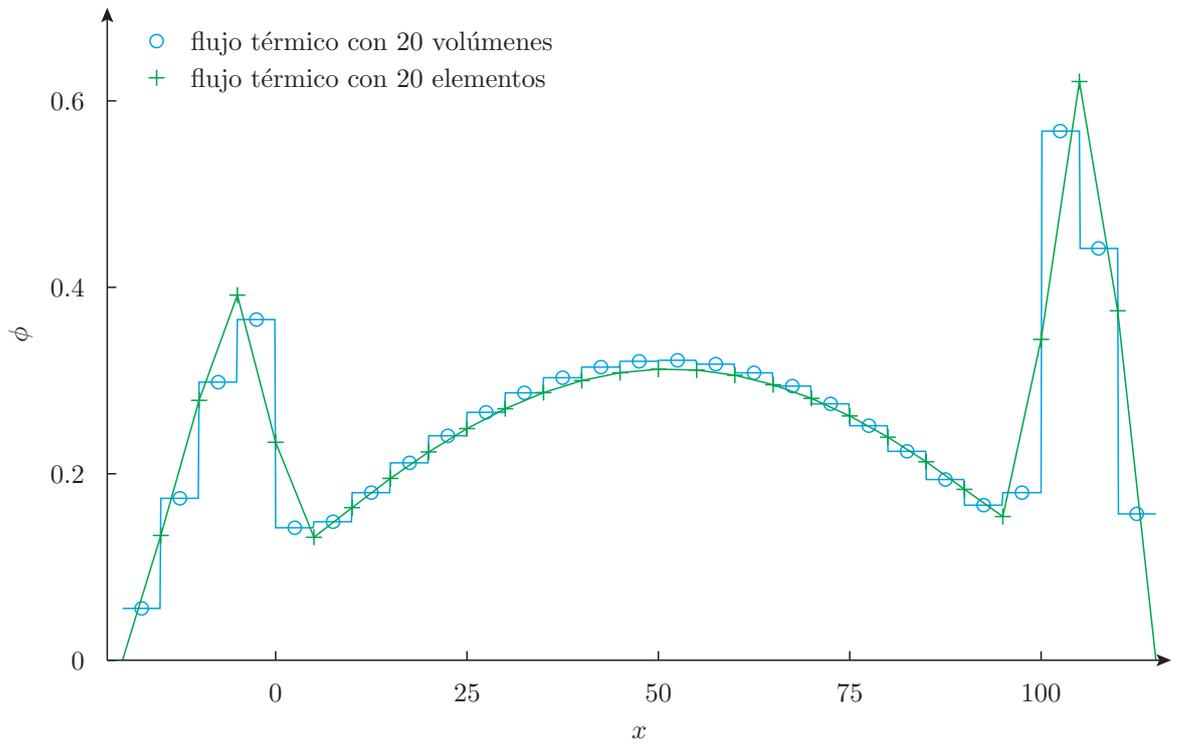


Figura 5.38 – Comparación entre volúmenes y elementos de flujos térmicos con $n = 20$

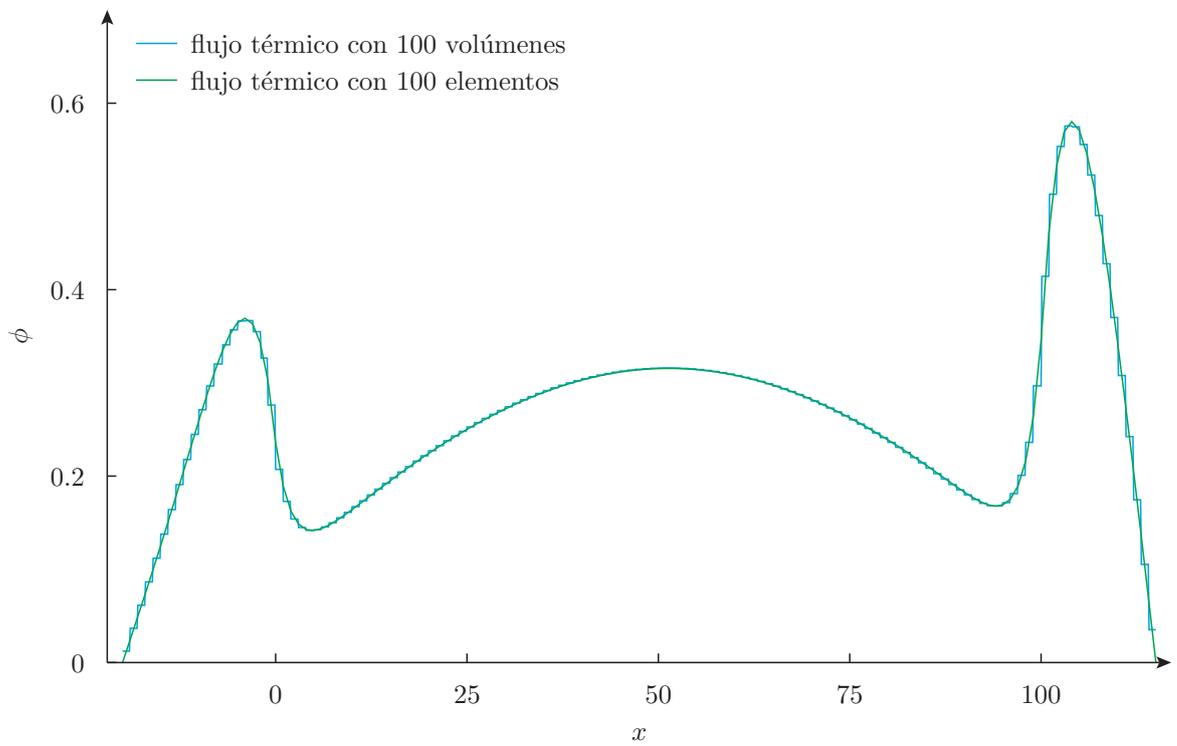


Figura 5.39 – Comparación entre volúmenes y elementos de flujos térmicos con $n = 100$



```
$ gnuplot keff-vs-refl-diff.gnuplot
$
```

Las figuras 5.40 y 5.41 muestran cómo depende el factor de multiplicación para un a fijo, en función de los espesores b y c de los reflectores izquierdo y derecho respectivamente. Como es de esperar, para espesores pequeños, el k_{eff} es pequeño y aumenta a medida que también lo hacen b y c . Y, como también es de esperar, la efectividad satura y a partir de un cierto espesor, ya no tiene sentido seguir agregando material para aumentar la reactividad estática. En principio, las dos figuras—una calculada con volúmenes y otra con elementos—parecen indistinguibles. Sin embargo, vemos que las curvas de isoreactividad proyectadas sobre el plano $b - c$ no coinciden. Para mejor comparar los métodos de volúmenes y elementos, graficamos en la figura 5.42 la diferencia entre el k_{eff}^{\square} calculado con volúmenes y el k_{eff}^{\times} calculado con elementos. No sólo vemos que esta diferencia aumenta a medida que hay más material reflector con respecto a una cantidad fija de material físil, sino también notamos que $k_{\text{eff}}^{\square} - k_{\text{eff}}^{\times} < 0$ por lo que el factor de multiplicación calculado con volúmenes es menor que el calculado con elementos. Este comportamiento es opuesto a lo que habíamos encontrado en la sección 5.1.1, en particular en la figura 5.4 y en la ecuación (5.7). Parecería que el carácter de exceso o defecto en el error cometido por las dos discretizaciones espaciales estudiadas depende de la relación entre la cantidad de material multiplicativo y no multiplicativo contenido en el dominio donde resolvemos la ecuación de difusión. Este incipiente resultado debería ser objeto de futuros y detallados estudios, que nos permitan entender mejor aún cómo funcionan los métodos numéricos que utilizamos para diseñar y analizar reactores nucleares reales.

5.1.3.3. Slab con reflector y absorbentes

Además de incluir reflectores en los extremos, en esta sección resolvemos un reactor tipo slab unidimensional con materiales absorbentes embebidos dentro del material físil base con la geometría que mostramos en la figura 5.43. En este problema, esperamos que el flujo se deprima alrededor de los absorbentes. Utilizando las secciones eficaces definidas en el archivo `materiales.was` mostrado en la página 144, el absorbente uno tiene secciones eficaces de absorción que son un orden de magnitud mayores que el absorbente dos por lo que a su vez hacemos que su espesor w_1 sea menor que el espesor w_2 del absorbente dos.

Además de calcular el k_{eff} , también estamos interesados ahora en calcular el factor de pico f_p de la distribución de potencia definido como el cociente entre el valor máximo y el valor medio del flujo térmico en el núcleo

$$f_p = \frac{\text{máx}_x \phi_2(x)}{\frac{1}{a} \int_0^a \phi_2(x) dx}$$

Esta vez, en lugar de usar un archivo `template` para generar tanto el archivo de geometría como el archivo de malla desde `milonga`, partimos de tres archivos de geometría fijos. Los tres definen la misma geometría continua de la figura 5.43 con la diferencia que cada uno de ellos asigna longitudes características diferentes a los puntos que definen los centros de los materiales absorbentes. Todos se basan en el archivo básico llamado `slab-abs.geo`, que deja abierta la posibilidad de que la malla pueda ser refinada alrededor de los absorbentes alterando la longitud característica base ℓ_c a través de factores f_1 y f_2 :

`slab-abs.geo`

```
a = 100; // ancho del slab
```

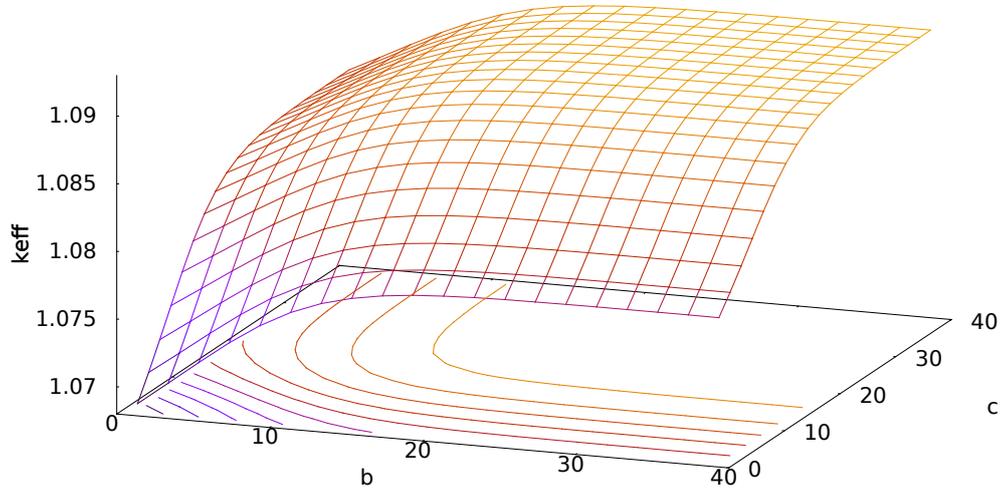


Figura 5.40 – Efectividad de reflectores bajo la forma de una superficie $k_{\text{eff}}(b, c)$ resolviendo un slab reflejado con volúmenes finitos

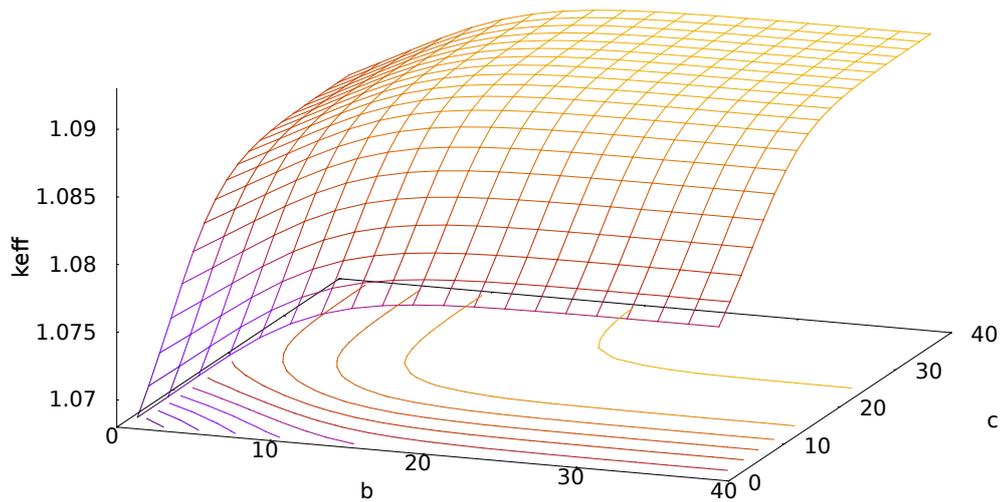


Figura 5.41 – Efectividad de reflectores bajo la forma de una superficie $k_{\text{eff}}(b, c)$ resolviendo un slab reflejado con elementos finitos

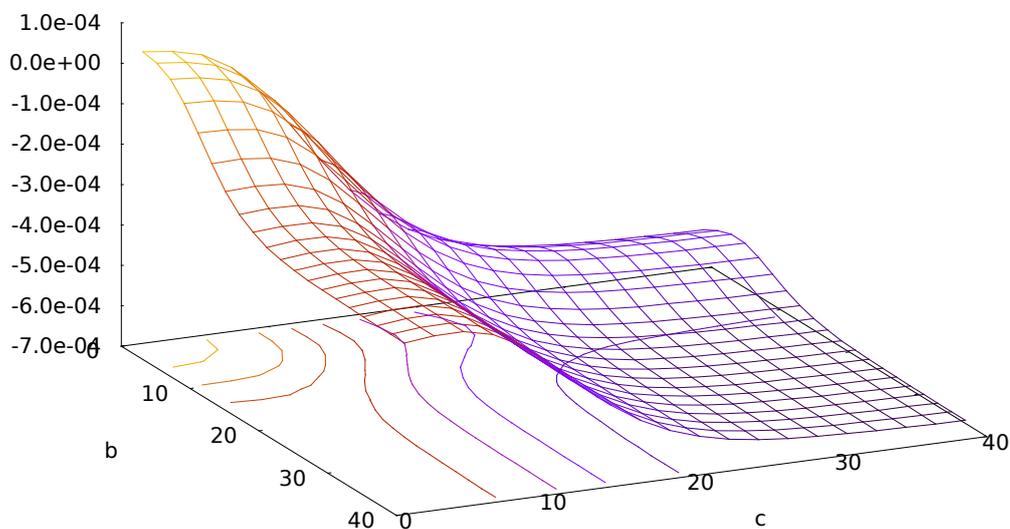


Figura 5.42 – Diferencia entre la efectividad de reflectores calculados con volúmenes y con elementos finitos.

```
b = 20; // long refl izquierdo
c = 15; // long refl derecho
x1 = 40; // centro del absorbente 1
w1 = 1; // ancho del absorbente 1
x2 = 70; // centro del absorbente 2
w2 = 6; // ancho del absorbente 2

Point(1) = { -b, 0, 0, lc };
Point(2) = { 0, 0, 0, lc };
Point(3) = { a, 0, 0, lc };
Point(4) = { a+c, 0, 0, lc };

Point(5) = { x1-0.5*w1, 0, 0, f1*lc };
Point(6) = { x1+0.5*w1, 0, 0, f1*lc };

Point(7) = { x2-0.5*w2, 0, 0, f2*lc };
Point(8) = { x2+0.5*w2, 0, 0, f2*lc };

Point(15) = { x1-w1, 0, 0, lc };
Point(16) = { x1+w1, 0, 0, lc };

Point(17) = { x2-w2, 0, 0, lc };
Point(18) = { x2+w2, 0, 0, lc };

Line(1) = { 1, 2 };
Line(2) = { 2, 15 };
Line(3) = { 15, 5 };
Line(4) = { 5, 6 };
Line(5) = { 6, 16 };
Line(6) = { 16, 17 };
```

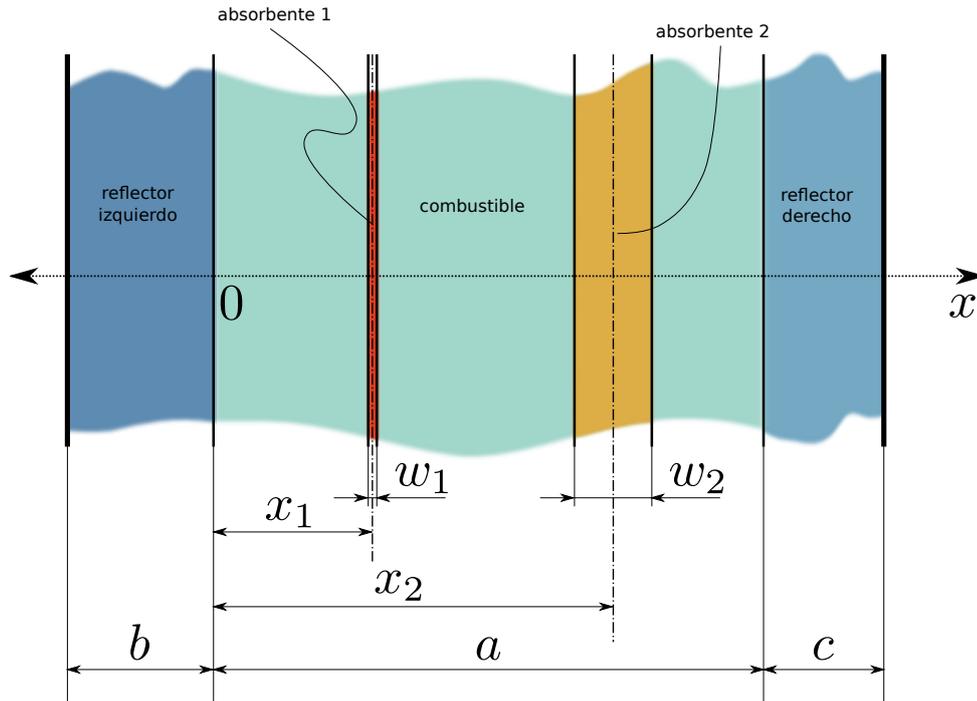


Figura 5.43 – Slab reflejado con absorbentes

```

Line(7) = {17, 7};
Line(8) = {7, 8};
Line(9) = {8, 18};
Line(10) = {18, 3};
Line(11) = {3, 4};

```

```

Physical Line("left_refl") = {1};
Physical Line("fuel") = {2, 3, 5, 6, 7, 9, 10};
Physical Line("abs1") = {4};
Physical Line("abs2") = {8};
Physical Line("right_refl") = {11};

```

```

Physical Point("left") = {1};
Physical Point("right") = {4};

```

La primer malla consiste en una discretización fina uniforme con $\ell_c = 0,02$, que tomamos como la solución referencia del problema:

slab-fino.geo

```

lc = 0.02;
f1 = 1;
f2 = 1;

Include "slab-abs.geo";

```

En la segunda, aumentamos el tamaño característico del elemento base $\ell_c = 3$ y se lo aplicamos a todos los puntos por igual haciendo $f_1 = f_2 = 1$:

slab-ref.geo



```
lc = 3;
f1 = 1;
f2 = 1;

Include "slab-abs.geo";
```

No obstante, debido a que ambos absorbentes está definidos por cuatro puntos

Finalmente, investigamos qué sucede si mantenemos $\ell_c = 3$ pero a la vez refinamos la malla alrededor de los absorbentes:

slab-refinado.geo

```
lc = 3;
f1 = 0.005;
f2 = 0.1;

Include "slab-abs.geo";
```

Para cada una de estos tres archivos de geometría, resolvemos el slab resultante con milonga utilizando primero volúmenes finitos y luego elementos finitos. Una vez realizados los seis cálculos, construimos una tabla en el archivo slab.txt que indica la reactividad estática y el factor de pico para cada caso, tal como mostramos en la terminal.

slab.was

```
# el archivo de malla lo leemos de la linea de comandos
MESH FILE_PATH $1 DIMENSIONS 1 DEGREES 2
SCHEME $2 # idem el esquema numerico

INCLUDE materials.was
MILONGA_SOLVE

a = 100

# como wasora no provee busqueda de maximo pero si de minimo ,
# buscamos donde esta el minimo de -phi_2(x)
# usamos como intervalo de busqueda [10,30]
xmax = func_min(-phi_2(x), x, 10, 30)

# el factor de pico se toma sobre el valor medio solo donde se
# genera potencia , que es en el intervalo [0,a]
f_p = phi_2(xmax)/(1/a * integral(phi_2(x), x, 0, a))

# mostramos en pantalla como resolvimos el problema
PRINT TEXT "$1_███" TEXT "$2_█" %.1f 1e5*(keff-1)/keff %.2f xmax %.3f f_p

# y escribimos la distribucino de flujo en un archivo
OUTPUT_FILE flujo $1-$2.dat
PRINT_FUNCTION phi_1 phi_2 FILE flujo
```

```
$ echo malla_____metodo_____rho_____xmax_____factor de pico > ←
    slab.txt
$ gmsht -v 0 -1 slab-fino.geo
```



```
$ milonga slab.was slab-fino.msh volumes >> slab.txt
$ milonga slab.was slab-fino.msh elements >> slab.txt
$ gmsh -v 0 -1 slab-sin-ref.geo
$ milonga slab.was slab-sin-ref.msh volumes >> slab.txt
$ milonga slab.was slab-sin-ref.msh elements >> slab.txt
$ gmsh -v 0 -1 slab-refinado.geo
$ milonga slab.was slab-refinado.msh volumes >> slab.txt
$ milonga slab.was slab-refinado.msh elements >> slab.txt
$ pyxplot slab-refinado.ppl
$ pyxplot slab-refinado-zoom.ppl
$ pyxplot slab-sin-ref.ppl
$ pyxplot slab-sin-ref-zoom.ppl
$ cat slab.txt
malla_____metodo_____rho_____xmax_____factor de pico
slab-fino.msh      volumes      -1163.3 19.72  1.239
slab-fino.msh      elements     -1163.5 19.74  1.239
slab-sin-ref.msh   volumes      -1032.0 18.00  1.270
slab-sin-ref.msh   elements     -1249.5 18.00  1.219
slab-refinado.msh  volumes      -1272.2 18.00  1.267
slab-refinado.msh  elements     -1115.7 21.00  1.224
$
```

Como vemos en la terminal, tanto el factor de multiplicación como el factor de pico calculado con volúmenes y con elementos prácticamente coinciden, con lo que tomamos a esta solución como la solución correcta. Notablemente, vemos que la reactividad estática calculada con elementos (volúmenes) usando la segunda malla con $\ell_c = 3$ uniforme tiene un error por exceso (defecto) con respecto a la solución de referencia, mientras que usando la tercera malla con $\ell_c = 3$ refinando alrededor de los absorbentes el error es por defecto (exceso). Sin embargo, el errores cometidos en el cálculo de f_p mantiene el signo para el mismo método para las diferentes mallas pero cambia de signo para la misma malla al cambiar de método (sic). Las figuras 5.44 y 5.45 comparan la solución de referencia con las obtenidas usando la segunda y la tercera, respectivamente. Hacemos un zoom alrededor del absorbente uno en las figuras 5.46 y 5.47, donde observamos que debido a que $w_1 < \ell_c$, existe un refinamiento implícito aún en la segunda malla. De las figuras y de la tabla que construimos en el archivo slab.txt mostrado en la terminal concluimos que la importancia del refinamiento selectivo es más patente en el cálculo del factor de multiplicación efectiva que para la distribución de flujo—y por ende del factor de pico—ya que los errores en k_{eff} disminuyen mucho más al refinar que los errores en f_p . Nuevamente, la extraña dependencia del signo del error de k_{eff} con la malla nos inducen a realizar en el futuro estudios más detallados sobre los métodos numéricos involucrados en la solución de la ecuación de difusión de neutrones.

5.1.4. Slab con distribución continua de propiedades

Para finalizar la sección sobre problemas unidimensionales, estudiamos cómo se comportan los métodos numéricos cuando las secciones eficaces, además de presentar discontinuidades debido a la aparición de diferentes materiales, varían continuamente dentro de un mismo material—que es el caso general que se presenta en problemas de interés ingenieril. Para ello, resolvemos un slab compuesto de un material fisil central con reflectores en los extremos pero con la particularidad de que las secciones eficaces macroscópicas del combustible dependen en forma continua de la posición x a través de distribuciones espaciales de propiedades. Un cálculo ingenieril debería incluir una geometría apropiada, distribuciones de propiedades como temperaturas, densidades, quemados del combustible, concentraciones de venenos, etc. adecuadas (caso que usualmente involucra cálculos no lineales) y secciones eficaces macroscó-

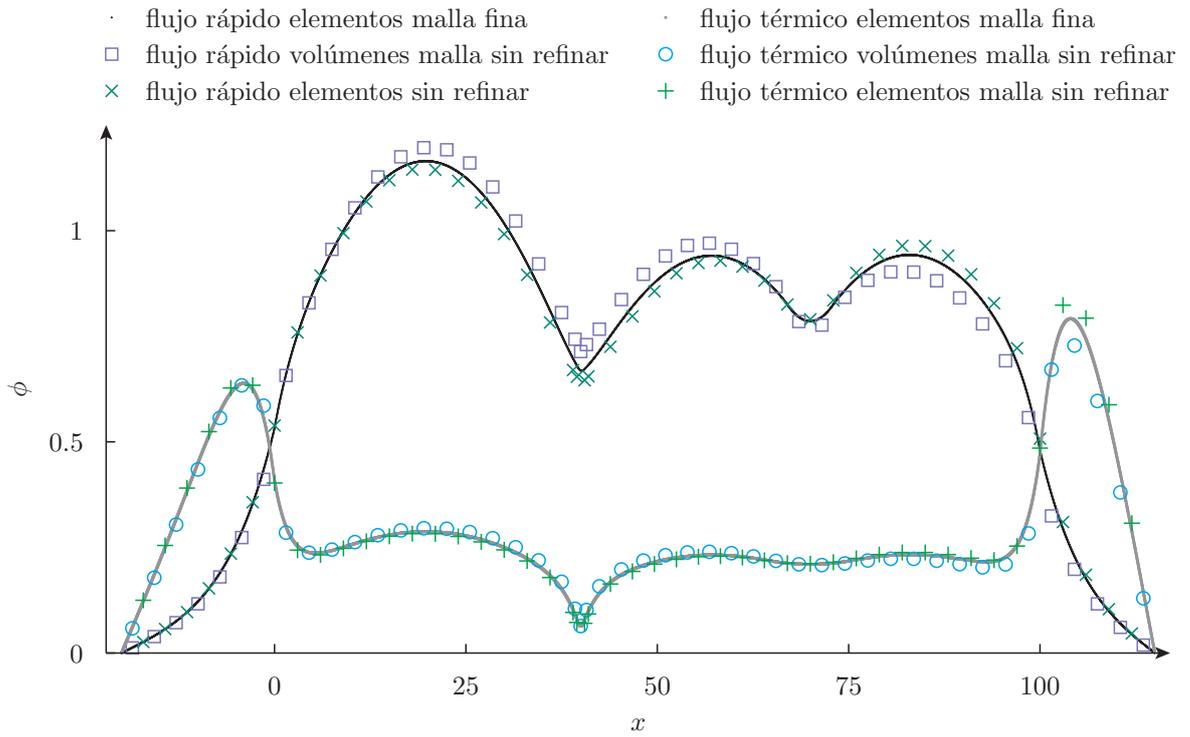


Figura 5.44 – Comparación entre la solución de referencia (línea llena) y las soluciones por volúmenes y por elementos sobre la malla gruesa sin refinar.

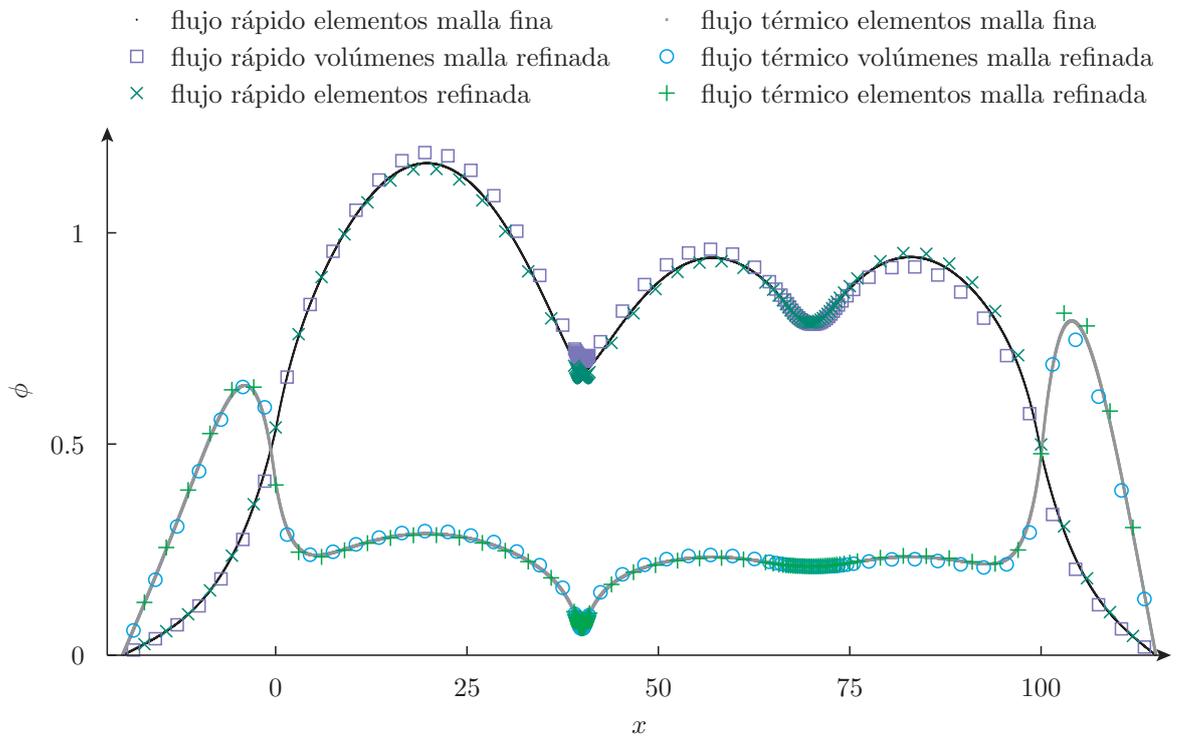


Figura 5.45 – Comparación entre la solución de referencia (línea llena) y las soluciones por volúmenes y por elementos sobre la malla gruesa refinada en los absorbentes.

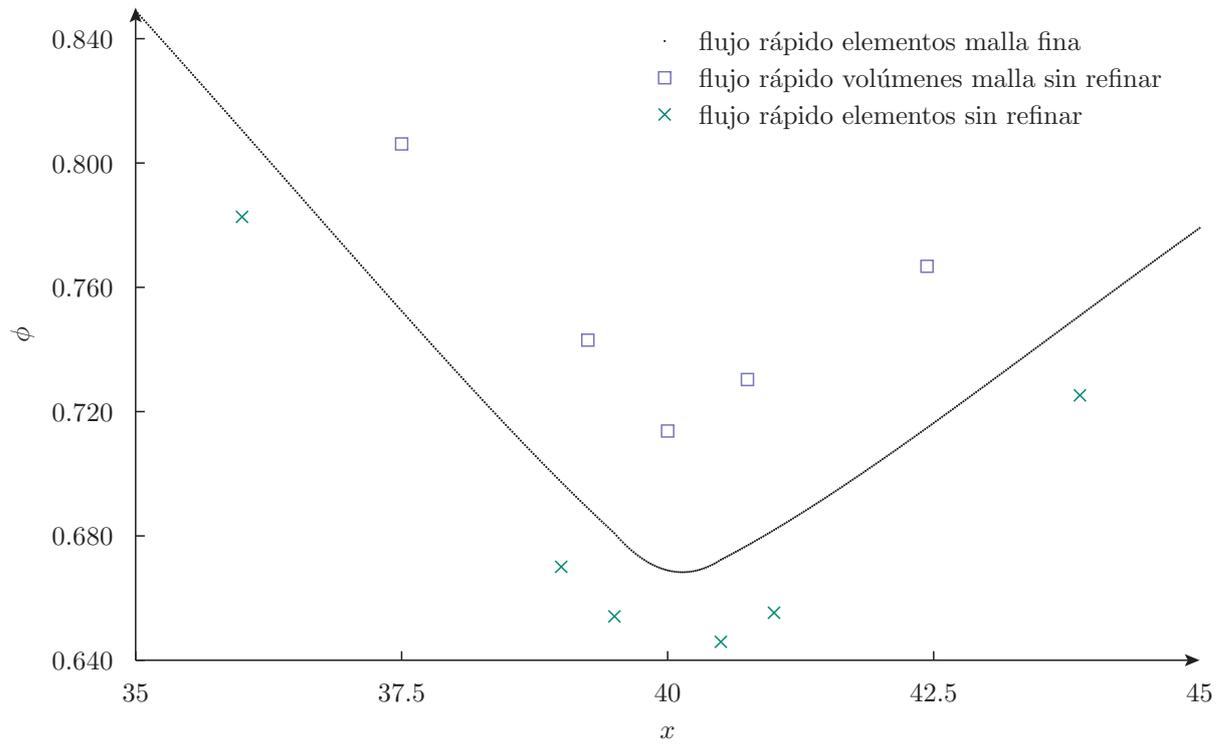


Figura 5.46 – Zoom de la figura 5.44 alrededor del absorbente uno

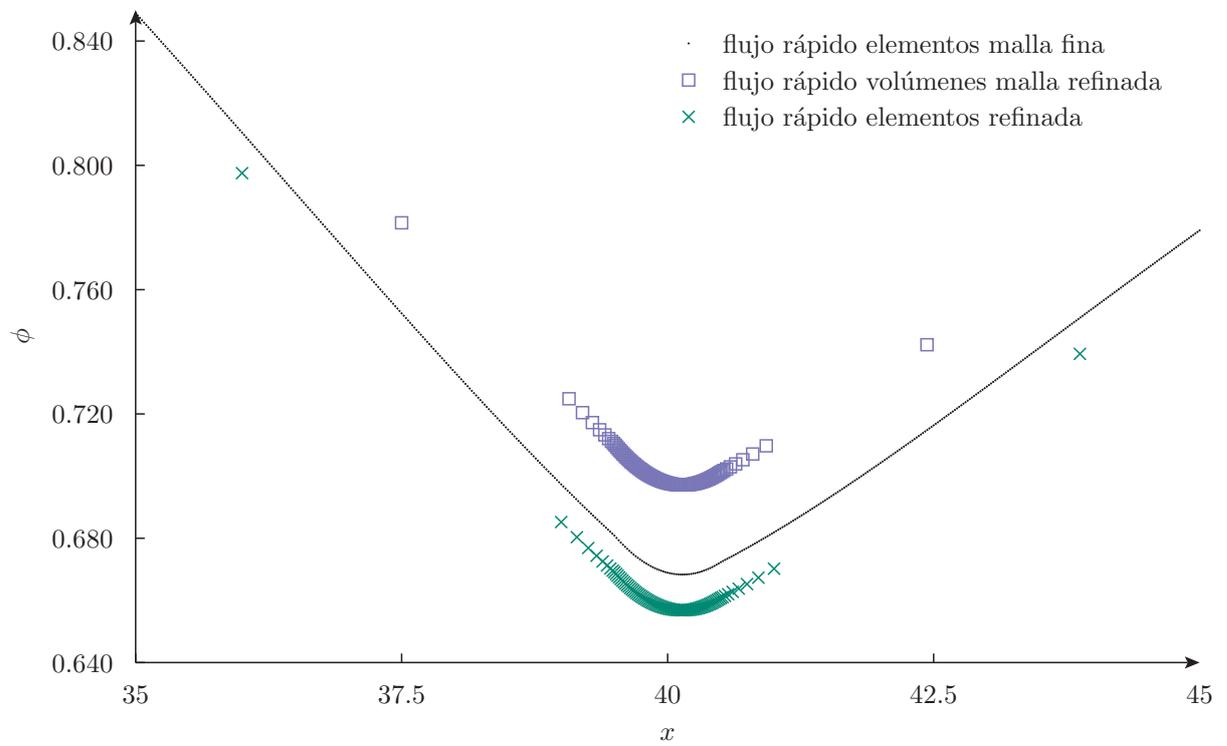


Figura 5.47 – Zoom de la figura 5.45 alrededor del absorbente uno

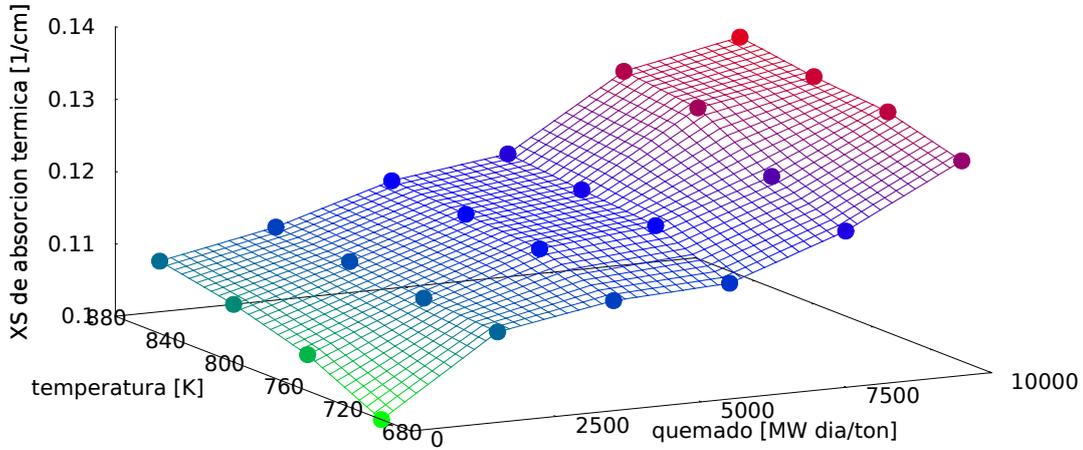


Figura 5.48 – Sección eficaz macroscópica de absorción en el grupo rápido en función del quemado y de la temperatura.

picas en función de dichas propiedades condensadas a partir de cálculos paramétricos realizados con códigos de neutrónicos de celda. En la resolución del problema que sigue, tanto la distribución de las propiedades como la forma en la que las secciones eficaces macroscópicas dependen de estas propiedades son completamente inventadas.

En particular, dejamos que las secciones eficaces del material fuel puedan depender de una cierta distribución de temperatura del combustible $T(x)$ y de una cierta distribución de quemado $q(x)$. A modo de ejemplo, inventamos que la sección eficaz $\Sigma_{a2}(q, T)$ depende del quemado y de la temperatura según una función definida por puntos (que podríamos haber calculado con un código de celda) que mostramos en la figura 5.48. Además, definimos—arbitrariamente—que podemos escribir $\nu\Sigma_{f2}(q, T)$ como

$$\nu\Sigma_{f2}(q, T) = \nu\Sigma_{f0}(q) - c_T (\sqrt{T} - \sqrt{T_0})$$

donde $\nu\Sigma_{f0}(q)$ la función definida por puntos (que también podríamos haber calculado con un código de celda) que mostramos en la figura 5.49, c es un coeficiente con unidades de longitud inversa por temperatura a la menos un medio, y T_0 es la temperatura a la cual corresponde la función $\nu\Sigma_{f0}(q)$. Por simplicidad, mantenemos el resto de las secciones eficaces del combustible constantes e independiente de x . Podemos implementar estas dependencias en milonga con el siguiente archivo que define los materiales y sus secciones eficaces que, como es de esperar, presenta una complejidad superior a las definiciones que estuvimos utilizando hasta el momento:

xs.was

```
# tabla de XS de fision en funcion del quemado a T0
# el quemado esta en MW-dia/tonU y la XS en 1/cm
FUNCTION nusigmaf0(q) INTERPOLATION akima DATA {
```

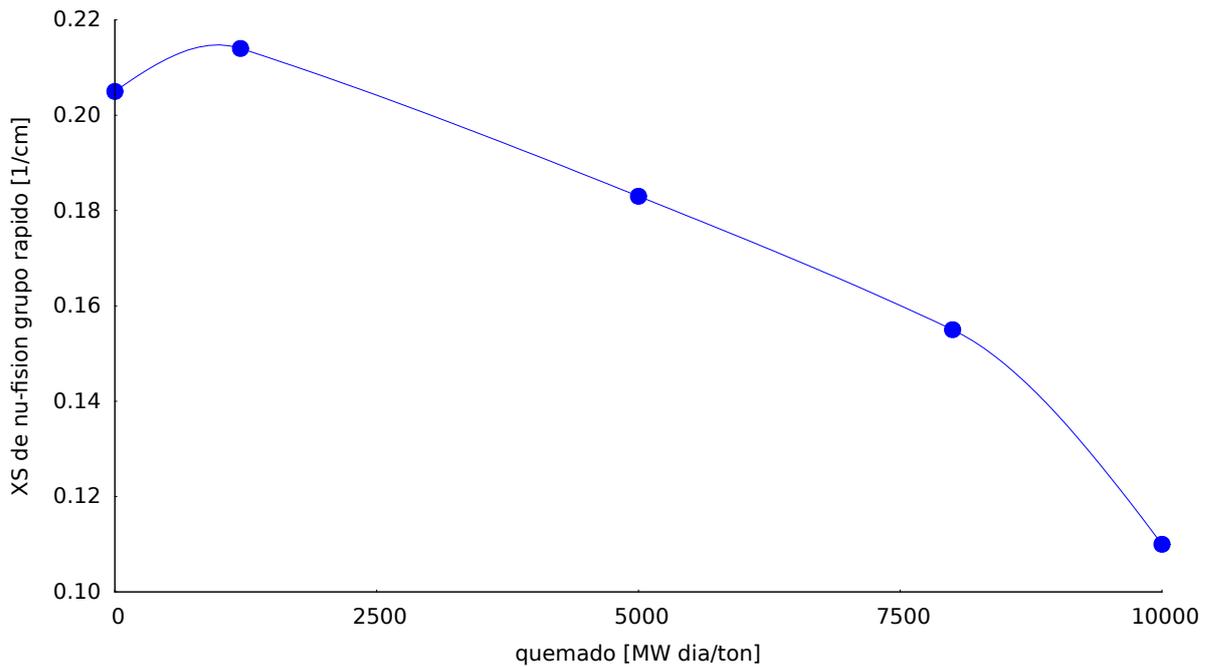


Figura 5.49 – Sección eficaz macroscópica de ν -fisiones en el grupo rápido en función del quemado para una temperatura constante igual a 700 K.

```

0      0.205
1200   0.214
5000   0.183
8000   0.155
10000  0.110
}

# temperatura central en kelvin
T0 = 700
# coeficiente de variacion con la temperatura
cT = 3.5e-3

# tabla de XS de absorcion en funcion del quemado
# y de la temperatura dada en un archivo separado
FUNCTION sigmaa(q,T) FILE_PATH sigmaa.xls INTERPOLATION rectangle

# definicion de materiales y sus XS en 1/cm
MATERIAL fuel {
  D_1      1.5
  D_2      0.4
  SigmaS_1->2 0.02
  SigmaA_1  0.01
  SigmaA_2  sigmaa(quem(x),temp(x))
  nuSigmaF_2 nusigmaf0(quem(x))-cT*(sqrt(temp(x))-sqrt(T0))
}

MATERIAL left_refl {
  D_1      2.0
  D_2      0.3
  SigmaS_1->2 0.04
  SigmaA_1  0.00
  SigmaA_2  0.01

```



```

}

MATERIAL right_refl {
  D_1      2.0
  D_2      0.3
  SigmaS_1->2 0.05
  SigmaA_1 0.00
  SigmaA_2 0.005
}

# generamos los datos para generar las figuras de SigmaA_2 y de
# nuSigmaF_2 con estas instrucciones pero no las necesitamos
# OUTPUT_FILE      absorcion sigmaa-cont.xls
# PRINT_FUNCTION FILE absorcion sigmaa      MIN 0 700 MAX 10000 850 STEP 200 5
#
# OUTPUT_FILE      fision      fision-cont.xls
# PRINT_FUNCTION FILE fision      nusigmaf0 MIN 0      MAX 10000      STEP 100

```

Así definidas, las secciones eficaces Σ_{a2} y $\nu\Sigma_{f2}$ del material fuel dependen de la coordenada espacial x —que es la única dependencia que pueden tener en un problema unidimensional— a través de las funciones intermedias $quem(x)$ y $temp(x)^2$ que deben ser definidas en algún lugar del archivo de entrada de milonga. Para analizar por separado el efecto de incorporar propiedades continuas a secciones eficaces que dependen de dichas propiedades, tomamos cuatro casos. El primero, que llamamos el caso base consiste en distribuciones uniformes tanto de quemado como de temperatura:

$$\begin{aligned}
quem_{\text{base}}(x) &= 0 \\
temp_{\text{base}}(x) &= 700 \text{ K}
\end{aligned}$$

y que implementamos en un archivo de milonga separado como

base.was

```

# caso base: ponemos quemado igual a cero y temperatura
# igual a 700K para cualquier x
quem(x) := 0
temp(x) := 700

```

En el segundo, damos una distribución senoidal de temperatura manteniendo quemado cero en todo el núcleo:

$$\begin{aligned}
quem_{\text{temp}}(x) &= 0 \\
temp_{\text{temp}}(x) &= 700 \text{ K} + 150 \text{ K} \cdot \sin\left(\frac{x}{a} \cdot \pi\right)
\end{aligned}$$

temperatura.was

```

# caso con temperatura solamente

```

²Estas funciones no pueden llamarse $q(x)$ y $T(x)$ porque los identificadores q y T ya fueron usados como variables.



```
# quemado cero
quem(x) := 0

# una distribucion de temperatura senoidal
# a partir de 700K con un pico de 850K
temp(x) := 700 + 150*sin(x/a*pi)
```

En el tercero, mantenemos la uniforme de temperatura pero generamos una distribución de quemados interpolando linealmente diez valores generados al azar distribuidos uniformemente en el intervalo $[0, a]$:

quemado.was

```
# caso con quemado solamente

# temperatura uniforme
temp(x) := 700

# quemado random (pero con semilla para hacerlo repetitivo)
# interpolando diez valores entre 0 y 9000
NUMBER N 10
VECTOR vecx N
VECTOR vecbu N

vecx(i) = (i-1/2)*a/N
vecbu(i) = random(0, 9000, 12345+123*i)

FUNCTION quem(x) VECTORS vecx vecbu
```

Finalmente, en el cuarto damos ambas distribuciones no triviales:

quemado-temp.was

```
# caso con quemado y con temperatura

# quemado random (pero con semilla para hacerlo repetitivo)
# interpolando diez valores entre 0 y 9000
NUMBER N 10
VECTOR vecx N
VECTOR vecbu N

vecx(i) = (i-1/2)*a/N
vecbu(i) = random(0, 9000, 12345+123*i)

FUNCTION quem(x) VECTORS vecx vecbu

# una distribucion de temperatura senoidal
# a partir de 700K con un pico de 850K
temp(x) := 700 + 150*sin(x*pi/a)
```

Debemos notar que si las secciones eficaces dependiesen de x indirectamente a través del flujo—como por ejemplo debería ser el caso de la temperatura del combustible, que depende de la potencia local que a su vez depende de $\phi(x)$ —entonces la formulación del problema dejaría



de ser lineal y tendríamos que resolverlo en forma iterativa. Como este tema está fuera del alcance de este trabajo (e incluso del alcance del curso de posgrado para el cual lo escribimos), todas las secciones eficaces inventadas son funciones explícitas de la posición espacial.

La geometría consiste en un slab unidimensional con un combustible central de ancho a , reflejado a la izquierda por un material reflector de ancho b y a la derecha por otro de ancho c , que ya definimos en la sección anterior. Dado que ahora estamos especificando que las secciones eficaces están en unidades de cm^{-1} , entonces las longitudes deben tener dimensiones de centímetros. El archivo de geometría que usamos en este caso es:

reflected.geo

```
// todas las longitudes estan en centimetros
a = 100;
b = 20;
c = 15;
lc = 2;

Point(1) = { -b, 0, 0, lc };
Point(2) = { 0, 0, 0, lc };
Point(3) = { a, 0, 0, lc };
Point(4) = { a+c, 0, 0, lc };

Line(1) = {1, 2};
Line(2) = {2, 3};
Line(3) = {3, 4};

Physical Line("left_refl") = {1};
Physical Line("fuel") = {2};
Physical Line("right_refl") = {3};

Physical Point("left") = {1};
Physical Point("right") = {4};
```

Para poder resolver cada uno de los cuatro casos con diferentes combinaciones de distribuciones de quemado y temperatura, incluimos desde el archivo de entrada principal el archivo de entrada secundario que indique el usuario en la línea de comando. De esta manera, al igual que venimos haciendo para poder utilizar tanto volúmenes como elementos finitos con el mismo input, podemos resolver los cuatro casos con ambos métodos con el mismo archivo de entrada principal:

slab.was

```
SCHEME $2

# el ancho del combustible
a = 100
b = 20
c = 15

# incluimos el archivo que define las distribuciones de quemado
# y de temperatura segun el argumento de la linea de comandos
INCLUDE $1.was

# las secciones eficaces son independientes de las distribuciones!
MESH DIMENSIONS 1 DEGREES 2 FILE_PATH reflected.msh
```



```
INCLUDE xs.was

MILONGA_SOLVE
PRINT %.8f keff

# escribimos en un archivo las distribuciones de temperatura y quemado
# y las secciones eficaces resultantes
OUTPUT_FILE xs xs-$1.dat
PRINT_FUNCTION FILE xs SigmaA_2 nuSigmaF_2 temp quem MIN -b MAX a+c STEP 1

# escribimos en otro archivo los flujos
# el resultado ahora depende del metodo de solucion
OUTPUT_FILE flujo $1-$2.dat
PRINT_FUNCTION FILE flujo phi_1 phi_2
```

```
$ gmsh -2 -v 0 reflected.geo
$ milonga slab.was base volumes
1.32219423
$ milonga slab.was base elements
1.32288864
$ milonga slab.was temperatura volumes
1.18308252
$ milonga slab.was temperatura elements
1.18467515
$ milonga slab.was quemado volumes
1.06928033
$ milonga slab.was quemado elements
1.07048909
$ milonga slab.was quemado-temp volumes
0.99862399
$ milonga slab.was quemado-temp elements
1.00067722
$ pyxplot quem.ppl
$ pyxplot temp.ppl
$ pyxplot sigmaa.ppl
$ pyxplot nusigmaf.ppl
$ pyxplot fast-volumes.ppl
$ pyxplot thermal-volumes.ppl
$ pyxplot fast-elements.ppl
$ pyxplot thermal-elements.ppl
$
```

Las figuras 5.50–5.53 muestran las distribuciones de temperatura y quemado en el núcleo y las secciones eficaces resultantes en todo el slab para cada uno de los cuatro casos. En las figuras 5.54 y 5.55 mostramos los flujos resultantes utilizando volúmenes finitos y en 5.56 y 5.57 los obtenidos con elementos finitos para las distribuciones de propiedades y las dependencias de las secciones eficaces que inventamos, de todas las posibles que podríamos llegar a haber elegido. Está claro que existe una gran cantidad de combinaciones de distribuciones y de secciones eficaces que podríamos estudiar, y para cada una de ella habría también una gran cantidad de resultados que podríamos obtener, los que a su vez nos harían llegar a otra gran cantidad de conclusiones.

En general, vemos que tanto el método de volúmenes finitos como de elementos finitos son capaces de resolver en forma más o menos satisfactoria todos los casos que estudiamos, con tal de que la malla sea razonablemente refinada. En general vemos que las soluciones encontradas con elementos finitos están más cercanas a las soluciones de referencia—sean éstas analíticas

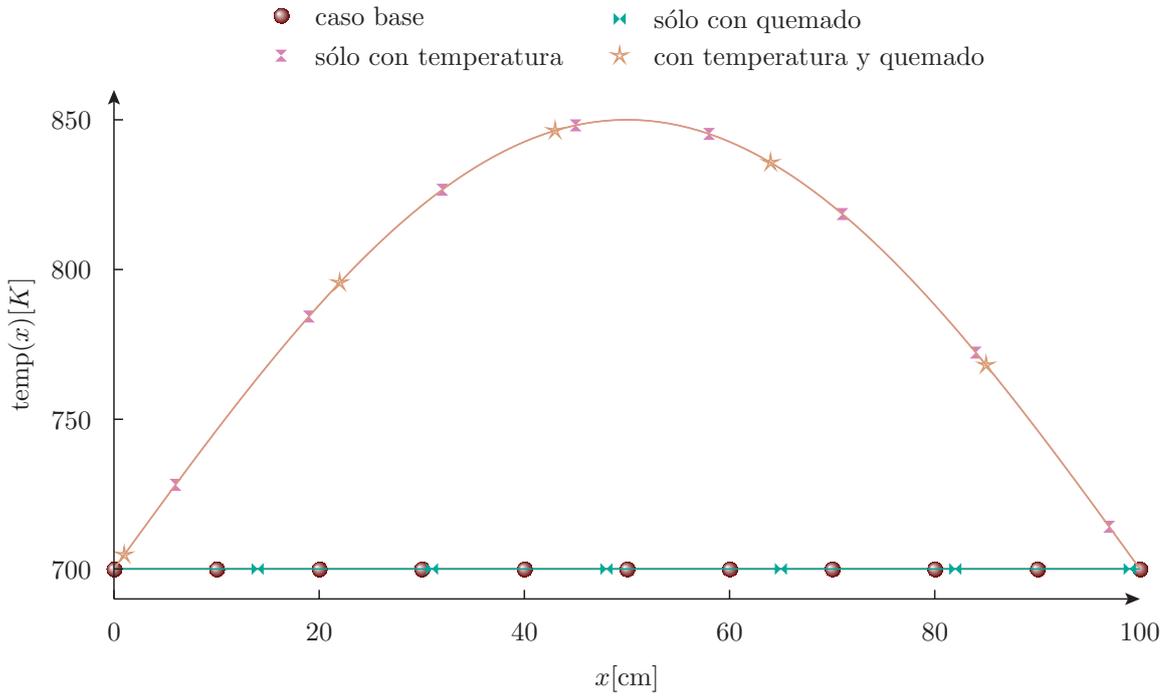


Figura 5.50 – Distribuciones de temperaturas en el núcleo

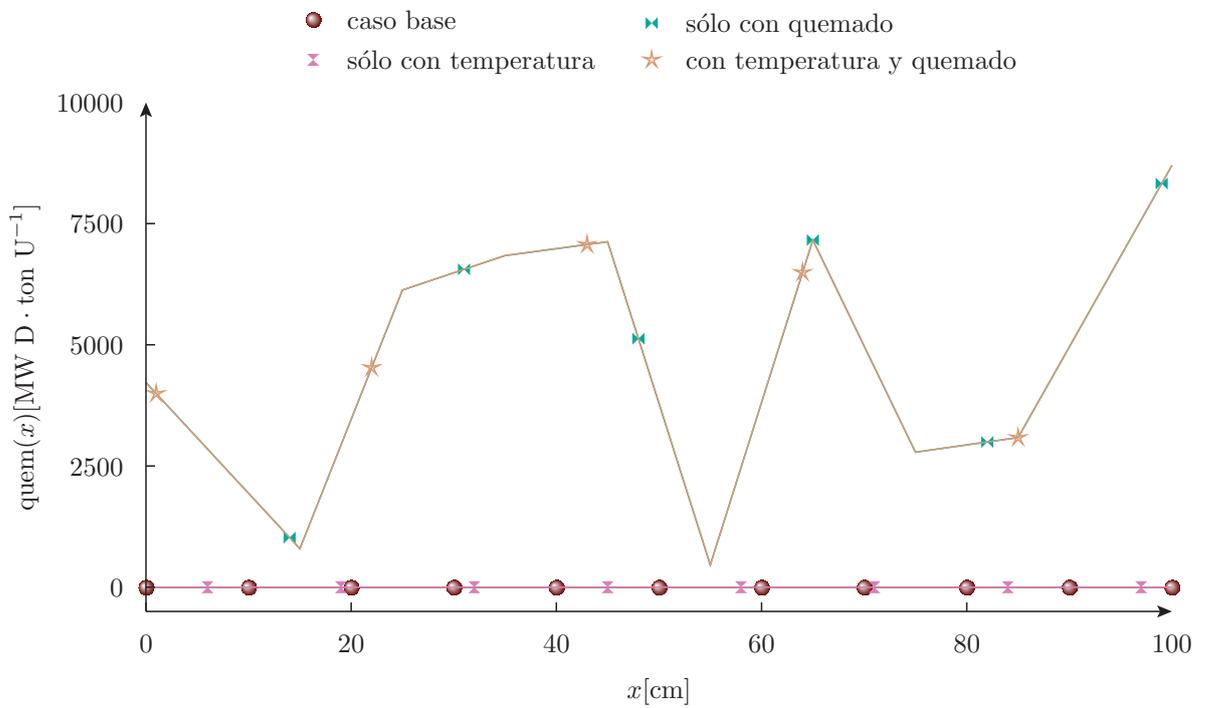


Figura 5.51 – Distribuciones de quemados en el núcleo

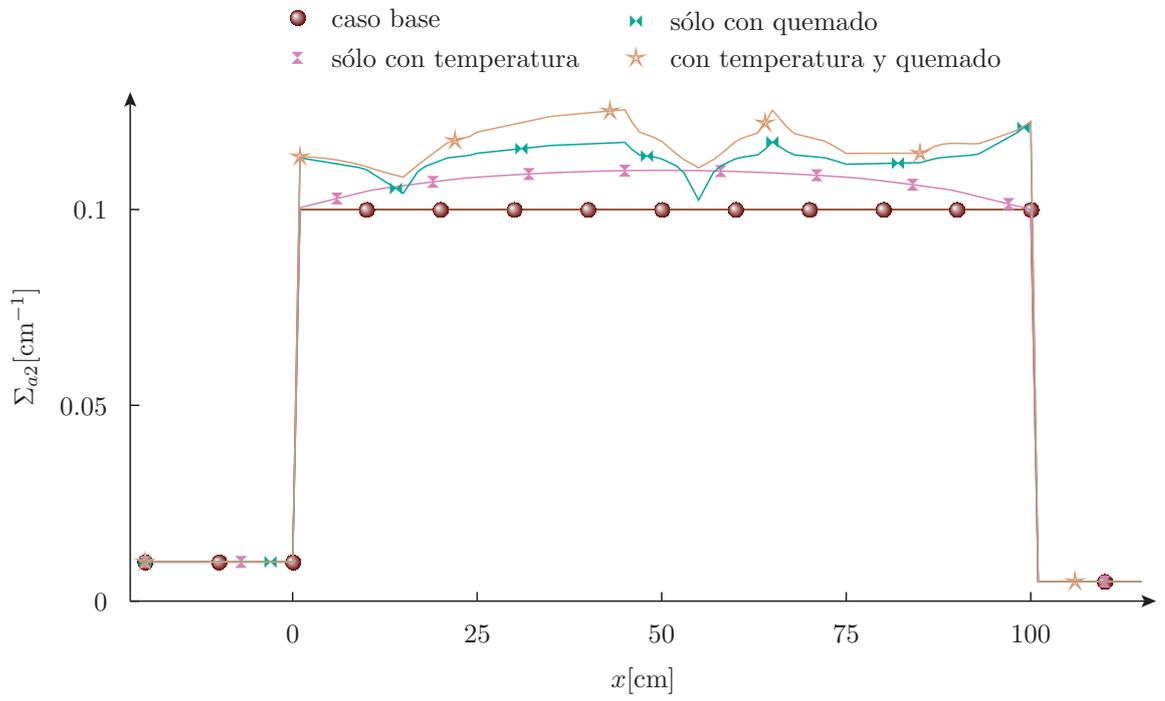


Figura 5.52 – Sección eficaz macroscópica de absorción del grupo térmico

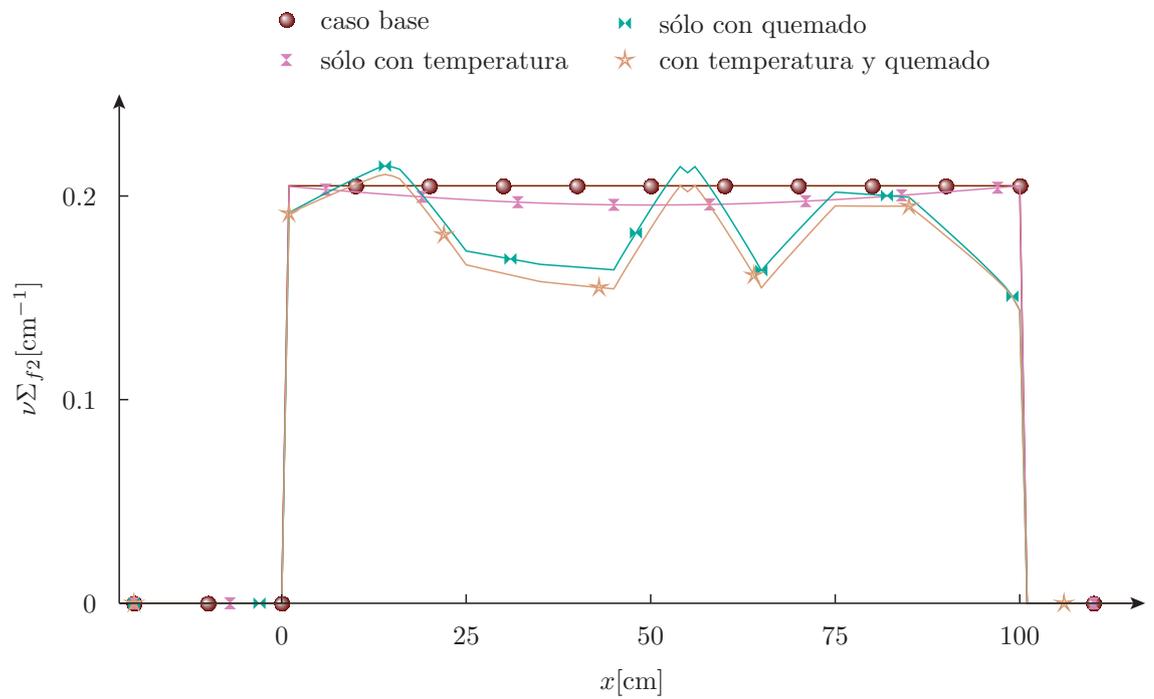


Figura 5.53 – Sección eficaz macroscópica de ν -fisiones del grupo térmico

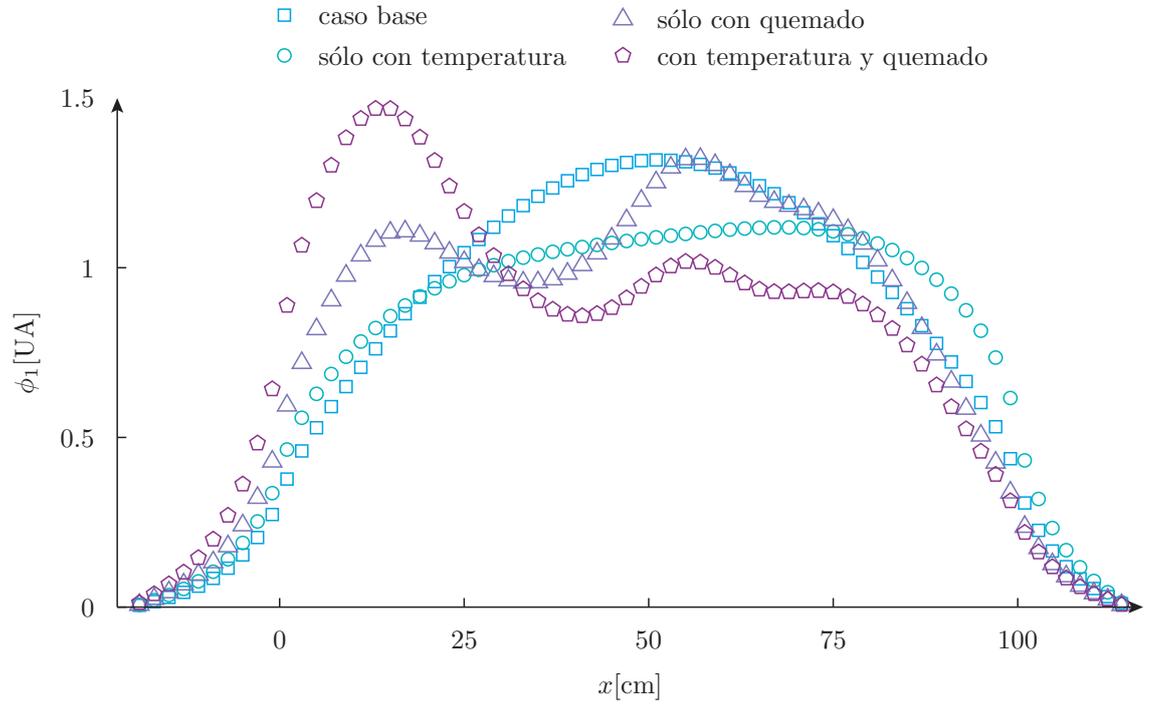


Figura 5.54 – Flujos rápidos resueltos con volúmenes finitos

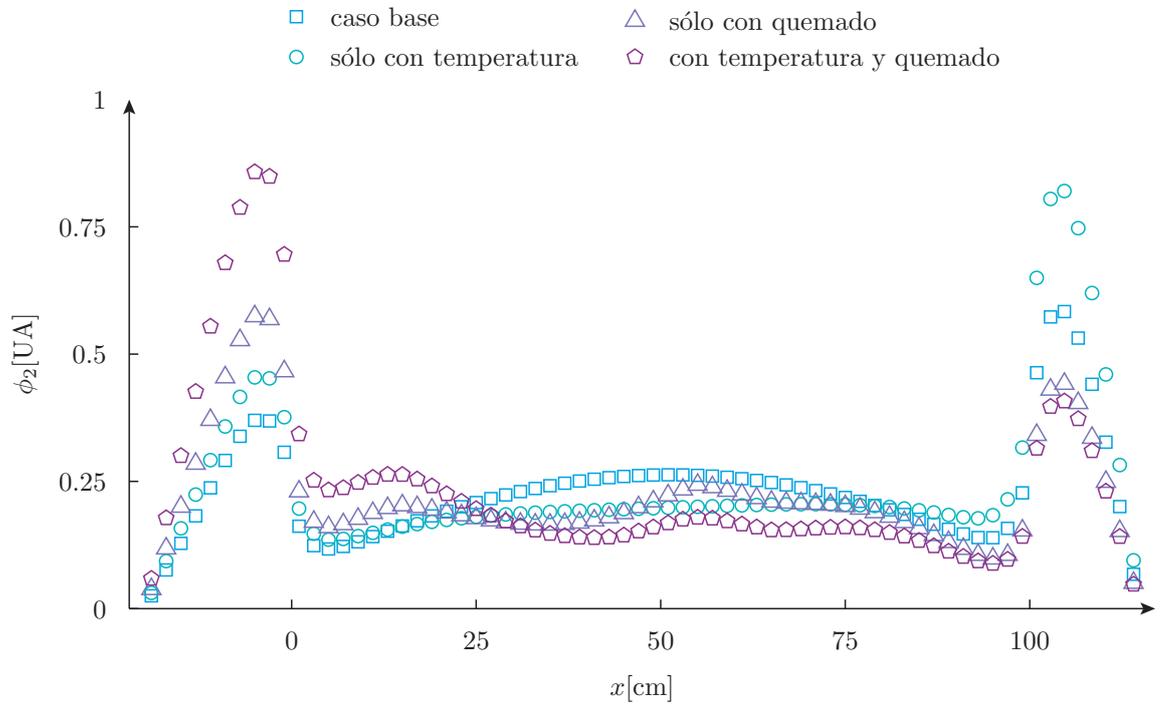


Figura 5.55 – Flujos térmicos resueltos con volúmenes finitos

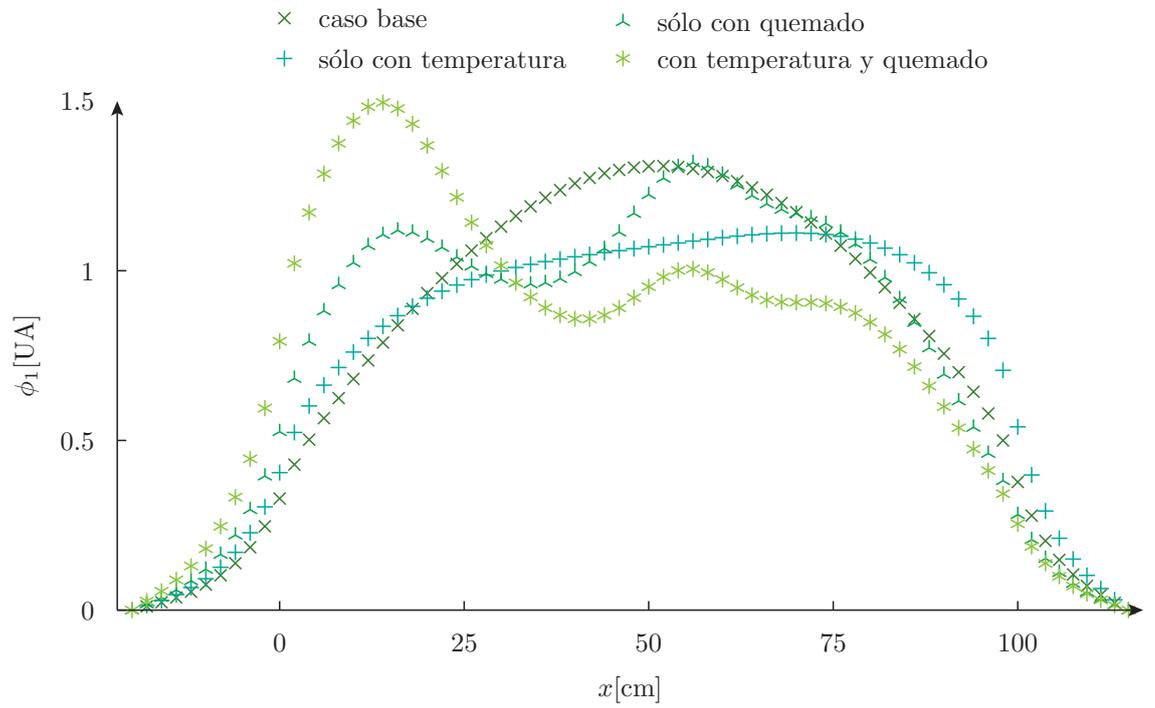


Figura 5.56 – Flujos rápidos resueltos con elementos finitos

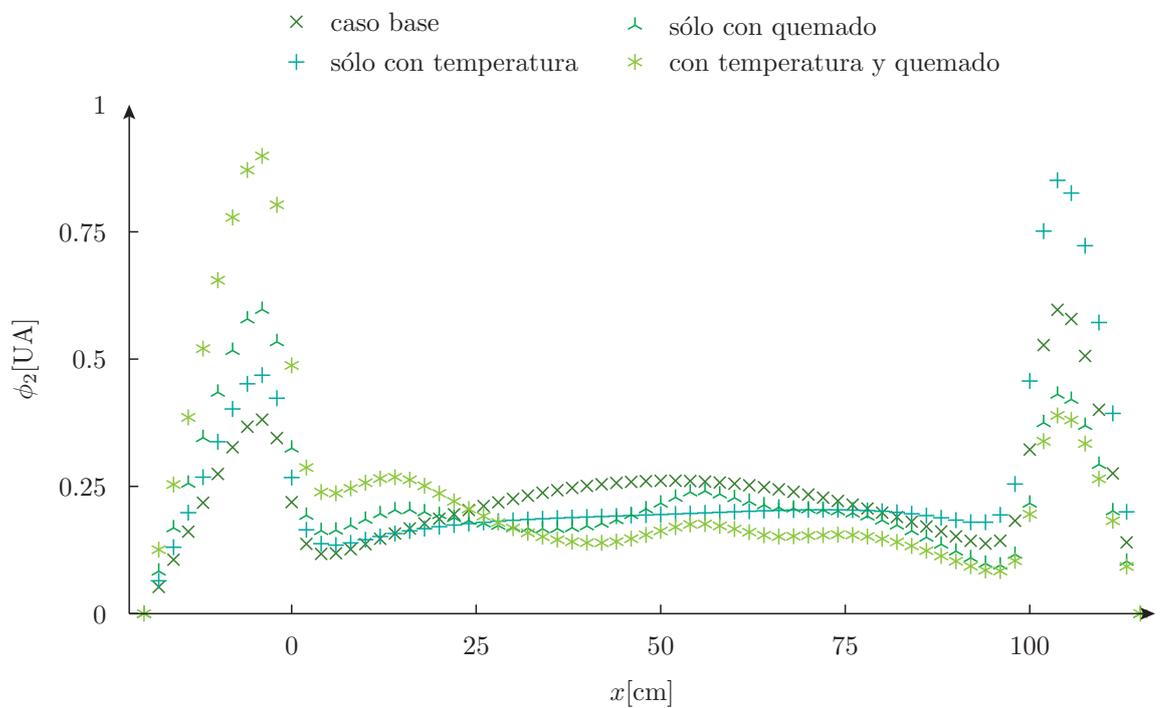


Figura 5.57 – Flujos térmicos resueltos con elementos finitos



u obtenidas con mallas extremadamente finas—que las que arroja el método de volúmenes finitos. Como los problemas unidimensionales no presentan una complejidad computacional suficientemente grande como para tener que decidir qué método seleccionar para resolver un cierto caso en base a limitaciones de tiempo de ejecución o de almacenamiento de datos, por lo que concluimos que el método de elementos finitos es ligeramente superior al de volúmenes finitos en una dimensión.

Notamos finalmente que aún en geometrías sencillas tipo slab, es posible observar efectos como los hombros térmicos o las discontinuidades en las derivadas del flujo que también aparecen en casos reales. De todas maneras, estos fenómenos son característicos de formulaciones a varios grupos y de secciones eficaces no uniformes, problemas que no es posible resolver analíticamente y que requieren necesariamente la aplicación de métodos numéricos. Es por eso que si bien resolver los primeros problemas de slabs con elementos finitos parecía una demasía, estos últimos problemas que involucran secciones eficaces que dependen de la posición tanto continuamente a través de propiedades como discontinuamente con cambios de materiales muestran que para observar aún en forma simplificada los efectos que realmente interesan en problemas tridimensionales necesitamos la aplicación de esquemas numéricos de discretización espacial tales como volúmenes o elementos finitos. Sin embargo, aunque los detalles ingenieriles están fuera del objetivo tanto de este capítulo como del trabajo, en problemas de aplicación industrial puede suceder de que dependiendo de la forma en la que se condensen y homogeneicen las secciones eficaces en los códigos de celda, los esquemas en volúmenes arrojen mejores resultados al utilizar también celdas en los cálculos de núcleo.

5.2. Problemas en dos dimensiones

Pasamos ahora a estudiar problemas en dos dimensiones, donde las diferencias entre las soluciones obtenidas utilizando esquemas espaciales basados en volúmenes finitos y en elementos finitos por el otro se hacen más patentes. Estos problemas introducen unas complejidades que son también características de problemas tridimensionales, mientras que al mismo tiempo las soluciones son más fáciles de estudiar y analizar. Ciertamente, los casos bidimensionales son más desafiantes—y divertidos—que los slabs ya los métodos numéricos son mandatorios.

5.2.1. Círculo desnudo homogéneo

El primer problema bidimensional que resolvemos consiste en un reactor circular de radio a a un grupo de energías con secciones eficaces uniformes (figura 5.58) con condición de contorno de flujo nulo en $r = \sqrt{x^2 + y^2} = a$, que es el caso bidimensional más sencillo que existe. Nos detenemos brevemente en este problema primero porque tiene solución analítica con lo que podemos evaluar y comparar no sólo los errores cometidos por los dos esquemas numéricos sino también como cambian estos errores con respecto al caso unidimensional de la sección 5.1.1. Además, la sencillez de la solución nos permite introducir y comparar la forma de reportar distribuciones multidimensionales en celdas y en nodos, que a partir de ahora realizamos con las facilidades de post-procesamiento de gmsh.

El factor de multiplicación efectivo analítico es

$$k_{\text{eff}} = \frac{\nu \Sigma_{f1}}{\Sigma_{a1} + D_1 \cdot \left(\frac{\nu_0}{a}\right)^2} \quad (5.10)$$

y la distribución de flujo normalizada de forma tal que el valor medio sea igual a la unidad resulta

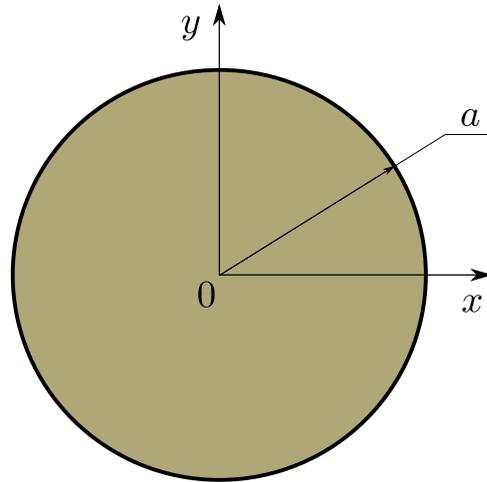


Figura 5.58 – Círculo bidimensional desnudo con propiedades homogéneas

$$\phi_1(r) = \frac{\nu_0^2}{2 \int_0^{\nu_0} J_0(r) \cdot r \, dr} \cdot J_0\left(\nu_0 \cdot \frac{r}{a}\right) \quad (5.11)$$

donde $J_0(r)$ es la función cilíndrica de Bessel de orden cero, $\nu_0 \approx 2,4048 \dots$ es su primer cero. Tanto el valor de ν_0 como la integral de normalización pueden ser calculados numéricamente desde el input de milonga.

5.2.1.1. Distribución de flujo

Lo primero que hacemos con el reactor circular desnudo uniforme a un grupo de energías de radio a es calcular la distribución de flujo tanto con volúmenes como con elementos finitos y compararla con la distribución analítica dada por la ecuación (5.11). Además, dado que insume sólo una línea de entrada, calculamos el k_{eff} analítico dado por la ecuación (5.10) y lo comparamos con el factor de multiplicación hallado numéricamente.

En dos dimensiones, no sólo debemos elegir el método de discretización espacial sino también la forma base de las celdas/elementos. Éstos pueden ser triángulos o cuadrángulos, que pueden aparecer mezclados para aproximar mejor una cierta geometría continua. Para estudiar cómo se comportan los métodos numéricos en dos dimensiones, resolvemos entonces el reactor circular tanto con triángulos como con cuadrángulos como figura base, mateniendo fija la longitud característica ℓ_c de la malla.

Para mallar una superficie bidimensional, el código gmsh siempre comienza aplicando algún algoritmo de mallado que genera triángulos. En caso de requerir cuadrángulos, éstos son recombinados utilizando otro algoritmo. Las combinaciones forma de celda/element (triángulo o cuadrángulo) y de esquema de discretización espacial (volúmenes o elementos) son cuatro. Para poder utilizar el mismo input de milonga para resolver los cuatro casos, seleccionamos el método y la forma básica desde la línea de comandos como venimos haciendo desde la sección anterior. Pero además, al resolver este problema ilustramos una manera de incorporar tanto la geometría continua como las instrucciones para generar figuras directamente en el input de milonga al generar todos los archivos necesarios con instrucciones PRINT. Es por eso que el input que usamos es particularmente largo, pero sólo para mostrar la flexibilidad del código.

Fijamos $a = 100$ y $\ell_c = 20$, generamos el archivo de geometría con esta información, llamamos a gmsh y resolvemos el problema numérico. Además, calculamos el k_{eff} analítico con



la ecuación (5.10) y definimos una función $\text{flux}(x, y)$ con la distribución espacial del flujo dada por la ecuación (5.11). Como esta distribución está normalizada, podemos compararla con la solución numérica $\phi_1(x, y)$ para generar una función $\text{error}(x, y)$. Las salidas consisten en los dos factores de multiplicación y su diferencia por la salida estándar y la generación de cuatro figuras para cada corrida: dos con gnuplot y dos con gmsh en modo post-procesamiento. La primera consiste en una superficie tridimensional que muestra los valores calculados del flujo en los centros de las celdas o en los nodos, según corresponda. La segunda es una evaluación del flujo numérico sobre el rango $[-a : a] \times [-a : a]$ con incrementos $\Delta x = \Delta y = a/400$, forzando de esta manera a milonga a interpolar la solución en puntos arbitrarios, utilizando para ello las funciones de forma discutidas en el capítulo 4 en caso de haber seleccionado el método de elementos finitos. Las últimas figuras consisten en dos vistas de post-procesamiento de gmsh del flujo numérico y del error cometido con respecto a la solución analítica.

flujo.was

```

SCHEME $1

a = 100 # radio del circulo
lc = 20 # longitud caracteristica

# en lugar de usar un template, directamente escribimos todo el
# archivo de geometria completo
OUTPUT_FILE geo circulo.geo
PRINT FILE geo TEXT "a_=" %g a TEXT ";" SEPARATOR " "
PRINT FILE geo TEXT "lc_=" %g lc TEXT ";" SEPARATOR " "
PRINT FILE geo TEXT "Mesh.Algorithm_=" 8;
PRINT FILE geo TEXT "Point(1)_=" "{0,0,lc}";
PRINT FILE geo TEXT "Point(2)_=" "{a,0,lc}";
PRINT FILE geo TEXT "Point(3)_=" "{0,a,lc}";
PRINT FILE geo TEXT "Point(4)_=" "{-a,0,lc}";
PRINT FILE geo TEXT "Point(5)_=" "{0,-a,lc}";
PRINT FILE geo TEXT "Circle(1)_=" "{2,1,3}";
PRINT FILE geo TEXT "Circle(2)_=" "{3,1,4}";
PRINT FILE geo TEXT "Circle(3)_=" "{4,1,5}";
PRINT FILE geo TEXT "Circle(4)_=" "{5,1,2}";
PRINT FILE geo TEXT "Line_Loop(5)_=" "{2,3,4,1}";
PRINT FILE geo TEXT "Plane_Surface(6)_=" "{5}";
PRINT FILE geo TEXT "Physical_Line(\"bc\")_=" "{1,2,3,4}";
PRINT FILE geo TEXT "Physical_Surface(\"fuel\")_=" "{6}";
# si el segundo argumento es quad, indicamos a gmsh que queremos
# que recombine los triangulos para obtener cuadrangulos
# (todavia no esta implementado el manejo de strings en wasora)
SHELL "if [ "$2" = "quad" ]; then echo "Mesh.RecombineAll_=" 1; \>>_ <-
    circulo.geo;_fi"

# mallamos
SHELL "gmsh_-v_0_-2_circulo.geo"
MESH DIMENSIONS 2 DEGREES 1 FILE_PATH circulo.msh

# un unico material a un grupo de energias
MATERIAL fuel D 1 nuSigmaF 0.1 SigmaA 0.101

# resolvemos el problema numerico
MILONGA SOLVE

# y el problema analitico
VAR r
nu0 = root(j0(r), r, 0, 3) # la primer raiz de j0(x)

```

```

k = nuSigmaF(0,0)/(SigmaA(0,0) + D(0,0)*(nu0/a)^2)

f = nu0^2/(2*integral(j0(r)*r, r, 0, nu0)) # factor de normalizacion
flux(x,y) := f * j0(nu0 * sqrt(x^2+y^2)/a) # distribucion de flujo
error(x,y) := phi_1(x,y) - flux(x,y) # error del metodo numerico

# escribimos en pantalla los keffs
PRINT %.8f TEXT "keff_analitico_=" k
PRINT %.8f TEXT "keff_numerico_=" keff
PRINT %.4e TEXT "error_=" keff-k TEXT "(con" %g unknowns TEXT " ←
incognitas)"

# dibujamos una superficie tridimensional con gnuplot
# de los puntos donde phi(x,y) fue calculada
OUTPUT_FILE flujo-dat flujo-$1-$2.dat
PRINT_FUNCTION FILE flujo-dat phi_1

OUTPUT_FILE gp flujo-post.gp
# PRINT FILE gp TEXT "set title \"flujo con $1 $2\""
PRINT FILE gp TEXT "set_ticslevel_0"
PRINT FILE gp TEXT "set_palette_defined_(0,\"blue\",0.5,\"yellow\",1,\"red ←
\")"
PRINT FILE gp TEXT "set_xtics_50"
PRINT FILE gp TEXT "set_ytics_50"
PRINT FILE gp TEXT "set_ztics_0.6"
PRINT FILE gp TEXT "set_zrange_[-1e-6:" nu0 TEXT "]"
PRINT FILE gp TEXT "set_cbrange_[-1e-6:" nu0 TEXT "]"
PRINT FILE gp TEXT "set_xlabel_\"x\""
PRINT FILE gp TEXT "set_ylabel_\"y\""
PRINT FILE gp TEXT "set_zlabel_\"flujo\""
PRINT FILE gp TEXT "unset_colorbox"
PRINT FILE gp TEXT "set_terminal_pdf_size_16cm,9cm"
PRINT FILE gp TEXT "set_output_\"flujo3d-$1-$2.pdf\""
PRINT FILE gp TEXT "set_pm3d_depthorder"
PRINT FILE gp TEXT "splot_\"flujo-$1-$2.dat\"_w_p_pt_59_ps_0.7_palette_ti_ ←
\""

SHELL "gnuplot_flujo-post.gp"

OUTPUT_FILE flujo-cont flujo-cont-$1-$2.dat
PRINT_FUNCTION FILE flujo-cont phi_1 MIN -a -a MAX +a +a STEP a/400 a ←
/400

OUTPUT_FILE gpcont flujo-cont.gp
# PRINT FILE gpcont TEXT "set title \"flujo interpolado con $1 $2\""
PRINT FILE gpcont TEXT "set_palette_defined_(0,\"blue\",0.5,\"yellow\",1, ←
\"red\")"
PRINT FILE gpcont TEXT "set_size_square"
PRINT FILE gpcont TEXT "set_xrange_[" %g -a TEXT ":" a TEXT "]"
PRINT FILE gpcont TEXT "set_yrange_[" %g -a TEXT ":" a TEXT "]"
PRINT FILE gpcont TEXT "set_cbtics" nu0/4
PRINT FILE gpcont TEXT "set_cbrange_[-1e-6:" nu0 TEXT "]"
PRINT FILE gpcont TEXT "set_terminal_pdf_size_10cm,10cm"
PRINT FILE gpcont TEXT "set_output_\"flujo-cont-$1-$2.pdf\""
PRINT FILE gpcont TEXT "plot_\"flujo-cont-$1-$2.dat\"_w_image"

SHELL "gnuplot_flujo-cont.gp"

# escribimos el flujo en el formato de post-procesamiento de gms
OUTPUT_FILE flujo-post flujo-$1-$2.pos

```

```

MESH_POST FILE flujo-post phi_1

# y ademas incluimos opciones para hacer directamente las figuras con gmsh
OUTPUT_FILE fpost flujo-post.geo
PRINT FILE fpost TEXT "Merge_\\"flujo-$1-$2.pos\\";"
PRINT FILE fpost TEXT "General.SmallAxes_=_0;"
PRINT FILE fpost TEXT "PostProcessing.HorizontalScales_=_0;"
PRINT FILE fpost TEXT "General.GraphicsWidth_=_800;"
PRINT FILE fpost TEXT "General.GraphicsHeight_=_600;"
PRINT FILE fpost TEXT "View.ColormapNumber_=_4;"
PRINT FILE fpost TEXT "View.RangeType_=_2;"
PRINT FILE fpost TEXT "View.NbIso_=_4;"
PRINT FILE fpost TEXT "View.CustomMin_=_0;"
PRINT FILE fpost TEXT "View.CustomMax_=_ nu0 TEXT ";"
PRINT FILE fpost TEXT "Draw;"
PRINT FILE fpost TEXT "Print_\\"flujo2d-$1-$2.pdf\\";"
PRINT FILE fpost TEXT "Exit;"

# llamamos a gmsh para post-procesar
SHELL "gmsh_flujo-post.geo"

# lo mismo para el error
OUTPUT_FILE error-post error-$1-$2.pos
MESH_POST FILE error-post error

OUTPUT_FILE epost error-post.geo
PRINT FILE epost TEXT "Merge_\\"error-$1-$2.pos\\";"
PRINT FILE epost TEXT "General.SmallAxes_=_0;"
PRINT FILE epost TEXT "PostProcessing.HorizontalScales_=_0;"
PRINT FILE epost TEXT "Mesh.ColorCarousel_=_3;"
PRINT FILE epost TEXT "View.ColormapNumber_=_2;"
PRINT FILE epost TEXT "View.RangeType_=_2;"
PRINT FILE epost TEXT "View.NbIso_=_4;"
PRINT FILE epost TEXT "View.CustomMin_=_ -0.13;"
PRINT FILE epost TEXT "View.CustomMax_=_ +0.13;"
PRINT FILE epost TEXT "Draw;"
PRINT FILE epost TEXT "Print_\\"error2d-$1-$2.pdf\\";"
PRINT FILE epost TEXT "Exit;"

SHELL "gmsh_error-post.geo"

```

```

$ milonga flujo.was volumes triang
keff analitico =      0.98446205
keff numerico  =      0.98433928
error          =     -1.2278e-04      (con   178   incognitas)
$ milonga flujo.was volumes quad
keff analitico =      0.98446205
keff numerico  =      0.98463500
error          =      1.7295e-04      (con    88   incognitas)
$ milonga flujo.was elements triang
keff analitico =      0.98446205
keff numerico  =      0.98437880
error          =     -8.3254e-05      (con   106   incognitas)
$ milonga flujo.was elements quad
keff analitico =      0.98446205
keff numerico  =      0.98435582
error          =     -1.0624e-04      (con   105   incognitas)
$

```

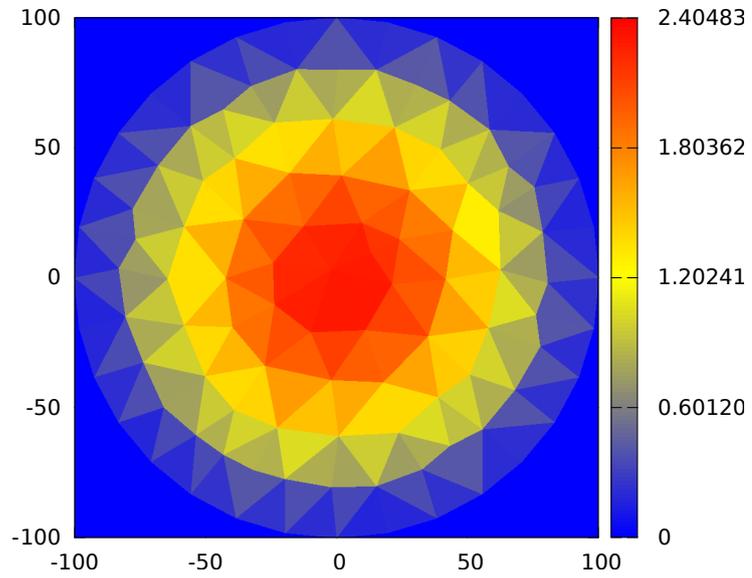


Figura 5.59 – Volúmenes finitos con triángulos: distribución de flujo interpolada por milonga en una malla fina con $\Delta x = \Delta y = a/400$.

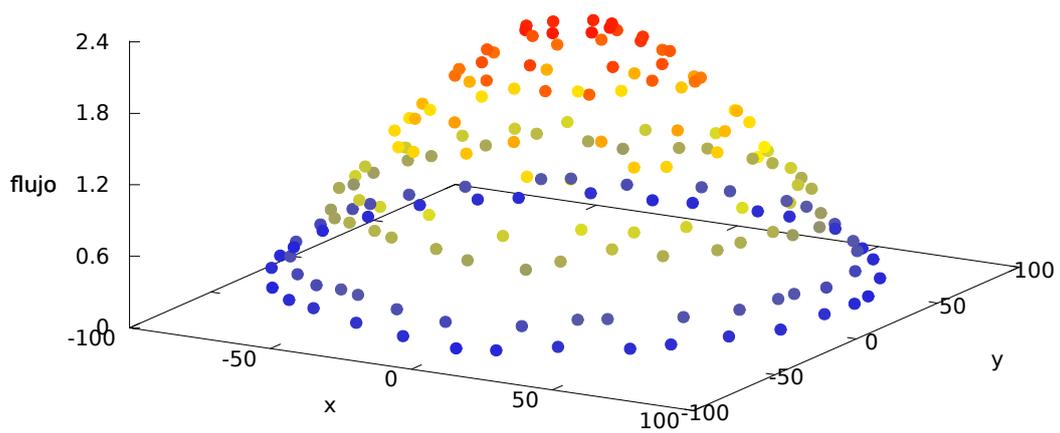


Figura 5.60 – Volúmenes finitos con triángulos: distribución de flujo evaluada en los centros de las celdas.

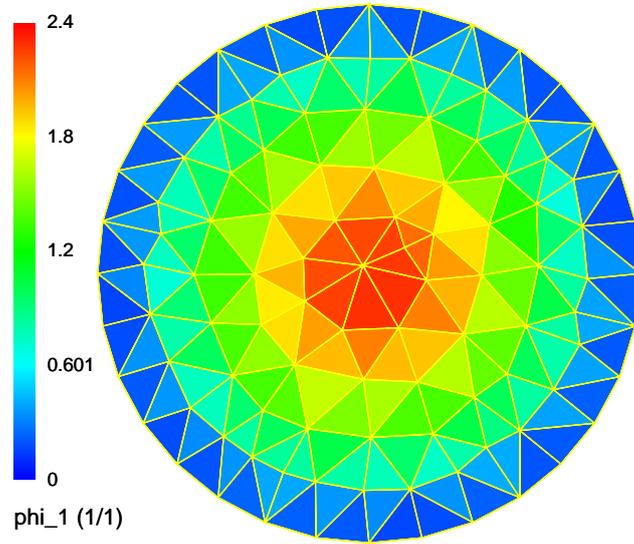


Figura 5.61 – Volúmenes finitos con triángulos: vista del flujo post-procesado con gmsh.

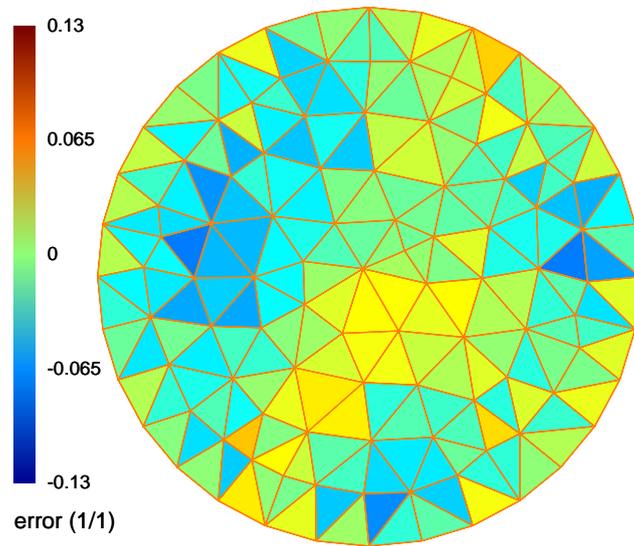


Figura 5.62 – Volúmenes finitos con triángulos: error del flujo evaluado en el centro de la celda con respecto a la solución analítica.

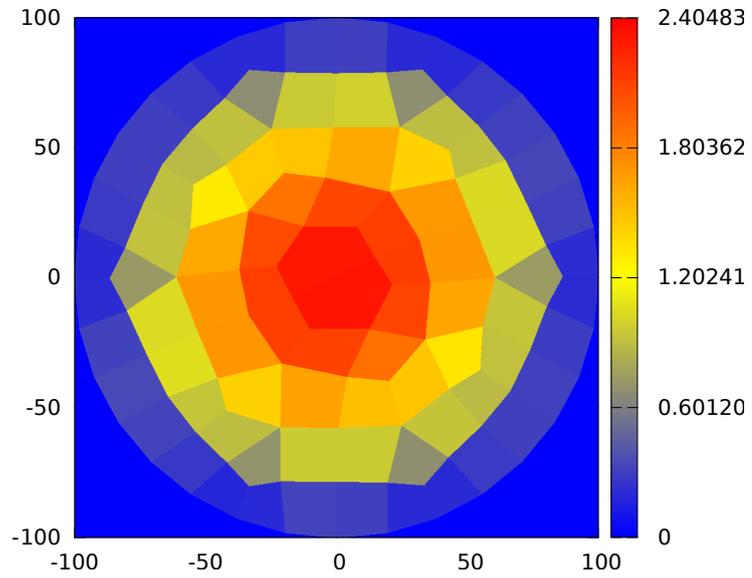


Figura 5.63 – Volúmenes finitos con cuadrángulos: distribución de flujo interpolada por milonga en una malla fina con $\Delta x = \Delta y = a/400$.

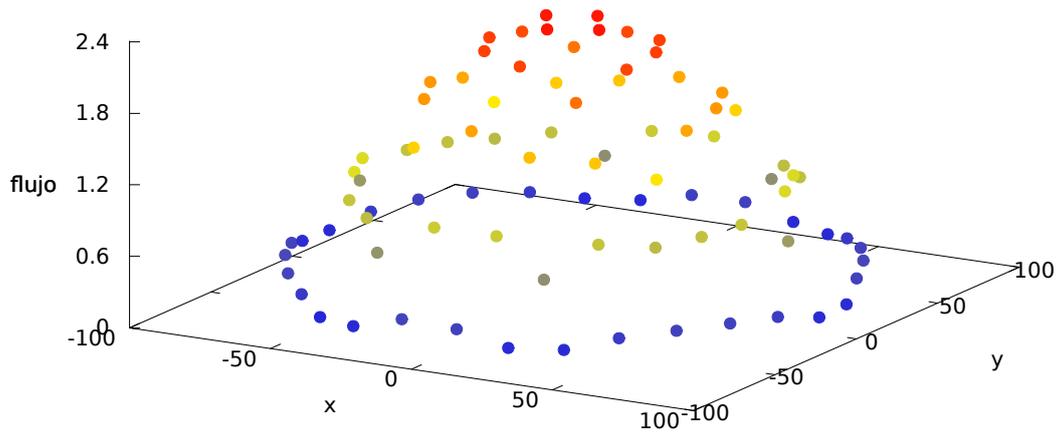


Figura 5.64 – Volúmenes finitos con cuadrángulos: distribución de flujo evaluada en los centros de las celdas.

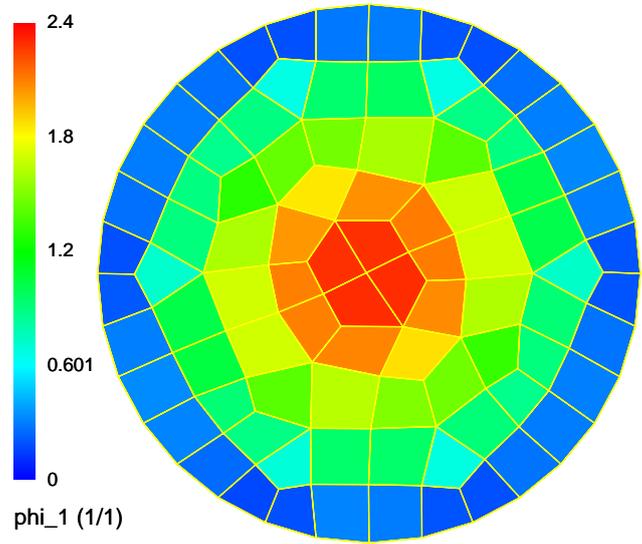


Figura 5.65 – Volúmenes finitos con cuadrángulos: vista del flujo post-procesado con gmsh.

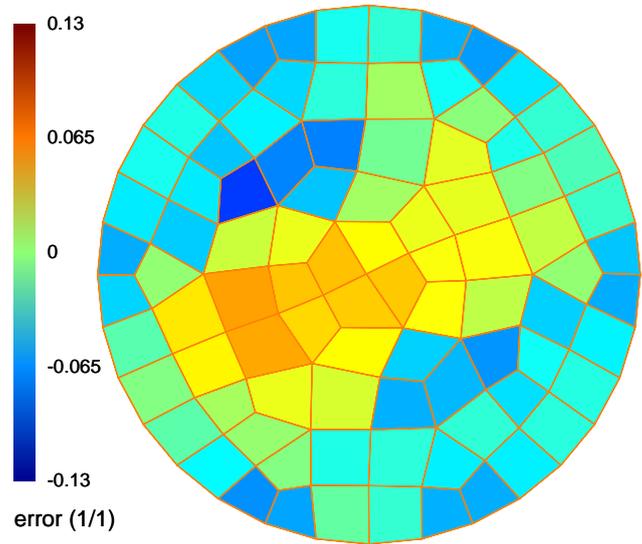


Figura 5.66 – Volúmenes finitos con cuadrángulos: error del flujo evaluado en el centro de la celda con respecto a la solución analítica.



En el mímico de la terminal podemos observar el error cometido en el cálculo del k_{eff} utilizando las cuatro combinaciones de triángulo/cuadrángulos con respecto a la solución analítica. Vemos que el error al utilizar el método de elementos finitos es casi un orden de magnitud más pequeño que el de volúmenes finitos. Además, el error al emplear volúmenes con triángulos o con cuadrángulos es similar—aunque difiere en signo—incluso cuando en el primer caso la cantidad de celdas es casi el doble, debido a que los cuadrángulos se obtienen de recombinar triángulos. No es exactamente el doble ya que después de recombinar, gmsh aplica un algoritmo de optimización de malla que puede mover, agregar, eliminar o volver a recombinar celdas/elementos.

Podemos explicar esta diferencia entre volúmenes y elementos de la siguiente manera: la precisión de la aproximación de las integrales de superficie que aparecen al aplicar el teorema de la divergencia al término de fugas depende, entre otros factores, del coseno del ángulo entre el vector normal a la superficie que separa dos celdas vecinas y el segmento que une sus baricentros. Como ilustramos en la figura 5.75, en general sucede que este ángulo es mayor para el caso de triángulos que para cuadrángulos. Por un lado, entonces, la malla basada en triángulos es desde este punto de vista inferior a la malla basada en cuadrados. Sin embargo, al no recombinar triángulos para formar cuadrángulos, la cantidad de celdas es mayor por lo que en cierto sentido la discretización es más fina. Como estos dos efectos son opuestos, es de esperar que ambas soluciones tengan errores similares con respecto al caso analítico.

Las figuras 5.59–5.74 son los gráficos generados por el input flujo.mil. La primera de cada uno de los cuatro conjuntos, (i.e. figura 5.59) consiste en la evaluación de la solución numérica $\phi_1(x, y)$ donde cada pixel abarca un cuadrado de tamaño $\Delta x \times \Delta y = a/400 \times a/400$ en las unidades de longitud del plano $x - y$. Por definición, $\phi_1(x, y) = 0$ si (x, y) está fuera del dominio. En las soluciones basadas en volúmenes, es claro que $\phi_1(x, y)$ es constante en cada celda y por lo tanto discontinua, mientras que en las soluciones basadas en elementos la función resultante es continua. Esto es así debido a que la solución del problema utilizando el método de elementos finitos no sólo da una receta sobre cómo evaluar $\phi_1(x, y)$ en los nodos sino que también indica cómo interpolar la función para coordenadas arbitrarias utilizando las mismas funciones de forma usadas para la construcción de las matrices R y F .

La segunda figura (i.e. 5.60) muestra la solución numérica como puntos dispersos. Para las soluciones en volúmenes, las coordenadas sobre el plano $x - y$ son los centros de las celdas y la distancia vertical indica el valor medio del flujo en ella. Para las soluciones en elementos, las coordenadas $x - y$ son las posiciones de los nodos y la distancia vertical el flujo evaluado en la posición del nodo. Es interesante notar que la condición de contorno de flujo nulo en $r = a$ se cumple exactamente en elementos—a menos de la precisión de la solución numérica del problema de autovalores—mientras que en volúmenes lo que se cumple es que la mejor estimación extrapolada del flujo es aproximadamente cero en $r = a$. En la solución de volúmenes y triángulos podemos observar que la densidad de puntos solución es mayor que en los otros casos. En volúmenes finitos la cantidad de incógnitas depende directamente—además de la cantidad de grupos de energía—de la cantidad de celdas, que a su vez depende de la forma base. Por el contrario, en elementos, la cantidad de incógnitas depende del número de nodos que no cambia al recombinar triángulos para generar cuadrángulos.

Las últimas dos figuras fueron construidas con las facilidades de post-procesamiento de gmsh, donde podemos observar tanto tanto las soluciones como la malla usada para obtenerlas. En la tercer figura (i.e. 5.61) vemos, en una escala de color, el flujo numérico $\phi_1(x, y)$. Otra vez, en volúmenes, el flujo es uniforme en cada celda. En elementos, dentro de cada elemento gmsh dibuja píxeles de colores de acuerdo a algún esquema de interpolación de los datos en los nodos que no necesariamente está basado en las mismas funciones de forma utilizadas para resolver el problema, por lo que debemos tomar esta figura sólo como ilustrativa y no como una solución

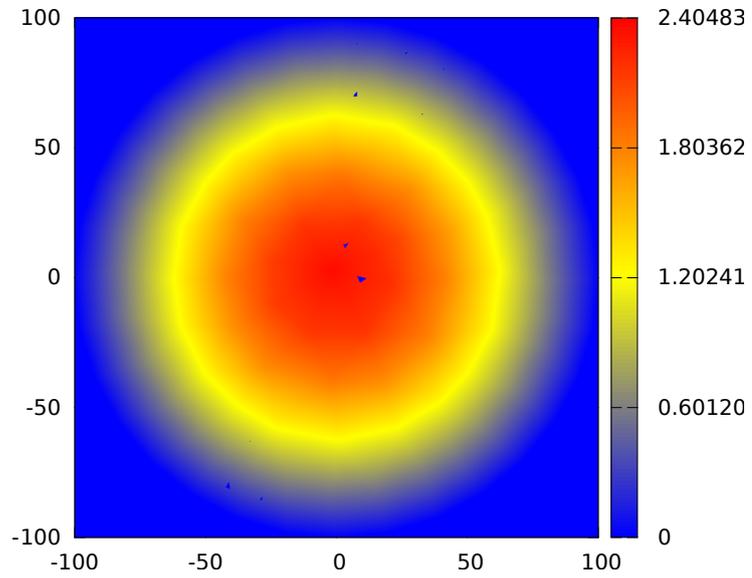


Figura 5.67 – Elementos finitos con triángulos: distribución de flujo interpolada por milonga en una malla fina con $\Delta x = \Delta y = a/400$.

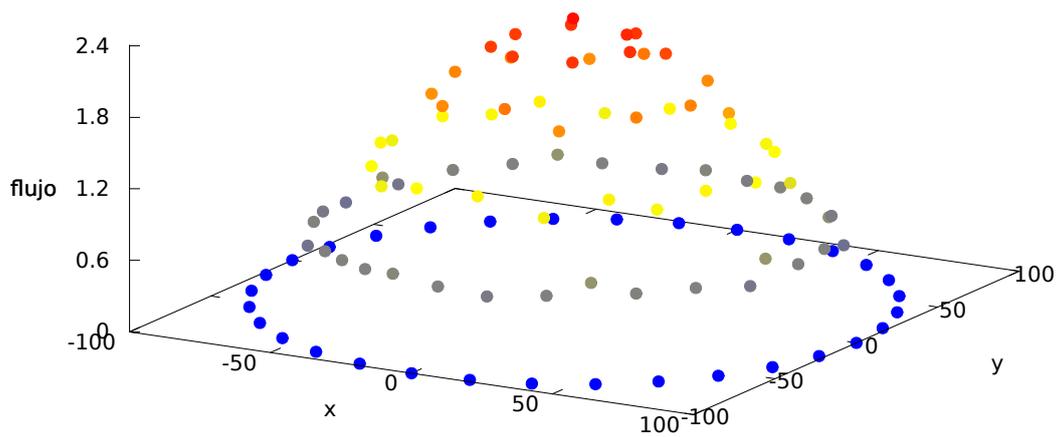


Figura 5.68 – Elementos finitos con triángulos: distribución de flujo evaluada en los centros de las celdas.

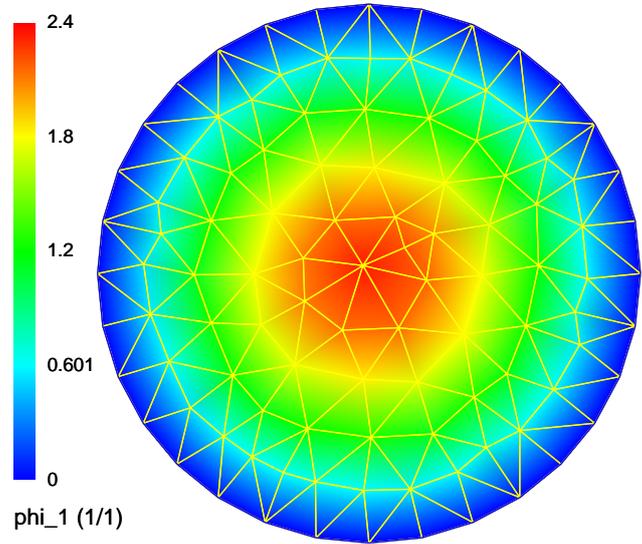


Figura 5.69 – Elementos finitos con triángulos: vista del flujo post-procesado con gmsh.

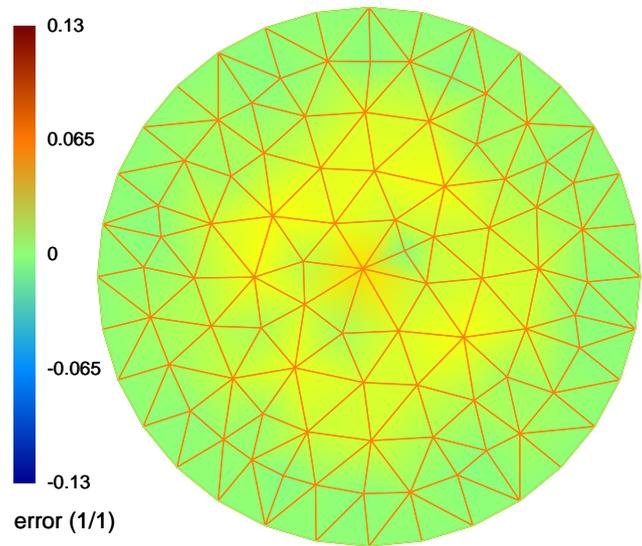


Figura 5.70 – Elementos finitos con triángulos: error del flujo evaluado en el centro de la celda con respecto a la solución analítica.

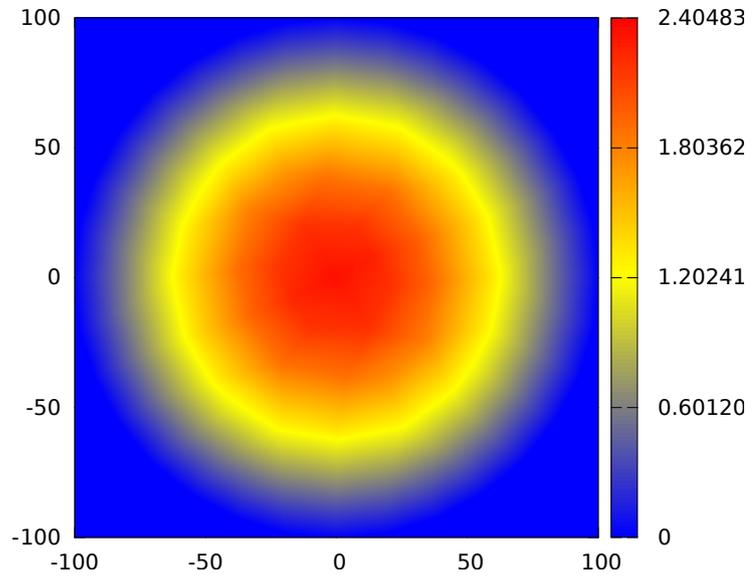


Figura 5.71 – Elementos finitos con cuadrángulos: distribución de flujo interpolada por milonga en una malla fina con $\Delta x = \Delta y = a/400$.

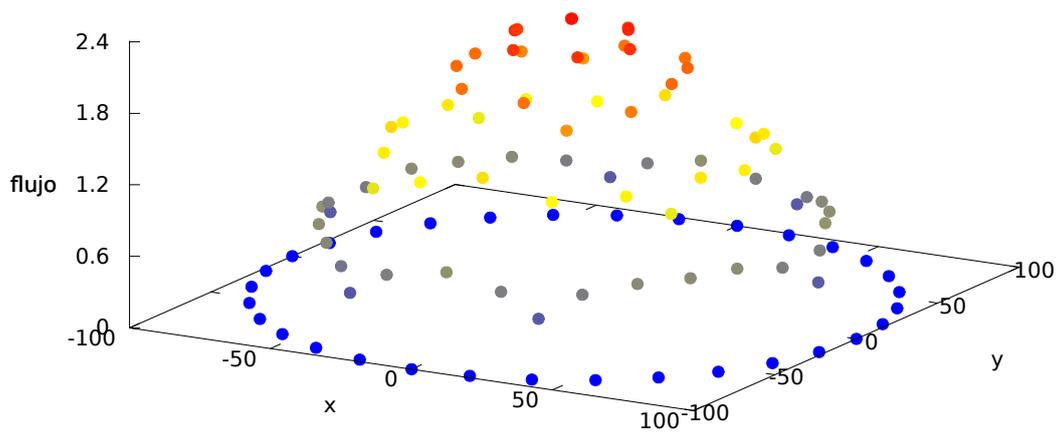


Figura 5.72 – Elementos finitos con cuadrángulos: distribución de flujo evaluada en los centros de las celdas.

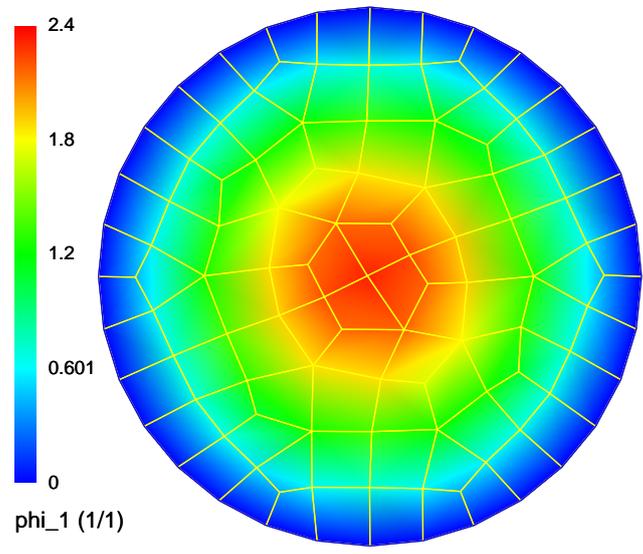


Figura 5.73 – Elementos finitos con cuadrángulos: vista del flujo post-procesado con gmsh.

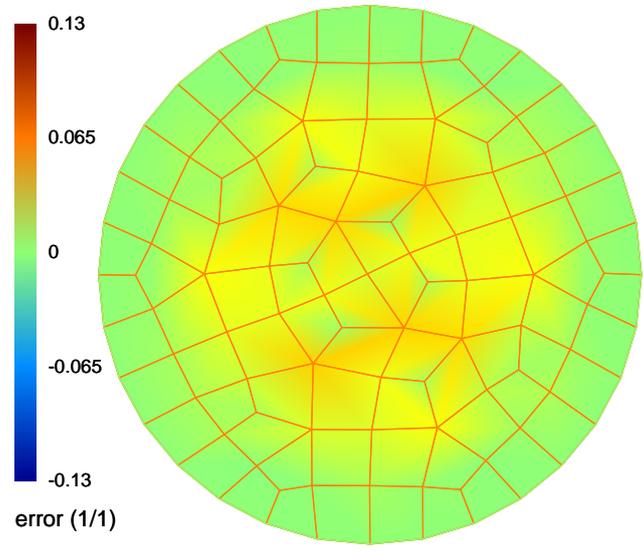


Figura 5.74 – Elementos finitos con cuadrángulos: error del flujo evaluado en el centro de la celda con respecto a la solución analítica.

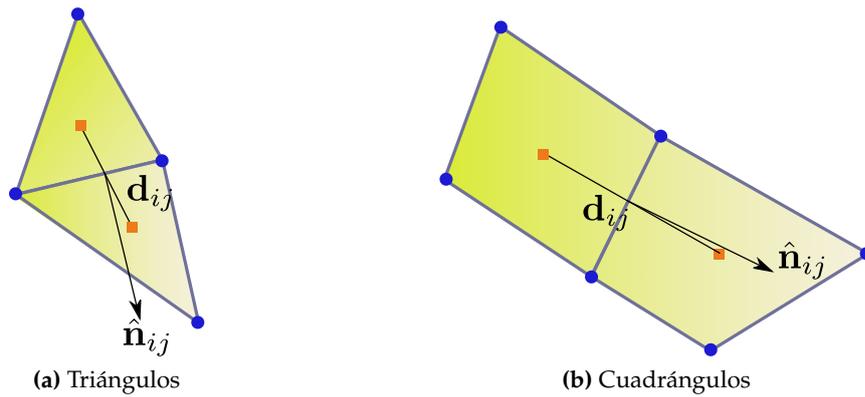


Figura 5.75 – Ángulo entre el segmento d_{ij} que une los baricentros de dos celdas vecinas y la normal \hat{n}_{ij} al segmento que los separa.

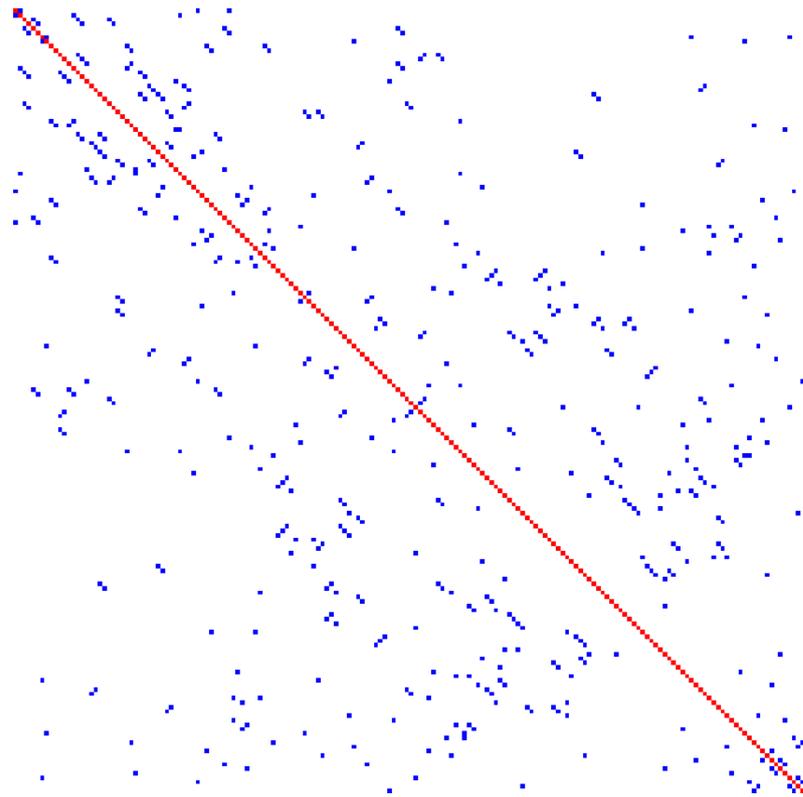
matemáticamente correcta. La cuarta figura (i.e. 5.62) ilustra, en una escala vertical igual para los cuatro casos, la diferencia entre la solución numérica $\phi_1(x, y)$ evaluada en los centros de las celdas o en los nodos, según corresponda. Notamos que dentro de cada celda/elemento, la figura no muestra el error con respecto a la solución analítica punto a punto sino que la escala de color es una interpolación de los errores cometidos en los puntos solución. De cualquier manera, es evidente que las soluciones en elementos arrojan errores mucho menores que las soluciones en volúmenes. Una vez más, debemos asociar este error más pequeño al esquema de discretización utilizado para resolver una ecuación diferencial en derivadas parciales con coeficientes uniformes. Conclusiones sobre errores cometidos en la aplicación de ingeniería de ambos métodos—involucrando homogeneización de secciones eficaces utilizando códigos de celda y aplicación a geometrías de reactores nucleares reales—deben realizarse con estudios más complejos que el propuesto en este trabajo.

5.2.1.2. Estructura de las matrices

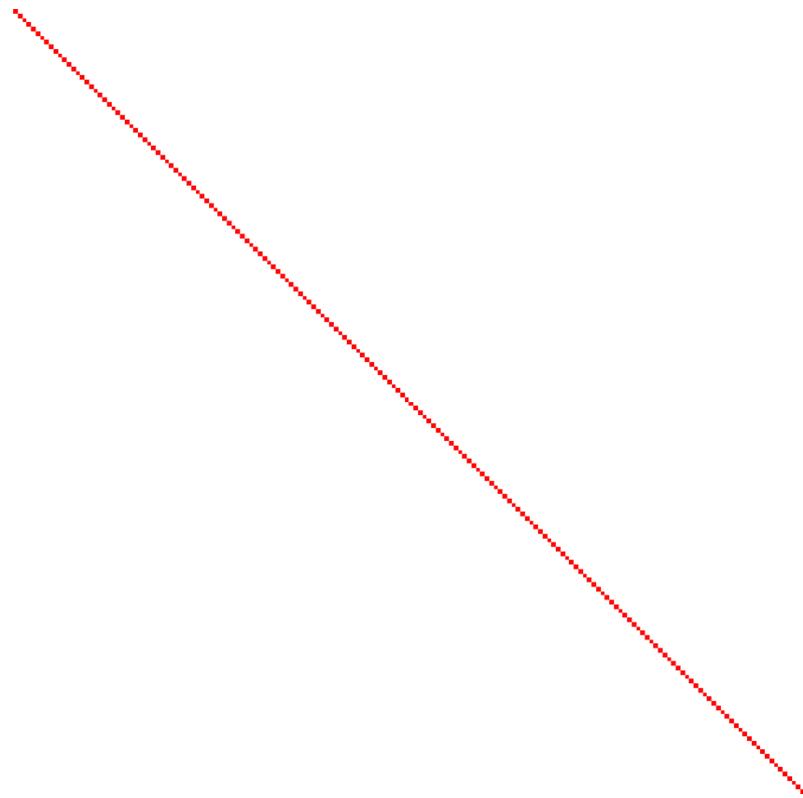
Como vimos en los capítulos 3 y 4, discretizar espacialmente la ecuación de difusión de neutrones multigrupo en estado estacionario consiste en reemplazar un sistema de G ecuaciones diferenciales homogénea en derivadas parciales por un problema de autovalores generalizados para dos matrices R y F de tamaño $LG \times LG$ tales que

$$R \cdot \hat{\phi} = \frac{1}{k_{\text{eff}}} \cdot F \cdot \hat{\phi} \quad (2.11)$$

donde L es la cantidad de incógnitas espaciales, que es la cantidad N de celdas en volúmenes finitos o la cantidad K de nodos en elementos finitos. En cualquier caso, problemas de interés real involucran valores de L del orden de decenas de miles. Afortunadamente, las matrices R y F tienen son ralas, por lo que una pequeña fracción de los $(LG)^2$ elementos son diferentes de cero en cada una, y es posible almacenarlas completamente en memoria RAM. Es interesante estudiar las diferencias entre las estructuras de elementos diferentes de cero para las matrices R y para F al resolver el problema del círculo desnudo. Por un lado, es importante estudiar las estructuras resultantes para entender mejor las diferencias entre volúmenes y elementos finitos. Pero además, el esfuerzo computacional necesario para resolver el problema numérico de autovalores, la forma óptima de construir las matrices en forma distribuida y otros detalles tales como el formato de almacenamiento en memoria depende de la estructura final de las matrices ralas del problema.

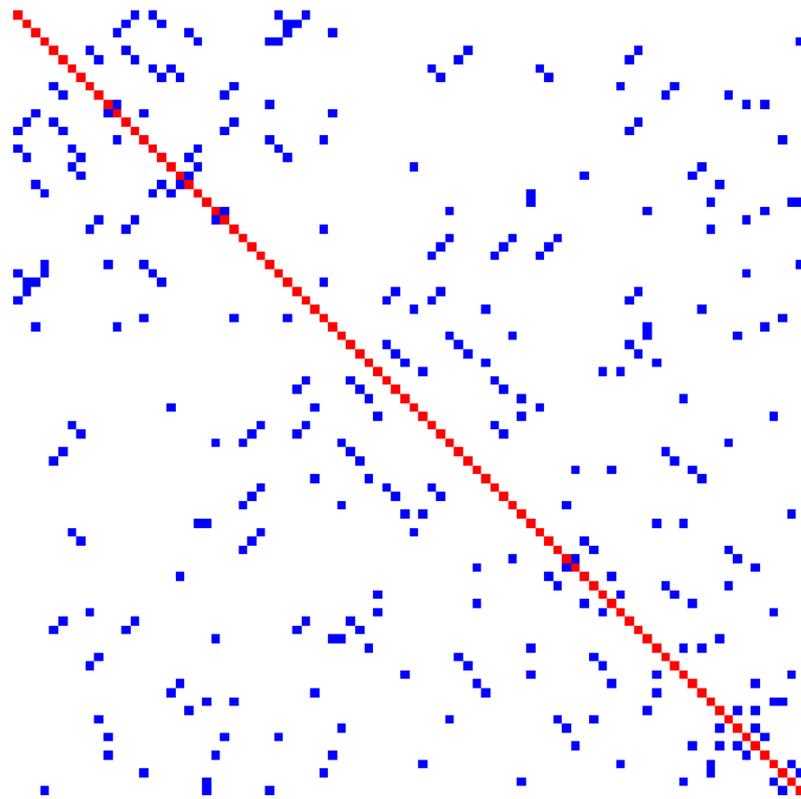


(a) R volúmenes triángulos 178×178

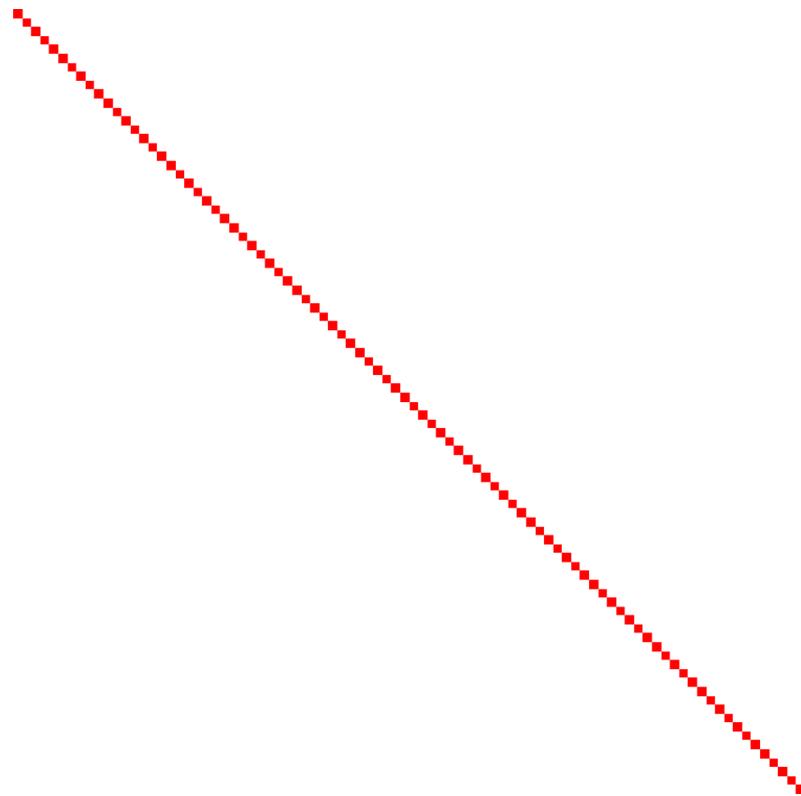


(b) F volúmenes triángulos 178×178

Figura 5.76 – Estructura de no-ceros en las matrices del problema del círculo desnudo resuelto con volúmenes finitos usando triángulos como figura base.

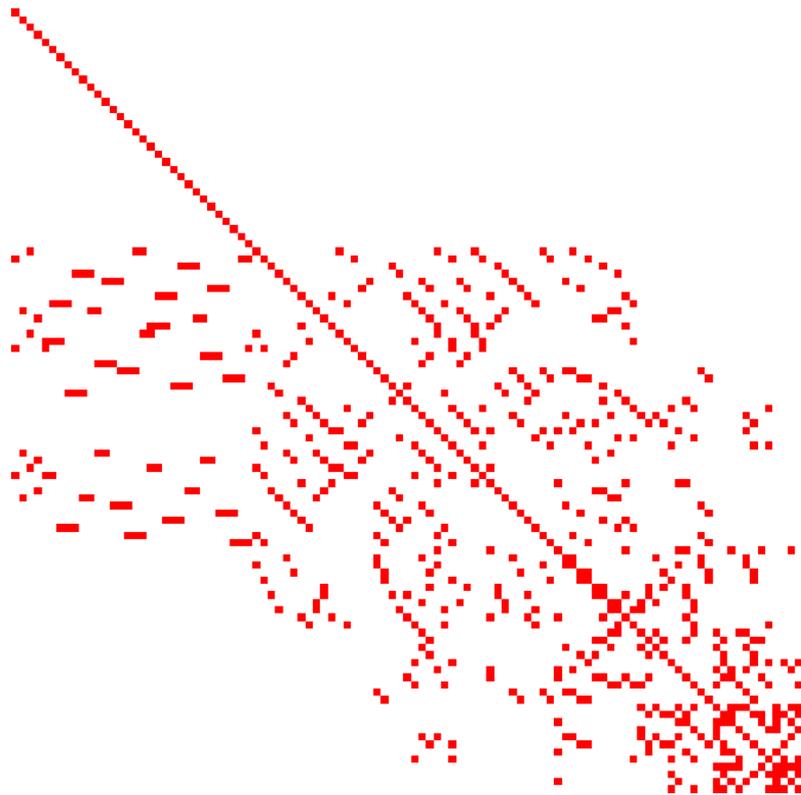


(a) R volúmenes cuadrángulos 88×88

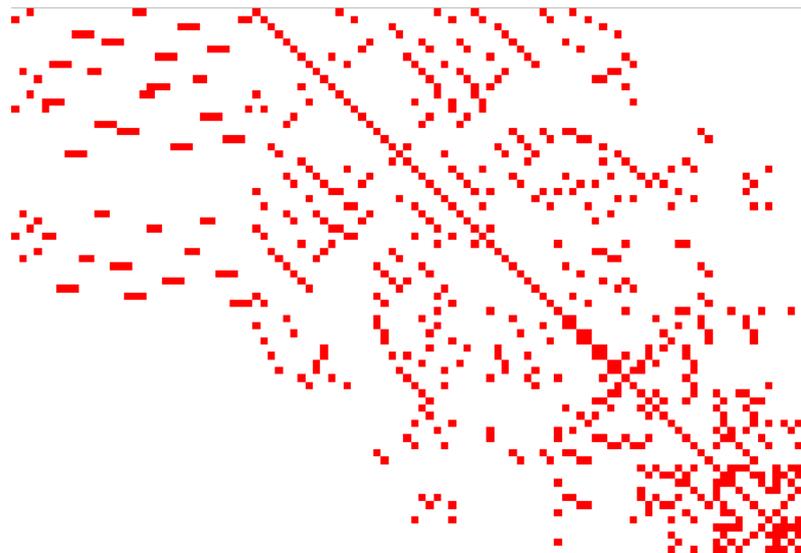


(b) F volúmenes cuadrángulos 88×88

Figura 5.77 – Estructura de no-ceros en las matrices del problema del círculo desnudo resuelto con volúmenes finitos usando cuadrángulos como figura base.

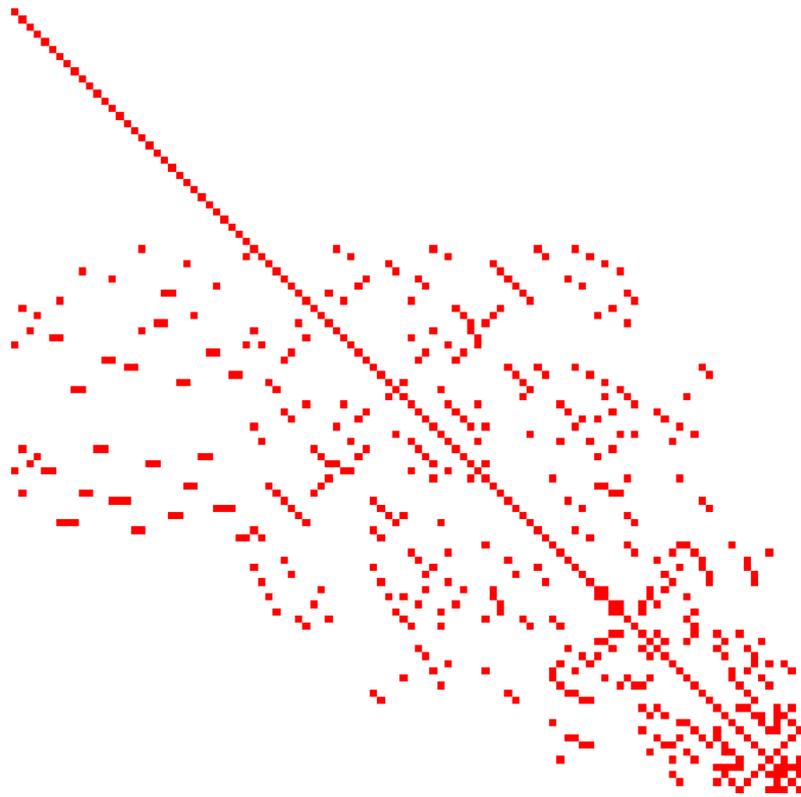


(a) R elementos triángulos 106×106

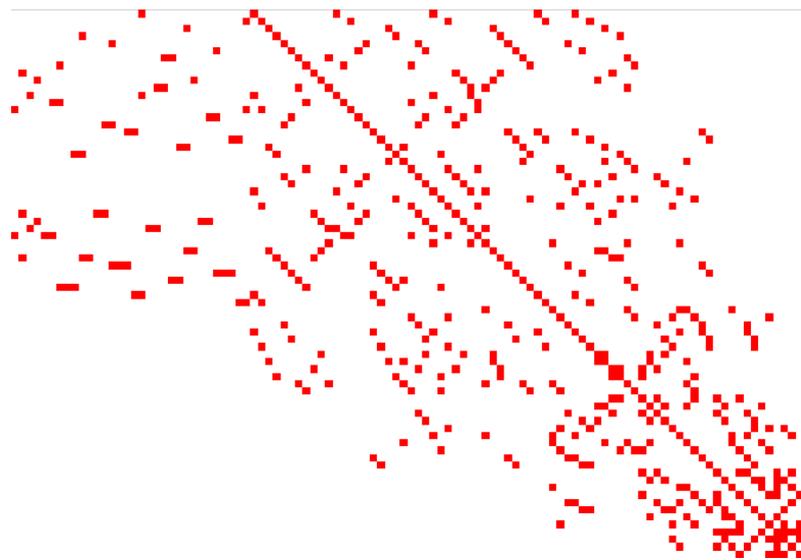


(b) F elementos triángulos 106×106

Figura 5.78 – Estructura de no-ceros en las matrices del problema del círculo desnudo resuelto con elementos finitos usando triángulos como figura base.



(a) R elementos cuadrángulos 105×105



(b) F elementos cuadrángulos 105×105

Figura 5.79 – Estructura de no-ceros en las matrices del problema del círculo desnudo resuelto con elementos finitos usando cuadrángulos como figura base.



Las figuras 5.76 y 5.76 muestran la estructura de las matrices del problema resuelto con volúmenes finitos con triángulos y cuadrángulos, respectivamente. Las figuras 5.78 y 5.78 hacen lo propio con elementos finitos. Los puntos azules indican valores positivos y los rojos valores negativos. El primer comentario consiste en notar que en volúmenes a un grupo de energía, la matriz F siempre es diagonal mientras que en elementos siempre tiene una estructura no trivial. En forma similar a lo que sucede con la matriz de masa en el problema elástico lineal transitorio, en volúmenes la fisión en cada celda depende sólo de las propiedades de la celda y de ningún otro flujo. Como para cada ecuación tenemos una única contribución a las fisiones, podemos elegir tanto el orden de los elementos del vector incógnita $\hat{\phi}$ como de las filas de las matrices de forma tal de que F sea diagonal. En cambio, en elementos finitos el ritmo de fisión en un elemento depende del flujo de los nodos que lo componen y por lo tanto no es posible elegir un arreglo que resulte en una matriz F diagonal. Además, dado que la malla es no estructurada, tampoco es posible en general obtener una estructura tipo banda, sino que la estructura está dada por la conectividad de los nodos que es elegida por el mallador y no por el solver.

La siguiente diferencia entre las matrices F tiene que ver con que la matriz de elementos finitos tiene una zona compuesta completamente por ceros, mientras que la matriz de volúmenes no. Esto es así dado que el problema que estamos resolviendo tiene condiciones de contorno de flujo nulo en el círculo externo, que en elementos fijamos reemplazando las ecuaciones correspondiente a los nodos que están en la superficie con esta condición de contorno de forma tal de que su solución sea $\phi_i = 0$. En cambio, en volúmenes finitos esta condición de contorno se ve reflejada como un término que aparece sumado al coeficiente de la diagonal de la matriz R correspondiente a aquellas celdas adyacentes al círculo externo.

Comparando ahora las figuras 5.76 y 5.76, vemos que—como ya discutimos—la solución basada en triángulos necesita aproximadamente el doble de incógnitas que si utilizamos cuadrángulos. La relación entre la cantidad de incógnitas en cada caso no es exactamente dos debido al proceso de optimización de malla que aplica gmsh luego de recombinar triángulos para obtener cuadrángulos al seleccionar el método de mallado delquad. La estructura de no-ceros en R depende de la conectividad entre celdas, que no es trivial para el caso de una malla no estructurada.

Por otro lado, sucede que las figuras 5.76 y 5.76 son esencialmente similares. La diferencia en el tamaño radica en que en el problema particular del círculo desnudo con $\ell_c = 20$, gmsh ha decidido eliminar un nodo luego de recombinar los triángulos para generar los cuadrángulos. La diferencia en la estructura fina de las matrices corresponde a que la conectividad de los triángulos es ligeramente diferente a la de los cuadrángulos, pero la estructura gruesa es similar en ambos casos.

Para terminar la sección, notamos primero que si cambiamos el tipo de condición de contorno de Dirichlet a Neumann o a Robin, la estructura de las matrices cambia. Y, por otro lado, solamente analizamos un caso con $G = 1$. En problemas a varios grupos de energía, aparecen sub-estructuras en la matriz R que tienen que ver con los terminos de scattering entre los diferentes grupos de energía y sub-estructuras en la matriz F correspondientes a las fisiones generadas por absorción de neutrones de un grupo que nacen con la energía de otro. Un problema similar al discutido en esta sección—pero con diferentes condiciones de contorno—fue analizado por el autor en la referencia [33], donde también analizamos la estructura resultante de las matrices con ambos esquemas de discretización.



5.2.1.3. Factor de multiplicación en función de la malla

Tal como hicimos con el slabs desnudos en la sección 5.1.1.3, en este problema también podemos estudiar cómo depende el error cometido en el k_{eff} con la longitud característica de la malla $\ell_c = a/n$. Para ello, resolvemos paramétricamente cada uno de los cuatro problemas variando n para cada caso, entre 5 y 50 con incrementos $\Delta n = 5$.

keff-vs-n.was

```
# estudio parametrico en funcion de n
PARAMETRIC n 10 50 2

a = 100 # radio del circulo
lc = a/n # longitud característica

# mallamos (y medimos el tiempo)
TEMPLATE circulo.tpl circulo.geo a lc
SHELL "if [ \$_1 \$_2 = \$_3 ] ; then echo \$_2 "Mesh.RecombineAll=1; \$_2 >> \$_3 <-
    circulo.geo; fi"
SHELL "gmsht -v \$_0 -2 \$_2 circulo.geo"
MESH DIMENSIONS 2 DEGREES 1 FILE_PATH circulo.msh

# resolvemos el keff numerico
MATERIAL fuel D 1 nuSigmaF 0.1 SigmaA 0.101
MILONGA SOLVE

# y el keff analitico
VAR r
nu0 = root(j0(r), r, 0, 3)
k = nuSigmaF(0,0)/(SigmaA(0,0) + D(0,0)*(nu0/a)^2)

# imprimimos los errores en funcion de n y de la cantidad de incognitas
PRINT %g n unknowns %3e keff-k abs(keff-k) init_time build_matrices_time <-
    solve_time SEPARATOR "  "
```

```
$ milonga keff-vs-n.was --volumes trian > keff-vs-n-vol-triang.dat
$ cat keff-vs-n-vol-triang.dat
10 714 -1.618e-04 1.618e-04 3.287e-03 7.060e-04 1.236e-02
12 1014 -9.558e-05 9.558e-05 6.675e-03 1.281e-03 1.229e-02
14 1360 -1.022e-04 1.022e-04 6.772e-03 1.051e-03 1.330e-02
16 1876 -8.636e-05 8.636e-05 1.712e-02 1.588e-03 1.800e-02
18 2342 -6.865e-05 6.865e-05 1.664e-02 1.853e-03 2.204e-02
20 2848 -8.659e-05 8.659e-05 2.618e-02 2.244e-03 2.707e-02
22 3408 -8.759e-05 8.759e-05 3.349e-02 2.747e-03 2.930e-02
24 4010 -5.795e-05 5.795e-05 5.328e-02 4.723e-03 4.142e-02
26 4650 -4.650e-05 4.650e-05 6.841e-02 3.955e-03 4.567e-02
28 5378 -3.551e-05 3.551e-05 9.457e-02 4.773e-03 5.035e-02
30 6350 -3.004e-05 3.004e-05 1.271e-01 5.466e-03 6.430e-02
32 7172 -3.866e-05 3.866e-05 1.682e-01 6.718e-03 6.672e-02
34 8064 -1.471e-05 1.471e-05 2.098e-01 7.369e-03 7.672e-02
36 8968 4.142e-06 4.142e-06 2.616e-01 8.282e-03 9.739e-02
38 9892 1.247e-05 1.247e-05 3.160e-01 9.311e-03 1.084e-01
40 10854 2.560e-05 2.560e-05 3.779e-01 1.037e-02 1.225e-01
42 11898 2.874e-05 2.874e-05 4.548e-01 1.157e-02 1.348e-01
44 13388 3.588e-05 3.588e-05 5.759e-01 1.398e-02 1.592e-01
46 14572 3.158e-05 3.158e-05 6.886e-01 1.447e-02 1.629e-01
48 15834 3.299e-05 3.299e-05 8.021e-01 1.586e-02 1.946e-01
```

```

50 16980 4.773e-05 4.773e-05 9.244e-01 1.721e-02 2.222e-01
$ milonga keff-vs-n.was --volumes quad > keff-vs-n-vol-quad.dat
$ cat keff-vs-n-vol-quad.dat
10 356 2.041e-04 2.041e-04 1.474e-03 3.631e-04 5.140e-03
12 569 1.412e-04 1.412e-04 2.623e-03 6.659e-04 8.870e-03
14 676 1.333e-04 1.333e-04 2.773e-03 6.371e-04 7.497e-03
16 924 1.234e-04 1.234e-04 5.775e-03 1.096e-03 1.451e-02
18 1235 9.474e-05 9.474e-05 7.178e-03 1.190e-03 1.234e-02
20 1406 9.864e-05 9.864e-05 1.417e-02 1.331e-03 1.556e-02
22 1787 7.437e-05 7.437e-05 1.376e-02 1.713e-03 1.943e-02
24 1985 8.207e-05 8.207e-05 2.037e-02 1.946e-03 2.199e-02
26 2409 9.226e-05 9.226e-05 2.418e-02 2.669e-03 2.495e-02
28 2655 6.923e-05 6.923e-05 3.161e-02 2.616e-03 2.725e-02
30 3142 4.469e-05 4.469e-05 3.796e-02 3.162e-03 3.533e-02
32 3694 6.784e-05 6.784e-05 5.388e-02 3.815e-03 4.277e-02
34 3979 4.220e-05 4.220e-05 6.002e-02 4.504e-03 4.471e-02
36 4580 5.142e-05 5.142e-05 8.469e-02 4.727e-03 4.950e-02
38 4902 5.475e-05 5.475e-05 9.453e-02 5.220e-03 5.872e-02
40 5578 4.609e-05 4.609e-05 1.294e-01 6.016e-03 7.278e-02
42 5904 4.817e-05 4.817e-05 1.434e-01 6.496e-03 6.578e-02
44 6644 5.185e-05 5.185e-05 1.885e-01 7.602e-03 8.238e-02
46 7428 4.225e-05 4.225e-05 2.292e-01 8.285e-03 9.309e-02
48 7843 3.715e-05 3.715e-05 2.624e-01 8.835e-03 9.923e-02
50 8649 2.461e-05 2.461e-05 3.169e-01 9.788e-03 1.157e-01
$ milonga keff-vs-n.was --elements trian > keff-vs-n-ele-triang.dat
$ cat keff-vs-n-ele-triang.dat
10 390 -2.033e-05 2.033e-05 3.259e-04 1.196e-02 1.216e-02
12 546 -1.437e-05 1.437e-05 1.130e-04 1.633e-02 1.131e-02
14 725 -1.080e-05 1.080e-05 1.230e-04 2.544e-02 1.061e-02
16 991 -7.807e-06 7.807e-06 1.280e-04 3.461e-02 1.475e-02
18 1230 -6.258e-06 6.258e-06 1.521e-04 4.084e-02 1.800e-02
20 1489 -5.152e-06 5.152e-06 1.450e-04 5.372e-02 2.162e-02
22 1775 -4.310e-06 4.310e-06 1.719e-04 8.700e-02 2.625e-02
24 2082 -3.662e-06 3.662e-06 1.600e-04 1.146e-01 3.044e-02
26 2408 -3.166e-06 3.166e-06 2.770e-04 1.437e-01 3.644e-02
28 2778 -2.748e-06 2.748e-06 1.841e-04 1.658e-01 4.269e-02
30 3272 -2.316e-06 2.316e-06 2.069e-04 2.030e-01 5.019e-02
32 3689 -2.050e-06 2.050e-06 2.141e-04 2.306e-01 5.730e-02
34 4141 -1.828e-06 1.828e-06 2.279e-04 2.678e-01 6.579e-02
36 4599 -1.649e-06 1.649e-06 2.332e-04 3.150e-01 7.777e-02
38 5067 -1.493e-06 1.493e-06 2.580e-04 3.483e-01 8.441e-02
40 5554 -1.361e-06 1.361e-06 2.739e-04 3.965e-01 9.273e-02
42 6082 -1.249e-06 1.249e-06 3.109e-04 4.681e-01 1.015e-01
44 6835 -1.105e-06 1.105e-06 3.190e-04 5.980e-01 1.188e-01
46 7433 -1.014e-06 1.014e-06 3.390e-04 6.857e-01 1.340e-01
48 8070 -9.370e-07 9.370e-07 4.840e-04 7.830e-01 1.523e-01
50 8649 -8.758e-07 8.758e-07 3.870e-04 8.560e-01 1.582e-01
$ milonga keff-vs-n.was --elements quad > keff-vs-n-ele-quad.dat
$ cat keff-vs-n-ele-quad.dat
10 389 -2.589e-05 2.589e-05 3.211e-04 7.128e-03 6.827e-03
12 610 -1.613e-05 1.613e-05 1.280e-04 1.669e-02 1.085e-02
14 721 -1.360e-05 1.360e-05 1.180e-04 1.491e-02 1.083e-02
16 977 -9.749e-06 9.749e-06 1.500e-04 2.883e-02 1.445e-02
18 1296 -7.316e-06 7.316e-06 1.440e-04 3.191e-02 1.997e-02
20 1471 -6.447e-06 6.447e-06 2.182e-04 4.228e-02 2.244e-02
22 1860 -5.059e-06 5.059e-06 1.612e-04 5.191e-02 2.977e-02
24 2062 -4.537e-06 4.537e-06 1.590e-04 5.703e-02 3.350e-02
26 2494 -3.719e-06 3.719e-06 2.799e-04 7.780e-02 4.117e-02
28 2744 -3.387e-06 3.387e-06 2.921e-04 8.699e-02 4.538e-02
30 3239 -2.843e-06 2.843e-06 3.130e-04 1.079e-01 5.579e-02
32 3799 -2.427e-06 2.427e-06 3.490e-04 1.311e-01 6.708e-02

```



34	4088	-2.243e-06	2.243e-06	3.610e-04	1.455e-01	7.557e-02
36	4697	-1.949e-06	1.949e-06	3.660e-04	1.726e-01	8.648e-02
38	5023	-1.814e-06	1.814e-06	2.570e-04	1.906e-01	9.529e-02
40	5707	-1.595e-06	1.595e-06	2.830e-04	2.283e-01	1.110e-01
42	6037	-1.508e-06	1.508e-06	2.961e-04	2.503e-01	1.232e-01
44	6785	-1.344e-06	1.344e-06	3.231e-04	2.936e-01	1.429e-01
46	7577	-1.202e-06	1.202e-06	3.490e-04	3.786e-01	1.637e-01
48	7996	-1.133e-06	1.133e-06	3.641e-04	4.376e-01	1.912e-01
50	8810	-1.027e-06	1.027e-06	5.150e-04	5.807e-01	2.124e-01
\$ pyxplot error-vs-n.ppl						
\$ pyxplot error-vs-incognitas.ppl						
\$ pyxplot read-vs-incognitas.ppl						
\$ pyxplot build-vs-incognitas.ppl						
\$ pyxplot solve-vs-incognitas.ppl						
\$ pyxplot total-vs-incognitas.ppl						
\$ pyxplot error-vs-time.ppl						
\$						

En la figura 5.80 podemos observar el valor absoluto del error cometido con respecto al k_{eff} analítico en la escala vertical logarítmica en función del parámetro n . Dado que el tamaño del problema numérico no es el mismo en cada uno de los cuatro casos estudiados, graficamos el mismo error pero en función de la cantidad de incógnitas en la figura 5.81. No sólo vemos que los errores cometidos por los métodos basados en elementos son siempre menores que los cometidos por volúmenes finitos, sino que éstos últimos no cambian monótonamente con n . De hecho, en la terminal podemos ver que incluso su signo cambia para mallas cada vez más finas. Podemos explicar este comportamiento basándonos de nuevo en el coseno del ángulo entre \hat{n}_{ij} y \mathbf{d}_{ij} . Al ir disminuyendo la longitud característica, el código gmsh podrá considerar sucesivamente mallas que hagan que este coseno pueda variar en forma no continua. Esto puede introducir grandes cambios en el k_{eff} resultante aún para pequeños cambios en ℓ_c . En el caso de elementos finitos, debemos notar que si bien la cantidad de incógnitas que resulta de aplicar una malla basada en triángulos con y sin recombinar son similares, el esfuerzo computacional requerido para construir las matrices R y F se incrementa con la cantidad de elementos, que en el caso de triángulos puros es aproximadamente el doble de cuadrángulos generados a partir de recombinar triángulos. Es por eso que en el mismo sentido en el que decíamos en la sección anterior que una malla triangular provee una discretización espacial más fina que una malla cuadrangular, observamos que la solución en elementos basada en triángulos da como resultado un error menor a costa de un mayor esfuerzo computacional.

En efecto, las figuras 5.82–5.85 comparan el tiempo insumido para interpretar la malla y buscar vecinos si el método seleccionado es volúmenes finitos, el insumido para construir las matrices, el insumido para resolver el problema de autovalores y la suma de cada uno de ellos, correspondientemente. Los tiempos indicados deben tomarse sólo como referencias, ya que son tiempos de pared (wall times) medidos por una versión de milonga que no está aún optimizada para realizar eficientemente muchas de las operaciones necesarias para resolver el problema. Sin embargo, es útil comparar cada uno de estos tiempos entre los diferentes casos. Los métodos basados en volúmenes necesitan evaluar qué celdas son vecinas de qué celdas, ya que gmsh no provee esa información en el archivo de malla. Si bien es posible optimizar el algoritmo que usa milonga, este paso insume un cierto esfuerzo que no es necesario para elementos finitos. Sin embargo, la construcción de las matrices R y F para volúmenes es más sencilla que para elementos, que implica la evaluación e integración de jacobinas más un proceso de ensamble que, otra vez, aunque pueda ser optimizado, no es necesario para volúmenes. La estructura resultante de las matrices para elementos finitos hace que el problema de autovalores necesite ligeramente más operaciones que para volúmenes. La figura 5.85 compara el tiempo

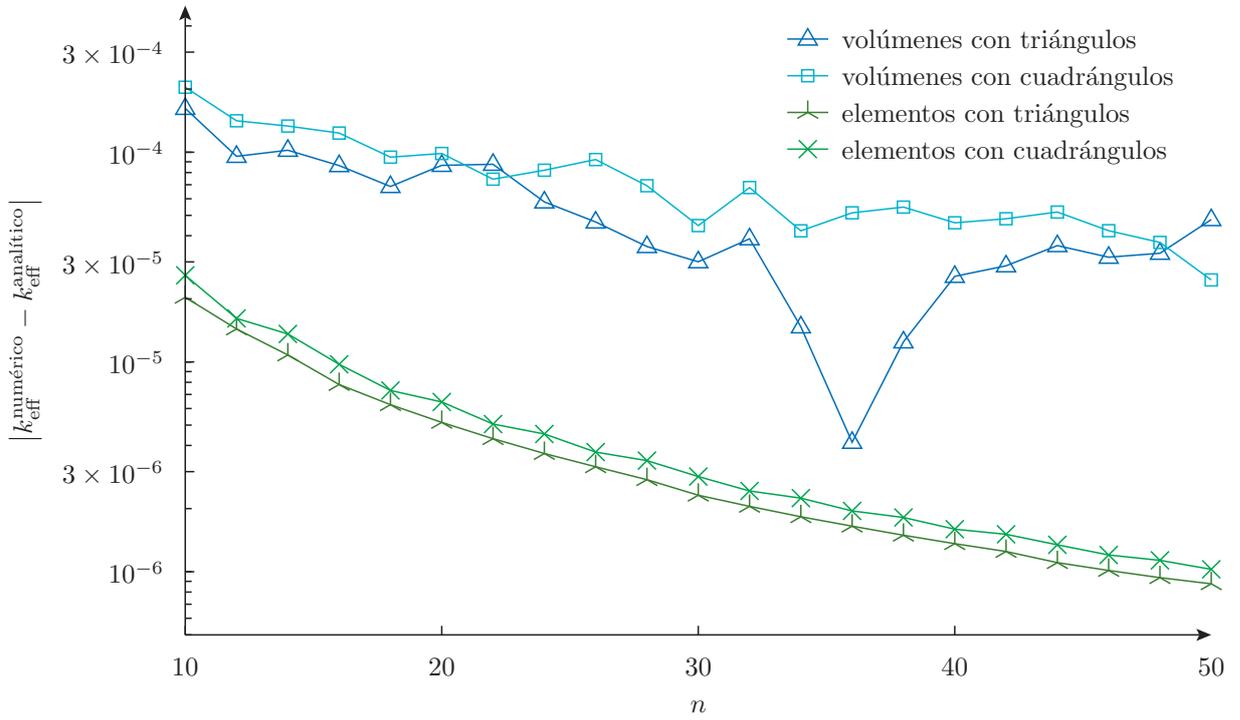


Figura 5.80 – Error cometido en el cálculo de k_{eff} en función del parámetro n utilizado en la definición de la longitud característica $\ell_c = a/n$ de la malla.

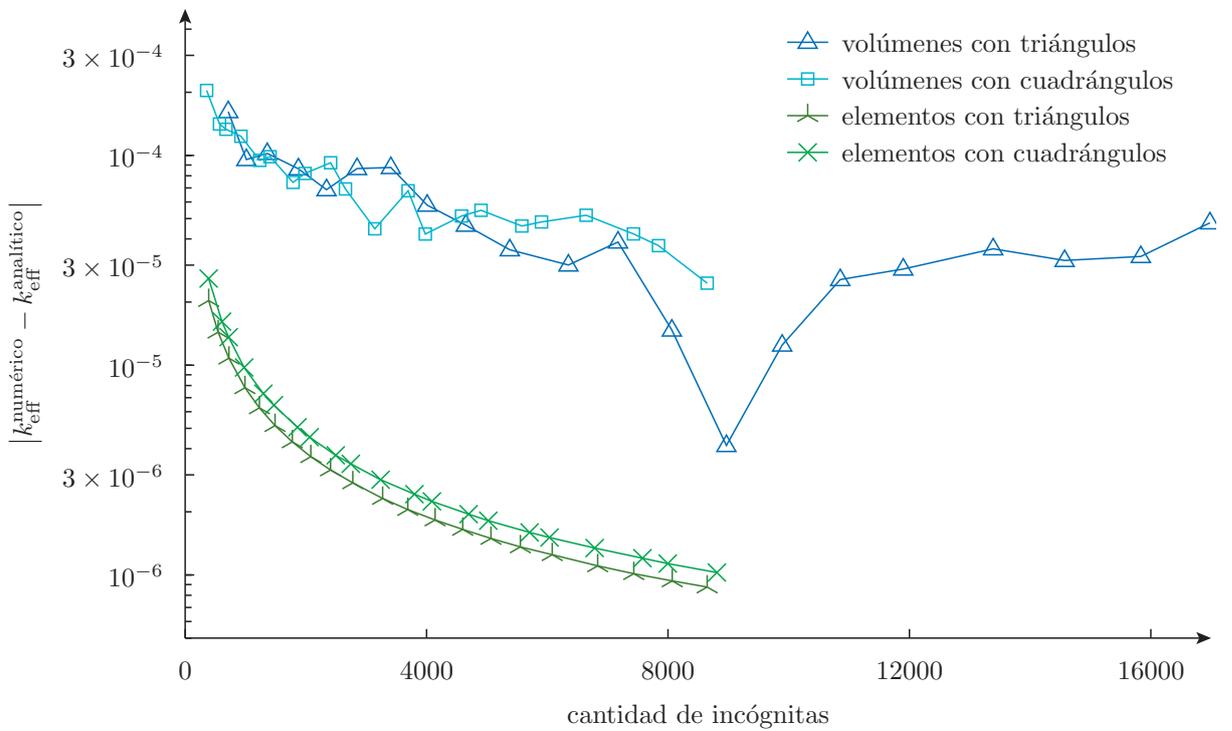


Figura 5.81 – Error cometido en el cálculo de k_{eff} en función de la cantidad de incógnitas del problema.

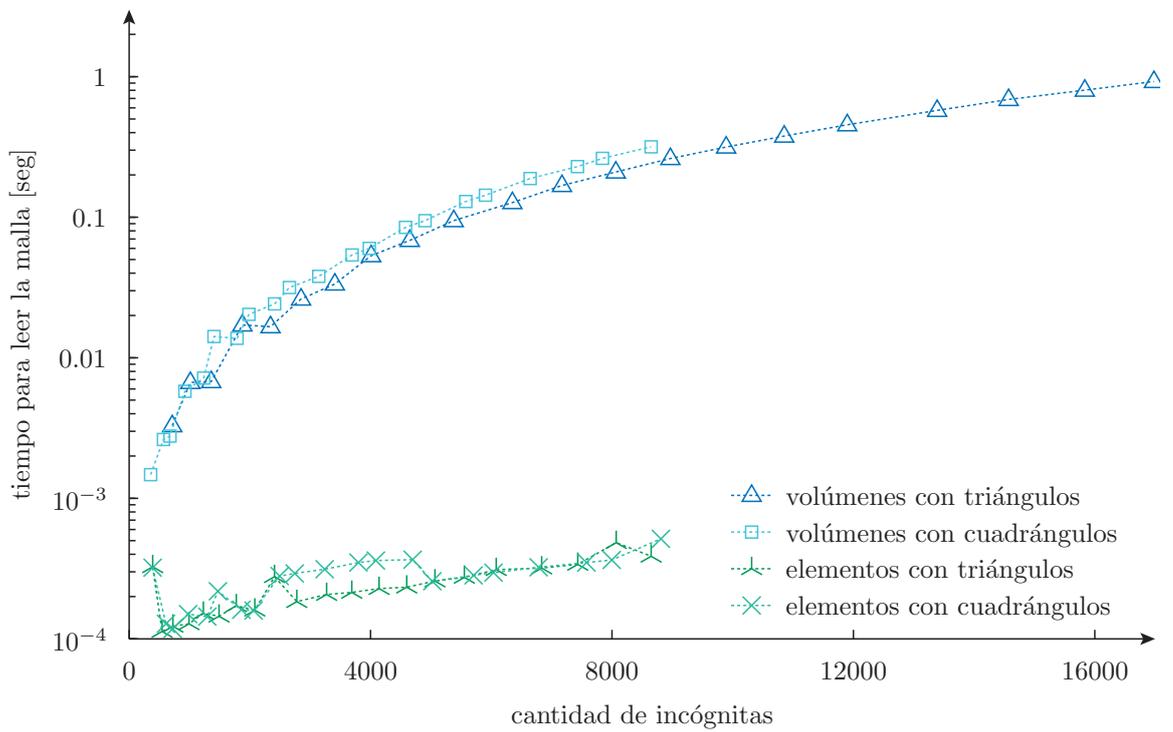


Figura 5.82 – Tiempo de cálculo insumido para interpretar la malla generada por gmsh (no para mallar ni para leer el archivo) en función de la cantidad de incógnitas.

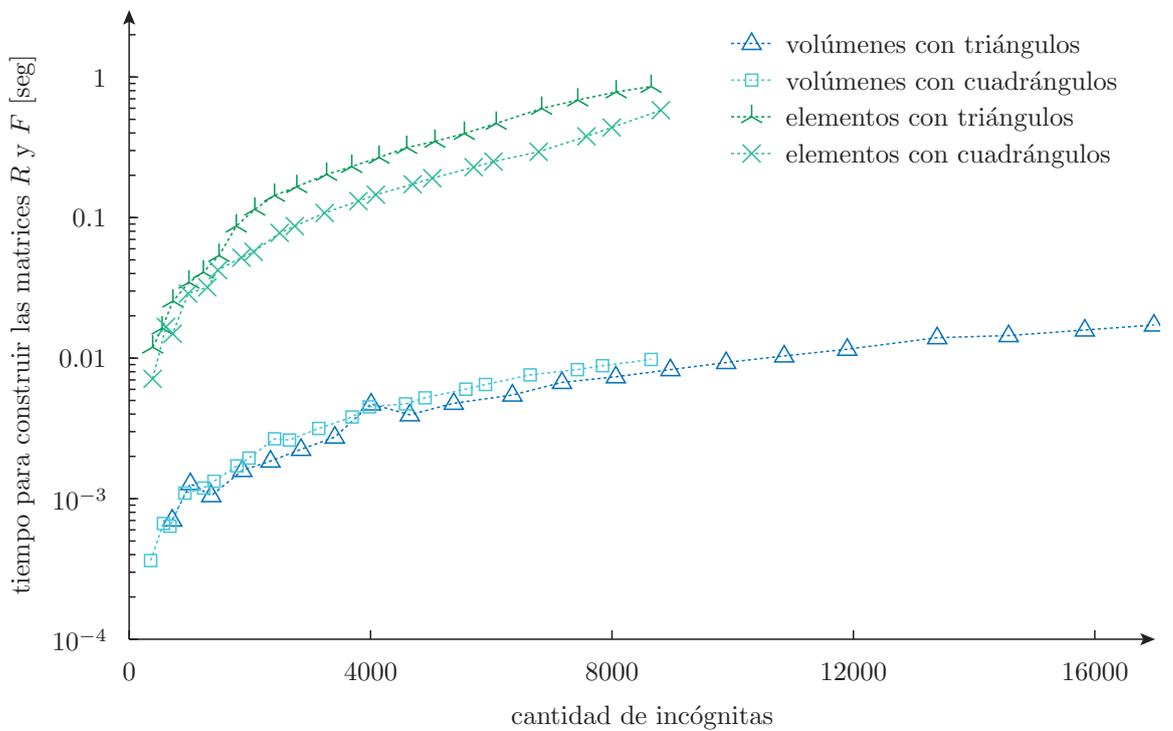


Figura 5.83 – Tiempo de cálculo insumido para construir las matrices R y F del problema discreto en función de la cantidad de incógnitas.

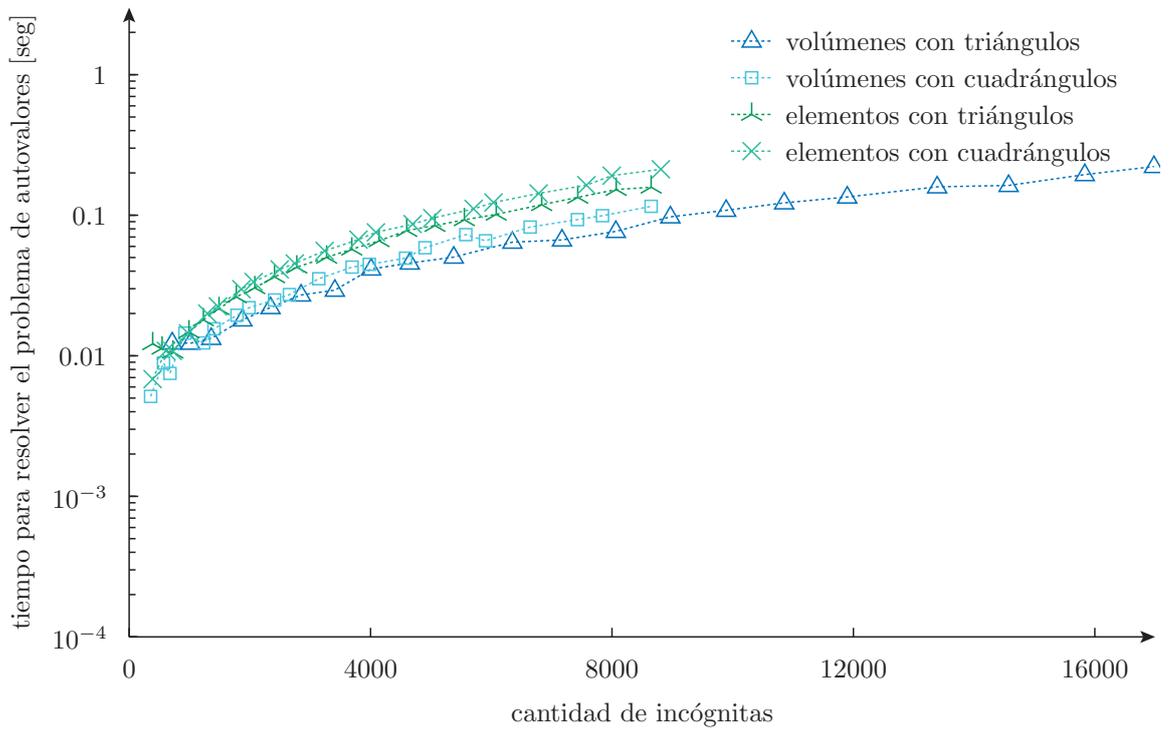


Figura 5.84 – Tiempo de cálculo insumido para resolver el problema de autovalores en función de la cantidad de incógnitas.

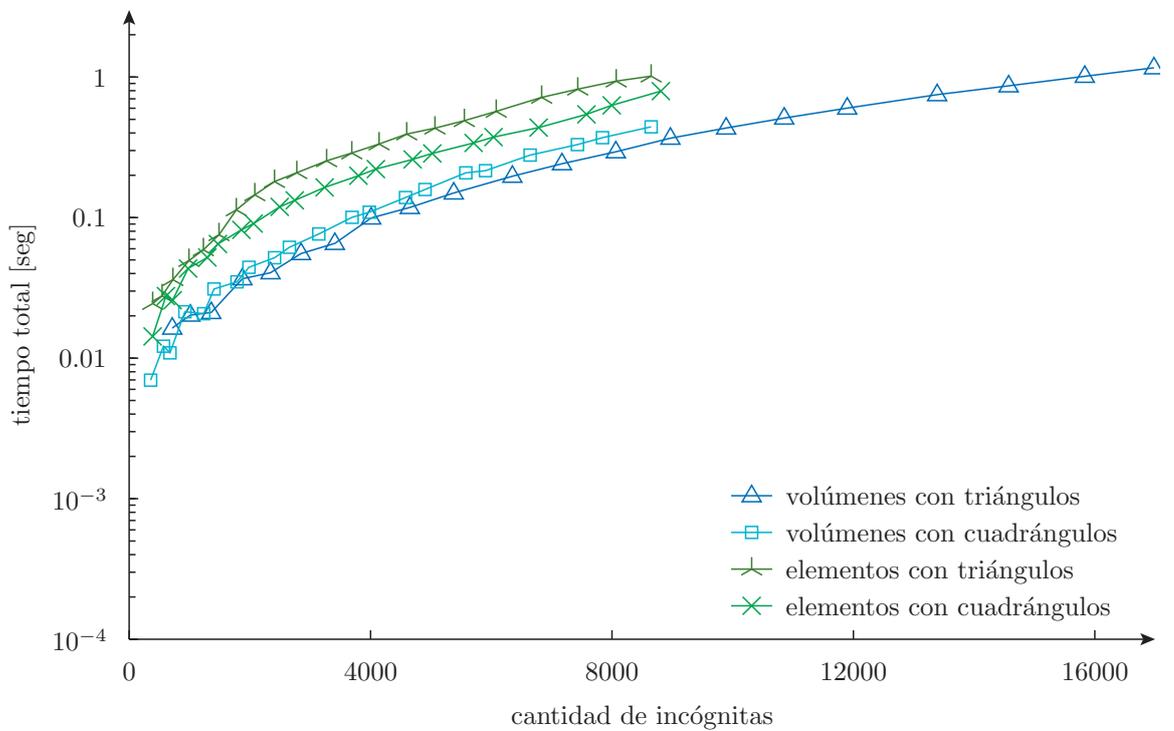


Figura 5.85 – Tiempo total de cálculo (sin contar el tiempo necesario para generar la malla por gmsh) en función de la cantidad de incógnitas.

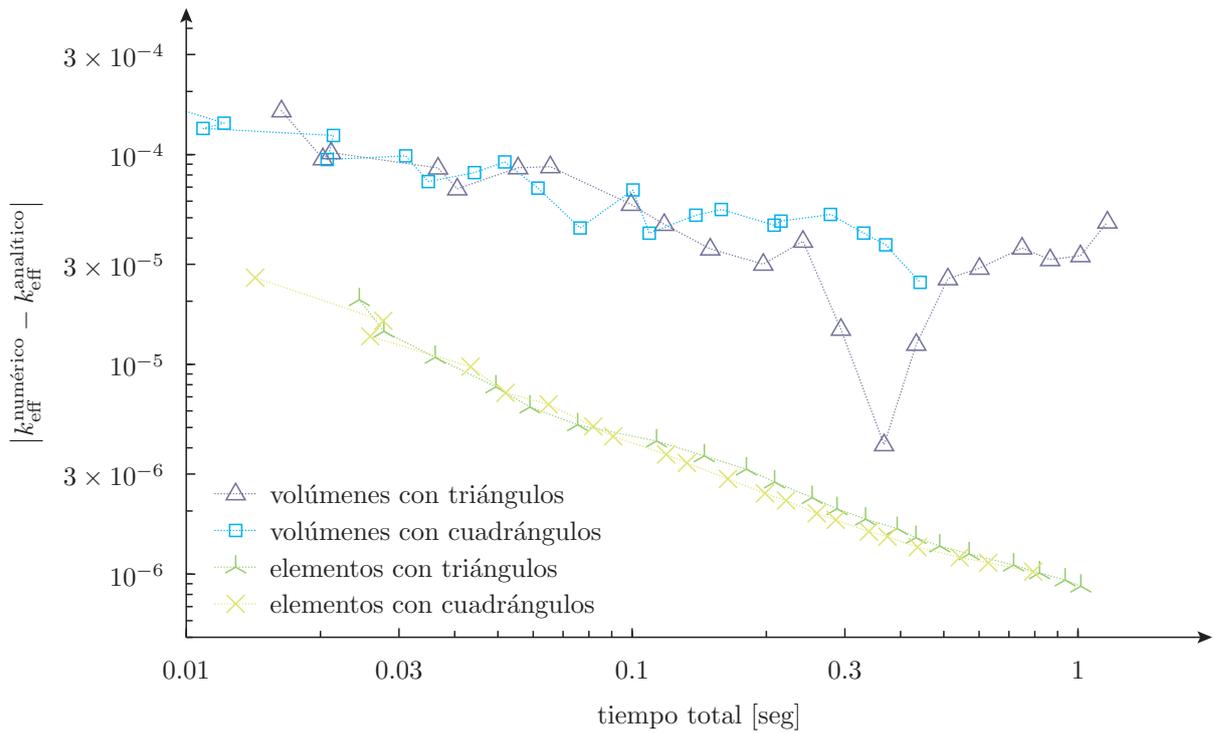


Figura 5.86 – Error en función del tiempo necesario para obtener la solución.

total insumido en resolver el problema completamente (exceptuando el tiempo necesario para generar la malla con gmsh) para los cuatro casos, y en función de la cantidad de incógnitas. En el problema del círculo uniforme desnudo en particular que estamos resolviendo con los algoritmos actuales del código milonga, el método de elementos finitos es en general computacionalmente más caro que el método en volúmenes para el mismo tamaño de problema. En efecto, en la figura 5.86 resumimos los resultados al graficar el error cometido en función del tiempo total insumido para lograr la solución. Observamos que, para los algoritmos utilizados por milonga, la relación error/costo es casi un orden de magnitud menor para los métodos basados en elementos que para volúmenes finitos.

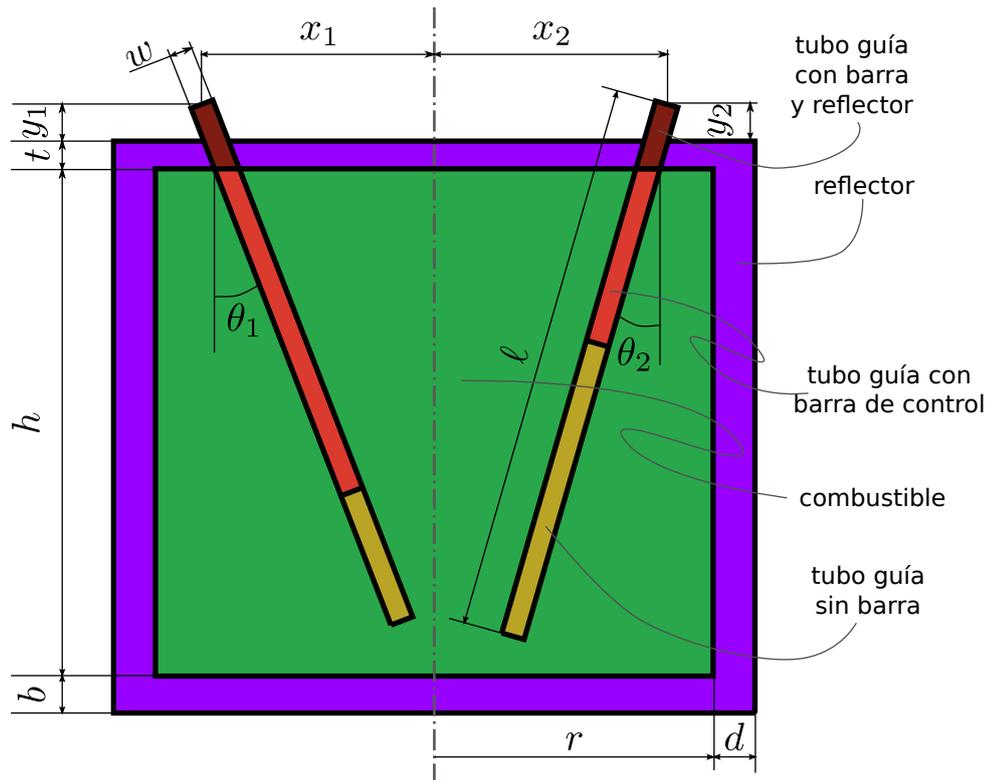


Figura 5.87 – Geometría de un corte transversal de un reactor con barras inclinadas

5.2.2. Corte transversal de un reactor con barras inclinadas

Si bien el problema que resolvemos en esta sección es también completamente inventado, ya deja de ser un ejemplo canónico y carente de interés ingenieril. La geometría, que mostramos en la figura 5.87, está inspirada en un hipotético reactor reflejado con barras de control que deslizan sobre tubos guía que entran al núcleo con un cierto ángulo de inclinación con respecto a la dirección del eje vertical. Como es un problema bidimensional, podemos pensar que la geometría es un corte transversal de un reactor que se prolonga infinitamente en la dirección perpendicular al papel o a la pantalla, dependiendo del medio de publicación de la monografía.

Como vemos en la figura 5.87, la geometría está completamente parametrizada. Podemos implementarla en gmesh a través del siguiente archivo template

corte.tpl

```

// longitud característica del elemento
lc = $1;
// algoritmo de mallado (1 = meshadapt, 8 = delquad)
Mesh.Algorithm = $2;
// recombina triangulos? (1 = cuadrangulos, 0 = triangulos)
//Mesh.RecombineAll = 1;

h = $3; // longitud activa
r = $4; // radio del nucleo
d = $5; // espesor del reflector radial
b = $6; // altura del reflector inferior
t = $7; // altura del reflector superior

```

```

w = $8;      // ancho de las barras de control
l = $9;      // longitud de las barras de control

// posicion de la punta de la barra de control
// con respecto al reflector superior
x1 = $10;
y1 = $11;
theta1 = $12;      // angulo de inclinacion
p1 = $13;          // insercion

x2 = $14;
y2 = $15;
theta2 = $16;      // angulo de inclinacion
p2 = $17;          // insercion

// nucleo
Point(1) = {0, 0, 0, lc};
Point(2) = {r, 0, 0, lc};
Point(3) = {-r, 0, 0, lc};
Point(4) = {r, h, 0, lc};
Point(5) = {-r, h, 0, lc};

// reflector
Point(6) = {r+d, -b, 0, lc};
Point(7) = {r+d, h+t, 0, lc};
Point(8) = {-(r+d), h+t, 0, lc};
Point(9) = {-(r+d), -b, 0, lc};

// barra de control 1
Point(10) = {x1+0.5*w*cos(theta1), h+t+y1+0.5*w*sin(theta1), 0, lc};
Point(11) = {x1-0.5*w*cos(theta1), h+t+y1-0.5*w*sin(theta1), 0, lc};

Point(12) = {x1+l*sin(theta1)+0.5*w*cos(theta1), h+t+y1-l*cos(theta1)+0.5*w* ←
    sin(theta1), 0, lc};
Point(13) = {x1+l*sin(theta1)-0.5*w*cos(theta1), h+t+y1-l*cos(theta1)-0.5*w* ←
    sin(theta1), 0, lc};

Point(14) = {x1+0.5*w*cos(theta1)+(y1+0.5*w*sin(theta1))/(Tan(Pi/2-theta1)), ←
    h+t, 0, lc};
Point(15) = {x1-0.5*w*cos(theta1)+(y1-0.5*w*sin(theta1))/(Tan(Pi/2-theta1)), ←
    h+t, 0, lc};

Point(16) = {x1+0.5*w*cos(theta1)+(y1+0.5*w*sin(theta1)+t)/(Tan(Pi/2-theta1) ←
    ), h, 0, lc};
Point(17) = {x1-0.5*w*cos(theta1)+(y1-0.5*w*sin(theta1)+t)/(Tan(Pi/2-theta1) ←
    ), h, 0, lc};

Point(18) = {x1+p1*l*sin(theta1)-0.5*w*cos(theta1), h+t+y1-p1*l*cos(theta1) ←
    -0.5*w*sin(theta1), 0, lc};
Point(19) = {x1+p1*l*sin(theta1)+0.5*w*cos(theta1), h+t+y1-p1*l*cos(theta1) ←
    +0.5*w*sin(theta1), 0, lc};

// barra de control 2
Point(20) = {x2+0.5*w*cos(theta2), h+t+y2+0.5*w*sin(theta2), 0, lc};
Point(21) = {x2-0.5*w*cos(theta2), h+t+y2-0.5*w*sin(theta2), 0, lc};

Point(22) = {x2+l*sin(theta2)+0.5*w*cos(theta2), h+t+y2-l*cos(theta2)+0.5*w* ←

```

```

Sin(theta2), 0, lc};
Point(23) = {x2+l*Sin(theta2)-0.5*w*Cos(theta2), h+t+y2-l*Cos(theta2)-0.5*w* ←
Sin(theta2), 0, lc};

Point(24) = {x2+0.5*w*Cos(theta2)+(y2+0.5*w*Sin(theta2))/(Tan(Pi/2-theta2)), ←
h+t, 0, lc};
Point(25) = {x2-0.5*w*Cos(theta2)+(y2-0.5*w*Sin(theta2))/(Tan(Pi/2-theta2)), ←
h+t, 0, lc};

Point(26) = {x2+0.5*w*Cos(theta2)+(y2+0.5*w*Sin(theta2)+t)/(Tan(Pi/2-theta2) ←
), h, 0, lc};
Point(27) = {x2-0.5*w*Cos(theta2)+(y2-0.5*w*Sin(theta2)+t)/(Tan(Pi/2-theta2) ←
), h, 0, lc};

Point(28) = {x2+p2*l*Sin(theta2)-0.5*w*Cos(theta2), h+t+y2-p2*l*Cos(theta2) ←
-0.5*w*Sin(theta2), 0, lc};
Point(29) = {x2+p2*l*Sin(theta2)+0.5*w*Cos(theta2), h+t+y2-p2*l*Cos(theta2) ←
+0.5*w*Sin(theta2), 0, lc};

Line(1) = {13, 18};
Line(2) = {18, 17};
Line(3) = {17, 15};
Line(4) = {15, 11};
Line(5) = {11, 10};
Line(6) = {10, 14};
Line(7) = {14, 16};
Line(8) = {16, 19};
Line(9) = {19, 12};
Line(10) = {12, 13};

Line(21) = {23, 28};
Line(22) = {28, 27};
Line(23) = {27, 25};
Line(24) = {25, 21};
Line(25) = {21, 20};
Line(26) = {20, 24};
Line(27) = {24, 26};
Line(28) = {26, 29};
Line(29) = {29, 22};
Line(30) = {22, 23};
Line(31) = {18, 19};
Line(32) = {28, 29};

Line(33) = {1, 2};
Line(34) = {2, 4};
Line(35) = {4, 26};
Line(36) = {27, 16};
Line(37) = {17, 5};
Line(38) = {5, 3};
Line(39) = {3, 1};
Line(40) = {6, 7};
Line(41) = {7, 24};
Line(42) = {25, 14};
Line(43) = {15, 8};
Line(44) = {8, 9};
Line(45) = {9, 6};

Line(46) = {17, 16};
Line(47) = {27, 26};

```



```
Line Loop(48) = {38, 39, 33, 34, 35, 28, 29, 30, 21, 22, 36, 8, 9, 10, 1, 2, ←
37};
Plane Surface(49) = {48};

Line Loop(50) = {34, 35, -27, -41, -40, -45, -44, -43, -3, 37, 38, 39, 33};
Plane Surface(51) = {50};

Line Loop(52) = {36, -7, -42, -23};
Plane Surface(53) = {52};

Line Loop(54) = {1, 31, 9, 10};
Plane Surface(55) = {54};

Line Loop(56) = {21, 32, 29, 30};
Plane Surface(57) = {56};

Line Loop(58) = {2, 46, 8, -31};
Plane Surface(59) = {58};

Line Loop(60) = {22, 47, 28, -32};
Plane Surface(61) = {60};

Line Loop(62) = {3, 4, 5, 6, 7, -46};
Plane Surface(63) = {62};

Line Loop(64) = {24, 25, 26, 27, -47, 23};
Plane Surface(65) = {64};

Physical Line("surface") = {44, 45, 40, 41, 26, 25, 24, 42, 6, 5, 4, 43};

Physical Surface("fuel") = {49};
Physical Surface("refl") = {51, 53};
Physical Surface("fuel+tube") = {55, 57};
Physical Surface("fuel+tube+rod") = {59, 61};
Physical Surface("refl+tube+rod") = {63, 65};

Recombine Surface(49);
Recombine Surface(51);
Recombine Surface(53);
```

en el cual los wildcards \$n son reemplazados por valores numéricos apropiados por milonga para generar el archivo de geometría corte.geo que gmesh lee para generar la malla según las condiciones solicitadas. Podemos juntar todos los parámetros geométricos como variables de milonga en un archivo de entrada separado, que llamamos geometria.mil y que incluimos desde el input principal:

geometria.mil

```
h = 550      # longitud activa
r = 300      # radio del nucleo
d = 45       # espesor del reflector radial
b = 40       # altura del reflector inferior
t = 30       # altura del reflector superior

w = 25       # ancho de las barras de control
l = 600      # longitud de las barras de control

# posicion de la punta de la barra de control
```



```
# con respecto al reflector superior
x1 = -250
y1 = 40
theta1 = 21*pi/180 # angulo de inclinacion barra 1
p1 = 0.75          # insercion barra 1

x2 = 250
y2 = 40
theta2 = -16*pi/180 # angulo de inclinacion barra 1
p2 = 0.45          # insercion barra 1
```

A modo de ilustración, en la figura 5.88 mostramos una captura de pantalla de la interfaz gráfica de gmsh al cargar la geometría con los parámetros propuestos. Notar que la inserción de las barras de control es un parámetro continuo y puede tomar cualquier valor real. La malla resultante será tal que los extremos de las barras de control coincidirán con bordes de celdas/elementos.

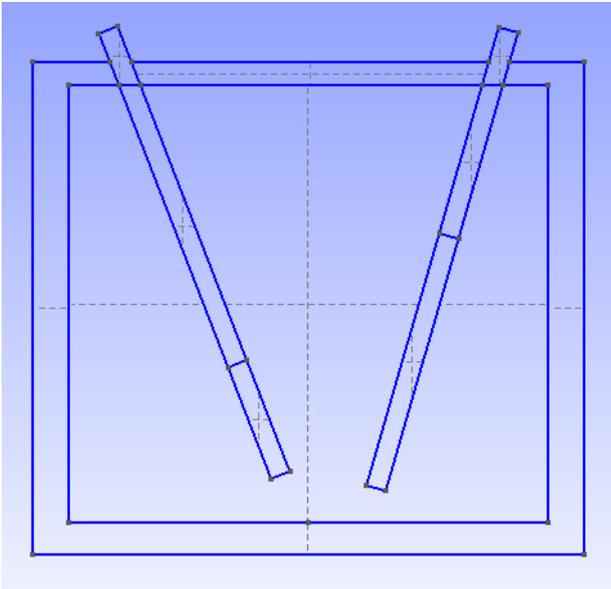


Figura 5.88 – Interpretación por parte de gmsh de la geometría continua dada por el archivo de template geometria.tpl con los parámetros del archivo geometria.mil.

Proponemos que las secciones eficaces del combustible dependan sólo del quemado $q(x, y)$, que a su vez tiene una cierta distribución en el espacio. A modo de ilustración, mostramos la dependencia de Σ_a y $\nu\Sigma_f$ con q en las figuras 5.89 y 5.90 con funciones definidas por interpolación lineal de puntos dispersos, que en cálculos reales deben ser calculados con códigos de celda.

Para comparar la distribución de flujo resultante teniendo en cuenta o no el quemado, proponemos una distribución de quemados $q(x, y)$ definida algebraicamente como

$$q(x, y) = \text{beq} \cdot \left\{ 1000 + 7000 \cdot \sin\left(\pi \cdot \frac{y}{h}\right) \cdot \left[1 - \left(\frac{x}{r}\right)^6 \right] \right\} \quad [\text{MW} \cdot \text{día} \cdot \text{tonU}^{-1}] \quad (5.12)$$

donde beq es un flag que vale cero si el núcleo es fresco (beginning of life) o uno si el núcleo alcanzó el quemado de equilibrio, que suponemos igual a la distribución continua propuesta

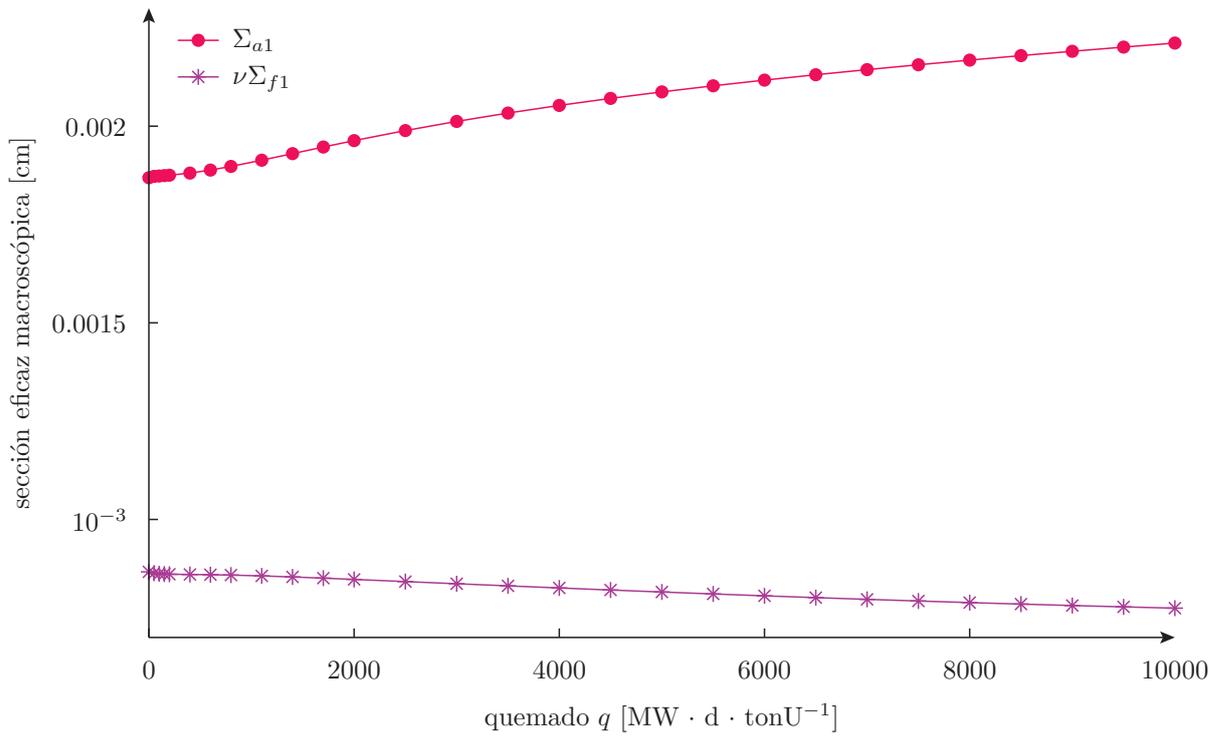


Figura 5.89 – Dependencia de la sección eficaz de absorción del grupo rápido y de la sección eficaz de ν -fisiones del grupo rápido con el quemado q .

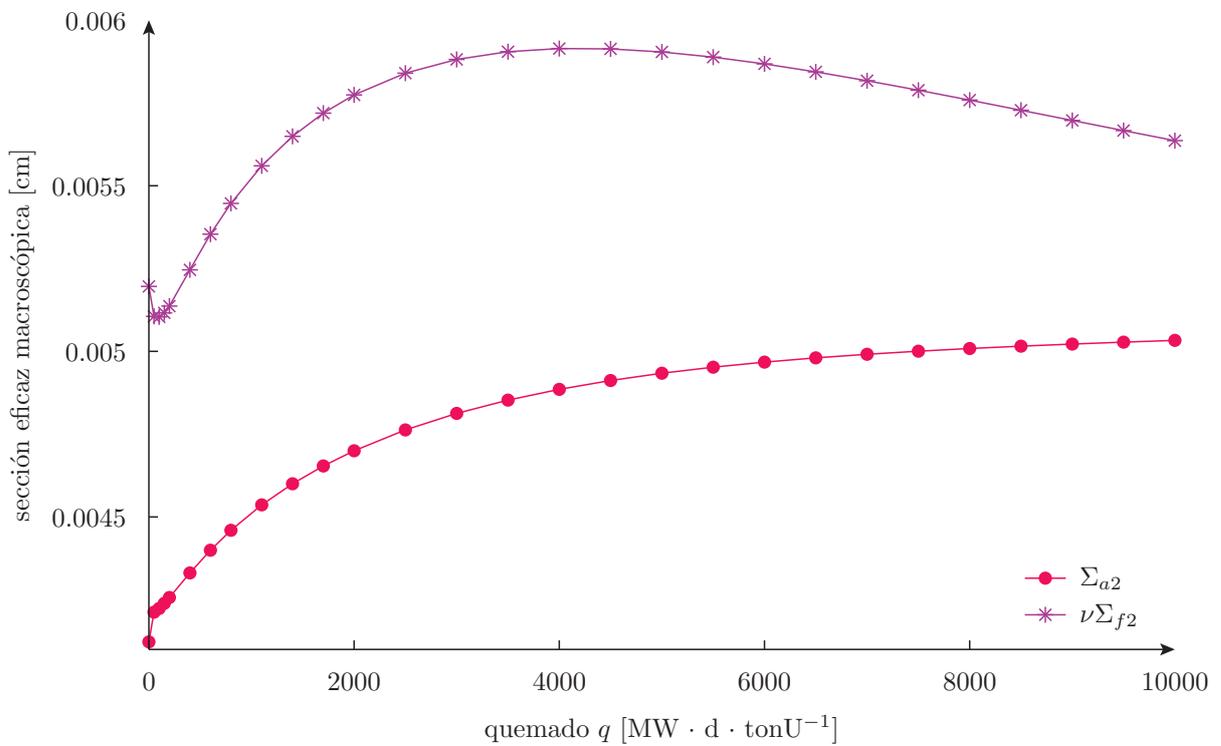


Figura 5.90 – Dependencia de la sección eficaz de absorción del grupo térmico y de la sección eficaz de ν -fisiones del grupo térmico con el quemado q .



dentro de las llaves. Además, dentro de milonga limitamos la función $q(x, y)$ a valores no nulos sólo en el intervalo $[-r, r] \times [0, h]$. Este límite no es necesario para el cálculo, ya que $q(x, y)$ tiene sólo sentido al calcular la sección eficaz macroscópica del combustible, pero por completitud y para poder graficar la distribución de quemado propuesta en gms, es que hacemos cero $q(x, y)$ si $\|x\| > r \wedge y < 0 \wedge y > h$. Preparamos por un lado un archivo llamado `xs.dat` con la dependencia de las diferentes secciones eficaces macroscópicas en función del quemado en columnas y por otro un archivo de entrada de milonga auxiliar que define todos los materiales asociados a las entidades físicas del archivo de geometría que llamamos `xs.mil`:

`xs.mil`

```
# distribucion de quemado, si beq = 0 entonces
# indica beginning of life y q(x,y) = 0
# hacemos q(x,y) = 0 si y < 0, y > h o |x| > r
# para que al escribir el quemado en funcion de
# x e y sepamos bien donde termina
q(x,y) := beq*(1000 + 7000*sin(pi*y/h)*(1-(x/r)^6)) * (y > 0 & y < h & abs(x ↔
) < r)

# XS en funcion del quemado
FUNCTION dif1(b)      FILE xs.dat COLUMNS 1 2
FUNCTION dif2(b)      FILE xs.dat COLUMNS 1 3
FUNCTION sigmaa1(b)   FILE xs.dat COLUMNS 1 4
FUNCTION sigma12(b)   FILE xs.dat COLUMNS 1 5
FUNCTION sigma21(b)   FILE xs.dat COLUMNS 1 6
FUNCTION sigmaa2(b)   FILE xs.dat COLUMNS 1 7
FUNCTION nusigf1(b)   FILE xs.dat COLUMNS 1 8
FUNCTION nusigf2(b)   FILE xs.dat COLUMNS 1 9

MATERIAL fuel {
  D_1      dif1(q(x,y))
  D_2      dif2(q(x,y))
  SigmaT_1 sigmaa1(q(x,y))+sigma12(q(x,y))
  SigmaS_1->2 sigma12(q(x,y))
  SigmaS_2->1 sigma21(q(x,y))
  SigmaT_2 sigmaa2(q(x,y))+sigma21(q(x,y))
  nuSigmaF_1 nusigf1(q(x,y))
  nuSigmaF_2 nusigf2(q(x,y))
}

MATERIAL refl {
  D_1      1.40
  D_2      0.95
  SigmaT_1 1.51e-4+1.23e-2
  SigmaS_1->2 1.23e-2
  SigmaS_2->1 6.89e-5
  SigmaT_2 1.06e-3+6.89e-5
}

MATERIAL fuel+tube INCREMENTAL fuel {
  D_1      0
  D_2      0
  SigmaT_1 -1.10e-4
  SigmaS_1->2 -1.3e-5
  SigmaS_2->1 +1.51e-06
  SigmaT_2 +9.84e-05
  nuSigmaF_1 -5.00e-06
  nuSigmaF_2 +1.11e-05
}
```



```

}

MATERIAL fuel+tube+rod INCREMENTAL fuel+tube {
  D_1          0
  D_2          0
  SigmaT_1     +2.32e-04
  SigmaS_1->2  +5.46e-05
  SigmaS_2->1  +2.90e-05
  SigmaT_2     +2.76e-03
  nuSigmaF_1   -2.75e-05
  nuSigmaF_2   +1.92e-04
}

MATERIAL refl+tube+rod INCREMENTAL refl {
  D_1          0
  D_2          0
  SigmaT_1     -1.10e-04
  SigmaS_1->2  -1.37e-04
  SigmaS_2->1  +1.51e-06
  SigmaT_2     +9.83e-05
}

```

Al utilizar el keyword incremental, estamos indicando que las expresiones dadas deben ser sumadas algebraicamente a las secciones eficaces del material base.

Si bien ya hemos definido muchos de los parámetros del problema, aún quedan algunos por fijar. Como venimos haciendo a lo largo del capítulo, queremos utilizar el mismo archivo de entrada de milonga para resolver varios casos y comparar resultados. Por lo tanto, las opciones que nos faltan las leemos desde la línea de comandos. El primer parámetro a modificar es el flag beq, que indica núcleo fresco (beq=0) o núcleo en equilibrio (beq=1), es decir con la distribución de quemados $q(x, y)$ dada por la expresión (5.12). La segunda es la longitud característica de la malla l_c , para que podamos estudiar una malla gruesa y una fina. La siguiente es el algoritmo que queremos que use gmsh para malla la geometría continua. En este caso, vamos a comparar los métodos meshadapt y delquad sin entrar en detalles sobre cada algoritmo, sólo para observar gráficamente las mallas resultantes. Y, finalmente, para no escaparnos al objetivo principal del trabajo, elegimos resolver la ecuación de difusión de neutrones con el esquema basado en volúmenes finitos del capítulo 3 o con el esquema basado en elementos finitos desarrollado en el capítulo 4.

cor.te.was

```

# resuelve el keff y la distribucion de flujo de un simil corte
# transversal de un reactor tipo PHWR con barras de control
# inclinadas a dos grupos de energia
# se puede elegir por linea de comandos:
# $1 - nucleo fresco o con una distribucion de quemado
#     dada por una expresion algebraica de x,y
# $2 - longitud caracteristica de la malla (mas chica -> mas fina)
# $3 - algoritmo de mallado
# $4 - metodo de solucion , volumes o elements

beq = $1      # flag que indica BEQ o BOL
lc = $2      # longitud caracteristica del elemento
alg = $3     # algoritmo de mallado (1 = meshadapt , 8 = delquad)
SCHEME $4   # volumes o elementos

# definicion de los parametros de la geometria

```

```

INCLUDE geometria.was

# condicion de contorno de robin
PHYSICAL_ENTITY NAME surface BC robin -0.5

# saludamos al usuario , porque somos un codigo polite
PRINT TEXT "=====
IF beq=0&alg=1 PRINT TEXT "\#_resolviendo_nucleo_fresco_con_$4,_lc=_$2_y_ ←
    alg=_meshadapt"
IF beq=0&alg=8 PRINT TEXT "\#_resolviendo_nucleo_fresco_con_$4,_lc=_$2_y_ ←
    alg=_delquad"
IF beq=1&alg=1 PRINT TEXT "\#_resolviendo_nucleo_en_equilibrio_con_$4,_lc=_ ←
    $2_y_alg=_meshadapt"
IF beq=1&alg=8 PRINT TEXT "\#_resolviendo_nucleo_en_equilibrio_con_$4,_lc=_ ←
    $2_y_alg=_delquad"

# mallamos (y medimos el tiempo)
t1 = clock()
PRINT TEXT "\#_+_mallando..._" NONEWLINE
TEMPLATE corte.tpl corte-$2-$3.geo lc alg h r d b t w l x1 y1 theta1 p1 ←
    x2 y2 theta2 p2
SHELL "gmsh-$2-$3-1_corte-$2-$3.geo>_/dev/null"
PRINT TEXT "listo!"
t2 = clock()
MESH DIMENSIONS 2 DEGREES 2 FILE_PATH corte-$2-$3.msh
read_mesh_time = clock()-t2
mesh_time = t2-t1
INCLUDE xs.was

# resolvemos (y medimos el tiempo)
PRINT TEXT "\#_+_resolviendo..." NONEWLINE
MILONGA SOLVE
PRINT TEXT "listo!"

PRINT TEXT "\#_keff=__" %.6f keff
PRINT TEXT "\#"
PRINT TEXT "\#_tiempos_(wall)"
PRINT TEXT "\#_+_mallado_(gmsh)_____=" %.2f mesh_time TEXT "_segs"
PRINT TEXT "\#_+_lectura_de_malla_____=" %.2f read_mesh_time+init_time ←
TEXT "_segs"
PRINT TEXT "\#_+_armado_de_matrices_____=" %.2f build_matrices_time TEXT ←
    "_segs"
PRINT TEXT "\#_+_solucion_del_problema_=" %.2f solve_time TEXT "_segs"
PRINT TEXT "\#_+_="
PRINT TEXT "\#_+_=" %.2f mesh_time+read_mesh_time+ ←
    init_time+build_matrices_time+solve_time TEXT "_segs"
PRINT TEXT "=====

OUTPUT_FILE flujo-post corte-$1-$2-$3-$4.pos
MESH_POST FILE flujo-post q nuSigmaF_1 phi_1 phi_2

```

```

$ cat loop.sh
for beq in 0 1; do
  for lc in 14 9; do
    for alg in 1 8; do
      for method in volumes elements; do
        milonga corte.was $beq $lc $alg $method
        gmsh corte-$beq-$lc-$alg-$method.pos corte.opt
        convert -trim 0.png corte-$beq-$lc-$alg-$method-bu.png

```

```

convert -trim 1.png corte-$beq-$lc-$alg-$method-sigmafl.png
convert -trim 2.png corte-$beq-$lc-$alg-$method-phi1.png
convert -trim 3.png corte-$beq-$lc-$alg-$method-phi2.png
done
done
done
done
$ ./loop.sh
=====
# resolviendo nucleo fresco con volumes, lc = 14 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.080495
#
# tiempos (wall)
# + mallado (gmsh)    = 1.05      segs
# + lectura de malla  = 0.04      segs
# + armado de matrices = 0.11      segs
# + solucion del problema = 0.09      segs
# -----
#                               1.29      segs
=====
# resolviendo nucleo fresco con elements, lc = 14 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.079742
#
# tiempos (wall)
# + mallado (gmsh)    = 0.97      segs
# + lectura de malla  = 0.01      segs
# + armado de matrices = 0.63      segs
# + solucion del problema = 0.15      segs
# -----
#                               1.76      segs
=====
# resolviendo nucleo fresco con volumes, lc = 14 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.080121
#
# tiempos (wall)
# + mallado (gmsh)    = 0.43      segs
# + lectura de malla  = 0.03      segs
# + armado de matrices = 0.10      segs
# + solucion del problema = 0.08      segs
# -----
#                               0.64      segs
=====
# resolviendo nucleo fresco con elements, lc = 14 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.079749
#
# tiempos (wall)
# + mallado (gmsh)    = 0.41      segs
# + lectura de malla  = 0.01      segs
# + armado de matrices = 0.59      segs
# + solucion del problema = 0.12      segs

```

```

# -----
#                               1.14      segs
=====
# resolviendo nucleo fresco con volumes, lc = 9 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.080402
#
# tiempos (wall)
# + mallado (gmsh)    = 2.25      segs
# + lectura de malla  = 0.22      segs
# + armado de matrices = 0.28      segs
# + solucion del problema = 0.31      segs
# -----
#                               3.06      segs
=====
# resolviendo nucleo fresco con elements, lc = 9 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.079796
#
# tiempos (wall)
# + mallado (gmsh)    = 2.26      segs
# + lectura de malla  = 0.03      segs
# + armado de matrices = 1.54      segs
# + solucion del problema = 0.51      segs
# -----
#                               4.34      segs
=====
# resolviendo nucleo fresco con volumes, lc = 9 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.079674
#
# tiempos (wall)
# + mallado (gmsh)    = 0.93      segs
# + lectura de malla  = 0.18      segs
# + armado de matrices = 0.24      segs
# + solucion del problema = 0.27      segs
# -----
#                               1.63      segs
=====
# resolviendo nucleo fresco con elements, lc = 9 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              1.079788
#
# tiempos (wall)
# + mallado (gmsh)    = 0.93      segs
# + lectura de malla  = 0.03      segs
# + armado de matrices = 1.36      segs
# + solucion del problema = 0.39      segs
# -----
#                               2.71      segs
=====
# resolviendo nucleo en equilibrio con volumes, lc = 14 y alg = meshadapt

```

```

# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.993343
#
# tiempos (wall)
# + mallado (gmsh)   = 0.97      segs
# + lectura de malla = 0.05      segs
# + armado de matrices = 0.12     segs
# + solucion del problema = 0.10   segs
# -----
#                               1.24      segs
=====
# resolviendo nucleo en equilibrio con elements, lc = 14 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992539
#
# tiempos (wall)
# + mallado (gmsh)   = 0.97      segs
# + lectura de malla = 0.01      segs
# + armado de matrices = 0.63     segs
# + solucion del problema = 0.16   segs
# -----
#                               1.77      segs
=====
# resolviendo nucleo en equilibrio con volumes, lc = 14 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992180
#
# tiempos (wall)
# + mallado (gmsh)   = 0.42      segs
# + lectura de malla = 0.04      segs
# + armado de matrices = 0.11     segs
# + solucion del problema = 0.09   segs
# -----
#                               0.66      segs
=====
# resolviendo nucleo en equilibrio con elements, lc = 14 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992570
#
# tiempos (wall)
# + mallado (gmsh)   = 0.42      segs
# + lectura de malla = 0.01      segs
# + armado de matrices = 0.58     segs
# + solucion del problema = 0.15   segs
# -----
#                               1.16      segs
=====
# resolviendo nucleo en equilibrio con volumes, lc = 9 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.993484
#
# tiempos (wall)

```

```

# + mallado (gms)      = 2.18   segs
# + lectura de malla  = 0.23   segs
# + armado de matrices = 0.26   segs
# + solucion del problema = 0.35   segs
# -----
#                               3.03   segs
=====
# resolviendo nucleo en equilibrio con elements, lc = 9 y alg = meshadapt
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992566
#
# tiempos (wall)
# + mallado (gms)    = 2.23   segs
# + lectura de malla = 0.03   segs
# + armado de matrices = 1.53   segs
# + solucion del problema = 0.50   segs
# -----
#                               4.30   segs
=====
# resolviendo nucleo en equilibrio con volumes, lc = 9 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992321
#
# tiempos (wall)
# + mallado (gms)    = 0.93   segs
# + lectura de malla = 0.18   segs
# + armado de matrices = 0.24   segs
# + solucion del problema = 0.29   segs
# -----
#                               1.64   segs
=====
# resolviendo nucleo en equilibrio con elements, lc = 9 y alg = delquad
# + mallando...      listo!
# + resolviendo...   listo!
# keff =              0.992562
#
# tiempos (wall)
# + mallado (gms)    = 0.94   segs
# + lectura de malla = 0.02   segs
# + armado de matrices = 1.38   segs
# + solucion del problema = 0.39   segs
# -----
#                               2.72   segs
=====
$ gms corte-14-1.geo corte-14-1.msh mesh.opt
$ inkscape -D -A corte-mesh-14-1.pdf mesh.svg
$ gms corte-9-1.geo corte-9-1.msh mesh.opt
$ inkscape -D -A corte-mesh-9-1.pdf mesh.svg
$ gms corte-14-8.geo corte-14-8.msh mesh.opt
$ inkscape -D -A corte-mesh-14-8.pdf mesh.svg
$ gms corte-9-8.geo corte-9-8.msh mesh.opt
$ inkscape -D -A corte-mesh-9-8.pdf mesh.svg
$
$

```

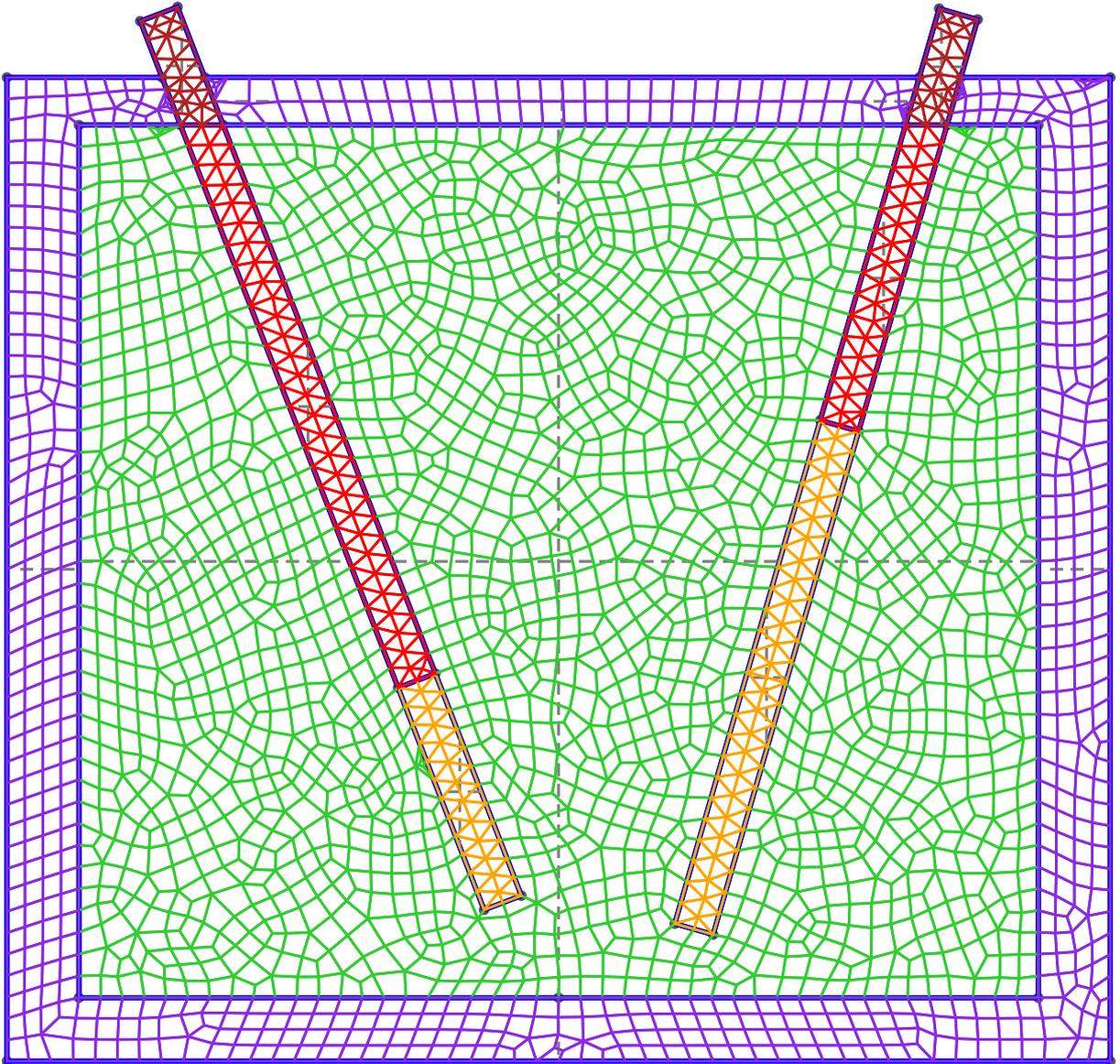


Figura 5.91 – Malla generada por gmsch con el algoritmo meshadapt para $\ell_c = 14$ cm

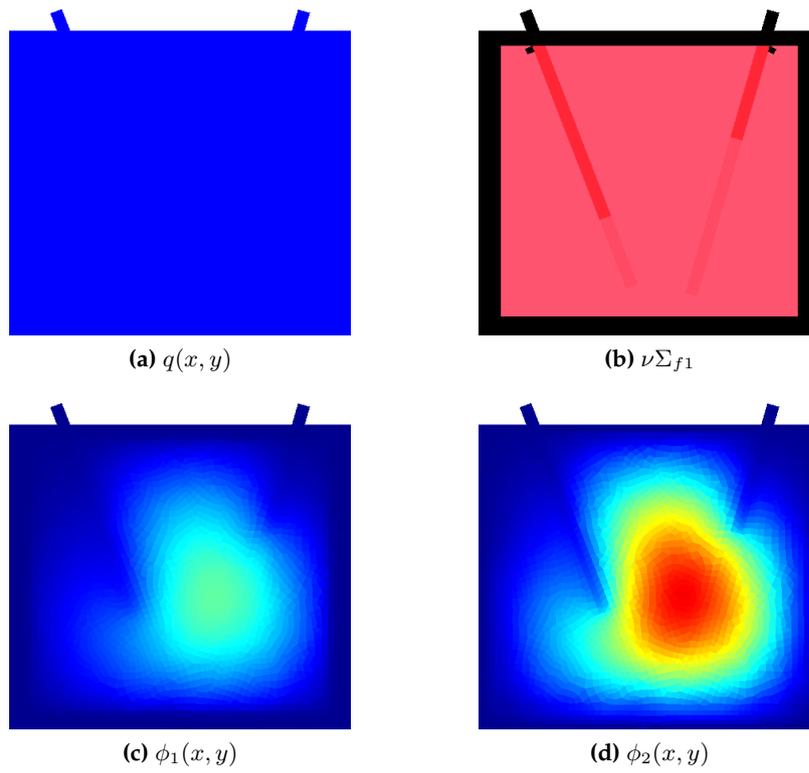


Figura 5.92 – Núcleo fresco, $\ell_c = 14$, algoritmo meshadapt, volúmenes finitos

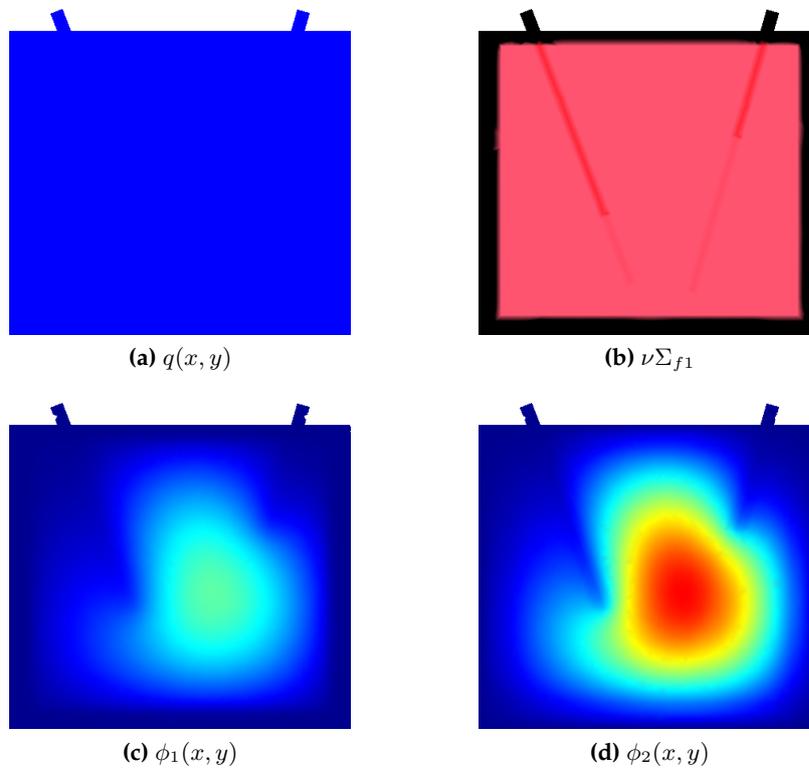


Figura 5.93 – Núcleo fresco, $\ell_c = 14$, algoritmo meshadapt, elementos finitos

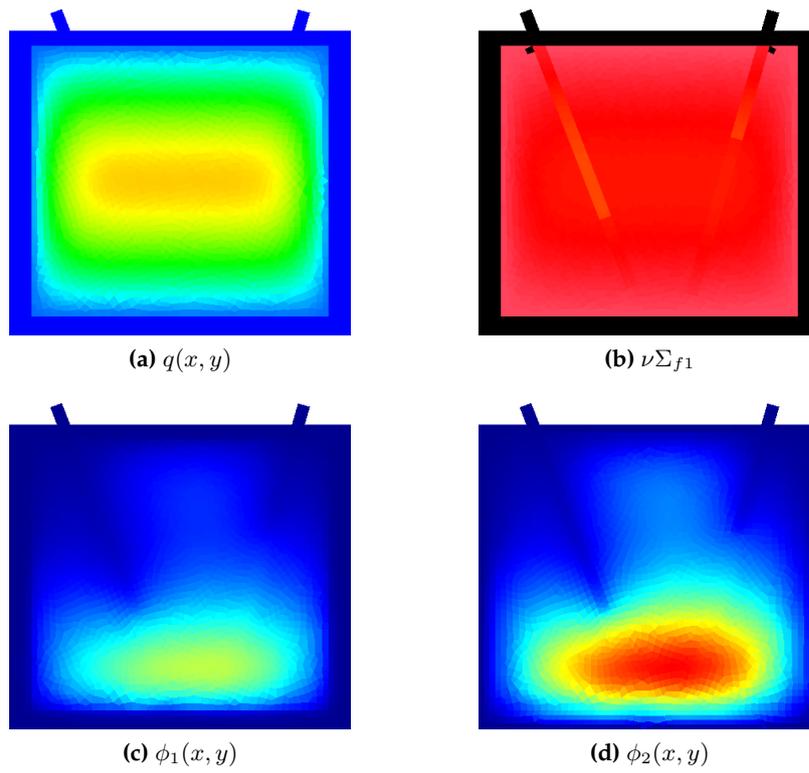


Figura 5.94 – Núcleo en equilibrio, $\ell_c = 14$, algoritmo meshadapt, volúmenes finitos

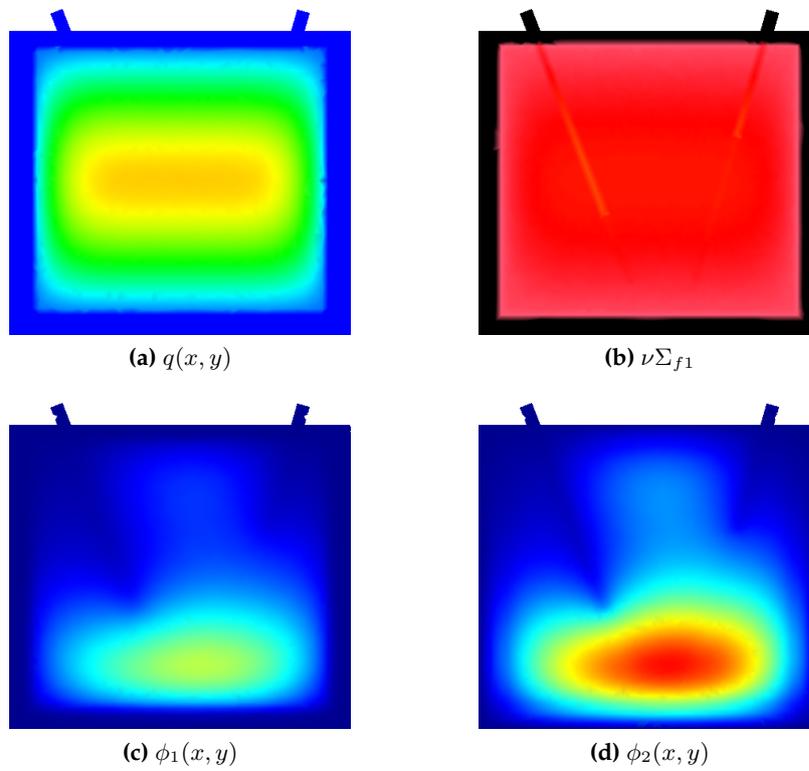


Figura 5.95 – Núcleo en equilibrio, $\ell_c = 14$, algoritmo meshadapt, elementos finitos

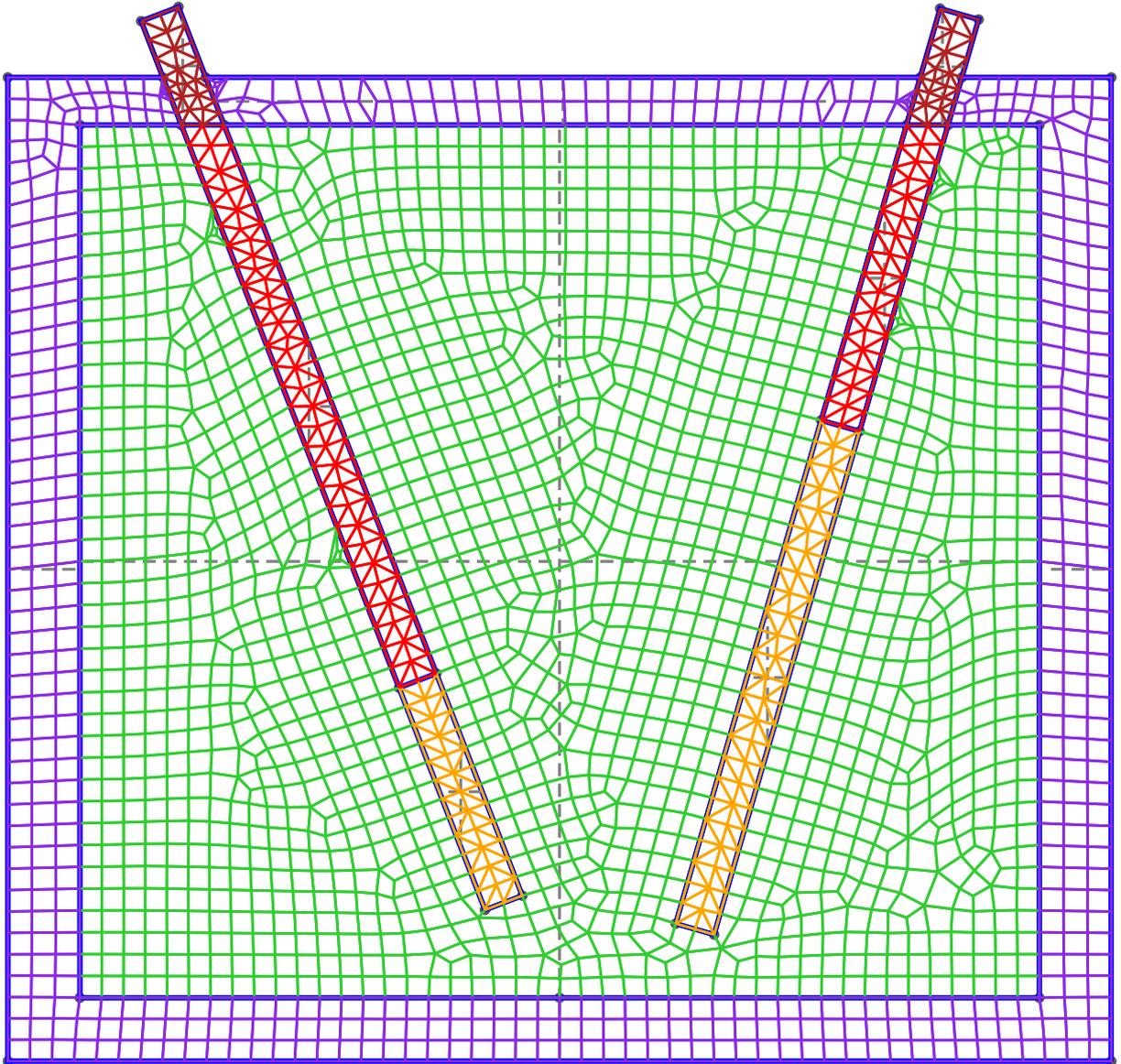


Figura 5.96 – Malla generada por gmsh con el algoritmo delquad para $\ell_c = 14$ cm

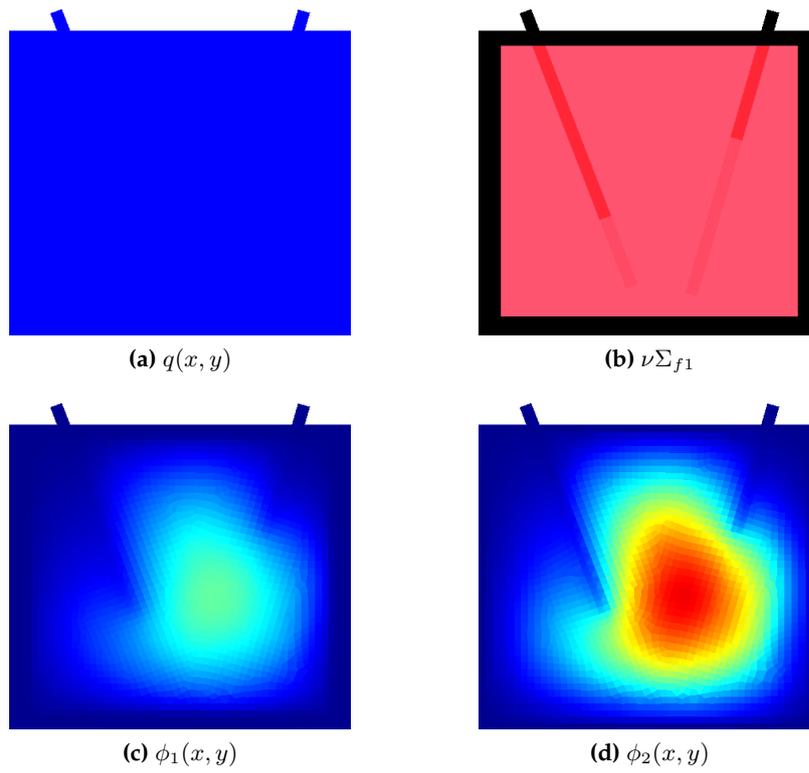


Figura 5.97 – Núcleo fresco, $\ell_c = 14$, algoritmo delquad, volúmenes finitos

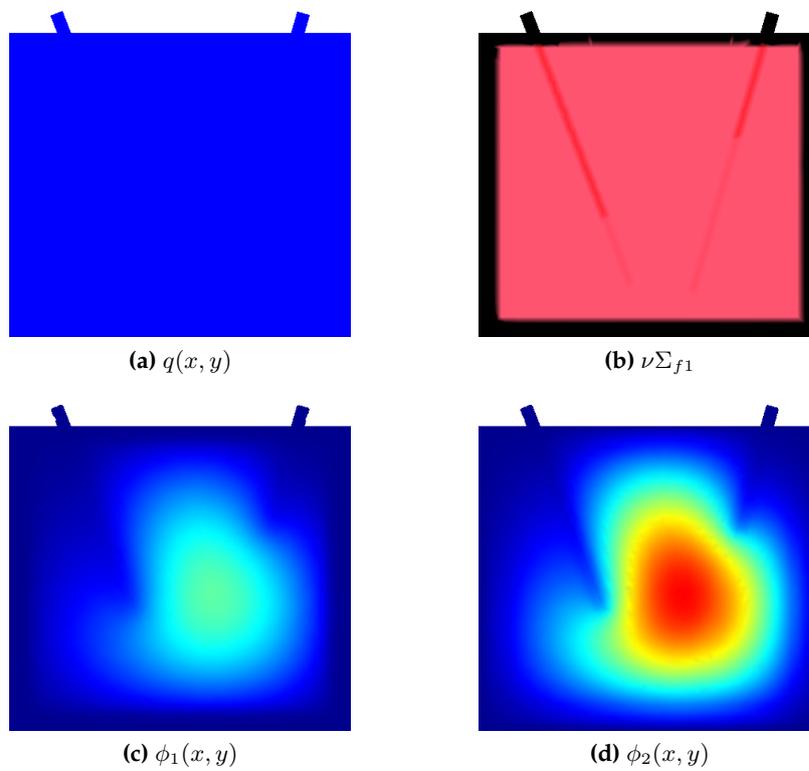


Figura 5.98 – Núcleo fresco, $\ell_c = 14$, algoritmo delquad, elementos finitos

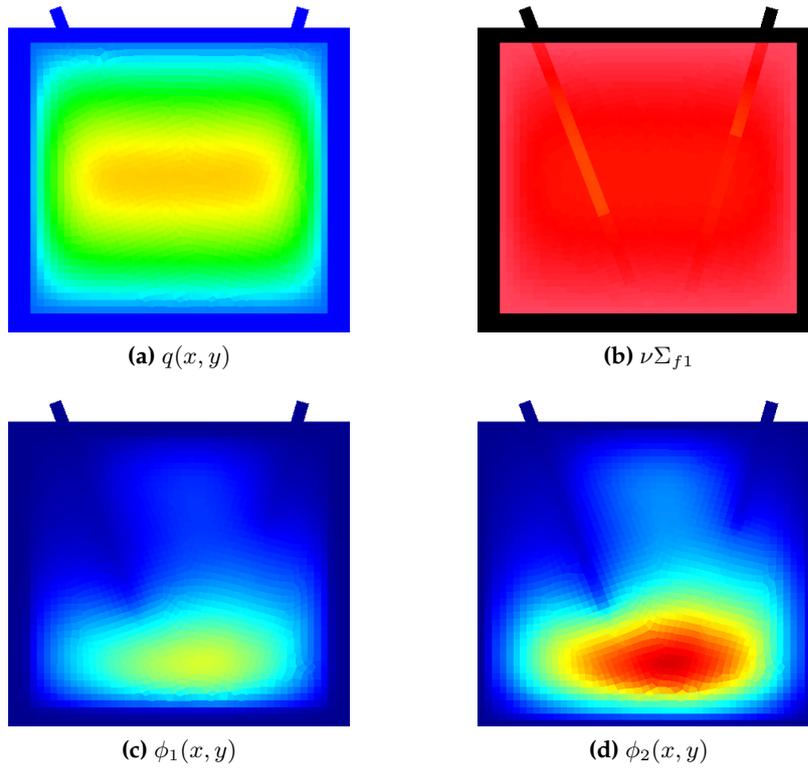


Figura 5.99 – Núcleo en equilibrio, $\ell_c = 14$, algoritmo delquad, volúmenes finitos

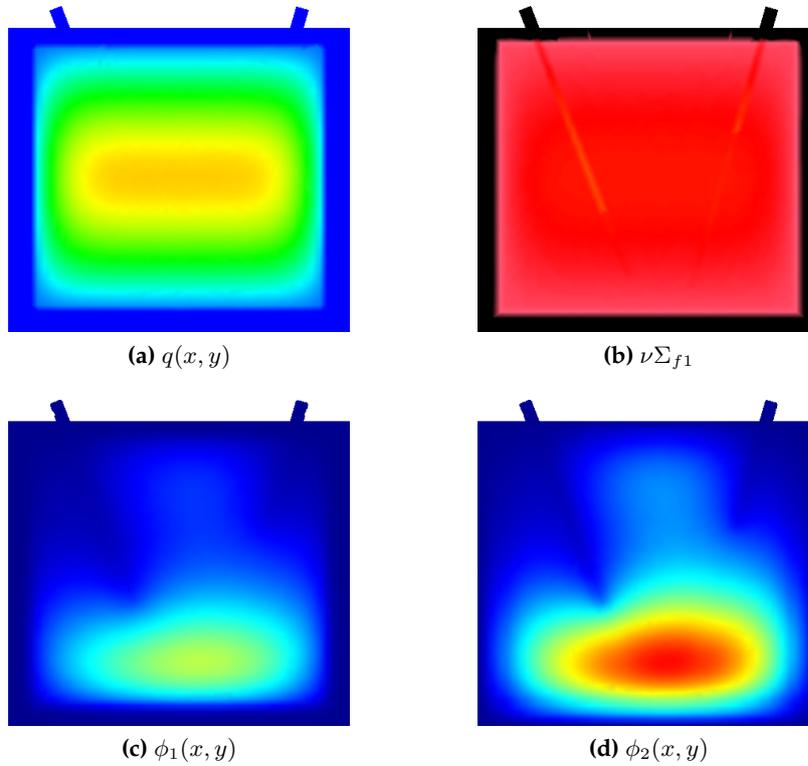


Figura 5.100 – Núcleo en equilibrio, $\ell_c = 14$, algoritmo delquad, elementos finitos

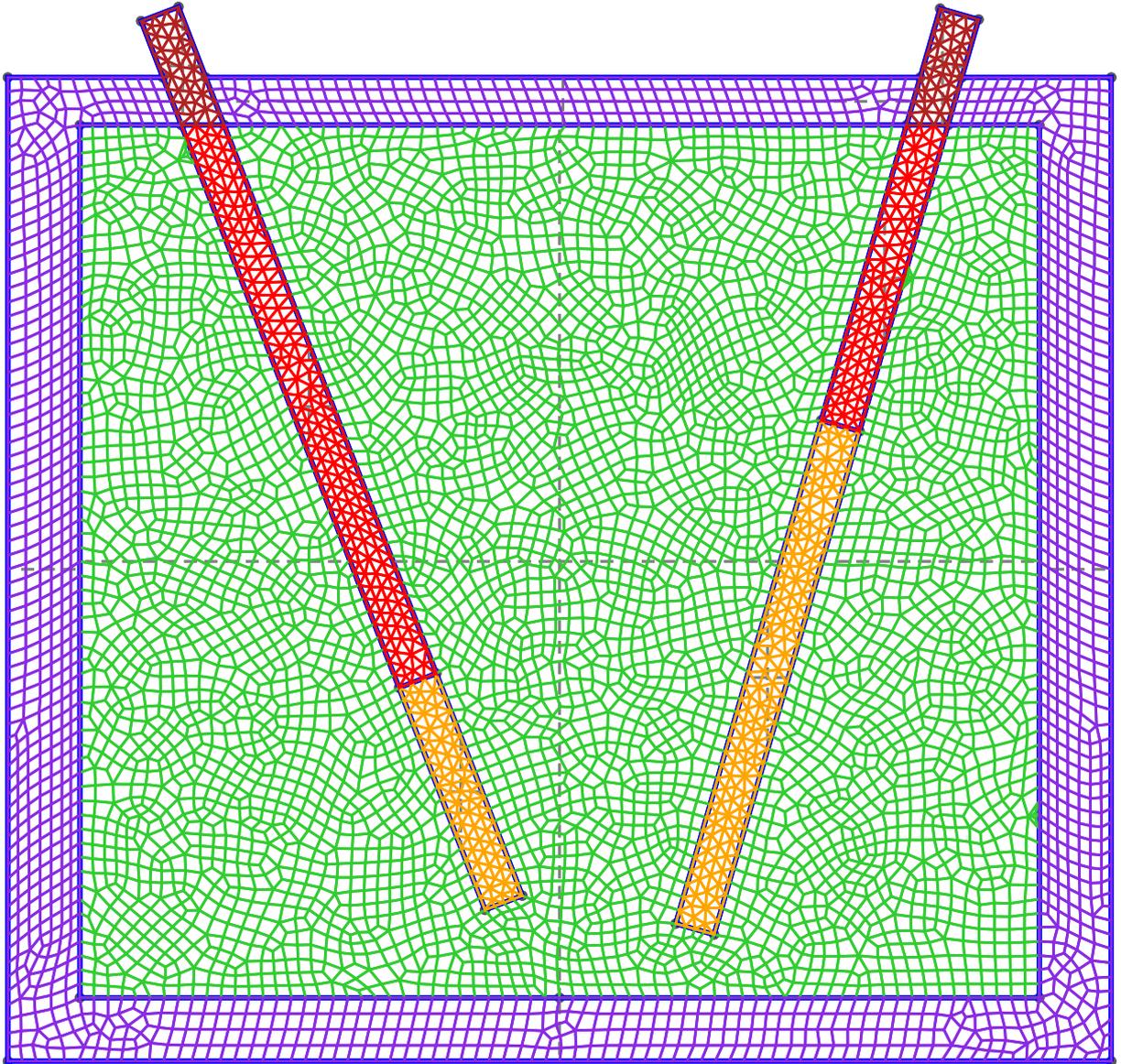


Figura 5.101 – Malla generada por gmsh con el algoritmo meshadapt para $\ell_c = 9$ cm

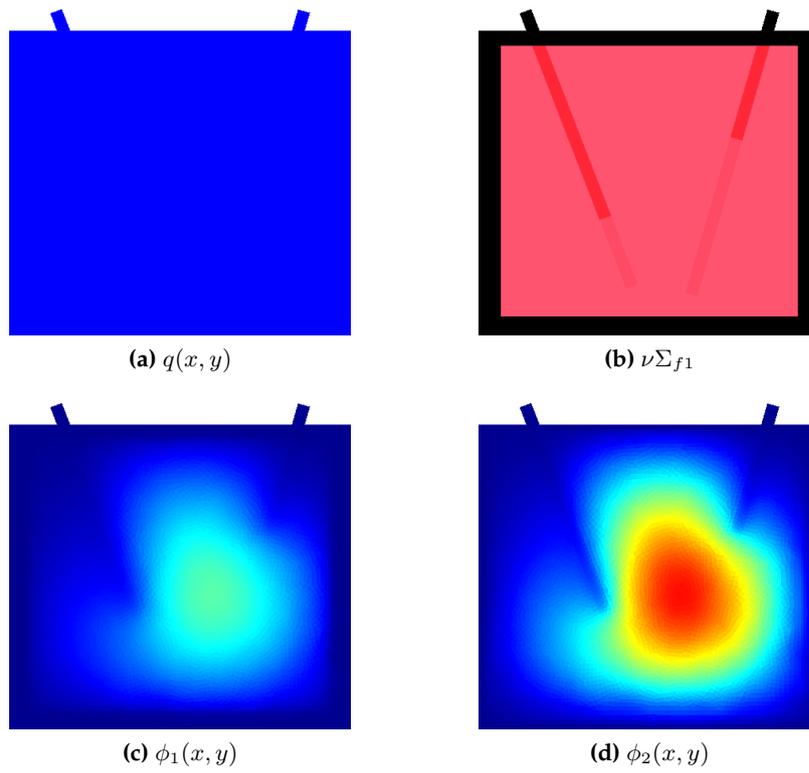


Figura 5.102 – Núcleo fresco, $\ell_c = 9$, algoritmo meshadapt, volúmenes finitos

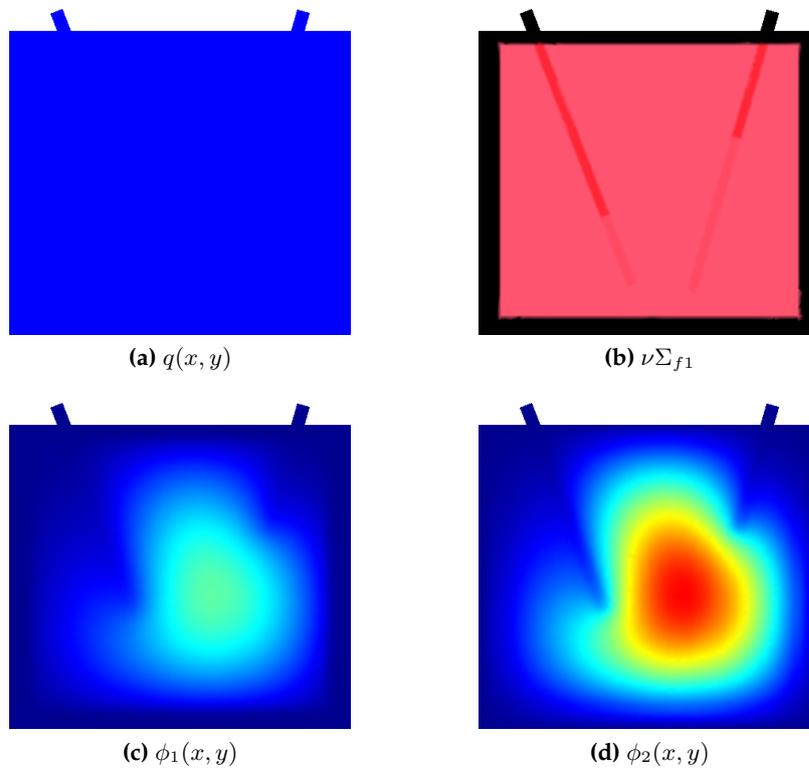


Figura 5.103 – Núcleo fresco, $\ell_c = 9$, algoritmo meshadapt, elementos finitos

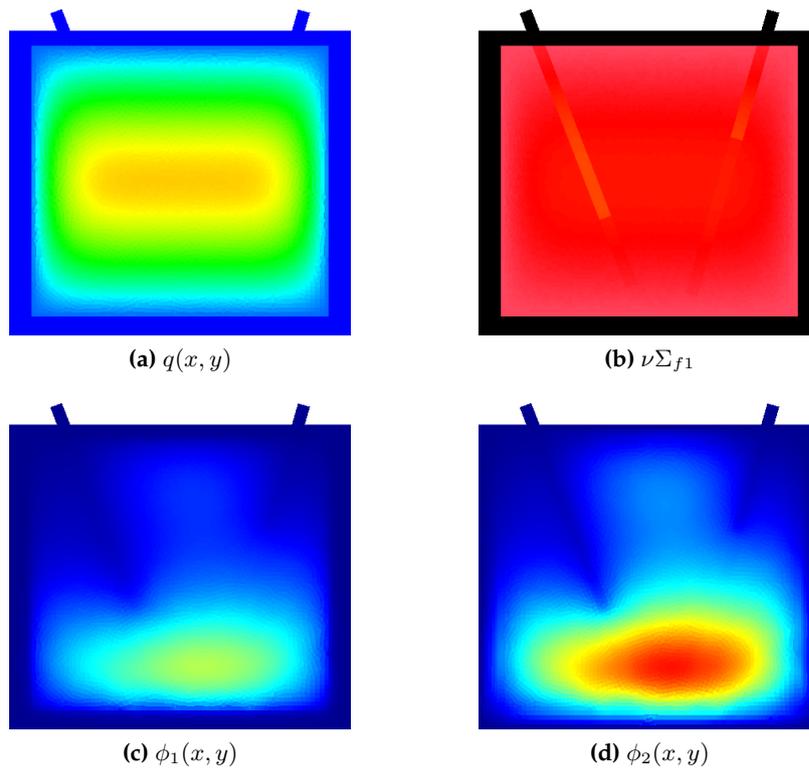


Figura 5.104 – Núcleo en equilibrio, $\ell_c = 9$, algoritmo meshadapt, volúmenes finitos

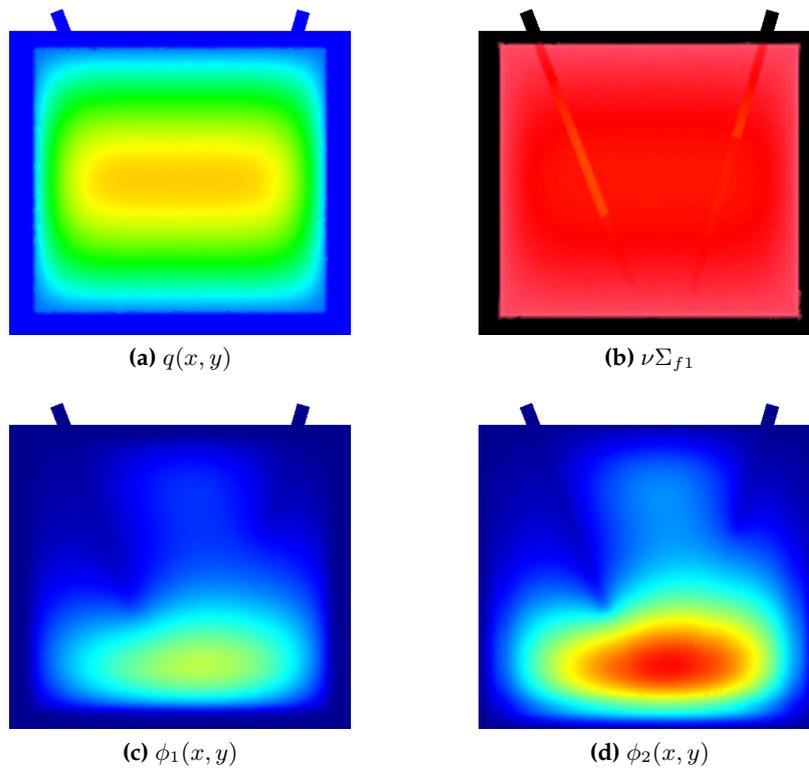


Figura 5.105 – Núcleo fresco, $\ell_c = 9$, algoritmo meshadapt, elementos finitos

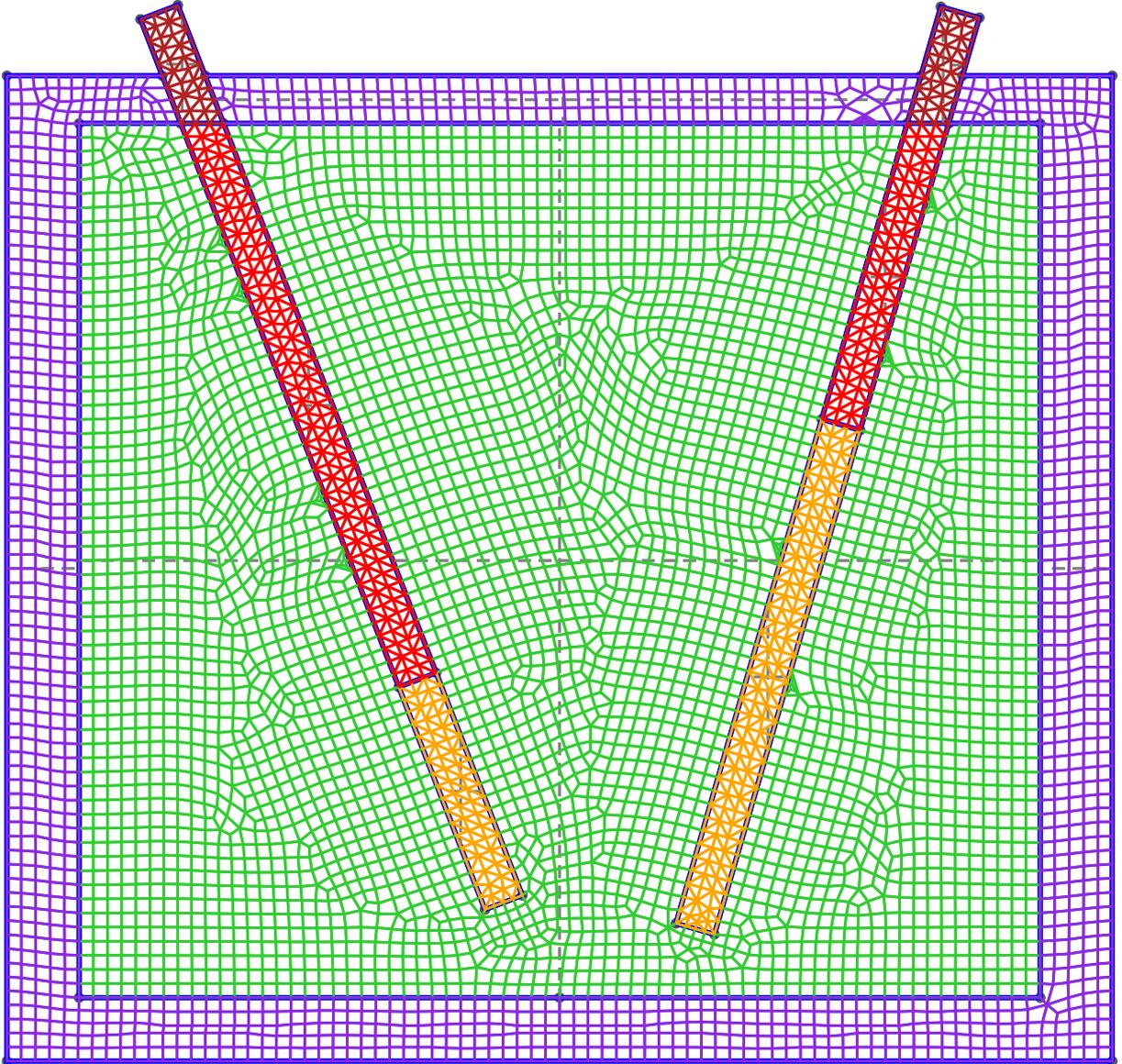


Figura 5.106 – Malla generada por gmsh con el algoritmo delquad para $\ell_c = 9$ cm

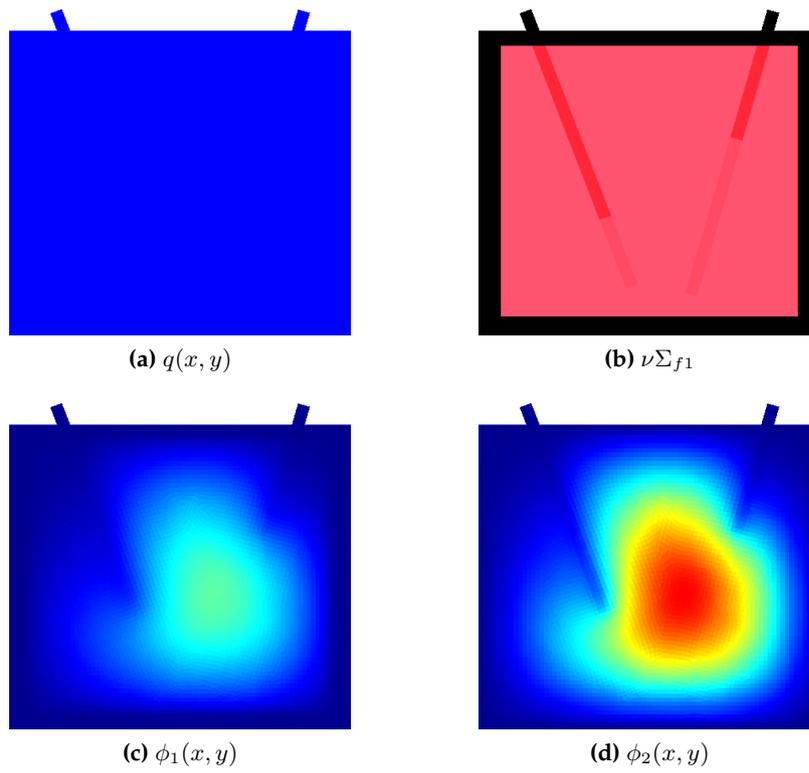


Figura 5.107 – Núcleo fresco, $\ell_c = 9$, algoritmo delquad, volúmenes finitos

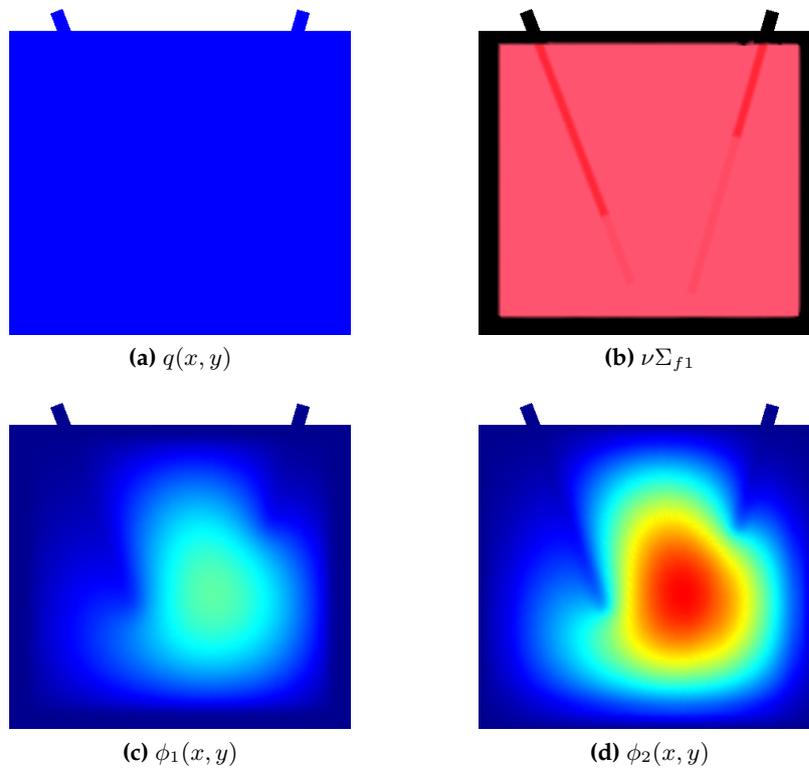


Figura 5.108 – Núcleo fresco, $\ell_c = 9$, algoritmo delquad, elementos finitos

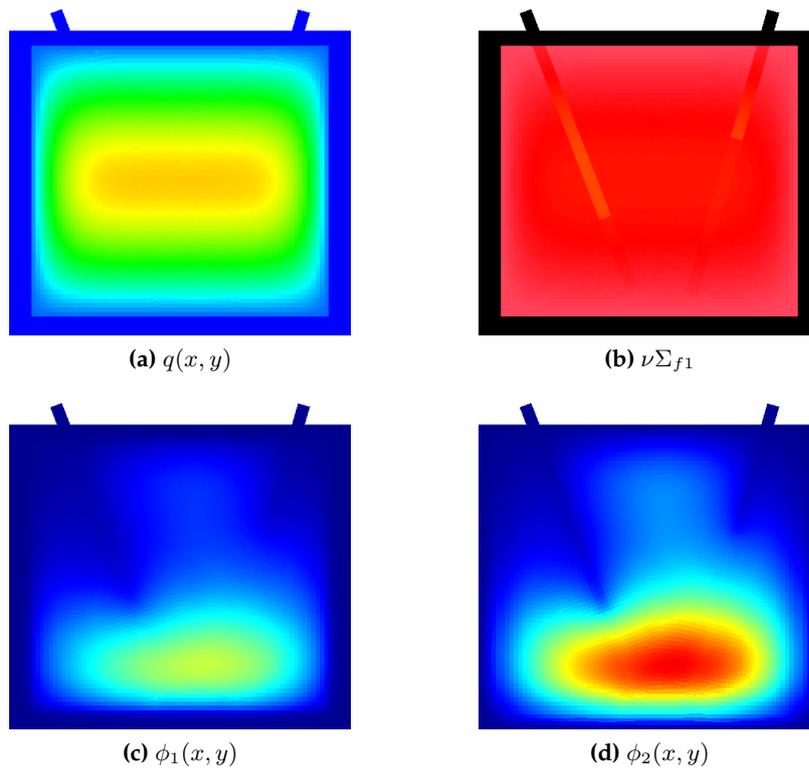


Figura 5.109 – Núcleo en equilibrio, $\ell_c = 9$, algoritmo delquad, volúmenes finitos

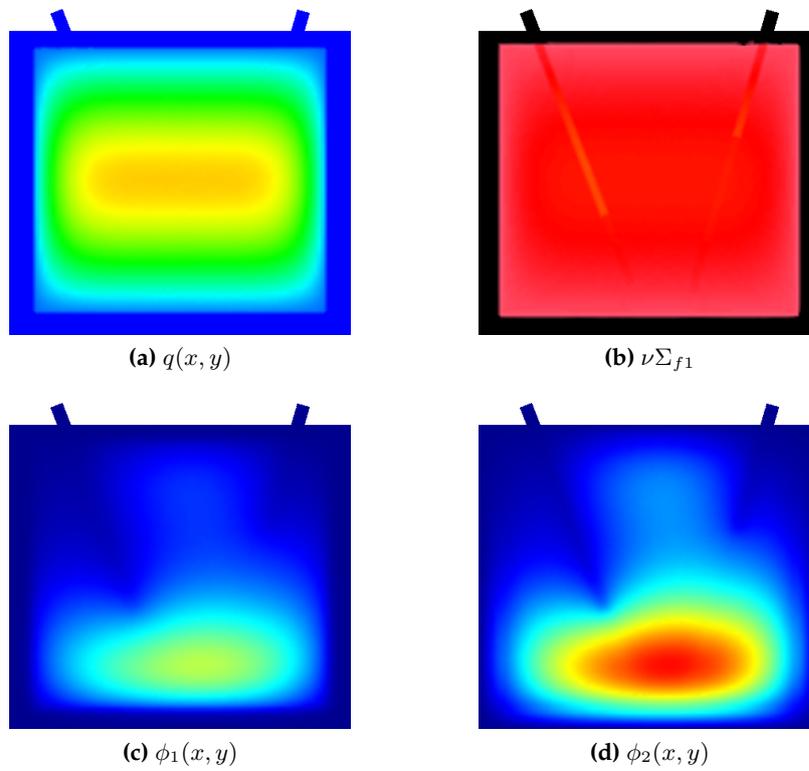


Figura 5.110 – Núcleo en equilibrio, $\ell_c = 9$, algoritmo delquad, elementos finitos



En el mímico de la terminal podemos observar no sólo el k_{eff} para cada una de las dieciséis combinaciones de parámetros sino también el tiempo insumido en la generación de la malla por gmsh, en la lectura de la malla, en el armado de las matrices y en la resolución numérica del problema de autovalores.

Mostramos los resultados de las dieciséis combinaciones en las figuras 5.91–5.108. La primera figura de cada grupo muestra en detalle las cuatro diferentes mallas generada por gmsh para las combinaciones de $\ell_c = 14\text{cm}/9\text{cm}$ y algoritmo de malla meshadapt/delquad. Debemos notar que las superficies que corresponden a las barras de control y a los tubos guías se generan a partir de triángulos mientras que el núcleo y el reflector están compuestas por cuadrángulos, que a su vez se generan recomblando triángulos básicos. A continuación mostramos, para dicha malla, algunos resultados espaciales para las cuatro combinaciones de núcleo fresco/equilibrio y volúmenes/elementos finitos. La subfigura (a) muestra la distribución espacial de quemado, que para núcleo fresco es idénticamente cero y para núcleo en equilibrio está dada por la ecuación (5.12). La subfigura (b) ilustra cuál es la distribución de $\nu\Sigma_{f1}$ resultante. Si el quemado es cero, las distribuciones de secciones eficaces macroscópicas dependen sólo de la geometría y son uniformes en cada material. En cambio para el caso de núcleo en equilibrio con distribución de quemado no trivial, las secciones eficaces cambian continuamente dentro de cada material y en forma discontinua en las interfaces. En las figuras, tanto $q(x, y)$ como $\nu\Sigma_{f1}(x, t)$ aparecen discretizadas como valores homogéneo sobre cada celda en volúmenes e interpolaciones de valores nodales sobre cada elemento. Es por eso que en las soluciones de elementos tanto el quemado la sección eficaz de fisión aparezcan como corruptas en los bordes: sucede que en las posiciones de los nodos que están en la interfaz digamos núcleo-reflector, ni $q(x, y)$ ni $\nu\Sigma_{f1}(x, t)$ están bien definidas. Dependiendo de cómo se realiza la comparación de números de coma flotante, los valores nodales serán los correspondientes a uno u otro material en forma más o menos aleatoria. Luego, cuando gmsh intenta interpolar sobre los elementos adyacentes a la interfaz, pueden aparecer colores intermedios entre las propiedades de ambos materiales. Sin embargo, ésta es sólo la representación en vista de post-procesamiento de gmsh. En la construcción las matrices del problema, milonga integra numéricamente las distribuciones continuas en función de x e y para obtener los elementos de R y de F . Como las integrales se realizan sobre la superficie del elemento sin llegar hasta el borde, en todo momento $\nu\Sigma_{f1}(x, y)$ —y todas las secciones eficaces—están bien definidas y por lo tanto las integrales son calculadas correctamente, dentro de la precisión del esquema numérico.

Las subfiguras (c) y (d) muestran las distribuciones de flujo rápido y térmico respectivamente. Para volúmenes finitos, asignamos el valor medio de cada celda obtenido en el cálculo numérico a toda la celda y para elementos finitos, interpolamos los valores nodales arrojados por el problema de autovalores/autovectores. Estas funciones $\phi_1(x, y)$ y $\phi_2(x, y)$ sí están discretizadas pero como son continuas, están bien definidas incluso en las interfaces por lo que no aparecen elementos con colores raros en la representación de post-procesamiento.

Hemos preparado cada una de las figuras subfiguras como mapa de bits con un tamaño razonable como para ser introducidas en un documento cuya finalidad es ser impreso en un soporte de papel con formato A4. Es por eso que algunos de los detalles que discutimos en el párrafo anterior y algunos otros que podrían llegar a interesarle al lector no puedan ser apreciados en el formato propuesto. El autor del trabajo proveerá gustosamente los archivos digitales originales en formato de post-procesamiento para gmsh con los que se pueden generar gráficos vectoriales en escalas arbitrarias sin pérdida de calidad. Una vez más debemos recordar que tanto la geometría como los datos de este problema fueron inventados completamente. Lo interesante del estudio es la comparación de los resultados obtenidos con ambos esquemas numéricos frente a un problema de una complejidad considerable más que los resultados en sí mismos.



5.2.3. El problema de los pescaditos

En esta sección invocamos un problema clásico de los cursos de física de reactores conocido como el problema del pescadito y lo generalizamos para dos y tres pescaditos.³ El enunciado original consiste en suponer que existe un cierto reactor líquido homogéneo esférico de radio A y que un pescadito también esférico de radio $a \ll A$ se mueve libremente dentro del reactor. La consigna es calcular, utilizando teoría de perturbaciones, cómo cambia la reactividad neta del reactor en función de la distancia r del pescadito al centro del reactor.

A continuación resolvemos primero el problema original de un pescadito de radio a moviéndose no en una esfera sino en un círculo de radio A y calculamos el cambio de reactividad en función de la distancia r no con teoría de perturbaciones sino resolviendo paramétricamente el problema de difusión al ubicar el pescadito en diferentes posiciones radiales r . Luego estudiamos qué sucede no cuando hay un único pescadito sino dos pescaditos que se mueven a lo largo de un mismo diámetro. Nuevamente, resolvemos la reactividad en función de la distancia r . Finalmente, insertamos tres pescaditos en el reactor. Esta vez, fijamos la posición de dos de ellos en forma no simétrica y nos planteamos el problema de encontrar dónde hay que ubicar el tercero para que la reactividad total sea mínima. A raíz de esta monografía, se ha presentado un artículo en la serie Mecánica Computacional [34] y un trabajo en la Reunión Anual de la Asociación Argentina de Tecnología Nuclear 2013 [35].

Fijamos $A = 50$ unidades arbitrarias de longitud, $a = 2$ y barremos paramétricamente r con incrementos $\Delta r = 1$. Definimos la longitud característica de las celdas/elementos $\ell_c = 2$, aunque dentro y alrededor del pescadito refinamos un poco la malla achicando a la mitad la longitud característica asociada a los puntos que lo definen. Suponemos un grupo de energías y nuevamente inventamos las secciones eficaces del reactor original y del pescadito:

```
# parametros de la geometria
NUMBER A      50      # radio del reactor
NUMBER a      2       # radio del pescadito
NUMBER deltar 1      # incremento en r
lc = 2        # longitud característica
```

```
# secciones eficaces
MATERIAL fuel      D 1      SigmaA 0.02      nuSigmaF 0.021
MATERIAL pescadito D 0.9    SigmaA 0.10
```

5.2.3.1. Un único pescadito

El problema tradicional del pescadito consiste en encontrar cómo cambia la reactividad de una configuración similar a la que mostramos en la figura 5.111 para diferentes valores de r . Podemos generar esta geometría con el siguiente archivo template:

```
A = $1;      // radio del reactor
a = $2;      // radio del pescadito
r = $3;      // posicion radial del pescadito
lc = $4;     // longitud característica de la malla
```

³A lo largo de esta sección usamos la palabra pescado no como participio pasado del verbo pescar sino como el sustantivo cuyo significado todos conocemos perfectamente.

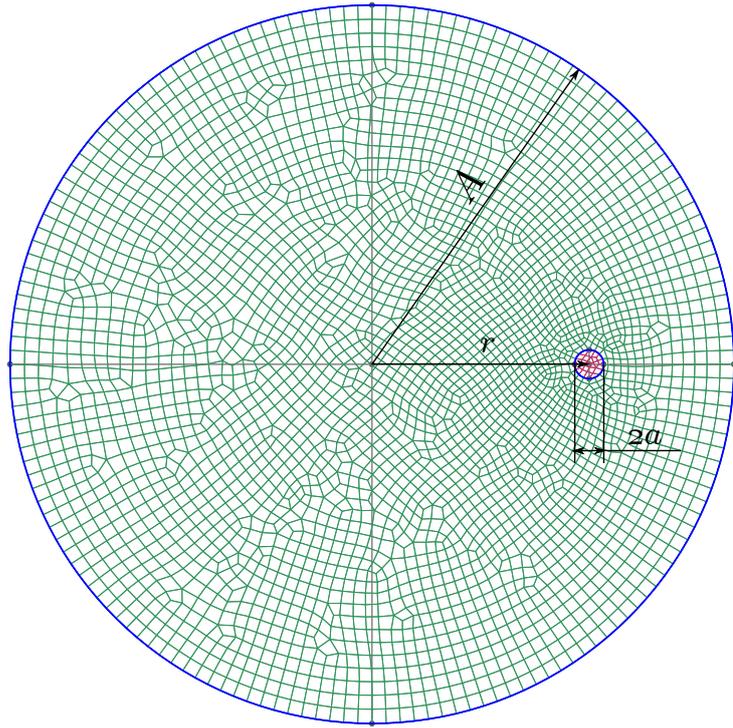


Figura 5.111 – Un pescadito de radio a se mueve dentro de un reactor de radio A a una distancia r del centro.

```

Mesh.Algorithm = 8; // algoritmo de mallado delaunay

Point(1) = { 0, 0, 0, lc }; // el círculo grande de radio A
Point(2) = { A, 0, 0, lc };
Point(3) = { -A, 0, 0, lc };
Point(4) = { 0, A, 0, lc };
Point(5) = { 0, -A, 0, lc };
Circle(1) = {5, 1, 2};
Circle(2) = {2, 1, 4};
Circle(3) = {4, 1, 3};
Circle(4) = {3, 1, 5};
Line Loop(13) = {2, 3, 4, 1};

Point(8) = {r, 0, 0, 0.5*lc}; // el pescadito de radio a
Point(10) = {r+a, 0, 0, 0.5*lc}; // y malla refinada
Point(11) = {r, a, 0, 0.5*lc};
Point(12) = {r-a, 0, 0, 0.5*lc};
Point(13) = {r, -a, 0, 0.5*lc};
Circle(5) = {13, 8, 10};
Circle(6) = {10, 8, 11};
Circle(7) = {11, 8, 12};
Circle(8) = {12, 8, 13};
Line Loop(14) = {6, 7, 8, 5};

Plane Surface(16) = {13, 14}; // las superficies (con los huecos)
Recombine Surface (16);
Plane Surface(17) = {14};
Recombine Surface (17);

Physical Line(1) = {2, 1, 4, 3}; // superficie externa
Physical Surface("fuel") = {16}; // fuel

```

```
Physical Surface("pescadito") = {17}; // pescadito

Color SeaGreen { Surface {16}; }
Color Maroon   { Surface {17}; }
```

donde los valores de A , a , r y ℓ_c son escritos por milonga antes de generar la malla.

Hacemos una corrida paramétrica variando r en el rango $[0, A - a - 1/2\Delta r]$ con incrementos Δr . Por un lado calculamos el k_{eff} para cada r tanto con elementos como con volúmenes, y por otro calculamos el $k_{\text{eff}0}$ que tendría el reactor grande en ausencia de cualquier pescadito. Luego calculamos el incremento de reactividad introducida por el pescadito como

$$\Delta\rho(r) = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} - \frac{k_{\text{eff}0} - 1}{k_{\text{eff}0}}$$

que es el resultado esperado del problema del pescadito.

un-pescadito.was

```
# estudio parametrico sobre r entre [0:A-a-deltar] con saltos
# de deltar (si el rango esta dado por expresiones deben ser
# de NUMBERS constantes y no de variables)
INCLUDE geo.was
PARAMETRIC r 0 A-a-0.5*deltar deltar

# generamos el archivo .geo con lo que valgan los parametros A, r y a
TEMPLATE un-pescadito.tpl un-pescadito.geo A a r lc
# llamamos a gmsh para generar la malla
SHELL "gmsh_v_0_2_un-pescadito.geo"

MESH FILE_PATH un-pescadito.msh DEGREES 1 DIMENSIONS 2
INCLUDE xs.was # geometria y secciones eficaces

# resolvemos el problema
MILONGA SOLVE

# calculamos el keff que tendria el circulo sin pescaditos
# para eso miramos las XS en un punto donde sepamos que no va a
# estar el pescadito
ke0 = nuSigmaF(0,0.9*A) / (SigmaA(0,0.9*A) + (D(0,0.9*A)*(2.405/A)^2))

# imprimimos en pantalla el delta rho que metio el pez vs. r
PRINT %g r 1e5*((keff-1)/keff-(ke0-1)/ke0)
```

```
$ milonga un-pescadito.was --volumes > un-pescadito-vol.dat
$ milonga un-pescadito.was --elements > un-pescadito-ele.dat
$ pyxplot un-pescadito.ppl
$
```

La figura 5.112 muestra el resultado del problema de un pescadito. Como era de esperar, el resultado al resolver el problema con elementos finitos se parece a una “curva S”: para $r \leq A$ el efecto del absorbente es pequeño y la reactividad introducida es casi cero, mientras que para $r = 0$ la reactividad negativa es máxima y su derivada es cero debido a la simetría del problema. Si bien la curva de volúmenes finitos sigue en líneas generales a la solución esperada, no es monótona con r ni tiene derivada cero en el centro. Podemos explicar estos wiggles

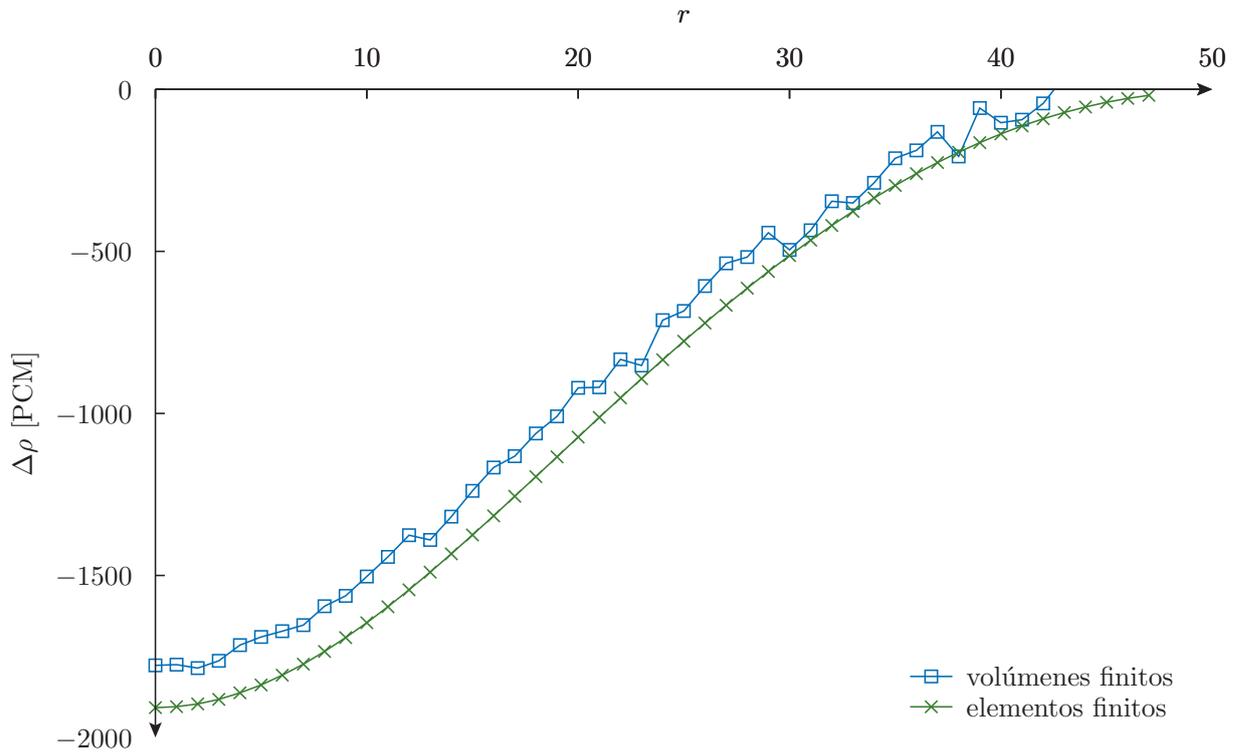


Figura 5.112 – Incremento de reactividad del conjunto reactor+pescadito con respecto al reactor sin pescadito, en función de la posición r del pescadito.

se con un argumento similar al utilizado en la sección 5.2.1. La solución de la formulación en volúmenes finitos depende del coseno del ángulo que forman el segmento que une los centros de dos celdas adyacentes y de la normal al segmento que las separa. Aunque r varíe sólo ligeramente, puede darse el caso de que gmesh cambie radicalmente la distribución de las celdas para acomodarse a la geometría continua, resultando grandes cambios en la precisión de la evaluación de los términos de superficie en la formulación de volúmenes. El método basado en elementos no tiene esta dependencia discontinua con la geometría y por lo tanto sus resultados son más suaves y tienen un mejor acuerdo con la solución que esperaríamos del problema del pescadito continuo.

5.2.3.2. Dos pescaditos

El problema que sigue es una variación del anterior, pero que ilustra tanto el efecto del conocido efecto de apantallamiento pero también el menos conocido y poco felizmente traducido efecto anti-shadow como anti-apantallamiento. En este caso, agregamos un segundo pescadito en todo idéntico al primero ubicado también a una distancia r del centro diametralmente opuesto al primer pez (figura 5.113).

dos-pescaditos.was

```

INCLUDE geo.was

PARAMETRIC r a+0.5*deltar A-a-0.5*deltar deltar
TEMPLATE dos-pescaditos.tpl dos-pescaditos.geo A a r lc
SHELL "gmsh_v_0_2_dos-pescaditos.geo"
MESH FILE_PATH dos-pescaditos.msh DEGREES 1 DIMENSIONS 2

```

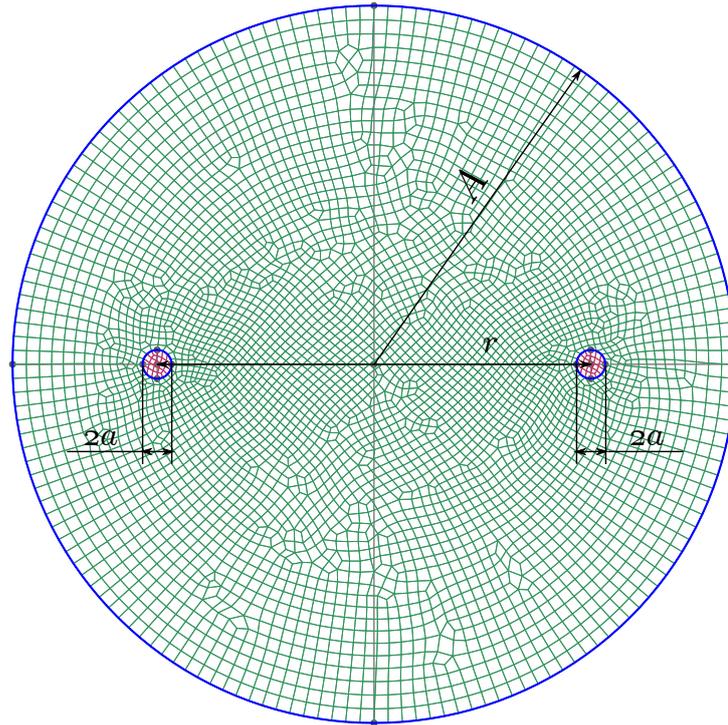


Figura 5.113 – Dos pescaditos de radio a se mueven en forma simétrica dentro de un reactor de radio A , cada uno a una distancia r del centro.

```
INCLUDE xs.was
MILONGA SOLVE
```

```
# si quisieramos un perfil a lo largo del eje x para
# cada r podriamos descomentar lo que sigue
# OUTPUT_FILE axial dos-pescaditos-axial-$1.dat
# phi(x) := phi_1(x,0)
# PRINT_FUNCTION FILE axial phi MIN -A MAX A STEP A/200
```

```
ke0 = nuSigmaF(0,0.9*A) / (SigmaA(0,0.9*A) + (D(0,0.9*A)*(2.405/A)^2))
```

```
# imprimimos en pantalla el delta rho que metio el pez vs. r
PRINT %g r 1e5*((keff-1)/keff-(ke0-1)/ke0)
```

```
$ milonga dos-pescaditos.was --volumes > dos-pescaditos-vol.dat
$ milonga dos-pescaditos.was --elements > dos-pescaditos-ele.dat
$ pyxplot dos-pescaditos-vol.ppl
$ pyxplot dos-pescaditos-ele.ppl
$
```

Estamos ahora interesados en comparar la reactividad neta introducida por ambos pescaditos en función de la distancia r y compararla con el doble de la reactividad introducida por un único pescadito, que es el resultado del problema anterior. Mostramos la solución obtenida con volúmenes finitos en la figura 5.114 y con elementos en la figura 5.115. Por un lado, si $r \leq A$ entonces las dos reactividades coinciden y son casi iguales a cero. Por otro lado, si r es pequeño (notar que si $r < a$ los dos pescaditos se chocan y gmsh no puede generar una malla válida), vemos que los dos pescaditos se apantallan entre sí y la reactividad neta es en valor

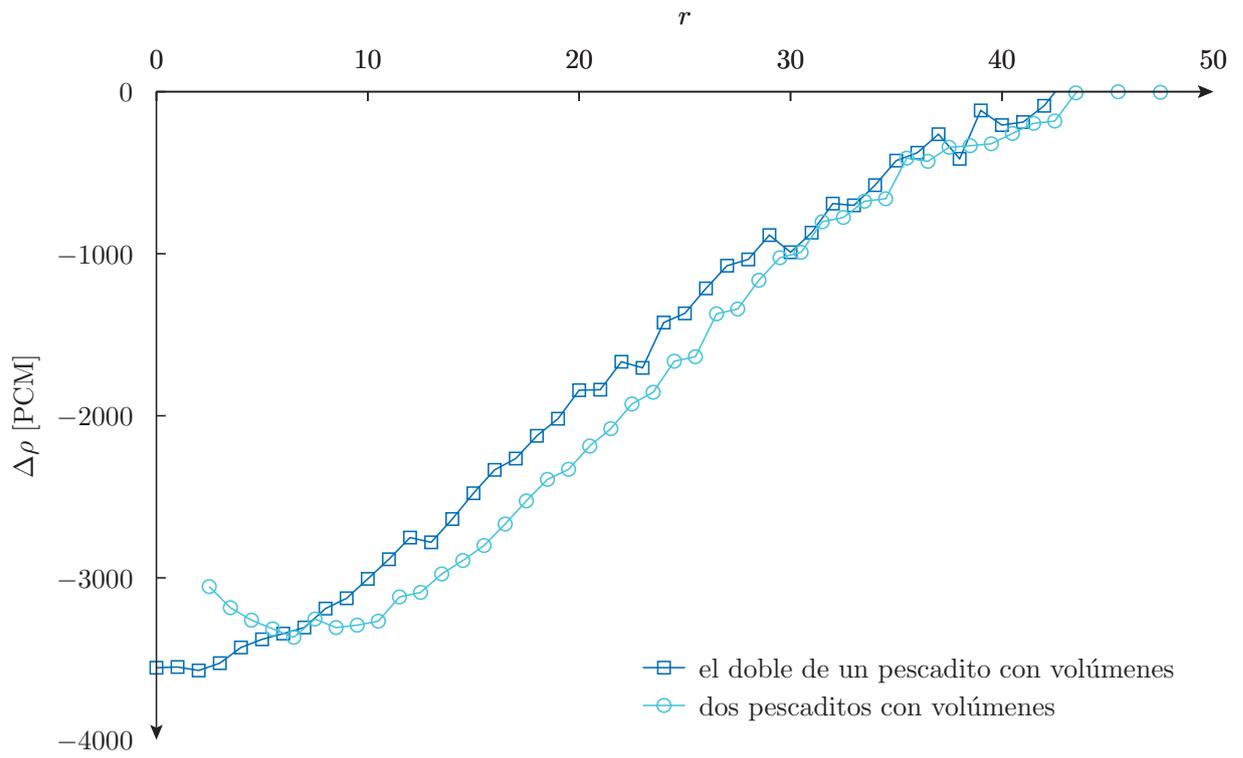


Figura 5.114 - Comparación entre la reactividad neta introducida por dos pescaditos y el doble de la reactividad introducida por uno solo calculada con volúmenes finitos.

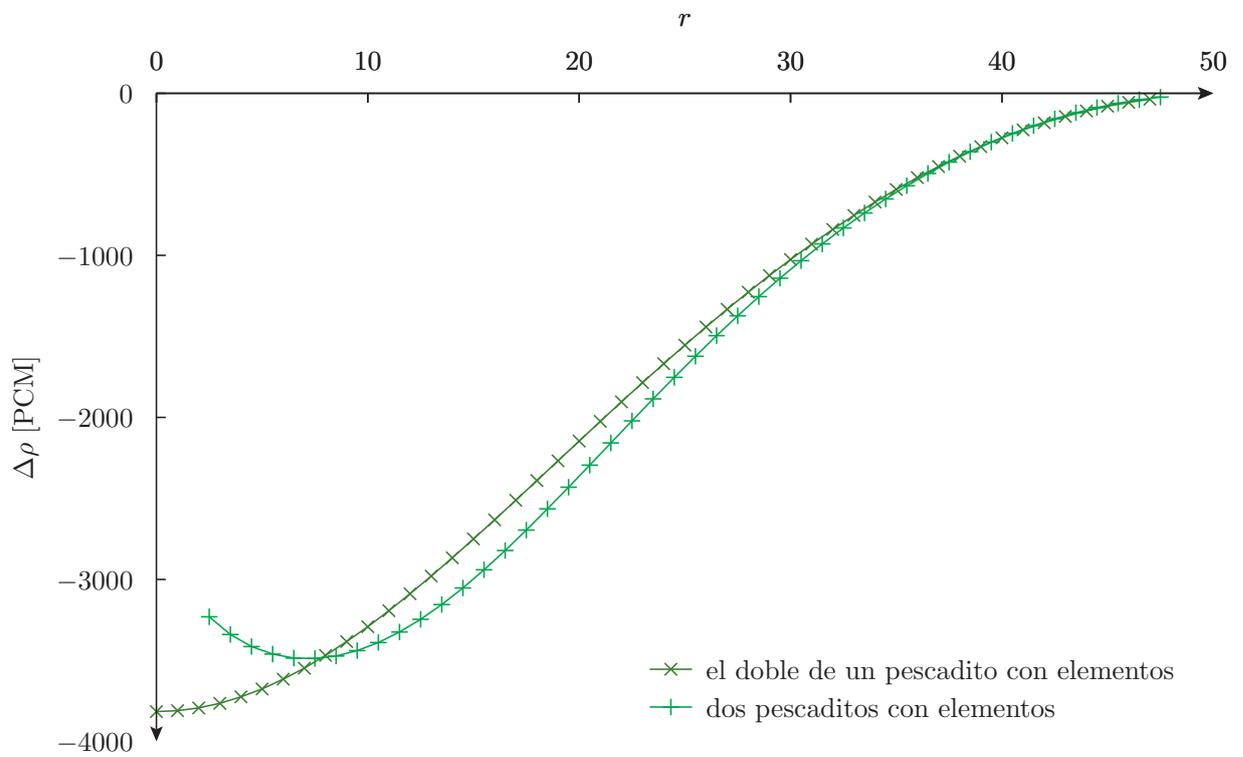


Figura 5.115 - Comparación entre la reactividad neta introducida por dos pescaditos y el doble de la reactividad introducida por uno solo calculada con elementos finitos.



absoluto menor que el doble de la que habría si tuviésemos dos veces solamente un pescadito de radio a en la posición r . Este es el efecto del apantallamiento, que se suele discutir en los cursos de física de reactores. Sin embargo, para valores de $r \approx 1/2A$ sucede que la reactividad neta introducida por ambos pescaditos es mayor en valor absoluto que el doble de un único pescadito. Este es el efecto de anti-apantallamiento. Puesto de otro modo, dos pescaditos absorbentes puestos a una cierta distancia son más efectivos para introducir reactividad negativa que si los tuviésemos por separados. Este resultado es conocido, pero recién aparece en el capítulo número 15 del libro de la referencia [20] donde incluso aparece un gráfico en todo similar a la figura 5.115 calculado al resolver cuasi-analíticamente la ecuación de difusión a un grupo. Como la mayoría de los estudiantes de grado no pasa del capítulo 9 de dicho libro, este efecto de anti-apantallamiento suele pasar desapercibido hasta la aparición de resultados de cálculos tridimensionales de núcleo realizados con contratos en moneda extranjera que no se pueden explicar si no fuese por una combinación de efectos neutrónicos reales y de defectos en la discretización espacial.

Desde el punto de vista de la comparación entre volúmenes y elementos finitos, al analizar las figuras 5.114 y 5.115 llegamos a la misma conclusión que para el problema de un único pescadito. Para este tipo de problemas en los que las secciones eficaces son uniformes y la geometría varía en forma continua, el método de elementos finitos del capítulo 4 es muy superior al método de volúmenes finitos del capítulo 3.

5.2.3.3. Tres pescaditos

El último de los problemas de esta sección involucra tres pescaditos iguales y consiste en encontrar, dadas las posiciones fijas \mathbf{x}_1 y $\mathbf{x}_2 \in \mathbb{R}^2$ de dos de ellos, la ubicación \mathbf{x}_3 del tercero para que la reactividad del conjunto sea mínima. Se trata de un problema de optimización, que no sólo milonga puede resolver sino que es este el objetivo de la tesis de doctorado del autor del código (y de este trabajo): la optimización de parámetros de diseño de reactores nucleares de potencia.

Los detalles de los algoritmos involucrados y de otras sutilezas y particularidades relacionadas a la optimización de la ecuación de difusión de neutrones involucraría una tesis de doctorado completa. En esta sección solamente ilustramos cómo es posible minimizar la función objetivo

$$\Delta\rho(x_3, y_3) = \left[\frac{k_{\text{eff}}(x_3, y_3) - 1}{k_{\text{eff}}(x_3, y_3)} - \frac{k_{\text{eff}0} - 1}{k_{\text{eff}0}} \right] \times 10^5 \text{ [PCM]}$$

encontrando la posición x_3, y_3 de un pescadito de radio a dentro de un reactor circular de radio A , dadas las posiciones fijas de dos pescaditos de radio a

$$\begin{aligned}x_1 &= 0,2A \\y_1 &= -0,1A \\x_2 &= 0 \\y_2 &= -0,15A\end{aligned}$$

Con estos valores, es de esperar que la solución óptima sea $x_3 < 0$ e $y_3 > 0$.

Para resolver el problema de optimización utilizamos el método de Nelder y Mead de minimización por simplex n -dimensionales [36] implementado en la biblioteca GNU Scientific Library [37], contra la que milonga debe linkearse al compilar. A partir de un cierto guess inicial, el algoritmo procede proponiendo nuevos valores dónde evaluar la función objetivo de



forma tal de el simplex se mueva hacia la solución óptima y eventualmente se contraiga hasta acotarla dentro de una cierta tolerancia especificada. Este método, a diferencia de los algoritmos basados en gradientes conjugados, no necesita evaluar derivadas parciales de la función objetivo con respecto a las variables a optimizar por lo que la cantidad de evaluaciones de la función $\rho(x_3, y_3)$ es menor que para gradientes conjugados. Sin embargo, el algoritmo de Nelder y Mead puede quedar atrapado en mínimos locales y por lo que en general no podemos asegurar que la solución encontrada sea efectivamente la óptima. Para ello existen otros métodos tales como recocido simulado o algoritmos genéticos, que estamos incorporando al framework wasora que utiliza milonga para resolver problemas de optimización.

En cualquier caso, fijada la posición de dos pescaditos, el problema de encontrar la posición de un tercero tiene solución única ya que si lo ubicamos muy lejos del centro su contribución negativa a la reactividad total será pequeña. Por otro lado, si intentamos ponerlo cerca de los dos primeros se apantallarán entre sí y la reactividad neta tampoco será muy negativa. La superficie $\Delta\rho(x_3, y_3)$ —que podríamos construir con milonga mediante un cálculo paramétrico sobre x_3 e y_3 —tiene un único mínimo local, que coincide con el mínimo global. Efectivamente, milonga es capaz de encontrar el punto óptimo dónde colocar un cierto absorbente para lograr una máxima reactividad negativa en esta configuración sencilla con el input que sigue. Es este un problema canónico, pero consideramos que es el primer paso para atacar el problema general de optimización del diseño de sistemas de seguridad para maximizar su eficacia ante transitorios en condiciones accidentales en reactores nucleares de potencia.

tres-pescaditos.was

```
INCLUDE geo.was

x1 = 0.2*A # ubicacion fija del primer pescadito
y1 = -0.1*A

x2 = 0 # ubicacion fija del segundo pescadito
y2 = -0.15*A

# guess inicial para x3 y3 en el tercer cuadrante para
# lustrar como milonga lo lleva al segundo
x3 = -0.3*A
y3 = -0.2*A

SCHEME $1
TEMPLATE tres-pescaditos.tpl tres-pescaditos-$1.geo A a lc x1 y1 x2 y2 x3 y3
SHELL "gmsht_v_0_2_tres-pescaditos-$1.geo"
MESH DEGREES 1 DIMENSIONS 2 FILE_PATH tres-pescaditos-$1.msh
INCLUDE xs.was
MILONGA SOLVE

# definimos una funcion que dependa de la posicion del nuevo pescadito
ke0 = (nuSigmaF(0,A) / (SigmaA(0,A) + (D(0,A)*(2.405/A)^2)))
deltarho(x3,y3) := 1e5*((keff-1)/keff - (ke0-1)/ke0)

# y le pedimos a milonga que la minimice con el algoritmo
# de los simplex que no necesita evaluar gradientes
MINIMIZE FUNCTION deltarho ALGORITHM nmsimplex2 STEP a a

# imprimimos en la salida estandar las soluciones parciales
PRINT %+.3f x3 y3 %.1f deltarho(x3,y3)

# y las ubicaciones de los dos pescaditos para dibujar
OUTPUT_FILE fijos pescaditos-fijos-$1.dat
```



```
IF done_outer PRINT x1 y1 FILE fijos  
IF done_outer PRINT x2 y2 FILE fijos
```

```
$ milonga tres-pescaditos.was volumes > tres-pescaditos-volumes.dat  
$ head -n 5 tres-pescaditos-volumes.dat  
-15.000 -10.000 -4074.1  
-15.000 -10.000 -4074.1  
-13.000 -10.000 -4121.9  
-15.000 -8.000 -4107.3  
-13.000 -8.000 -4226.0  
$ tail -n 5 tres-pescaditos-volumes.dat  
-7.781 +9.156 -4651.3  
-7.427 +9.142 -4639.1  
-7.443 +9.076 -4663.8  
-7.545 +9.146 -4649.4  
-7.578 +9.016 -4694.7  
$ milonga tres-pescaditos.was elements > tres-pescaditos-elements.dat  
$ head -n 5 tres-pescaditos-elements.dat  
-15.000 -10.000 -4245.2  
-15.000 -10.000 -4245.2  
-13.000 -10.000 -4272.7  
-15.000 -8.000 -4324.5  
-13.000 -8.000 -4353.1  
$ tail -n 5 tres-pescaditos-elements.dat  
-5.309 +8.396 -4864.4  
-5.413 +8.519 -4864.6  
-5.332 +8.725 -4863.9  
-5.315 +8.478 -4864.4  
-5.413 +8.519 -4864.6  
$ gnuplot tres-pescaditos-vol.gnuplot  
$ gnuplot tres-pescaditos-ele.gnuplot  
$
```

Como hasta el momento, resolvemos exactamente el mismo problema tanto con volúmenes como con elementos finitos. Partiendo de un cierto guess inicial para $x_3 < 0$ e $y_3 < 0$, vemos en la terminal que ambos esquemas comienzan aproximadamente tomando los mismos pasos aunque la función objetivo no sea exactamente igual en ambos casos. Luego de una cierta cantidad de iteraciones, el algoritmo de optimización llega a una cierta posición x_3, y_3 donde se satisfacen las condiciones de convergencia—por defecto ya que en el input no las especificamos—y milonga considera que el problema ha sido resuelto. Observamos en la terminal que las soluciones encontradas con volúmenes y con elementos coinciden a todos los fines prácticos. En efecto, en las figuras 5.116 y 5.117 mostramos todos los pasos intermedios tomados por el algoritmo de optimización según la evaluación de la función objetivo $\Delta\rho(x_3, y_3)$ con volúmenes y con elementos finitos, respectivamente. Aunque es de esperar que existan diferencias puntuales entre ambos esquemas, vemos que cualitativamente el algoritmo de optimización se comporta de manera similar en ambos casos. La solución óptima encontrada consiste en ubicar el pescadito en el segundo cuadrante a una cierta distancia del centro, ni muy alejada como para que no contribuya a la reactividad negativa ni muy cercana a los otros pescaditos como para apantallarse entre sí.

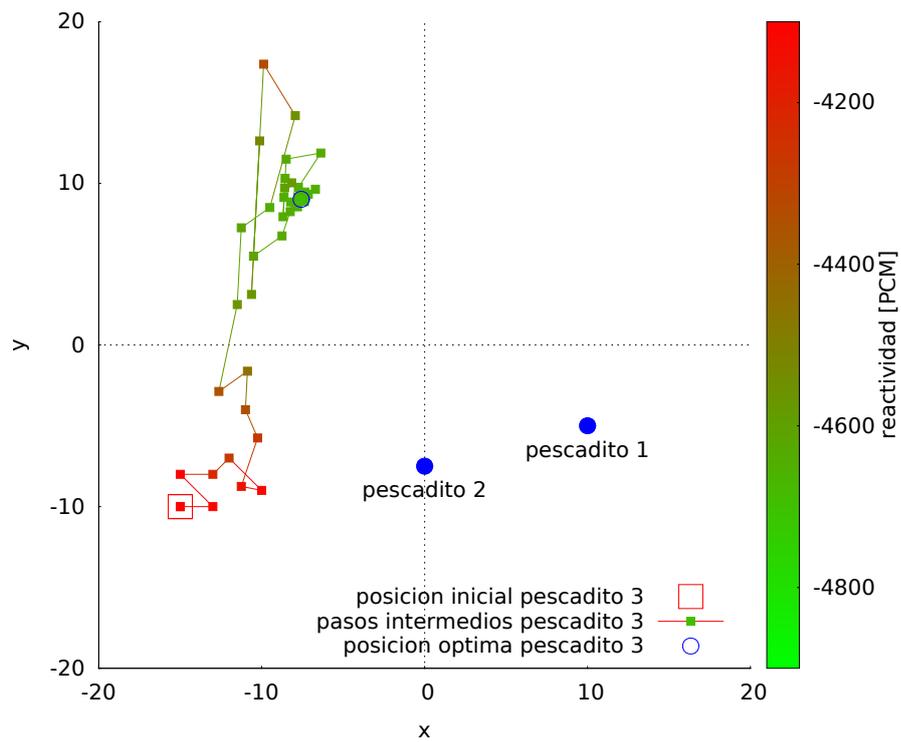


Figura 5.116 – Solución del problema de ubicar un pescadito dada la posición de otros dos fijos para minimizar la reactividad neta total con volúmenes finitos.

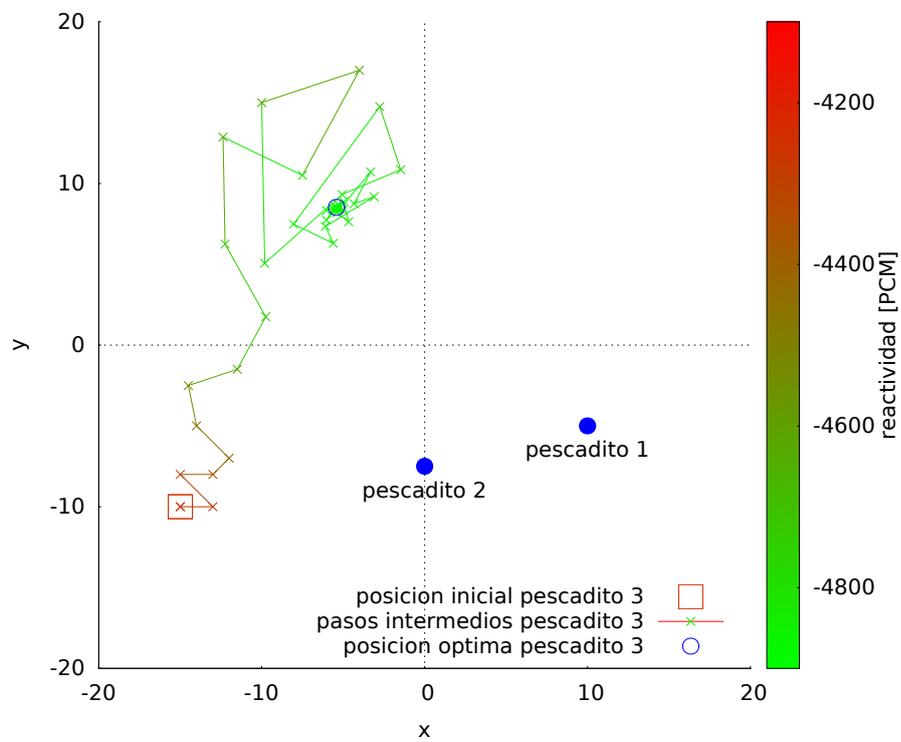


Figura 5.117 – Solución del problema de ubicar un pescadito dada la posición de otros dos fijos para minimizar la reactividad neta total con elementos finitos.



5.2.4. Two-dimensional PWR IAEA Benchmark

Finalizamos el capítulo de resultados resolviendo un problema clásico propuesto por la Agencia Internacional de Energía Atómica en 1976 [38] para comparar y verificar códigos de cálculo que resuelven la ecuación de difusión de neutrones multigrupo, que es justamente lo que es milonga. La geometría original propone una simetría de cuarto de núcleo con superficies externas planas, pensada para ser resuelta utilizando una malla estructurada, que de hecho es la forma en la que fueron remitidas las soluciones originales que hoy en día se toman como referencia. Incluso recientemente el código milonga fue utilizado para resolver el problema en su modo de malla estructurada [39], que era el único disponible en 2011. Estrictamente, la geometría presenta una simetría de un octavo de núcleo lo que reduciría las incógnitas del problema aproximadamente a la mitad y el esfuerzo computacional para resolverlo cuatro veces. Pero la imposibilidad de discretizar satisfactoriamente un sector de núcleo de cuarenta y cinco grados con una malla estructurada hizo que tradicionalmente se mantuviera el cuarto de núcleo como geometría del problema. A raíz de la presente monografía, se ha publicado un trabajo reciente [33] que ilustra la ventaja de utilizar mallas no estructuradas para resolver este problema que ya es considerado clásico de la física de reactores.

A continuación citamos el enunciado original del problema:

Reduction of source situation:

1. Two-group diffusion theory
2. Two-dimensional (x, y) -geometry

Two-group diffusion equations:

$$\begin{aligned}
 -\nabla(D_1\nabla\phi_1) + (\Sigma_{a1} + \Sigma_{1\rightarrow 2} + D_1B_{z1}^2) \cdot \phi_1 &= \frac{1}{\lambda} \nu\Sigma_{f2} \cdot \phi_2 \\
 -\nabla(D_2\nabla\phi_2) + (\Sigma_{a2} + D_2B_{z2}^2) \cdot \phi_2 &= \Sigma_{1\rightarrow 2} \cdot \phi_1
 \end{aligned}$$

Configuration:

Data (two-group constants):

	D_1	D_2	$\Sigma_{1\rightarrow 2}$	Σ_{a1}	Σ_{a2}	$\nu\Sigma_{f2}$	Material
1	1.5	0.4	0.02	0.01	0.080	0.135	Fuel 1
2	1.5	0.4	0.02	0.01	0.085	0.135	Fuel 2
3	1.5	0.4	0.02	0.01	0.130	0.135	Fuel 2 + Rod
4	2.0	0.3	0.04	0	0.010	0	Reflector

Tabla 5.1 – Macroscopic cross sections (units are not stated in the original reference, but they are assumed to be in cm^{-1} or cm as appropriate.)

Axial buckling $B_{z,g}^2 = 0,8 \cdot 10^{-4}$ for all regions and energy groups.

Boundary conditions:

$$J_g^{\text{in}} = 0 \quad \text{No incoming current at external boundaries}$$

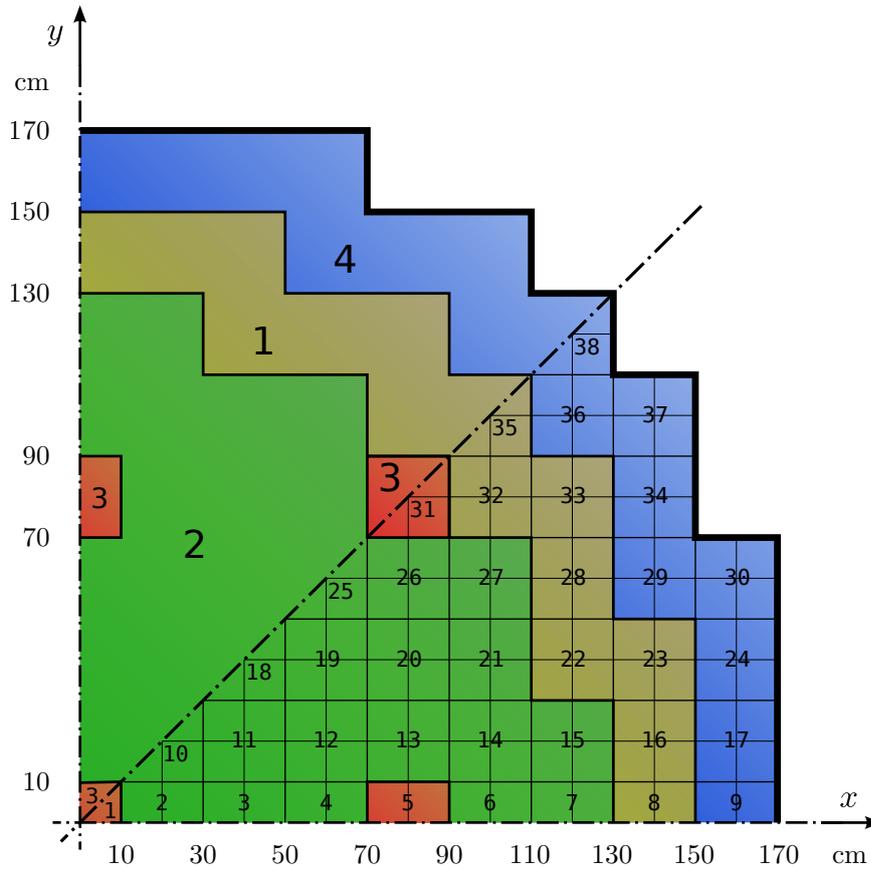


Figura 5.118 – Upper octant: region assignments. Lower octant: fuel assembly identification.

For finite difference diffusion theory codes the following form is considered equivalent:

$$\frac{\partial \phi_g}{\partial n} = -\frac{0,4692}{D_g} \cdot \phi_g$$

where n is the outward directed normal to the surface. At symmetry boundaries:

$$\frac{\partial \phi_g}{\partial n} = 0$$

Expected primary results:

1. Maximum eigenvalue
2. Fundamental flux distributions
 - a) Radial flux traverses $\phi_g(x,0)$ and $\phi_g(x,x)$.
Note: The fluxes shall be normalized such that

$$\frac{1}{V_{\text{core}}} \int_{V_{\text{core}}} \sum_g \nu \Sigma_{fg} \cdot \phi_g dV = 1$$



b) Value and location of maximum power density. This corresponds to maximum of ϕ_2 in the core. It is recommended that the maximum values of ϕ_2 both in the inner core and at the core/reflector interface be given.

3. Average subassembly powers P_k

$$P_k = \frac{1}{V_k} \int_{V_k} \sum_g \nu \Sigma_{fg} \cdot \phi_g dV$$

where V_k is the volume of the k -th subassembly and k designates the fuel subassemblies as shown in figure 5.118.

4. Number of unknowns in the problem, number of iterations, total and outer
5. Total computing time, iteration time, IO-time, computer used
6. Type and numerical values of convergence criteria
7. Table of average group fluxes for a square mesh grid of 20×20 cm
8. Dependence of results on mesh spacing

5.2.4.1. Solución al estilo milonga

Estamos interesados en resolver el problema, por supuesto tanto con volúmenes como con elementos finitos, pero también aprovechando el hecho de que usamos mallas no estructurada queremos comparar también la geometría original de un cuarto de núcleo con la geometría de un octavo de núcleo. Además, resulta interesante comparar los resultados obtenidos tomando como figura base triángulos y cuadrángulos, mallando con los algoritmos de delaunay y delquad provistos por gms. Finalmente, tal como solicita el ítem número ocho, estamos interesados en estudiar qué sucede al variar el tamaño de la malla. Una vez más, como hemos hecho a lo largo de este capítulo, queremos resolver todos estos casos con un único input de milonga por lo que le pasamos a través de la línea de comandos las siguientes selecciones:

1. número de caso: $n = 1, \dots, 80$
2. tipo de geometría: quarter / octave
3. algoritmo de mallado: delaunay / delquad
4. forma básica: triangs / quads
5. esquema de discretización: volumes / elements
6. longitud característica de la malla: $\ell_c = 4, 3, 2, 1, 0,75$

Esta situación da lugar a $80 = 2 \times 2 \times 2 \times 2 \times 5$ casos, que corremos con milonga usando en mismo input pero diferentes opciones en la línea de comandos.


```

SHELL "gmsH_-2_-$1-$2-$3-$4-$5-$6/benchmark.geo_>_/dev/null"
t2 = clock()

# ask wasora to read a two-dimensional two-groups problem
MESH DIMENSIONS 2 DEGREES 2 FILE_PATH $1-$2-$3-$4-$5-$6/benchmark.msh
t3 = clock()

# materials and cross sections according to the two-group constants
# each material corresponds to a physical entity in the geometry file

Bg2 = 0.8e-4 # axial geometric buckling in the z direction
MATERIAL fuel1          D_1 1.5          D_2 0.4 \
  SigmaA_1 0.010+D_1(x,y)*Bg2  SigmaS_1->2 0.02 \
  SigmaA_2 0.080+D_2(x,y)*Bg2  nuSigmaF_2 0.135  eSigmaF_2 nuSigmaF_2(x,y)

MATERIAL fuel2          D_1 1.5          D_2 0.4 \
  SigmaA_1 0.010+D_1(x,y)*Bg2  SigmaS_1->2 0.02 \
  SigmaA_2 0.085+D_2(x,y)*Bg2  nuSigmaF_2 0.135  eSigmaF_2 nuSigmaF_2(x,y)

MATERIAL fuel2+rod      D_1 1.5          D_2 0.4 \
  SigmaA_1 0.010+D_1(x,y)*Bg2  SigmaS_1->2 0.02 \
  SigmaA_2 0.130+D_2(x,y)*Bg2  nuSigmaF_2 0.135  eSigmaF_2 nuSigmaF_2(x,y)

MATERIAL reflector      D_1 2.0          D_2 0.3 \
  SigmaA_1 0.000+D_1(x,y)*Bg2  SigmaS_1->2 0.04 \
  SigmaA_2 0.010+D_2(x,y)*Bg2

# boundary conditions as requested by the problem, applied
# to appropriate physical entities defined in the geometry file
PHYSICAL_ENTITY_NAME external BC robin -0.4692
PHYSICAL_ENTITY_NAME mirror   BC neumann

# to force milonga to normalize the fluxes as requested, we
# can set a power setpoint equal to the core volume and fix
# eSigmaF = nuSigmaF (as we did in the MATERIAL section)

eps = 4e-4 # relative allowed integration error
key = 1    # GSL_INTEG_GAUSS15 (see GSL's documentation)
# we take the core as the geometric place of the
# points (x,y) that have a non-zero fission XS
power = integral(integral(nuSigmaF_2(x,y) > 0, x, 0, 170, eps, key), y, 0, ←
  170, eps, key)

# we finally ask milonga to solve the eigenvalue problem
MILONGA SOLVE

# keep track of the effort needed
mesh_time = t2-t1
read_mesh_time = t3-t2+init_time

# and then we proceed to answer the requested items

#-----
# item 1
# we save the effective multiplication factor with
# six decimal digits in a in a text file
OUTPUT_FILE keff $1-$2-$3-$4-$5-$6/1-keff.txt
PRINT FILE keff %.6f keff
# and the equivalent static reactivity in another one
rho = 1e5*(keff-1)/keff
OUTPUT_FILE rho $1-$2-$3-$4-$5-$6/1-rho.txt

```

```

PRINT FILE rho %.2f rho

# by the way, inform the user we are almos done
wall = mesh_time+read_mesh_time+build_matrices_time+solve_time
PRINT NOSEP %.2f rho TEXT "_pcm_"(" wall TEXT "_segs)" }

#-----
# item 2 (not asked)
# write the two-dimensional flux distribution both in
# ASCII and in gmsh post-processing format
# the diffusion coefficient is included to help decide
# if a point is in the core or in the reflector
OUTPUT_FILE flux-dist-ascii $1-$2-$3-$4-$5-$6/2-flux-dist.dat
OUTPUT_FILE flux-dist-post $1-$2-$3-$4-$5-$6/2-flux-dist.pos

PRINT_FUNCTION FILE flux-dist-ascii phi_1 phi_2 pow nuSigmaF_2(x,y)>0
MESH_POST FILE flux-dist-post phi_1 phi_2 pow

#-----
# item 2a
# the radial flux traverses phi(x,0) and phi(x,x)
OUTPUT_FILE flux-axis $1-$2-$3-$4-$5-$6/2a-flux-axis.dat
OUTPUT_FILE flux-diag $1-$2-$3-$4-$5-$6/2a-flux-diag.dat

phi1_axis(x) := phi_1(x,0)
phi2_axis(x) := phi_2(x,0)

phi1_diag(x) := phi_1(x,x)
phi2_diag(x) := phi_2(x,x)

PRINT_FUNCTION FILE flux-axis phi1_axis phi2_axis MIN 0 MAX 170 STEP 0.1
PRINT_FUNCTION FILE flux-diag phi1_diag phi2_diag MIN 0 MAX 170 STEP 0.1

#-----
# item 2b
# the maximum thermal flux has to be located in one of the
# solution points, and as computing the maximum of a 2-dimensional
# function is rather expensive, we use GNU sort over the
# ASCII file computed in item 2, filtering between the core
# and the reflector by the value of the fast diffusion coefficient
VAR x_max y_max phi1_max phi2_max

OUTPUT_FILE maximums $1-$2-$3-$4-$5-$6/2b-maximums.txt

SHELL "cat_$1-$2-$3-$4-$5-$6/2-flux-dist.dat|_grep_1\.000000e+00\$_|_sort_ ←
g_k3_r_|_head_n1_>_$1-$2-$3-$4-$5-$6/2b-maximum_core.dat"
READ ASCII_FILE_PATH $1-$2-$3-$4-$5-$6/2b-maximum_core.dat x_max y_max ←
phi1_max phi2_max
PRINT FILE maximums TEXT "maximum_thermal_flux_in_core_ is_phi2=" ←
phi2_max TEXT "at_x=" x_max TEXT "y=" y_max

SHELL "cat_$1-$2-$3-$4-$5-$6/2-flux-dist.dat|_grep_0\.000000e+00\$_|_sort_ ←
g_k3_r_|_head_n1_>_$1-$2-$3-$4-$5-$6/2b-maximum_refl.dat"
READ ASCII_FILE_PATH $1-$2-$3-$4-$5-$6/2b-maximum_refl.dat x_max y_max ←
phi1_max phi2_max
PRINT FILE maximums TEXT "maximum_thermal_flux_in_reflector_is_phi2=" ←
phi2_max TEXT "at_x=" x_max TEXT "y=" y_max

#-----
# item 3 (and 7)

```

```

# we build a table with the average subassembly powers as item 7
# asks the average fluxes , we compute the two of them here

# the file assemblies.coords contains five columns with the
# coordinates that define each of the 38 subassemblies
FUNCTION xmin(k) FILE_PATH assemblies.coords COLUMNS 1 2
FUNCTION xmax(k) FILE_PATH assemblies.coords COLUMNS 1 3
FUNCTION ymin(k) FILE_PATH assemblies.coords COLUMNS 1 4
FUNCTION ymax(k) FILE_PATH assemblies.coords COLUMNS 1 5

# to allow the 8-symmetry geometry, we define "symmetric
# functions" that mirror the power and the fluxes around the
# line y = x, otherwise for those assemblies that contain a
# small area with y > x, their averages would be underestimated
simpow(x,y) := if(x > y, pow(x,y), pow(y,x))
simp1(x,y) := if(x > y, phi_1(x,y), phi_1(y,x))
simp2(x,y) := if(x > y, phi_2(x,y), phi_2(y,x))

# volume of the k-th subassembly
vol(k) := (xmax(k)-xmin(k))*(ymax(k)-ymin(k))

# number of points used in the gauss-legendre quadrature
points = 8
mean_power(k) := 1/vol(k) * gauss_legendre(gauss_legendre( simpow(x,y), x, ←
    xmin(k), xmax(k), points), y, ymin(k), ymax(k), points)
mean_flux_1(k) := 1/vol(k) * gauss_legendre(gauss_legendre(simp1(x,y), x, ←
    xmin(k), xmax(k), points), y, ymin(k), ymax(k), points)
mean_flux_2(k) := 1/vol(k) * gauss_legendre(gauss_legendre(simp2(x,y), x, ←
    xmin(k), xmax(k), points), y, ymin(k), ymax(k), points)

OUTPUT_FILE table $1-$2-$3-$4-$5-$6/3-table.dat
PRINT TEXT "\#_k_P_k_fast_k_thermal_k" FILE table
PRINT_FUNCTION mean_power mean_flux_1 mean_flux_2 MIN xmin_a MAX xmax_b ←
STEP 1 FILE table FORMAT %.3f

#-----
# items 4, 5 & 6
# the requested information is written by milonga if
# the keyword DEBUG is used
OUTPUT_FILE info $1-$2-$3-$4-$5-$6/4-info.txt
DEBUG FILE info

#-----
# item 7 is included in point 3

#-----
# item 8
# it is not clear what does "results" mean, but we
# write a text file with some information we can later
# use to compare all the cases as a function of the
# size of the problem matrices (number of unknowns)
OUTPUT_FILE times $1-$2-$3-$4-$5-$6/8-results.dat
PRINT FILE times {
    %g unknowns nodes cells elements
    %e keff rho
    mesh_time read_mesh_time build_matrices_time solve_time
    %.2f wall }

# -----&----- milonga's solution ends here -----&-----

```

```

$ cat run.sh
ncase=0
for geo in quarter eighth; do
  for alg in delaunay delquad; do
    for shape in triangls quads; do
      for method in volumes elements; do
        for lc in 4 3 2 1 0.75; do
          ncase=$(printf %03d `echo $ncase+1 | bc`)
          milonga benchmark.was $ncase $geo $alg $shape $method $lc
          ./figuras.sh $ncase $geo $alg $shape $method $lc > /dev/null
        done
      done
    done
  done
done
./x-vs-mesh.sh
$ ./run.sh
case 001-quarter-delaunay-triangs-volumes-4      2862.69 pcm (0.31 segs)
case 002-quarter-delaunay-triangs-volumes-3      2882.35 pcm (0.68 segs)
case 003-quarter-delaunay-triangs-volumes-2      2893.52 pcm (1.74 segs)
case 004-quarter-delaunay-triangs-volumes-1      2894.04 pcm (18.43 segs)
case 005-quarter-delaunay-triangs-volumes-0.75   2889.26 pcm (74.94 segs)
case 006-quarter-delaunay-triangs-elements-4     2884.70 pcm (0.46 segs)
case 007-quarter-delaunay-triangs-elements-3     2879.64 pcm (0.85 segs)
case 008-quarter-delaunay-triangs-elements-2     2876.27 pcm (1.89 segs)
case 009-quarter-delaunay-triangs-elements-1     2874.35 pcm (8.66 segs)
case 010-quarter-delaunay-triangs-elements-0.75  2874.16 pcm (17.23 segs)
case 011-quarter-delaunay-quads-volumes-4        2871.06 pcm (0.41 segs)
case 012-quarter-delaunay-quads-volumes-3        2917.05 pcm (0.68 segs)
case 013-quarter-delaunay-quads-volumes-2        2908.85 pcm (1.49 segs)
case 014-quarter-delaunay-quads-volumes-1        2899.41 pcm (10.02 segs)
case 015-quarter-delaunay-quads-volumes-0.75    2892.88 pcm (26.90 segs)
case 016-quarter-delaunay-quads-elements-4       2886.69 pcm (0.56 segs)
case 017-quarter-delaunay-quads-elements-3       2879.41 pcm (0.91 segs)
case 018-quarter-delaunay-quads-elements-2       2877.03 pcm (2.10 segs)
case 019-quarter-delaunay-quads-elements-1       2874.43 pcm (10.79 segs)
case 020-quarter-delaunay-quads-elements-0.75   2874.13 pcm (23.23 segs)
case 021-quarter-delquad-triangs-volumes-4      2881.40 pcm (0.30 segs)
case 022-quarter-delquad-triangs-volumes-3      2927.16 pcm (0.56 segs)
case 023-quarter-delquad-triangs-volumes-2      2930.07 pcm (1.45 segs)
case 024-quarter-delquad-triangs-volumes-1      2929.26 pcm (11.78 segs)
case 025-quarter-delquad-triangs-volumes-0.75   3016.98 pcm (38.76 segs)
case 026-quarter-delquad-triangs-elements-4     2889.16 pcm (0.40 segs)
case 027-quarter-delquad-triangs-elements-3     2881.59 pcm (0.75 segs)
case 028-quarter-delquad-triangs-elements-2     2876.16 pcm (1.57 segs)
case 029-quarter-delquad-triangs-elements-1     2874.46 pcm (7.27 segs)
case 030-quarter-delquad-triangs-elements-0.75  2874.22 pcm (14.73 segs)
case 031-quarter-delquad-quads-volumes-4        2831.10 pcm (0.26 segs)
case 032-quarter-delquad-quads-volumes-3        2848.87 pcm (0.43 segs)
case 033-quarter-delquad-quads-volumes-2        2861.67 pcm (0.88 segs)
case 034-quarter-delquad-quads-volumes-1        2870.02 pcm (5.64 segs)
case 035-quarter-delquad-quads-volumes-0.75    2870.26 pcm (14.70 segs)
case 036-quarter-delquad-quads-elements-4       2889.96 pcm (0.34 segs)
case 037-quarter-delquad-quads-elements-3       2881.83 pcm (0.60 segs)
case 038-quarter-delquad-quads-elements-2       2877.18 pcm (1.21 segs)
case 039-quarter-delquad-quads-elements-1       2874.65 pcm (5.81 segs)
case 040-quarter-delquad-quads-elements-0.75    2874.30 pcm (11.77 segs)
case 041-eighth-delaunay-triangs-volumes-4     2864.21 pcm (0.25 segs)
case 042-eighth-delaunay-triangs-volumes-3     2879.68 pcm (0.31 segs)
case 043-eighth-delaunay-triangs-volumes-2     2892.66 pcm (0.68 segs)

```



case 044-eighth-delaunay-triangs-volumes-1	2888.78	pcm	(5.43 segs)
case 045-eighth-delaunay-triangs-volumes-0.75	2890.65	pcm	(15.02 segs)
case 046-eighth-delaunay-triangs-elements-4	2886.31	pcm	(0.28 segs)
case 047-eighth-delaunay-triangs-elements-3	2879.62	pcm	(0.43 segs)
case 048-eighth-delaunay-triangs-elements-2	2876.48	pcm	(0.91 segs)
case 049-eighth-delaunay-triangs-elements-1	2874.37	pcm	(4.02 segs)
case 050-eighth-delaunay-triangs-elements-0.75	2874.16	pcm	(7.87 segs)
case 051-eighth-delaunay-quads-volumes-4	2873.97	pcm	(0.24 segs)
case 052-eighth-delaunay-quads-volumes-3	2906.59	pcm	(0.34 segs)
case 053-eighth-delaunay-quads-volumes-2	2911.10	pcm	(0.66 segs)
case 054-eighth-delaunay-quads-volumes-1	2889.21	pcm	(3.32 segs)
case 055-eighth-delaunay-quads-volumes-0.75	2888.52	pcm	(8.84 segs)
case 056-eighth-delaunay-quads-elements-4	2885.55	pcm	(0.29 segs)
case 057-eighth-delaunay-quads-elements-3	2879.77	pcm	(0.45 segs)
case 058-eighth-delaunay-quads-elements-2	2877.23	pcm	(1.00 segs)
case 059-eighth-delaunay-quads-elements-1	2874.37	pcm	(4.18 segs)
case 060-eighth-delaunay-quads-elements-0.75	2874.17	pcm	(9.39 segs)
case 061-eighth-delquad-triangs-volumes-4	2883.21	pcm	(0.19 segs)
case 062-eighth-delquad-triangs-volumes-3	2912.78	pcm	(0.27 segs)
case 063-eighth-delquad-triangs-volumes-2	2888.07	pcm	(0.60 segs)
case 064-eighth-delquad-triangs-volumes-1	2912.07	pcm	(3.78 segs)
case 065-eighth-delquad-triangs-volumes-0.75	3009.34	pcm	(10.25 segs)
case 066-eighth-delquad-triangs-elements-4	2889.55	pcm	(0.28 segs)
case 067-eighth-delquad-triangs-elements-3	2882.63	pcm	(0.40 segs)
case 068-eighth-delquad-triangs-elements-2	2876.61	pcm	(0.83 segs)
case 069-eighth-delquad-triangs-elements-1	2874.53	pcm	(3.40 segs)
case 070-eighth-delquad-triangs-elements-0.75	2874.26	pcm	(6.53 segs)
case 071-eighth-delquad-quads-volumes-4	2836.73	pcm	(0.17 segs)
case 072-eighth-delquad-quads-volumes-3	2852.21	pcm	(0.24 segs)
case 073-eighth-delquad-quads-volumes-2	2862.13	pcm	(0.45 segs)
case 076-eighth-delquad-quads-elements-4	2891.11	pcm	(0.22 segs)
case 077-eighth-delquad-quads-elements-3	2882.23	pcm	(0.36 segs)
case 078-eighth-delquad-quads-elements-2	2877.38	pcm	(0.66 segs)
case 079-eighth-delquad-quads-elements-1	2874.66	pcm	(2.72 segs)
case 080-eighth-delquad-quads-elements-0.75	2874.29	pcm	(5.32 segs)
§			

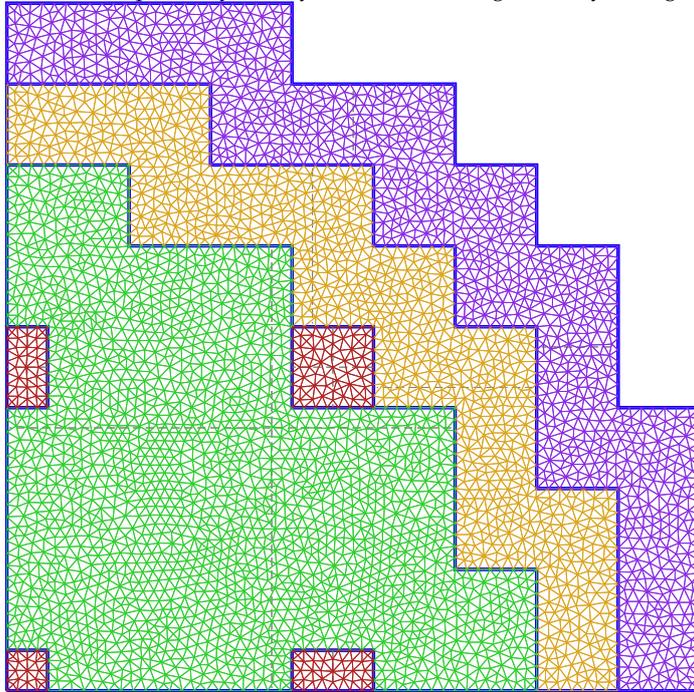
El script `figuras.sh`, que llama a `gms` y prepara archivos de texto para procesar con \LaTeX obtener los resultados solicitados en el enunciado. De hecho tampoco es posible incluir todas las figuras generadas y sólo mostramos algunas a modo de ilustración (figuras 5.119–5.133). De hecho, debido al soporte en papel A4 de esta monografía, es posible que no sea posible observar los detalles de las figuras generadas que hacen a la comparación de las distribuciones obtenidas no sólo entre volúmenes y elementos finitos sino también entre mallas, formas y longitudes características. La referencia [33] contiene otro juegos de figuras, ligeramente modificadas para satisfacer los requerimientos del revisor. Las ochenta figuras originales en alta calidad están disponible en la página web de milonga, accesible a través del código QR de la figura 5.1.

Las figuras 5.134–5.139 muestran algunas respuestas al ítem ocho del problema, que pide indicar cómo cambian los resultados del problema—que no queda claro a cuáles se refiere—con la malla. Estas figuras fueron preparadas por el autor de este trabajo para su publicación en una revista científica internacional [33]—donde se discuten algunos temas que aquí no—y por ello mantenemos aquí el epígrafe en inglés. El color verde indica volúmenes finitos mientras que el azul elementos finitos. Las líneas llenas corresponden a simetría 1/4 y las líneas de trazos a simetría 1/8. Si el bullet es lleno entonces la geometría fue mallada con el algoritmo delaunay y si es vacío con el algoritmo delquad, y su forma indica la figura básica de la celda/elemento (triángulo o cuadrángulo).



milonga's 2D LWR IAEA Benchmark Problem case #002

quarter-symmetry core meshed using delaunay (triangs, $\ell_c = 3$) solved with finite volumes

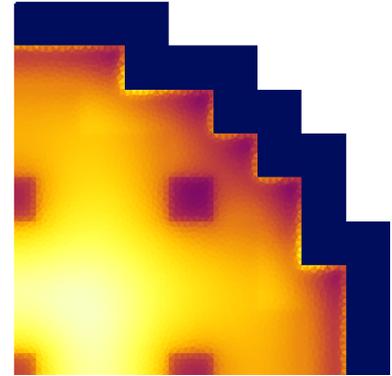
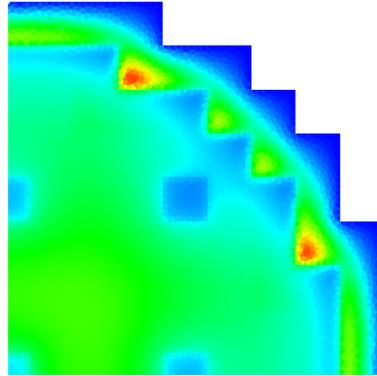
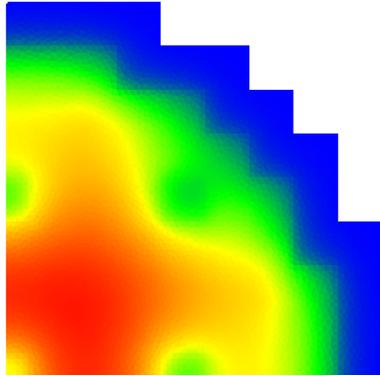


largest eigenvalue k_{eff}	1.029679 (2882.35 pcm)
$\max \phi_2(x, y)$ @core	11.34 @ (30.66, 30.79)
$\max \phi_2(x, y)$ @reflector	13.06 @ (51.71, 130.76)
number of unknowns	15736
outer iterations	3
linear iterations	32
inner iterations	1967
residual norm	2.673×10^{-8}
relative error	1.317×10^{-8}
error estimate	5.071×10^{-9}
memory used	77620 kB
soft page faults	20792
hard page faults	0
total CPU time	1.568 seconds

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.75	32.80	5.55	20	1.18	37.29	8.75
2	1.33	42.39	9.83	21	1.07	33.62	7.91
3	1.47	46.41	10.89	22	0.97	28.88	7.17
4	1.23	39.16	9.10	23	0.66	16.25	4.90
5	0.61	26.85	4.55	24	—	2.33	5.49
6	0.93	29.88	6.92	25	1.20	37.94	8.90
7	0.93	29.22	6.88	26	0.97	30.93	7.17
8	0.74	20.01	5.50	27	0.90	28.38	6.68
9	—	3.21	7.55	28	0.83	22.18	6.18
10	1.45	45.94	10.77	29	—	5.66	12.13
11	1.50	47.26	11.09	30	—	0.68	2.58
12	1.33	41.95	9.84	31	0.47	20.41	3.48
13	1.07	34.34	7.96	32	0.68	20.51	5.03
14	1.03	32.65	7.66	33	0.58	14.12	4.29
15	0.94	29.65	6.99	34	—	2.22	5.62
16	0.72	19.52	5.33	35	0.57	13.69	4.23
17	—	3.05	7.34	36	—	3.85	8.17
18	1.49	46.91	11.01	37	—	0.53	2.03
19	1.36	42.86	10.06	38	—	0.61	2.29

(a) Geometry and mesh: triangs with delaunay and $\ell_c = 3$

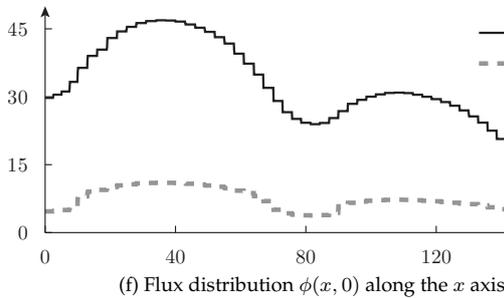
(b) Results with the finite volumes method



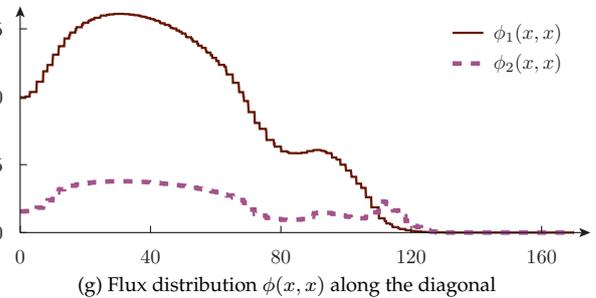
(c) Fast flux distribution

(d) Thermal flux distribution

(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis

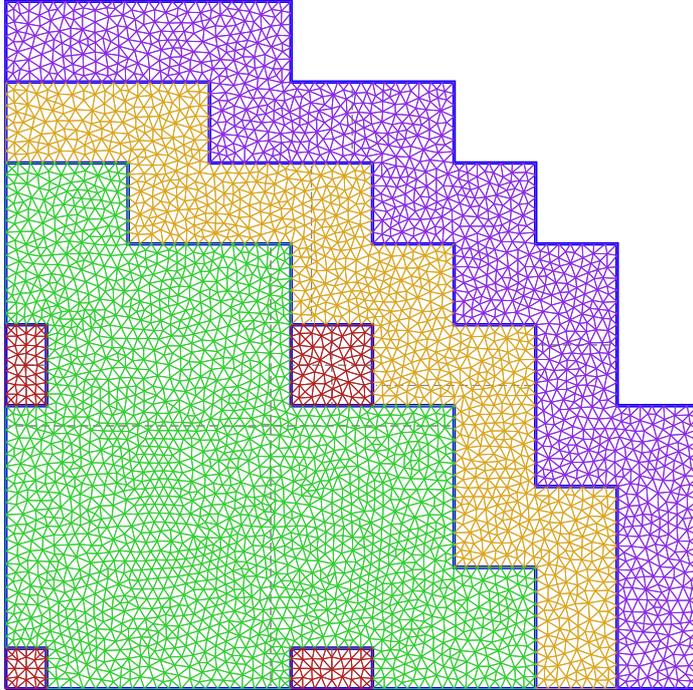


(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.119



milonga's 2D LWR IAEA Benchmark Problem case #007
 quarter-symmetry core meshed using delaunay (triangs, $\ell_c = 3$) solved with finite elements

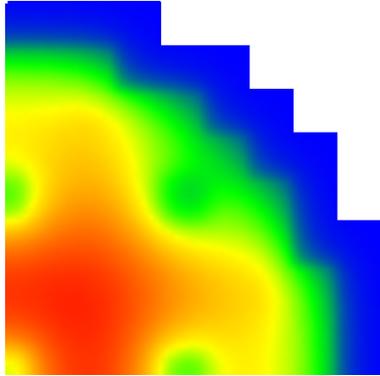


largest eigenvalue k_{eff}	1.029650 (2879.64 pcm)
$\max \phi_2(x, y)$ @core	11.11 @ (31.85, 31.58)
$\max \phi_2(x, y)$ @reflector	15.73 @ (132.20, 52.20)
number of unknowns	8106
outer iterations	3
linear iterations	32
inner iterations	1013
residual norm	1.171×10^{-8}
relative error	5.768×10^{-9}
error estimate	3.458×10^{-9}
memory used	84332 kB
soft page faults	22232
hard page faults	0
total CPU time	1.564 seconds

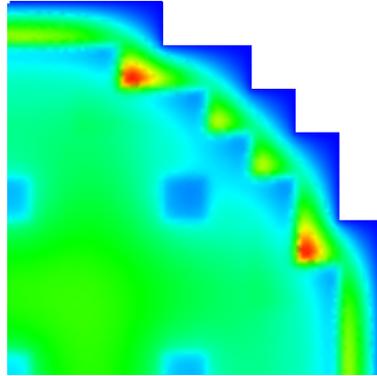
k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.15	5.49	20	1.18	37.07	8.70
2	1.29	41.44	9.59	21	1.07	33.71	7.93
3	1.44	45.44	10.67	22	0.98	29.25	7.26
4	1.20	38.36	8.89	23	0.59	16.88	5.24
5	0.61	26.45	4.52	24	—	2.53	5.96
6	0.93	29.90	6.91	25	1.19	37.47	8.79
7	0.94	29.52	6.95	26	0.96	30.80	7.14
8	0.70	20.53	5.67	27	0.91	28.56	6.73
9	—	3.48	8.09	28	0.79	22.79	6.35
10	1.42	44.87	10.52	29	—	6.14	12.57
11	1.47	46.28	10.86	30	—	0.74	2.89
12	1.30	41.21	9.66	31	0.47	20.43	3.50
13	1.05	33.69	7.79	32	0.69	20.88	5.10
14	1.03	32.70	7.67	33	0.51	14.62	4.52
15	0.95	29.97	7.07	34	—	2.50	6.15
16	0.68	20.05	5.53	35	0.49	14.19	4.43
17	—	3.33	7.71	36	—	4.14	8.44
18	1.46	46.00	10.80	37	—	0.59	2.30
19	1.34	42.17	9.90	38	—	0.67	2.63

(a) Geometry and mesh: triangs with delaunay and $\ell_c = 3$

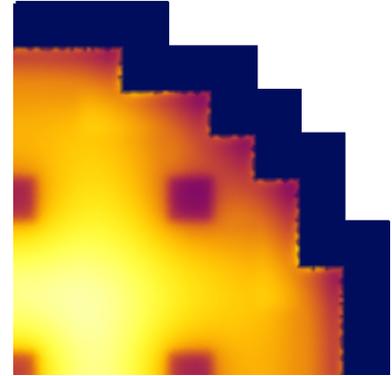
(b) Results with the finite elements method



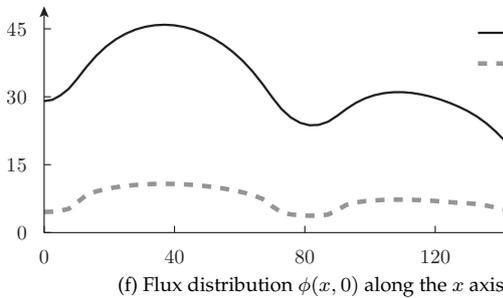
(c) Fast flux distribution



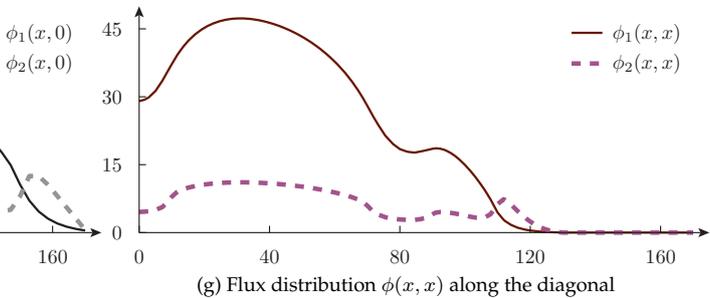
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis

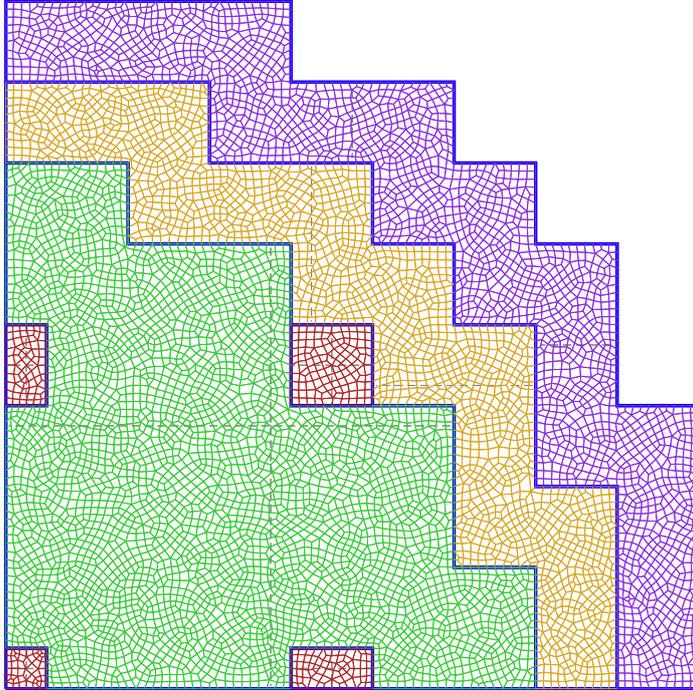


(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.120



milonga's 2D LWR IAEA Benchmark Problem case #013
 quarter-symmetry core meshed using delaunay (quads, $\ell_c = 2$) solved with finite volumes

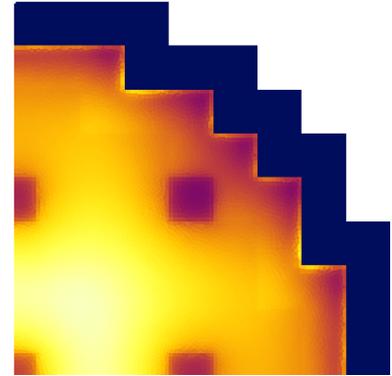
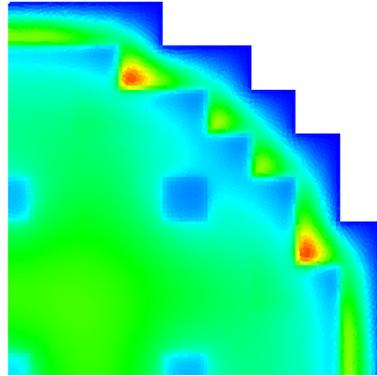
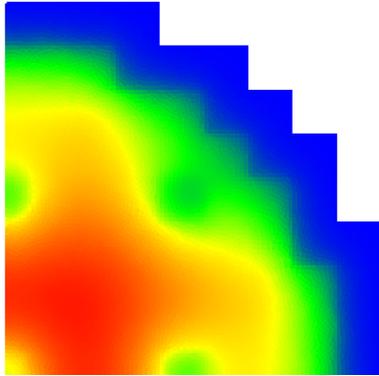


largest eigenvalue k_{eff}	1.029960 (2908.85 pcm)
max $\phi_2(x, y)$ @core	11.30 @ (30.99, 31.49)
max $\phi_2(x, y)$ @reflector	12.35 @ (130.98, 50.98)
number of unknowns	15612
outer iterations	3
linear iterations	32
inner iterations	1951
residual norm	7.316×10^{-9}
relative error	3.604×10^{-9}
error estimate	2.09×10^{-9}
memory used	86012 kB
soft page faults	23332
hard page faults	0
total CPU time	2.012 seconds

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.43	5.49	20	1.19	37.59	8.82
2	1.32	42.26	9.81	21	1.07	33.87	7.97
3	1.47	46.37	10.88	22	0.98	29.19	7.25
4	1.23	39.30	9.14	23	0.67	16.70	4.98
5	0.60	26.54	4.45	24	—	2.22	5.40
6	0.95	30.22	7.02	25	1.21	38.12	8.94
7	0.95	29.79	7.01	26	0.98	31.17	7.25
8	0.76	20.70	5.63	27	0.91	28.52	6.71
9	—	3.19	7.87	28	0.83	22.47	6.14
10	1.45	45.81	10.74	29	—	5.51	12.38
11	1.49	47.11	11.06	30	—	0.67	2.62
12	1.33	41.93	9.83	31	0.45	19.77	3.29
13	1.09	34.77	8.08	32	0.67	20.35	4.99
14	1.04	32.98	7.74	33	0.56	14.19	4.16
15	0.96	30.20	7.12	34	—	2.23	5.66
16	0.73	20.18	5.43	35	0.54	13.58	4.00
17	—	2.96	7.33	36	—	3.59	8.18
18	1.48	46.78	10.98	37	—	0.53	2.06
19	1.36	42.83	10.05	38	—	0.61	2.39

(a) Geometry and mesh: quads with delaunay and $\ell_c = 2$

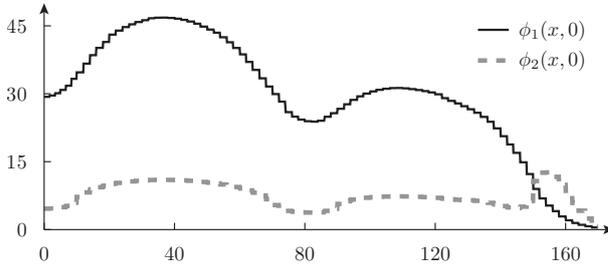
(b) Results with the finite volumes method



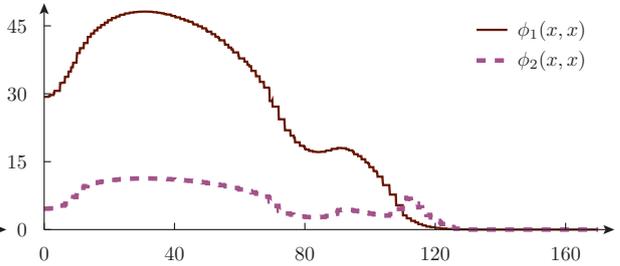
(c) Fast flux distribution

(d) Thermal flux distribution

(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



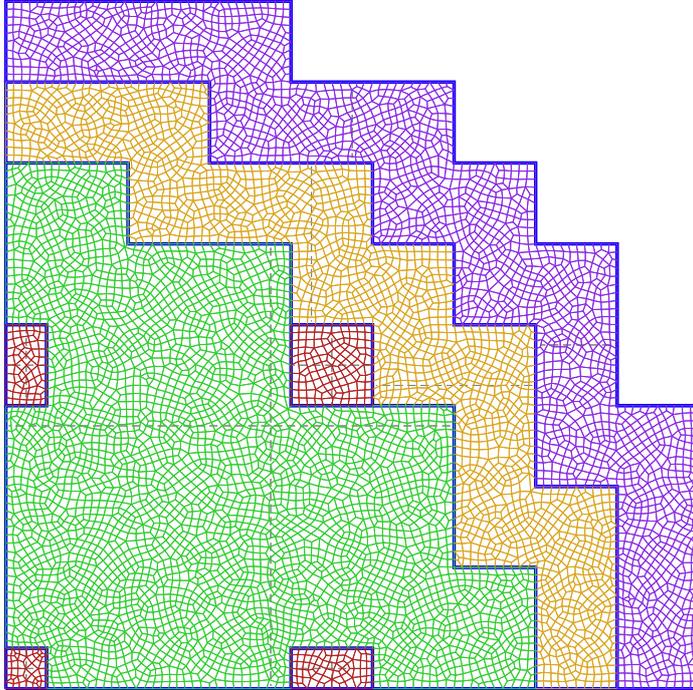
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.121



milonga's 2D LWR IAEA Benchmark Problem case #018

quarter-symmetry core meshed using delaunay (quads, $\ell_c = 2$) solved with finite elements

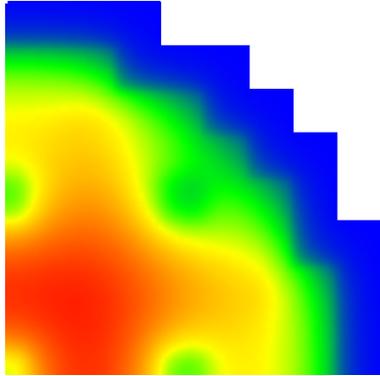


largest eigenvalue k_{eff}	1.029623 (2877.03 pcm)
$\max \phi_2(x, y)$ @core	11.15 @ (30.78, 30.31)
$\max \phi_2(x, y)$ @reflector	13.41 @ (51.85, 131.12)
number of unknowns	15958
outer iterations	3
linear iterations	32
inner iterations	1994
residual norm	1.156×10^{-8}
relative error	5.695×10^{-9}
error estimate	4.777×10^{-9}
memory used	149748 kB
soft page faults	40397
hard page faults	0
total CPU time	2.292 seconds

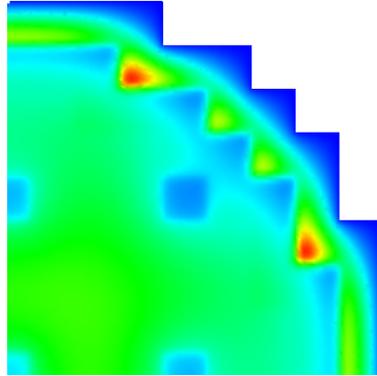
k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.31	5.51	20	1.18	37.14	8.72
2	1.30	41.66	9.64	21	1.07	33.71	7.93
3	1.45	45.66	10.72	22	0.98	29.18	7.25
4	1.21	38.52	8.93	23	0.62	16.78	5.17
5	0.61	26.51	4.52	24	—	2.48	5.91
6	0.93	29.92	6.92	25	1.19	37.57	8.81
7	0.94	29.47	6.93	26	0.96	30.85	7.15
8	0.71	20.43	5.62	27	0.91	28.53	6.72
9	—	3.43	8.05	28	0.80	22.71	6.31
10	1.43	45.10	10.57	29	—	6.05	12.53
11	1.47	46.49	10.91	30	—	0.73	2.85
12	1.31	41.37	9.70	31	0.47	20.41	3.49
13	1.07	34.12	7.90	32	0.69	20.83	5.09
14	1.04	32.71	7.67	33	0.54	14.55	4.47
15	0.95	29.91	7.06	34	—	2.47	6.11
16	0.70	19.96	5.48	35	0.52	14.12	4.37
17	—	3.27	7.66	36	—	4.06	8.41
18	1.46	46.18	10.84	37	—	0.58	2.27
19	1.34	42.30	9.93	38	—	0.65	2.60

(a) Geometry and mesh: quads with delaunay and $\ell_c = 2$

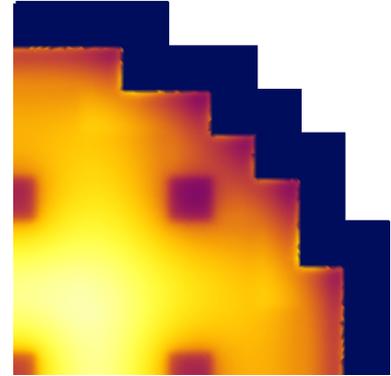
(b) Results with the finite elements method



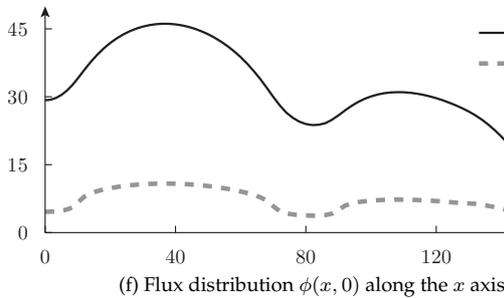
(c) Fast flux distribution



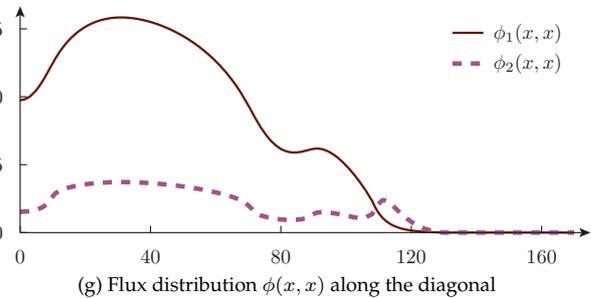
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis

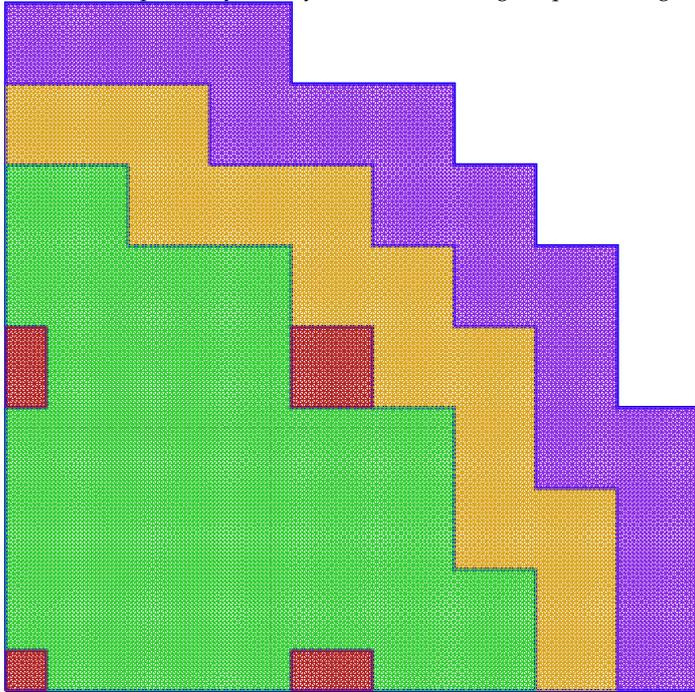


(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.122



milonga's 2D LWR IAEA Benchmark Problem case #024
 quarter-symmetry core meshed using delquad (triangs, $\ell_c = 1$) solved with finite volumes

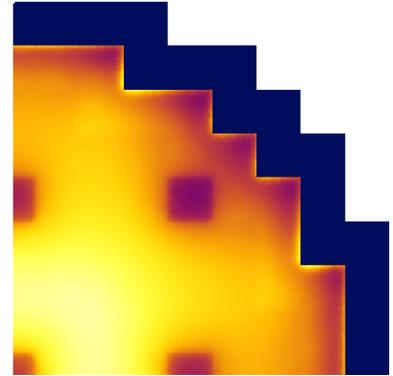
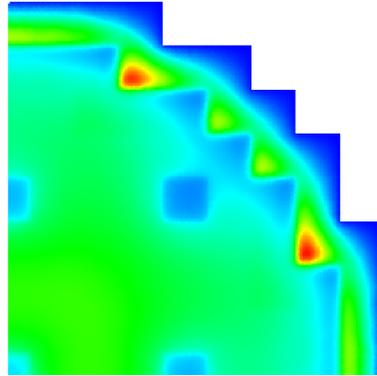
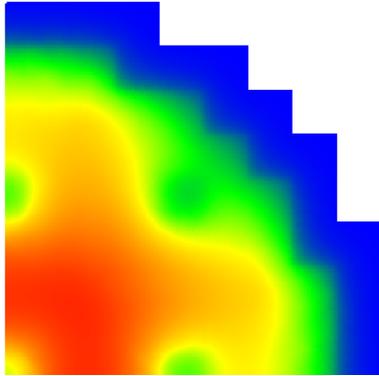


largest eigenvalue k_{eff}	1.030177 (2929.26 pcm)
max $\phi_2(x, y)$ @core	11.02 @ (31.67, 29.33)
max $\phi_2(x, y)$ @reflector	10.11 @ (130.33, 50.33)
number of unknowns	96400
outer iterations	3
linear iterations	32
inner iterations	12050
residual norm	9.282×10^{-9}
relative error	4.572×10^{-9}
error estimate	4.197×10^{-9}
memory used	259228 kB
soft page faults	74177
hard page faults	0
total CPU time	14.5 seconds

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.72	31.88	5.36	20	1.17	36.91	8.66
2	1.31	41.97	9.74	21	1.08	34.02	8.00
3	1.46	46.05	10.81	22	0.99	29.65	7.36
4	1.22	38.97	9.05	23	0.70	17.17	5.16
5	0.59	26.16	4.40	24	—	2.31	5.68
6	0.93	29.86	6.92	25	1.17	37.08	8.70
7	0.93	29.34	6.90	26	0.97	30.90	7.18
8	0.74	20.03	5.46	27	0.92	28.99	6.83
9	—	3.20	7.91	28	0.85	23.06	6.32
10	1.42	44.87	10.52	29	—	5.79	12.64
11	1.46	46.21	10.85	30	—	0.70	2.74
12	1.30	41.11	9.64	31	0.46	2	—
13	1.07	34.16	7.93	32	0.69	20.92	5.13
14	1.04	32.84	7.70	33	0.60	14.77	4.42
15	0.96	30.14	7.11	34	—	2.35	5.99
16	0.73	19.93	5.42	35	0.57	14.01	4.23
17	—	3.08	7.42	36	—	3.82	8.38
18	1.44	45.27	10.63	37	—	0.55	2.18
19	1.31	41.33	9.70	38	—	0.60	2.45

(a) Geometry and mesh: triangles with delquad and $\ell_c = 1$

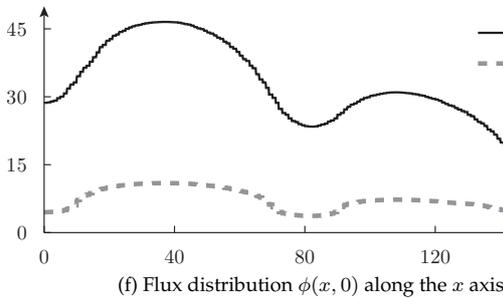
(b) Results with the finite volumes method



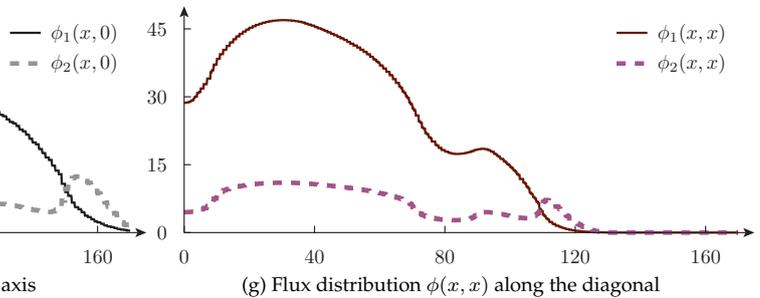
(c) Fast flux distribution

(d) Thermal flux distribution

(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



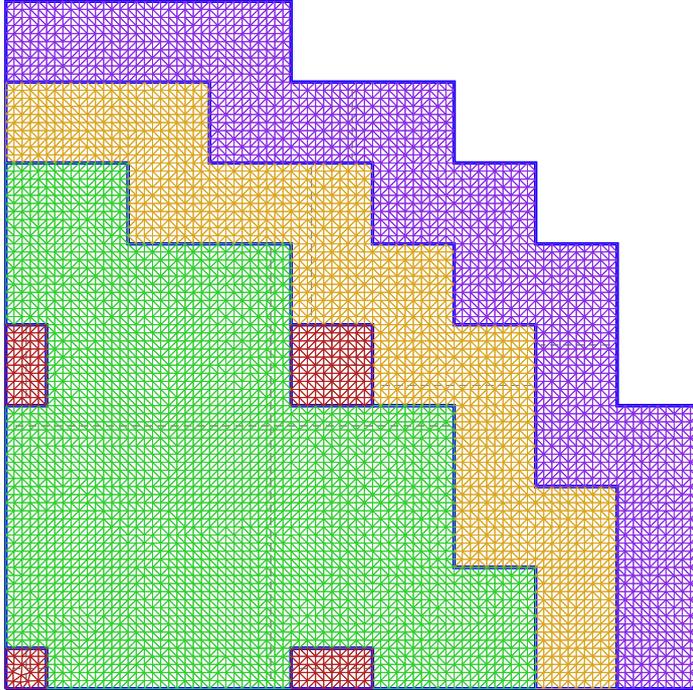
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.123



milonga's 2D LWR IAEA Benchmark Problem case #028

quarter-symmetry core meshed using delquad (triangs, $\ell_c = 2$) solved with finite elements

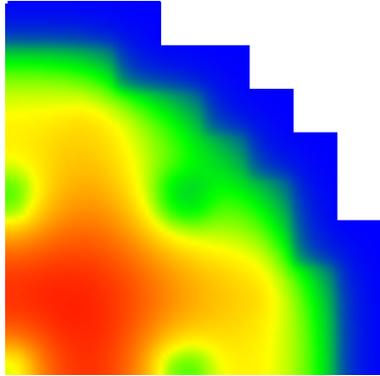


largest eigenvalue k_{eff}	1.029613 (2876.16 pcm)
$\max \phi_2(x, y)$ @core	11.15 @ (32.00, 30.00)
$\max \phi_2(x, y)$ @reflector	15.17 @ (52.00, 132.00)
number of unknowns	12392
outer iterations	3
linear iterations	32
inner iterations	1549
residual norm	2.551×10^{-8}
relative error	1.257×10^{-8}
error estimate	7.602×10^{-9}
memory used	103964 kB
soft page faults	27648
hard page faults	0
total CPU time	2.132 seconds

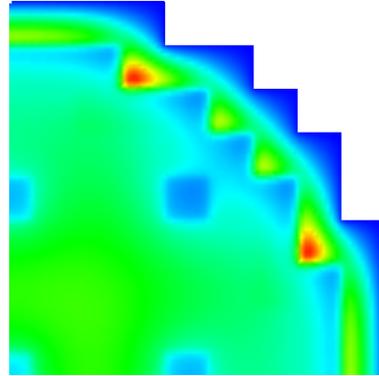
k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.27	5.51	20	1.18	37.12	8.71
2	1.30	41.60	9.63	21	1.07	33.71	7.93
3	1.45	45.60	10.70	22	0.98	29.20	7.25
4	1.20	38.49	8.92	23	0.62	16.81	5.17
5	0.61	26.49	4.51	24	—	2.50	5.94
6	0.93	29.91	6.92	25	1.19	37.55	8.81
7	0.94	29.48	6.94	26	0.96	30.84	7.15
8	0.72	20.46	5.62	27	0.91	28.54	6.72
9	—	3.43	8.08	28	0.80	22.73	6.31
10	1.43	45.05	10.56	29	—	6.07	12.56
11	1.47	46.45	10.90	30	—	0.72	2.87
12	1.31	41.34	9.69	31	0.47	20.41	3.49
13	1.07	34.10	7.90	32	0.69	20.83	5.09
14	1.04	32.71	7.67	33	0.54	14.57	4.46
15	0.95	29.93	7.06	34	—	2.47	6.13
16	0.70	19.99	5.48	35	0.52	14.14	4.38
17	—	3.28	7.70	36	—	4.08	8.44
18	1.46	46.15	10.83	37	—	0.58	2.28
19	1.34	42.28	9.92	38	—	0.66	2.61

(a) Geometry and mesh: triangles with delquad and $\ell_c = 2$

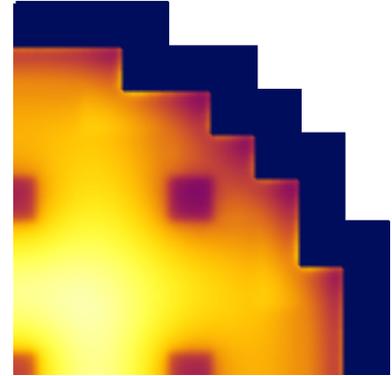
(b) Results with the finite elements method



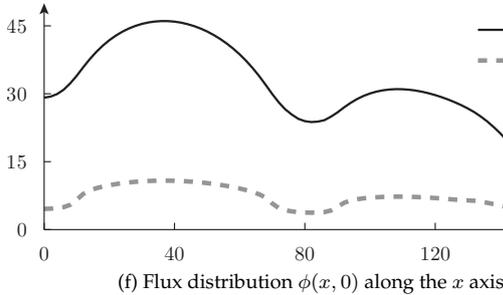
(c) Fast flux distribution



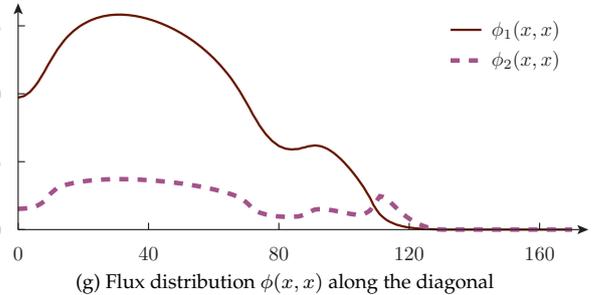
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis

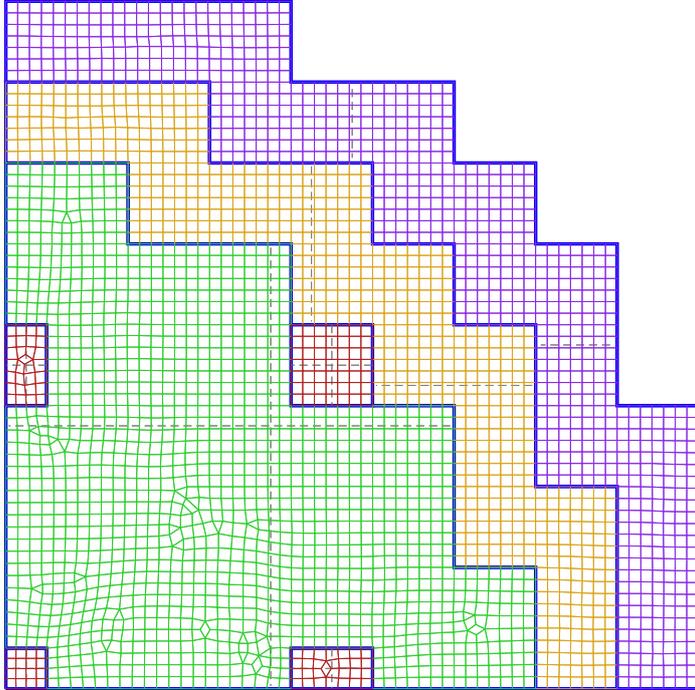


(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.124



milonga's 2D LWR IAEA Benchmark Problem case #037
 quarter-symmetry core meshed using delquad (quads, $\ell_c = 3$) solved with finite elements

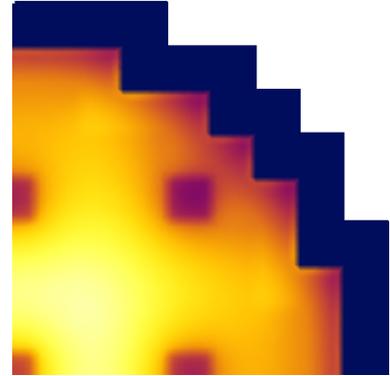
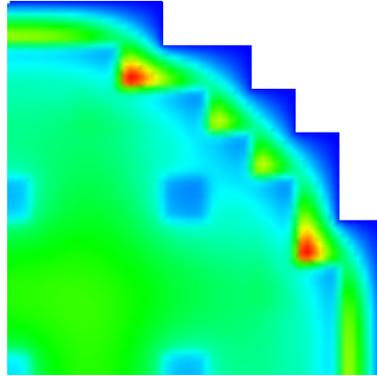
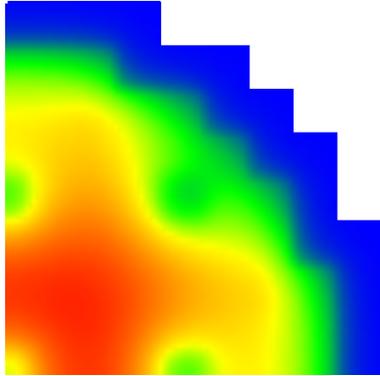


largest eigenvalue k_{eff}	1.029673 (2881.83 pcm)
$\max \phi_2(x, y)$ @core	11.08 @ (31.79, 31.77)
$\max \phi_2(x, y)$ @reflector	17.72 @ (132.86, 52.86)
number of unknowns	6140
outer iterations	3
linear iterations	32
inner iterations	767
residual norm	1.786×10^{-8}
relative error	8.8×10^{-9}
error estimate	5.639×10^{-9}
memory used	81996 kB
soft page faults	21490
hard page faults	0
total CPU time	1.14 seconds

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.06	5.48	20	1.17	37.04	8.70
2	1.29	41.34	9.56	21	1.07	33.73	7.94
3	1.44	45.34	10.64	22	0.98	29.31	7.28
4	1.20	38.29	8.86	23	0.56	16.94	5.30
5	0.61	26.42	4.53	24	—	2.56	5.97
6	0.93	29.91	6.91	25	1.19	37.42	8.77
7	0.94	29.57	6.96	26	0.96	30.79	7.13
8	0.69	20.59	5.71	27	0.91	28.58	6.74
9	—	3.52	8.12	28	0.76	22.85	6.39
10	1.42	44.75	10.49	29	—	6.20	12.57
11	1.46	46.17	10.84	30	—	0.75	2.90
12	1.30	41.13	9.64	31	0.48	20.43	3.53
13	1.06	34.00	7.86	32	0.69	20.91	5.11
14	1.03	32.70	7.67	33	0.49	14.67	4.57
15	0.96	30.02	7.09	34	—	2.53	6.16
16	0.67	20.12	5.57	35	0.47	14.22	4.47
17	—	3.37	7.72	36	—	4.16	8.43
18	1.45	45.90	10.77	37	—	0.60	2.31
19	1.33	42.10	9.88	38	—	0.67	2.64

(a) Geometry and mesh: quads with delquad and $\ell_c = 3$

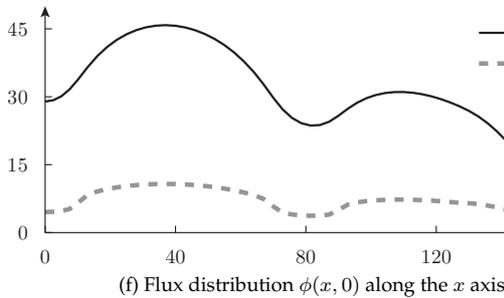
(b) Results with the finite elements method



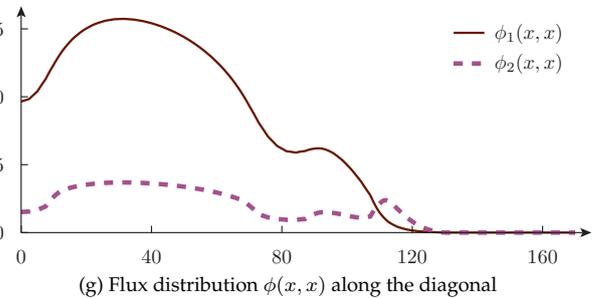
(c) Fast flux distribution

(d) Thermal flux distribution

(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



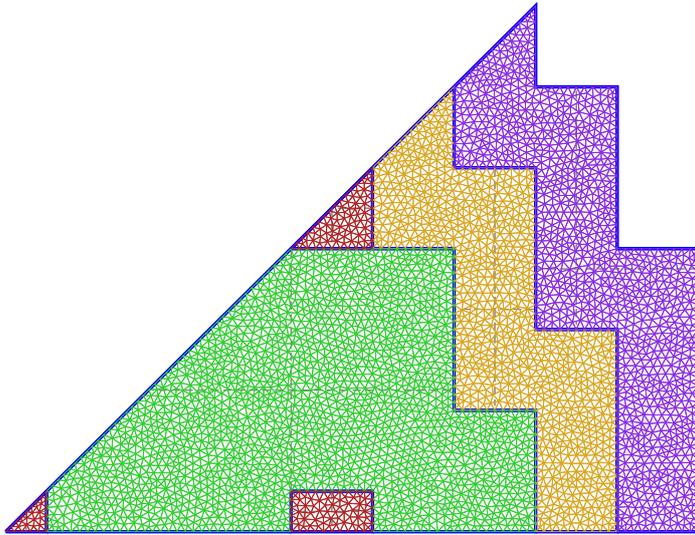
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.125



milonga's 2D LWR IAEA Benchmark Problem case #043
 eighth-symmetry core meshed using delaunay (triangs, $\ell_c = 2$) solved with finite volumes

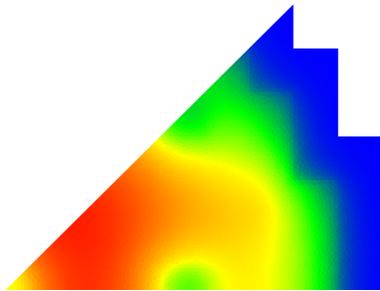
largest eigenvalue k_{eff}	1.029788 (2892.66 pcm)
$\max \phi_2(x, y)$ @core	11.19 @ (31.49, 30.11)
$\max \phi_2(x, y)$ @reflector	10.90 @ (130.38, 51.14)
number of unknowns	15776
outer iterations	3
linear iterations	32
inner iterations	1972
residual norm	2.548×10^{-12}
relative error	1.255×10^{-12}
error estimate	8.491×10^{-13}
memory used	151492 kB
soft page faults	39163
hard page faults	0
total CPU time	2.996 seconds



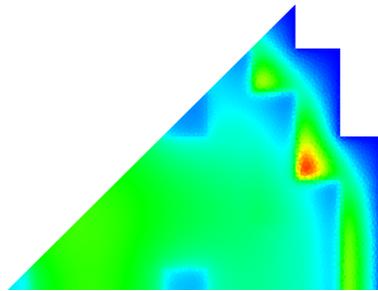
(a) Geometry and mesh: triangs with delaunay and $\ell_c = 2$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.40	5.47	20	1.18	37.19	8.73
2	1.31	41.87	9.71	21	1.07	33.68	7.92
3	1.45	45.81	10.75	22	0.97	29.11	7.22
4	1.22	38.77	9.00	23	0.67	16.48	4.98
5	0.60	26.42	4.44	24	—	2.34	5.69
6	0.93	29.83	6.91	25	1.19	37.62	8.82
7	0.93	29.38	6.91	26	0.96	30.77	7.15
8	0.74	20.28	5.52	27	0.91	28.48	6.71
9	—	3.23	7.78	28	0.83	22.56	6.17
10	1.43	45.22	10.60	29	—	5.77	12.27
11	1.48	46.66	10.95	30	—	0.68	2.65
12	1.31	41.44	9.71	31	0.46	20.18	3.42
13	1.07	34.21	7.94	32	0.68	20.61	5.06
14	1.03	32.66	7.66	33	0.59	14.44	4.36
15	0.95	29.90	7.05	34	—	2.33	5.89
16	0.73	19.86	5.38	35	0.57	13.83	4.21
17	—	3.12	7.51	36	—	3.79	8.25
18	1.47	46.36	10.88	37	—	0.55	2.11
19	1.34	42.42	9.96	38	—	0.61	2.41

(b) Results with the finite volumes method



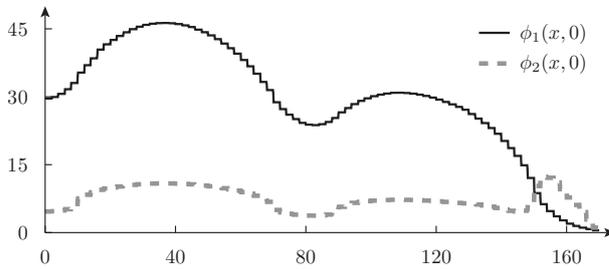
(c) Fast flux distribution



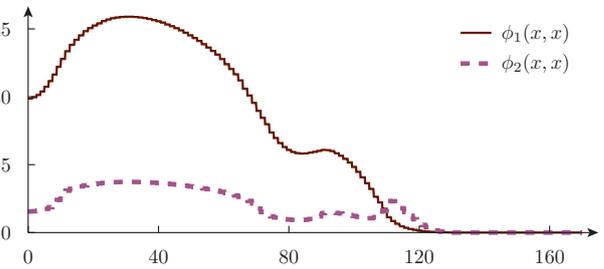
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



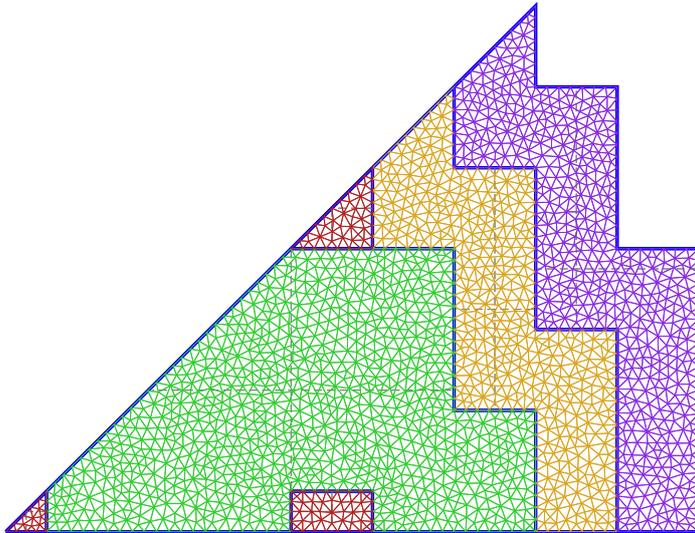
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.126



milonga's 2D LWR IAEA Benchmark Problem case #047
 eighth-symmetry core meshed using delaunay (triangs, $\ell_c = 3$) solved with finite elements

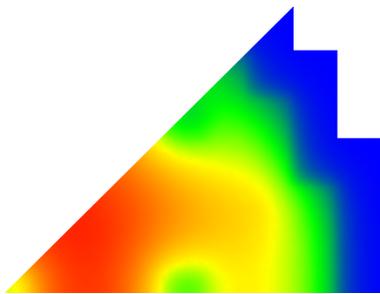
largest eigenvalue k_{eff}	1.029650 (2879.62 pcm)
$\max \phi_2(x, y)$ @core	11.08 @ (30.69, 30.69)
$\max \phi_2(x, y)$ @reflector	15.69 @ (132.20, 52.20)
number of unknowns	4040
outer iterations	2
linear iterations	24
inner iterations	505
residual norm	1.913×10^{-8}
relative error	9.425×10^{-9}
error estimate	6.785×10^{-9}
memory used	88336 kB
soft page faults	22725
hard page faults	0
total CPU time	1.496 seconds



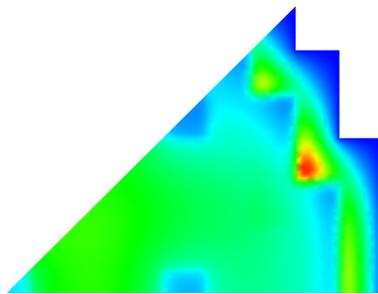
(a) Geometry and mesh: triangs with delaunay and $\ell_c = 3$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.08	5.47	20	1.17	36.96	8.68
2	1.29	41.34	9.57	21	1.07	33.61	7.91
3	1.44	45.33	10.64	22	0.97	29.16	7.24
4	1.20	38.26	8.86	23	0.58	16.83	5.23
5	0.61	26.38	4.51	24	—	2.52	5.94
6	0.93	29.82	6.89	25	1.18	37.36	8.76
7	0.94	29.44	6.93	26	0.96	30.71	7.12
8	0.70	20.46	5.65	27	0.91	28.47	6.71
9	—	3.47	8.07	28	0.79	22.72	6.33
10	1.42	44.76	10.50	29	—	6.12	12.53
11	1.46	46.16	10.84	30	—	0.73	2.88
12	1.30	41.10	9.64	31	0.47	20.36	3.49
13	1.06	33.94	7.85	32	0.68	20.80	5.08
14	1.03	32.61	7.65	33	0.51	14.57	4.51
15	0.95	29.88	7.05	34	—	2.49	6.13
16	0.68	19.99	5.51	35	0.49	14.14	4.41
17	—	3.32	7.70	36	—	4.12	8.40
18	1.45	45.88	10.77	37	—	0.59	2.29
19	1.33	42.05	9.87	38	—	0.66	2.63

(b) Results with the finite elements method



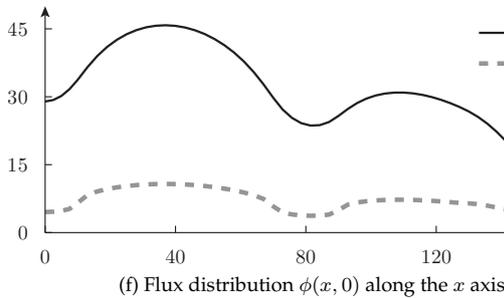
(c) Fast flux distribution



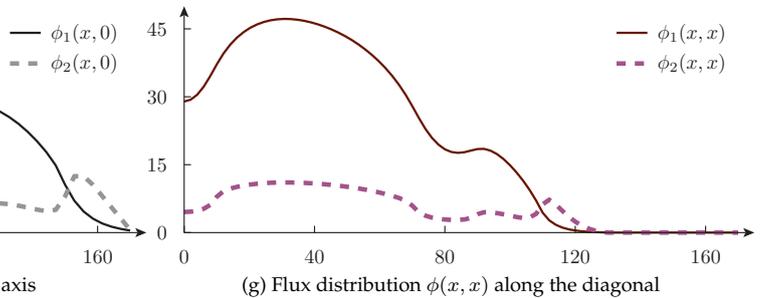
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



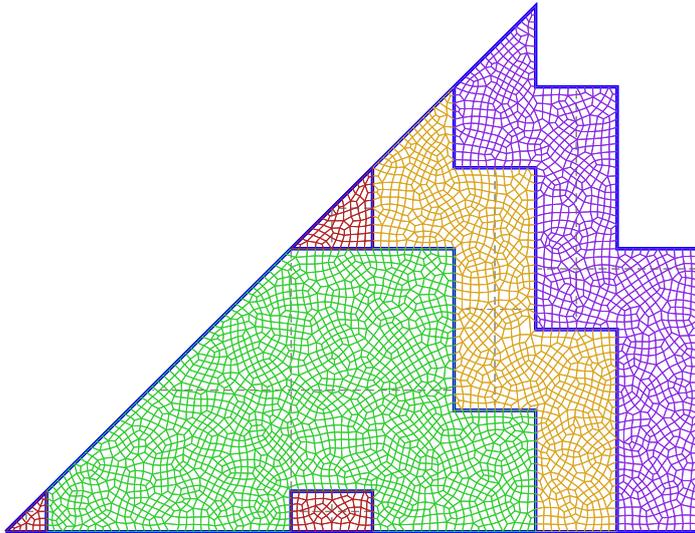
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.127



milonga's 2D LWR IAEA Benchmark Problem case #053
 eighth-symmetry core meshed using delaunay (quads, $\ell_c = 2$) solved with finite volumes

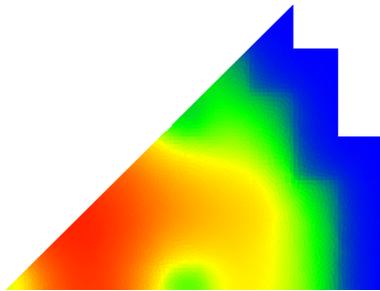
largest eigenvalue k_{eff}	1.029984 (2911.10 pcm)
$\max \phi_2(x, y)$ @core	11.11 @ (31.85, 30.64)
$\max \phi_2(x, y)$ @reflector	11.70 @ (130.75, 50.81)
number of unknowns	7936
outer iterations	3
linear iterations	32
inner iterations	992
residual norm	2.074×10^{-12}
relative error	1.022×10^{-12}
error estimate	6.981×10^{-13}
memory used	110852 kB
soft page faults	28611
hard page faults	0
total CPU time	2.164 seconds



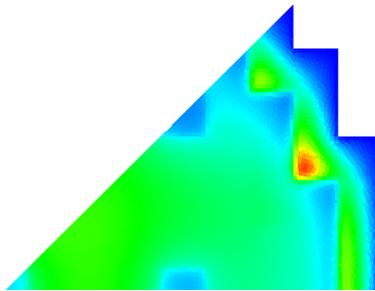
(a) Geometry and mesh: quads with delaunay and $\ell_c = 2$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.72	31.79	5.35	20	1.18	37.16	8.72
2	1.30	41.51	9.64	21	1.07	33.83	7.96
3	1.44	45.45	10.67	22	0.98	29.23	7.26
4	1.21	38.42	8.93	23	0.68	16.76	5.00
5	0.59	26.24	4.40	24	—	2.31	5.58
6	0.94	30.16	7.00	25	1.19	37.52	8.80
7	0.94	29.63	6.97	26	0.98	31.14	7.24
8	0.75	20.42	5.55	27	0.92	28.83	6.79
9	—	3.13	7.75	28	0.84	22.77	6.22
10	1.42	44.94	10.54	29	—	5.60	12.19
11	1.47	46.29	10.87	30	—	0.67	2.62
12	1.31	41.29	9.68	31	0.46	20.16	3.38
13	1.07	34.20	7.95	32	0.69	20.94	5.13
14	1.04	32.89	7.71	33	0.57	14.48	4.26
15	0.96	30.06	7.08	34	—	2.17	5.62
16	0.72	19.90	5.35	35	0.56	14.16	4.18
17	—	2.93	7.32	36	—	3.67	8.15
18	1.46	46.03	10.80	37	—	0.54	2.04
19	1.34	42.34	9.94	38	—	0.60	2.36

(b) Results with the finite volumes method



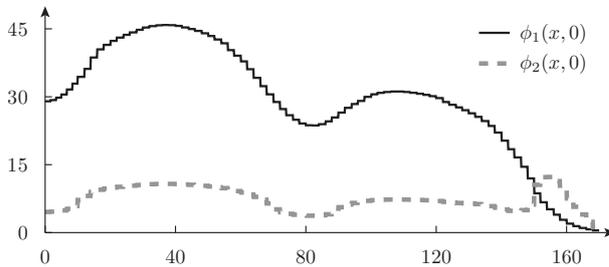
(c) Fast flux distribution



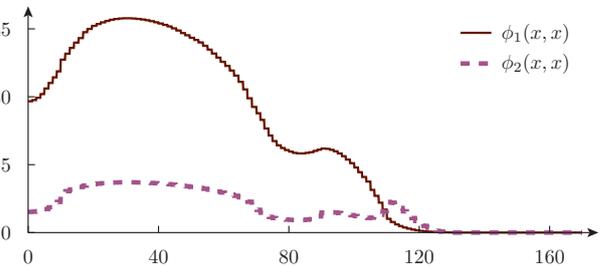
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



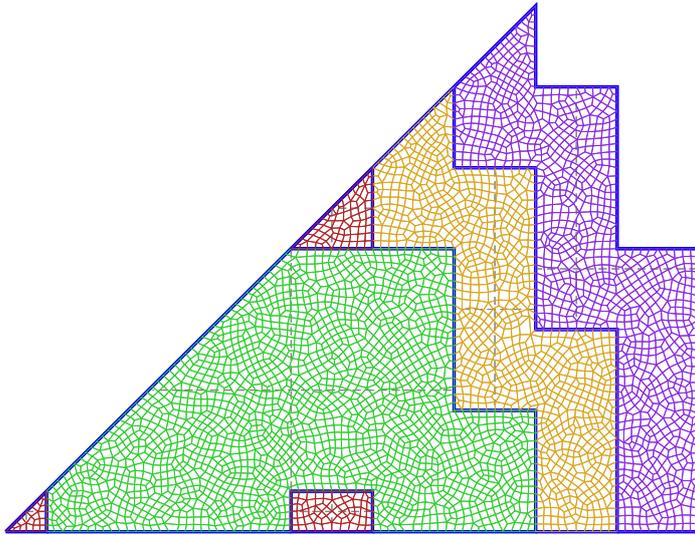
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.128



milonga's 2D LWR IAEA Benchmark Problem case #058
 eighth-symmetry core meshed using delaunay (quads, $\ell_c = 2$) solved with finite elements

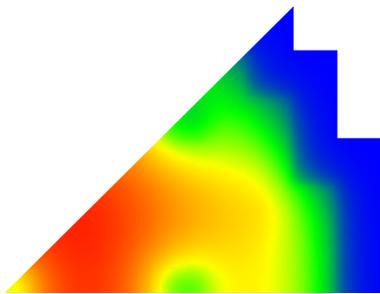
largest eigenvalue k_{eff}	1.029625 (2877.23 pcm)
$\max \phi_2(x, y)$ @core	11.12 @ (30.45, 30.45)
$\max \phi_2(x, y)$ @reflector	12.89 @ (130.96, 51.72)
number of unknowns	8210
outer iterations	3
linear iterations	32
inner iterations	1026
residual norm	3.788×10^{-12}
relative error	1.866×10^{-12}
error estimate	1.54×10^{-12}
memory used	143360 kB
soft page faults	37353
hard page faults	0
total CPU time	2.328 seconds



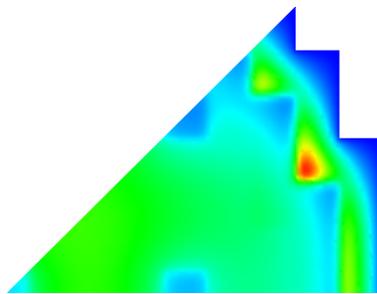
(a) Geometry and mesh: quads with delaunay and $\ell_c = 2$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.20	5.49	20	1.17	37.02	8.69
2	1.30	41.52	9.61	21	1.07	33.60	7.91
3	1.44	45.50	10.68	22	0.97	29.09	7.22
4	1.20	38.40	8.90	23	0.62	16.73	5.16
5	0.61	26.42	4.51	24	—	2.47	5.89
6	0.93	29.82	6.90	25	1.19	37.45	8.78
7	0.93	29.38	6.91	26	0.96	30.75	7.13
8	0.71	20.37	5.61	27	0.91	28.45	6.70
9	—	3.42	8.03	28	0.79	22.64	6.29
10	1.42	44.95	10.54	29	—	6.03	12.50
11	1.47	46.33	10.88	30	—	0.72	2.85
12	1.30	41.23	9.67	31	0.47	20.34	3.48
13	1.06	34.01	7.87	32	0.68	20.77	5.08
14	1.03	32.61	7.65	33	0.53	14.51	4.45
15	0.95	29.82	7.03	34	—	2.46	6.09
16	0.70	19.90	5.47	35	0.52	14.08	4.35
17	—	3.26	7.64	36	—	4.05	8.38
18	1.46	46.03	10.80	37	—	0.58	2.27
19	1.34	42.17	9.90	38	—	0.65	2.59

(b) Results with the finite elements method



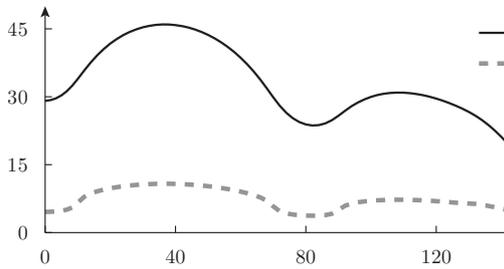
(c) Fast flux distribution



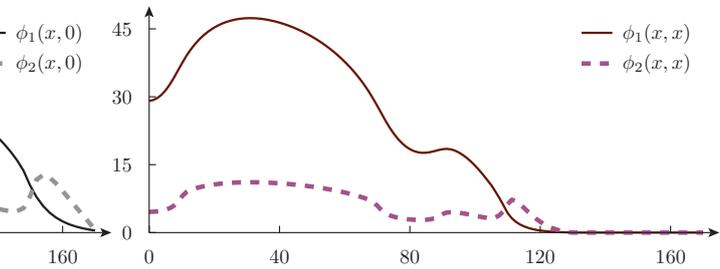
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



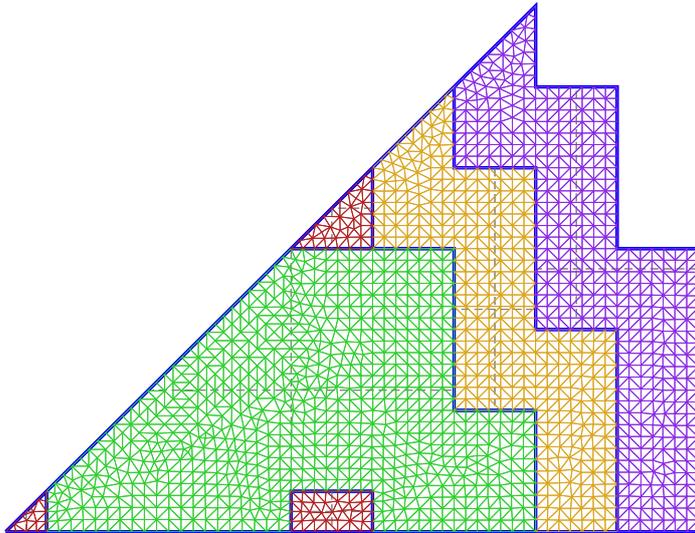
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.129



milonga's 2D LWR IAEA Benchmark Problem case #062
 eighth-symmetry core meshed using delquad (triangs, $\ell_c = 3$) solved with finite volumes

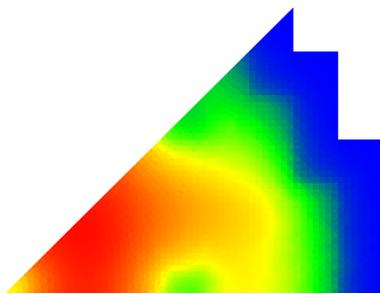
largest eigenvalue k_{eff}	1.030002 (2912.78 pcm)
$\max \phi_2(x, y)$ @core	11.56 @ (32.07, 30.69)
$\max \phi_2(x, y)$ @reflector	12.30 @ (130.95, 50.95)
number of unknowns	6016
outer iterations	3
linear iterations	32
inner iterations	752
residual norm	1.136×10^{-11}
relative error	5.597×10^{-12}
error estimate	1.91×10^{-12}
memory used	158876 kB
soft page faults	40337
hard page faults	0
total CPU time	2.588 seconds



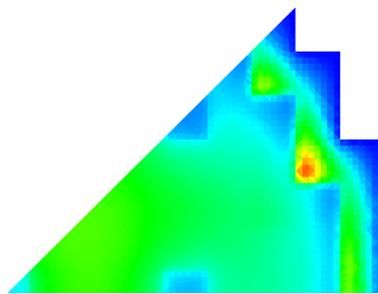
(a) Geometry and mesh: triangs with delquad and $\ell_c = 3$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.75	33.13	5.57	20	1.19	37.62	8.83
2	1.34	42.89	9.95	21	1.07	33.66	7.92
3	1.49	47.02	11.04	22	0.96	28.53	7.09
4	1.22	38.86	9.01	23	0.65	16.02	4.81
5	0.60	26.48	4.49	24	—	2.12	5.19
6	0.92	29.50	6.84	25	1.21	38.11	8.94
7	0.92	28.94	6.81	26	0.97	30.93	7.18
8	0.72	19.64	5.34	27	0.90	28.29	6.66
9	—	3.01	7.32	28	0.81	21.92	6.01
10	1.47	46.56	10.92	29	—	5.40	11.57
11	1.53	48.21	11.32	30	—	0.69	2.53
12	1.34	42.30	9.92	31	0.46	19.97	3.38
13	1.08	34.45	7.99	32	0.67	20.20	4.95
14	1.03	32.42	7.60	33	0.56	13.79	4.15
15	0.93	29.28	6.90	34	—	2.17	5.34
16	0.70	19.21	5.20	35	0.54	13.14	3.99
17	—	2.90	6.73	36	—	3.67	7.77
18	1.52	47.85	11.23	37	—	0.50	1.89
19	1.37	43.29	10.16	38	—	0.55	2.10

(b) Results with the finite volumes method



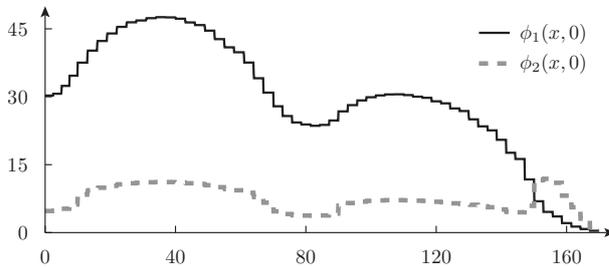
(c) Fast flux distribution



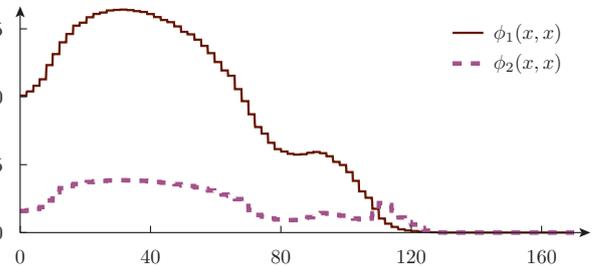
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



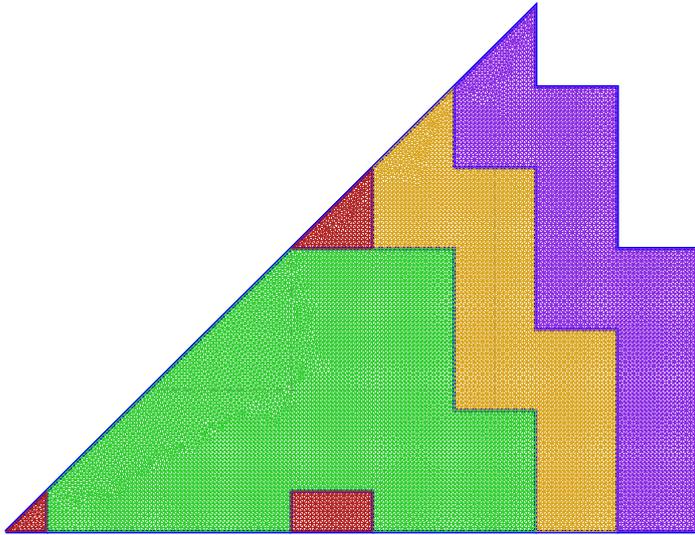
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.130



milonga's 2D LWR IAEA Benchmark Problem case #069
eighth-symmetry core meshed using delquad (triangs, $\ell_c = 1$) solved with finite elements

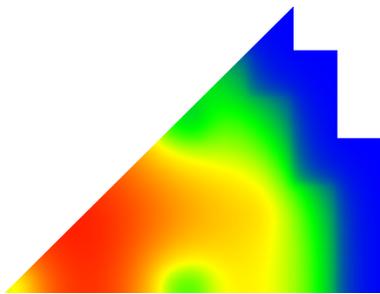
largest eigenvalue k_{eff}	1.029596 (2874.53 pcm)
$\max \phi_2(x, y)$ @core	11.15 @ (30.47, 30.47)
$\max \phi_2(x, y)$ @reflector	11.85 @ (131.00, 51.00)
number of unknowns	24894
outer iterations	3
linear iterations	32
inner iterations	3111
residual norm	1.484×10^{-12}
relative error	7.313×10^{-13}
error estimate	8.543×10^{-13}
memory used	197960 kB
soft page faults	53088
hard page faults	0
total CPU time	4.588 seconds



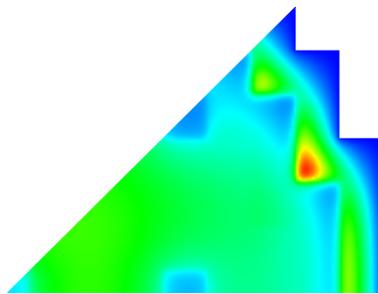
(a) Geometry and mesh: triangs with delquad and $\ell_c = 1$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.74	32.31	5.50	20	1.18	37.07	8.70
2	1.30	41.68	9.65	21	1.07	33.60	7.91
3	1.45	45.66	10.72	22	0.97	29.03	7.21
4	1.21	38.52	8.93	23	0.63	16.65	5.13
5	0.61	26.46	4.50	24	—	2.45	5.86
6	0.93	29.84	6.90	25	1.19	37.53	8.80
7	0.93	29.34	6.91	26	0.96	30.79	7.14
8	0.72	20.30	5.59	27	0.90	28.42	6.70
9	—	3.38	7.99	28	0.80	22.56	6.26
10	1.43	45.11	10.58	29	—	5.97	12.43
11	1.47	46.48	10.91	30	—	0.71	2.82
12	1.31	41.35	9.70	31	0.47	20.33	3.48
13	1.06	34.08	7.89	32	0.68	20.72	5.07
14	1.03	32.62	7.65	33	0.55	14.43	4.42
15	0.95	29.77	7.02	34	—	2.43	6.05
16	0.70	19.83	5.45	35	0.53	14.01	4.34
17	—	3.23	7.60	36	—	4.01	8.33
18	1.46	46.16	10.84	37	—	0.57	2.25
19	1.34	42.27	9.92	38	—	0.65	2.57

(b) Results with the finite elements method



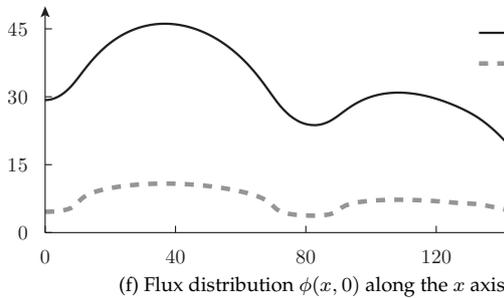
(c) Fast flux distribution



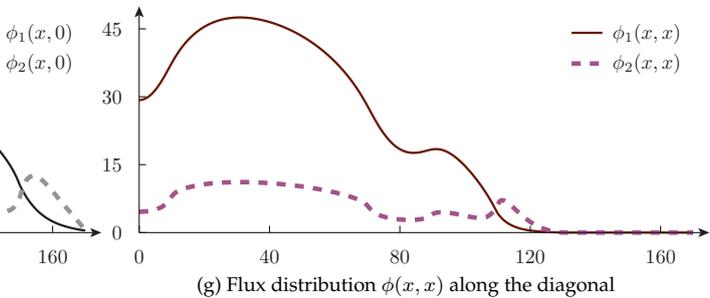
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



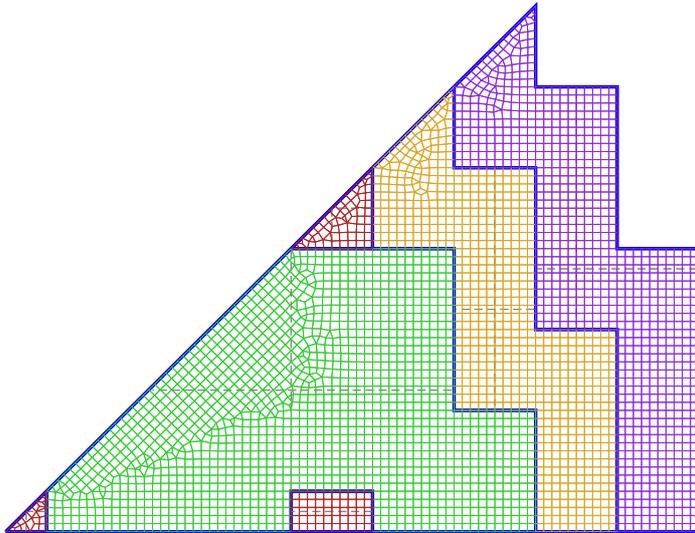
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.131



milonga's 2D LWR IAEA Benchmark Problem case #073
 eighth-symmetry core meshed using delquad (quads, $\ell_c = 2$) solved with finite volumes

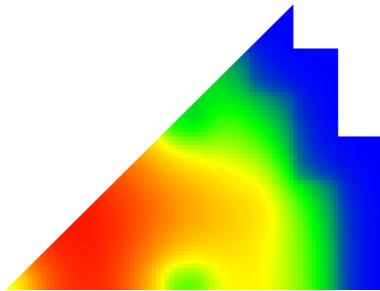
largest eigenvalue k_{eff}	1.029465 (2862.13 pcm)
$\max \phi_2(x, y)$ @core	11.35 @ (30.93, 29.53)
$\max \phi_2(x, y)$ @reflector	12.33 @ (131.00, 51.00)
number of unknowns	6234
outer iterations	2
linear iterations	24
inner iterations	779
residual norm	3.804×10^{-8}
relative error	1.875×10^{-8}
error estimate	7.96×10^{-9}
memory used	83448 kB
soft page faults	21581
hard page faults	0
total CPU time	1.532 seconds



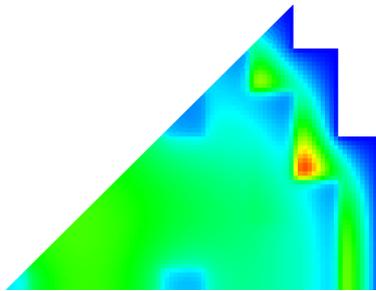
(a) Geometry and mesh: quads with delquad and $\ell_c = 2$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.75	32.94	5.55	20	1.19	37.46	8.79
2	1.34	42.66	9.92	21	1.07	33.59	7.90
3	1.48	46.57	10.93	22	0.96	28.80	7.14
4	1.23	39.23	9.13	23	0.64	16.36	4.75
5	0.59	26.42	4.40	24	—	2.17	5.63
6	0.94	29.95	6.96	25	1.20	37.94	8.90
7	0.93	29.13	6.85	26	0.98	31.14	7.25
8	0.72	20.02	5.34	27	0.90	28.31	6.66
9	—	3.01	7.73	28	0.81	22.29	6.00
10	1.46	46.03	10.79	29	—	5.36	12.57
11	1.50	47.33	11.11	30	—	0.62	2.50
12	1.33	42.08	9.87	31	0.45	20.06	3.36
13	1.08	34.44	8.01	32	0.68	20.47	5.02
14	1.03	32.69	7.67	33	0.55	14.17	4.10
15	0.94	29.53	6.96	34	—	2.16	5.79
16	0.70	19.53	5.21	35	0.55	13.63	4.10
17	—	2.88	7.35	36	—	3.61	8.39
18	1.49	46.88	11.01	37	—	0.50	2.00
19	1.36	42.84	10.06	38	—	0.61	2.35

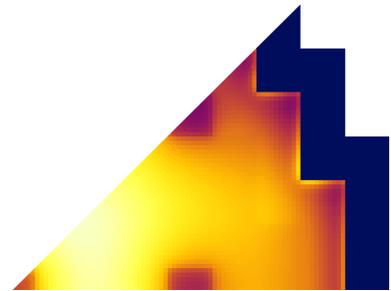
(b) Results with the finite volumes method



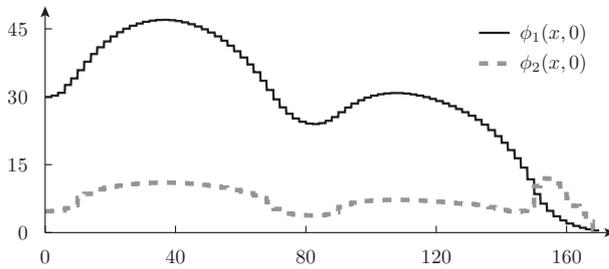
(c) Fast flux distribution



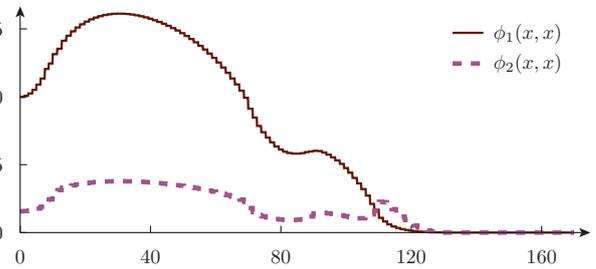
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



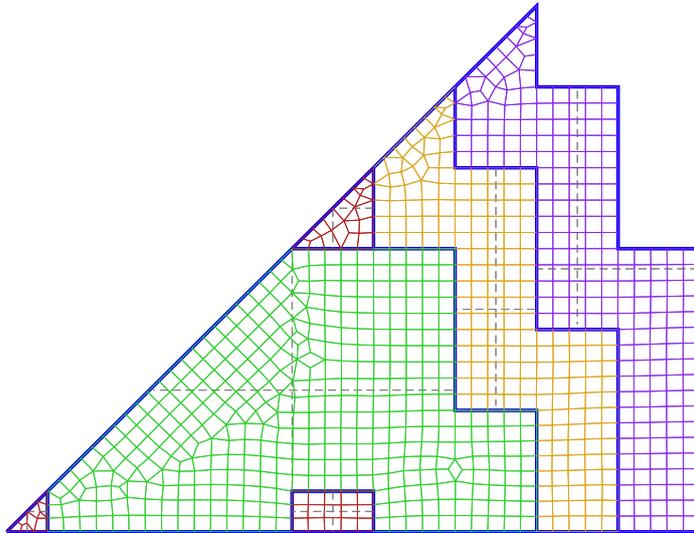
(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.132



milonga's 2D LWR IAEA Benchmark Problem case #076
 eighth-symmetry core meshed using delquad (quads, $\ell_c = 4$) solved with finite elements

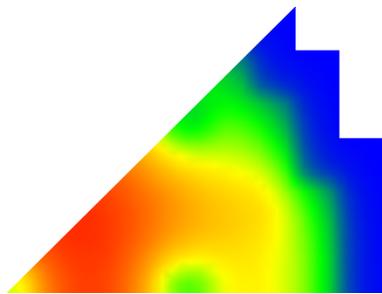
largest eigenvalue k_{eff}	1.029772 (2891.11 pcm)
$\max \phi_2(x, y)$ @core	10.94 @ (31.82, 31.82)
$\max \phi_2(x, y)$ @reflector	20.16 @ (134.00, 54.00)
number of unknowns	1750
outer iterations	3
linear iterations	32
inner iterations	218
residual norm	3.266×10^{-12}
relative error	1.609×10^{-12}
error estimate	9.738×10^{-13}
memory used	93700 kB
soft page faults	23897
hard page faults	0
total CPU time	1.416 seconds



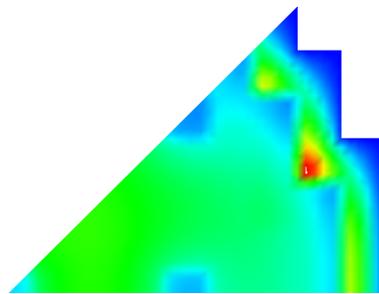
(a) Geometry and mesh: quads with delquad and $\ell_c = 4$

k	P_k	ϕ_{1k}	ϕ_{2k}	k	P_k	ϕ_{1k}	ϕ_{2k}
1	0.73	31.66	5.42	20	1.17	36.78	8.63
2	1.27	40.77	9.43	21	1.07	33.64	7.92
3	1.42	44.76	10.51	22	0.97	29.41	7.30
4	1.18	37.81	8.74	23	0.53	17.15	5.38
5	0.61	26.20	4.51	24	—	2.62	6.09
6	0.93	29.77	6.87	25	1.17	37.09	8.70
7	0.94	29.60	6.97	26	0.96	30.61	7.08
8	0.67	20.74	5.77	27	0.91	28.58	6.74
9	—	3.60	8.26	28	0.73	23.03	6.46
10	1.40	44.16	10.35	29	—	6.35	12.77
11	1.45	45.59	10.70	30	—	0.77	2.98
12	1.29	40.65	9.53	31	0.48	20.41	3.53
13	1.05	33.71	7.79	32	0.68	20.99	5.12
14	1.03	32.57	7.64	33	0.46	14.86	4.64
15	0.96	30.07	7.10	34	—	2.60	6.29
16	0.66	20.29	5.62	35	0.45	14.39	4.53
17	—	3.44	7.86	36	—	4.28	8.58
18	1.44	45.37	10.65	37	—	0.62	2.37
19	1.32	41.67	9.78	38	—	0.69	2.71

(b) Results with the finite elements method



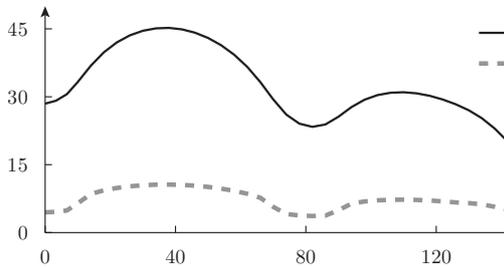
(c) Fast flux distribution



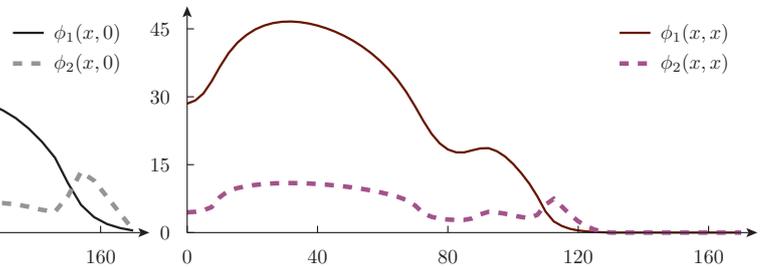
(d) Thermal flux distribution



(e) Power distribution



(f) Flux distribution $\phi(x, 0)$ along the x axis



(g) Flux distribution $\phi(x, x)$ along the diagonal

Figure 5.133

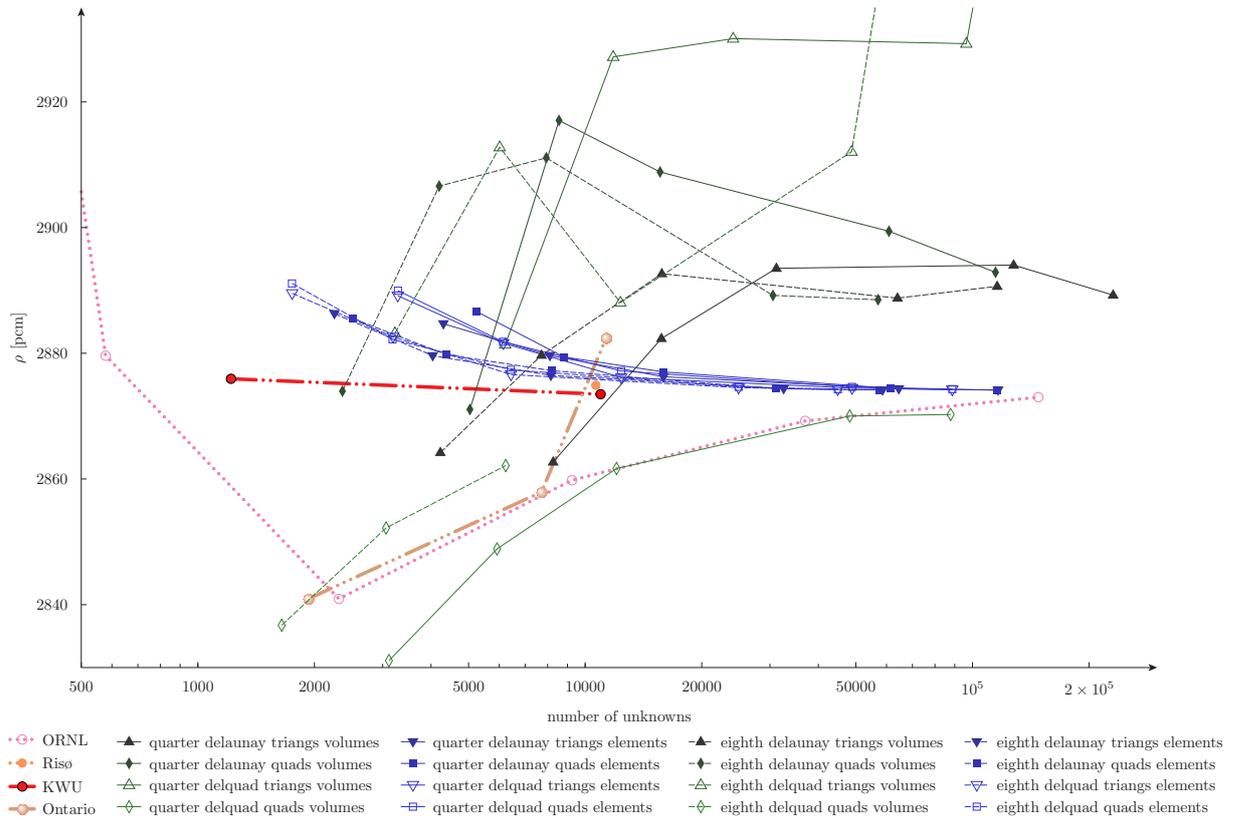


Figure 5.134 – Static reactivity vs. number of unknowns. The four original solutions as published in 1976 [38] are included as reference.

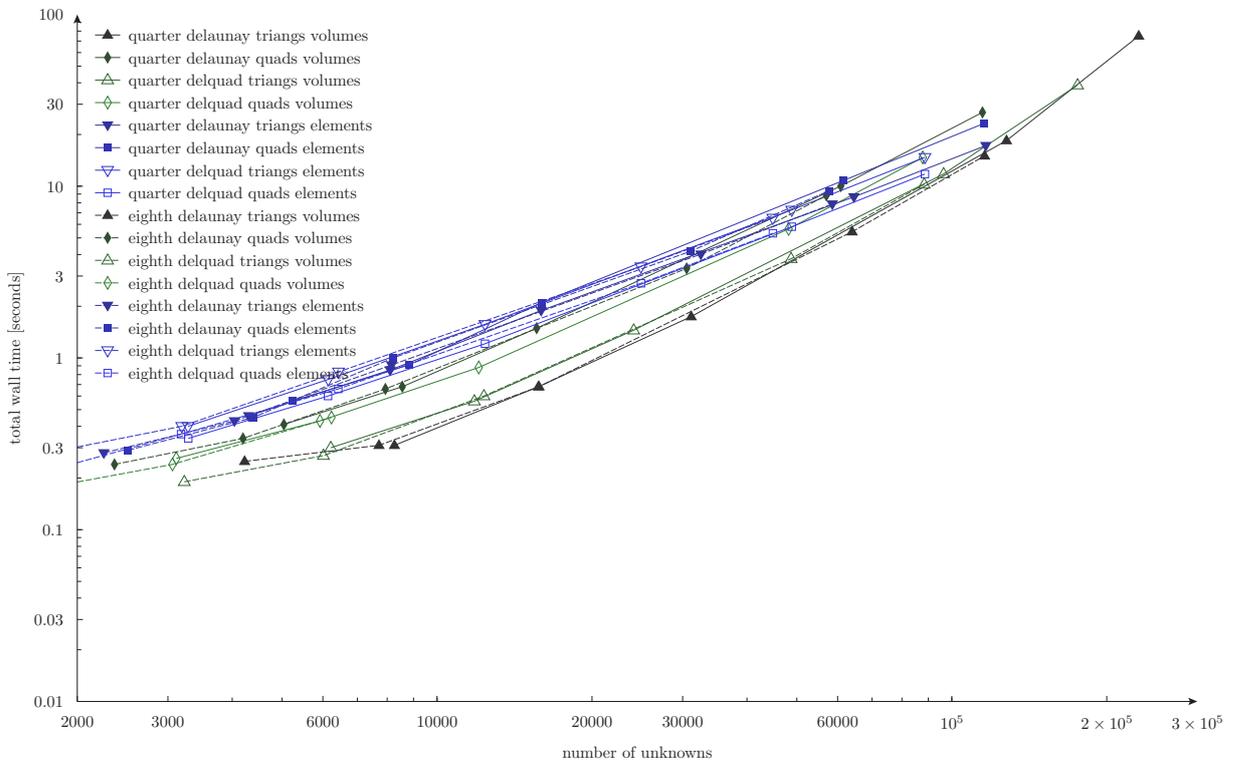


Figure 5.135 – Total wall time vs. number of unknowns

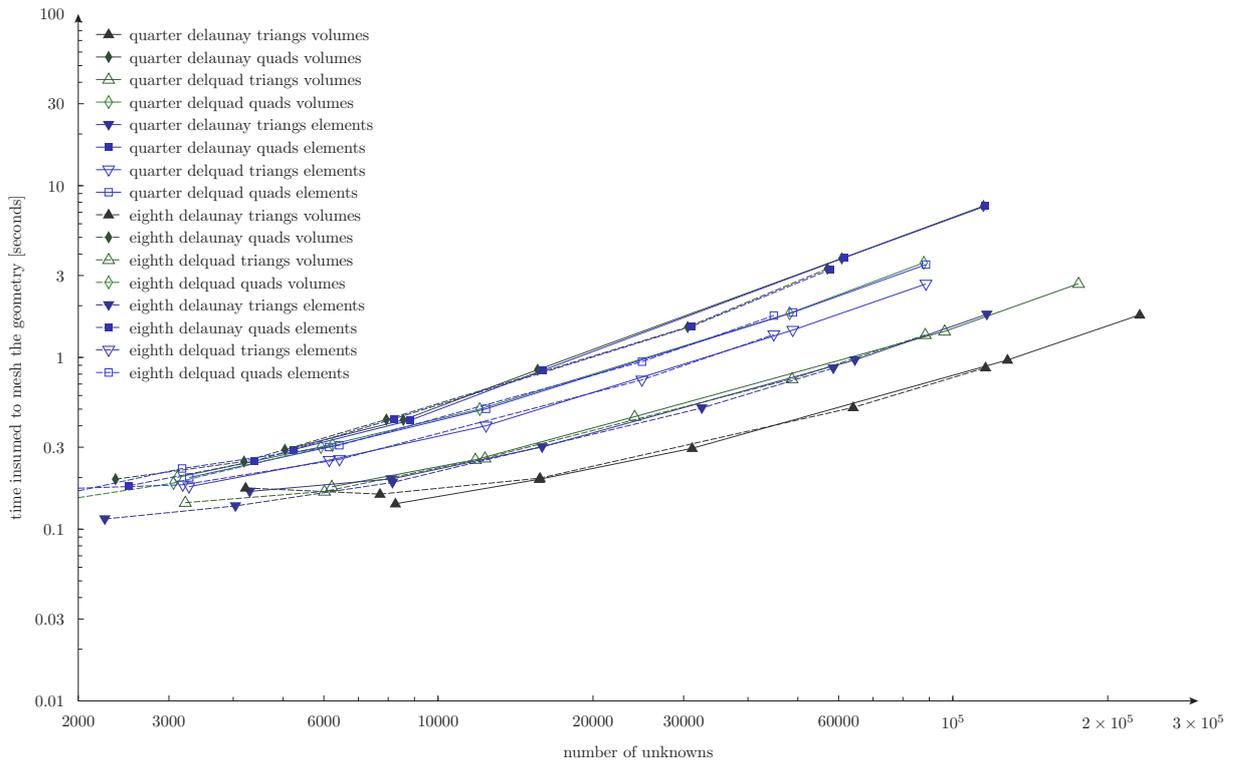


Figure 5.136 – Time needed to mesh the geometry vs. number of unknowns

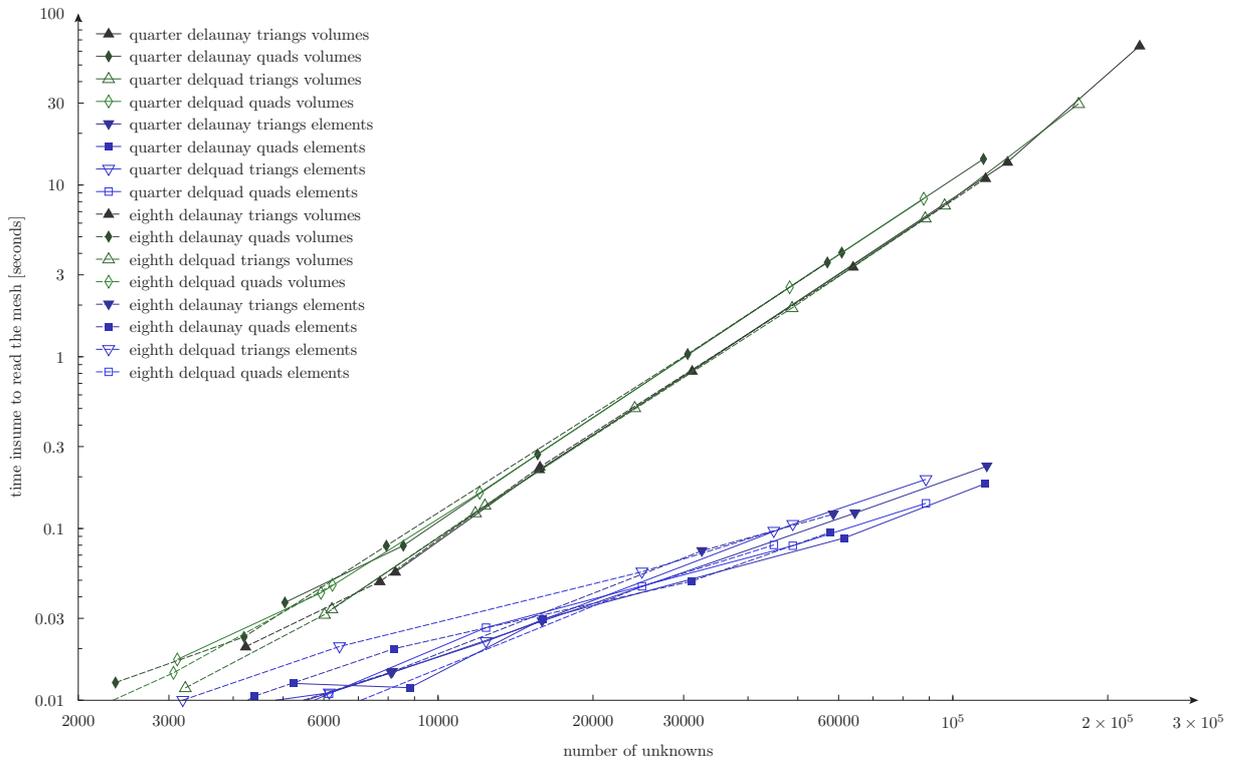


Figure 5.137 – Time needed to read the mesh vs. number of unknowns

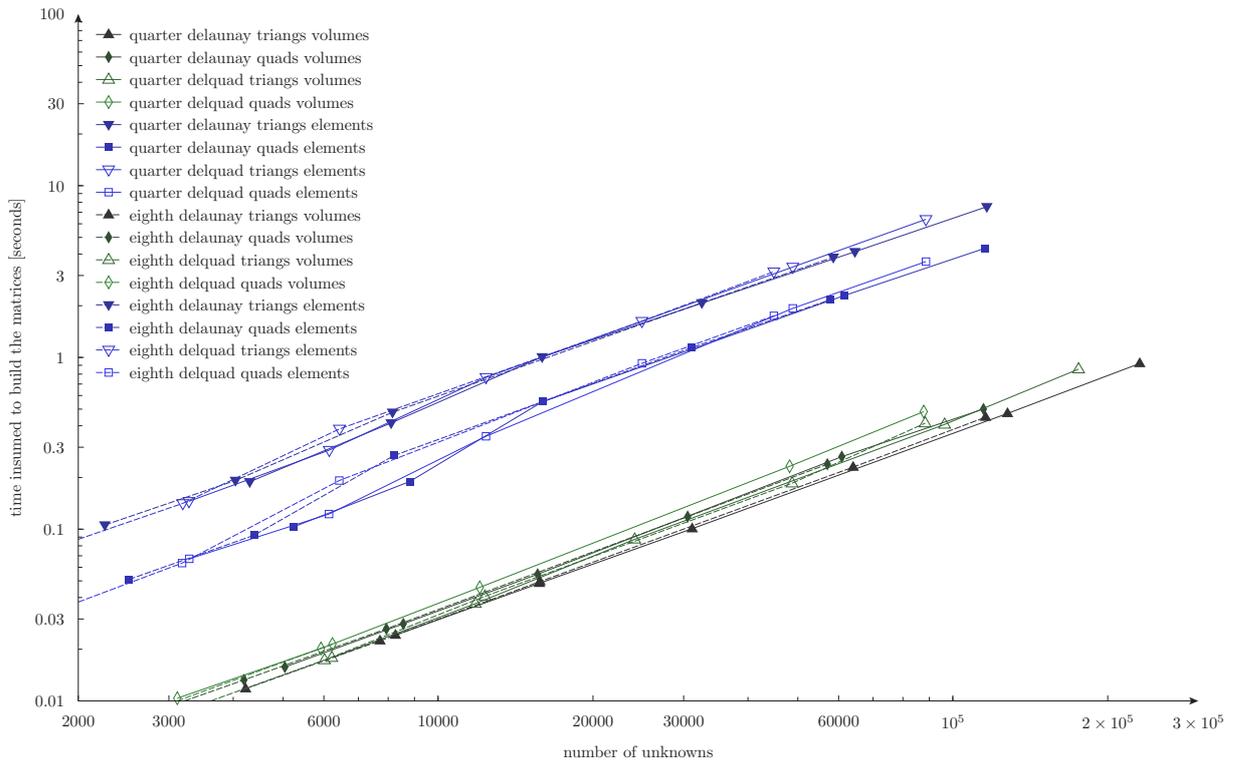


Figure 5.138 – Time needed to build the matrices vs. number of unknowns

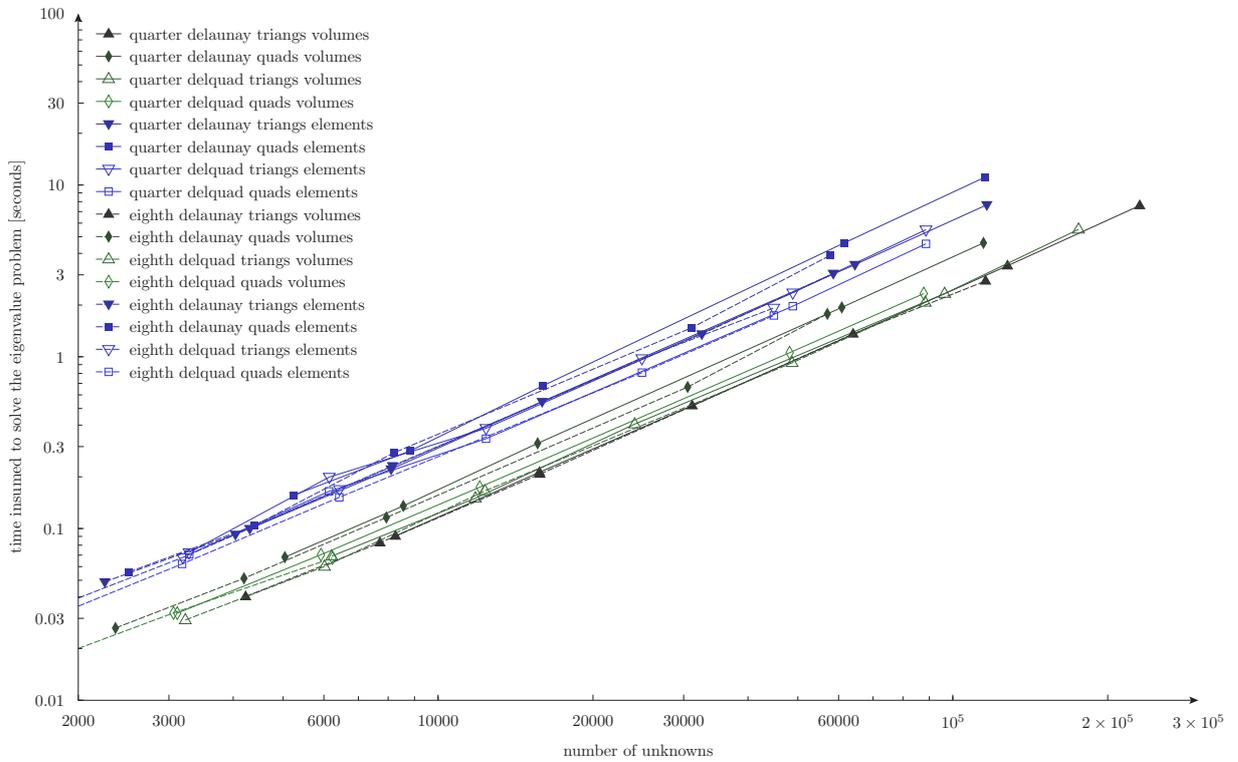


Figure 5.139 – Time needed to solve the eigenvalue problem vs. number of unknowns

Conclusiones

Il semble que la perfection soit atteinte
non quand il n'y a plus rien à ajouter,
mais quand il n'y a plus rien à retrancher.

Parecería que la perfección es alcanzada
no cuando no hay más nada que agregar,
sino cuando no hay mas nada que quitar.

Antoine de Saint Exupéry, Terre des Hommes, 1939

Aún cuando hayamos comenzado esta monografía hemos introducido las bases matemáticas de la ecuación de difusión de neutrones para el lector no familiarizado con las particularidades de la física de reactores, remarcamos una vez más que este trabajo es estrictamente sobre matemática. Se trata de comparar dos esquemas diferentes que se pueden utilizar para discretizar ecuaciones diferenciales en derivadas parciales en general y la ecuación de difusión de neutrones multigrupo en particular. La aplicación de los métodos matemáticos aquí introducidos y desarrollados a problemas de interés—tanto académicos como industriales—desde el punto de vista ingenieril están fuera del alcance, y serán objeto de nuevos y diferentes estudios.

La mayor parte del trabajo se basó en el desarrollo y la aplicación de un cierto código computacional escrito desde cero por el autor—aunque depende de ciertas librerías libres, en particular la GNU Scientific Library [37], la biblioteca PETSc [8] y la biblioteca SLEPc [10]—llamado milonga y distribuido libremente bajo la licencia general GNU. Los capítulos 3 y 4 fueron escritos para dar un sustento matemático a las línea de código de milonga. De hecho fueron encontrados—y con un poco de suerte corregidos—numerosos errores de programación. El capítulo 5 consistió en aplicar el código desarrollado a un conjunto de problemas introducidos en un orden de complejidad creciente. Aunque la longitud del escrito parezca—y tal vez lo sea—excesiva, incluso han quedado muchos interrogantes por estudiar y analizar. Incluso hay muchos problemas que han quedado fuera del capítulo de resultados y cuya diferencias al resolverlos con uno y otro método podrían arrojan nuevas e interesantes conclusiones, como por ejemplo el estudio de la influencia en la estructura de las matrices sobre el esfuerzo computacional necesario para resolver el problema de autovalores asociado.

En el capítulo 5 encontramos que para los problemas con solución analítica las soluciones que arroja el método basado en elementos son más precisos que las encontradas con volúmenes finitos para una tamaño de problema numérico equivalente. Además, encontramos que en dos dimensiones, la solución en elementos converge monótonamente con el refinamiento de la malla, mientras que la solución en volúmenes no lo hace. La explicación de este fenómeno reside en que al ir disminuyendo el tamaño característico Δ de las celdas/elementos en forma cuasi-continua, la topología de la malla resultante no varía en forma continua y puede darse



el caso de que pequeños cambios en Δ provoque una reorganización mayor de las celdas/elementos. Como el método de volúmenes finitos se basa en aproximar el flujo en las caras de las celdas a partir de consideraciones geométricas, si la geometría cambia en forma discontinua con Δ , es de esperar que el resultado del problema también lo haga. El método de elementos finitos, al depender básicamente de jacobianos de funciones de forma, es capaz de suavizar estos cambios en la topología y proveer resultados más continuos con respecto a Δ , lo que puede llegar a ser una ventaja con respecto a volúmenes finitos en algunos casos. Sin embargo—y aunque ya hemos dicho que las consideraciones ingenieriles están fuera del alcance—esto no quiere decir que un esquema basado en elementos dará siempre resultados más precisos ya que puede darse el caso de que una formulación en volúmenes sea más adecuada para representar la física detrás de la matemática de la ecuación de difusión de neutrones, basada en secciones eficaces homogenizadas a partir de cálculos de celda realizados justamente usando celdas y no elementos como célula básica. Análisis detallados de este tipo son necesarios para determinar en qué caso conviene aplicar uno u otro esquema de discretización espacial.

6.1. Trabajos futuros

El principal trabajo a futuro consiste en continuar desarrollando el código milonga para lograr obtener una herramienta potente y con eventuales aplicaciones en ingeniería. Hasta el momento solamente el código puede manejar geometrías en una o en dos dimensiones. Los problemas tridimensionales—que finalmente son los que pueden llegar a tener alguna potencial aplicación industrial—son el principal objetivo a implementar en el corto plazo.

Aprovechando las facilidades provistas por las bibliotecas numéricas de las que depende milonga, otro objetivo a mediano plazo consiste en paralelizar tanto la construcción de las matrices como la solución numérica del problema de autovalores. Incluso deberíamos evaluar la posibilidad de utilizar GPUs en lugar de CPUs para problemas grandes. Incluso hay muchos algoritmos ad-hoc dentro del código que pueden ser optimizados, ya que como suele suceder en programas diseñados bajo la filosofía UNIX [2], las primeras versiones suelen estar programadas “para que ande” en lugar de que “para que ande rápido”. Un ejemplo de este tipo de optimización podría ser implementar la construcción de las matrices elementales utilizando algún esquema de integración adaptiva para intentar o bien reducir el número de evaluaciones de los jacobianos o bien mantener acotado el error cometido en la cuadratura numérica. Otras sofisticaciones tales como intentar no empeorar el radio espectral de la matriz de remociones R al introducir condiciones de contorno de Dirichlet o tratar de mantener la simetría después de colocar las condiciones de contorno también deberían ser estudiadas e incorporadas.

En lugar de utilizar la aproximación de difusión, algunos códigos computacionales industriales utilizan alguna otra formulación basada en la ecuación de transporte de Maxwell, tal como la formulación S_N . Eventualmente podríamos incorporar a milonga la posibilidad de manejar este tipo de ecuaciones para aumentar su espectro de aplicación.

A más largo plazo, la posibilidad de resolver problemas transitorios debería ser incorporada eventualmente al código, una vez que se haya logrado cierta madurez, estabilidad y eficiencia en la resolución de problemas estacionarios. Aún más deseable que resolver cinética espacial resultaría el poder calcular con milonga los coeficientes que definen la formulación de cinética multi-puntual equivalente del problema tridimensional completo.

El código milonga está siendo desarrollado en el marco de una Tesis de Doctorado en Ingeniería Nuclear sobre temas de optimización de parámetros de diseño en reactores nucleares. Si bien en el ejemplo 5.2.3.3 intentamos introducir la forma en la que milonga puede utilizarse para resolver problemas complejos de optimización, para poder considerarlo un código apto para la optimización de parámetros de diseño de reactores nucleares reales. Pero es ciertamente



éste el principal objetivo a largo plazo. Y solamente este objetivo será alcanzado si a lo largo de todo el camino seguimos las bases del software libre, en el que tanto los usuarios como el programador se benefician. Los primeros porque tienen la posibilidad de ejecutar el programa, de regalarlo—y por lo tanto recibirlo—libremente, de estudiar como funciona y de cambiarlo para que se ajuste a sus necesidades (libertades cero, uno y dos). Y el autor se beneficia cuando los usuarios ejercitan su libertad número cuatro de redistribuir las versiones modificadas, ya que sus errores tendrán una mayor posibilidad de ser corregidos, configurando un lazo con realimentación positiva y aumentando también las chances de que la ciencia y la tecnología nuclear avancen un poquito más rápido que antes.

Algunos resultados de la geometría del plano

Este apéndice enuncia sin demostrar algunas fórmulas que utiliza el código milonga para calcular algunos resultados geométricos necesarios para la implementación del método de volúmenes finitos en dos dimensiones.

A.1. La normal a un segmento

Dado un segmento en el plano x - y definido por dos puntos $\mathbf{x}_1 = (x_1, y_1)$ y $\mathbf{x}_2 = (x_2, y_2)$ existen dos vectores unitarios $\hat{\mathbf{n}}$ y $\hat{\mathbf{n}}'$ cuya dirección es perpendicular a dicho segmento (figura A.1).

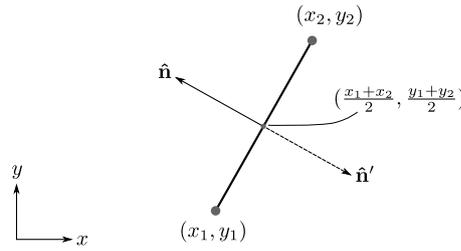


Figura A.1 – Vector normal a un segmento

Las componentes de estos vectores son

$$\hat{\mathbf{n}} = \left(\frac{y_1 - y_2}{\|\mathbf{x}_2 - \mathbf{x}_1\|}, \frac{x_2 - x_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|} \right) \quad (\text{A.1})$$

$$\hat{\mathbf{n}}' = \left(\frac{y_2 - y_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|}, \frac{x_1 - x_2}{\|\mathbf{x}_2 - \mathbf{x}_1\|} \right) \quad (\text{A.2})$$

$$(\text{A.3})$$

Si el segmento $\mathbf{x}_2 - \mathbf{x}_1$ es el borde de una celda i , entonces estamos interesados en el vector normal externo a la celda i . Por ejemplo, en la figura A.2, para la celda i el vector normal externo es $\hat{\mathbf{n}}'$.

Para saber cómo elegir el vector apropiado para una cierta celda i , calculamos las coordenadas del vector $\hat{\mathbf{n}}$ según la ecuación (A.1) y calculamos el producto interno

$$\hat{\mathbf{n}} \cdot (\mathbf{x}_i - \mathbf{x}_{ij})$$

entre el vector normal que proponemos y el vector que une el centro \mathbf{x}_i de la celda i con el centro \mathbf{x}_{ij} del segmento que separa la celda i de la celda j . Si este producto interno es negativo,

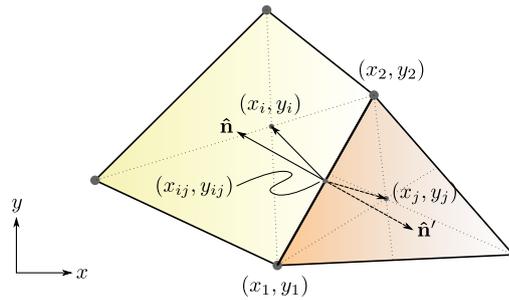


Figura A.2 – Vector normal externo a una celda

entonces el vector \hat{n} propuesto es el que está en la dirección normal externa. De lo contrario, el vector que estamos buscando es \hat{n}' .

A.2. El área del triángulo

Dado un triángulo en el plano x - y definido por tres puntos $\mathbf{x}_1 = (x_1, y_1)$, $\mathbf{x}_2 = (x_2, y_2)$ y $\mathbf{x}_3 = (x_3, y_3)$ (figura A.3), es posible demostrar que su área es igual a

$$A = \frac{1}{2} [x_1 \cdot (y_2 - y_3) + x_2 \cdot (y_3 - y_1) + x_3 \cdot (y_1 - y_2)] \quad (\text{A.4})$$

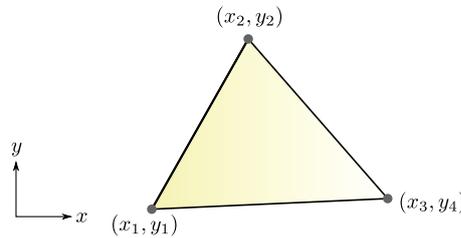


Figura A.3 – Triángulo definido por tres puntos

A.3. El área del cuadrángulo

Si ahora consideramos un cuadrángulo definido por cuatro puntos (figura A.4), podemos calcular su área si lo dividimos en dos triángulos, calculamos sus áreas según la ecuación (A.4) y las sumamos:

$$A = \frac{1}{2} [x_1 \cdot (y_2 - y_3) + x_2 \cdot (y_3 - y_1) + x_3 \cdot (y_1 - y_2)] + \frac{1}{2} [x_3 \cdot (y_4 - y_1) + x_4 \cdot (y_1 - y_3) + x_1 \cdot (y_4 - y_4)]$$

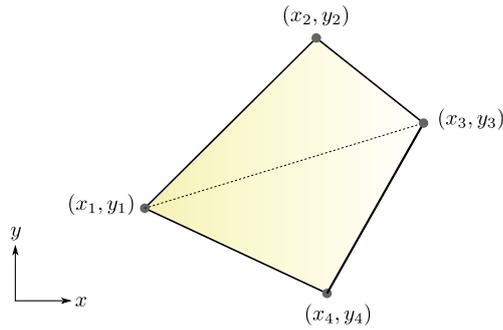


Figura A.4 – Cuadrángulo definido por cuatro puntos

A.4. Pertenencia de un punto a un triángulo

Dado un triángulo en el plano x - y definido por tres puntos $\mathbf{x}_1 = (x_1, y_1)$, $\mathbf{x}_2 = (x_2, y_2)$ y $\mathbf{x}_3 = (x_3, y_3)$ (figura A.3) y un punto arbitrario $\mathbf{x} = (x, y)$, queremos saber si el punto \mathbf{x} pertenece o no al triángulo. Para ello, calculamos las componentes z de los siguientes tres productos vectoriales:

$$\begin{aligned} z_1 &= (\mathbf{x}_2 - \mathbf{x}_1) \times \mathbf{x} \\ z_2 &= (\mathbf{x}_3 - \mathbf{x}_2) \times \mathbf{x} \\ z_3 &= (\mathbf{x}_1 - \mathbf{x}_3) \times \mathbf{x} \end{aligned}$$

Si los signos de estas tres cantidades z_1 , z_2 y z_3 son iguales, entonces el punto \mathbf{x} pertenece al triángulo. De lo contrario, si hay dos con igual signo y una con signo contrario, \mathbf{x} no pertenece al triángulo.

A.5. Pertenencia de un punto a un cuadrángulo

Podemos extender el test de pertenencia de un punto \mathbf{x} a un cuadrángulo definido por cuatro puntos, suponemos nuevamente que el cuadrángulo está formado por dos triángulos, el primero dado por los puntos $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ y el segundo por $(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_1)$. Luego, primero verificamos si el punto \mathbf{x} pertenece al primer triángulo calculando

$$\begin{aligned} z_1 &= (\mathbf{x}_2 - \mathbf{x}_1) \times \mathbf{x} \\ z_2 &= (\mathbf{x}_3 - \mathbf{x}_2) \times \mathbf{x} \\ z_3 &= (\mathbf{x}_1 - \mathbf{x}_3) \times \mathbf{x} \end{aligned}$$

Si los signos de estas tres cantidades z_1 , z_2 y z_3 son iguales, entonces el punto \mathbf{x} pertenece al primer triángulo, y por ende al cuadrángulo. Si no, calculamos las cantidades



$$z_4 = (\mathbf{x}_4 - \mathbf{x}_3) \times \mathbf{x}$$

$$z_5 = (\mathbf{x}_1 - \mathbf{x}_4) \times \mathbf{x}$$

$$z_6 = (\mathbf{x}_3 - \mathbf{x}_1) \times \mathbf{x}$$

y volvemos a verificar si sus signos coinciden. Si coinciden, entonces el punto \mathbf{x} pertenece al segundo triángulo, y por ende al cuadrángulo.

Identidades de Green

En la derivación del esquema basado en elementos finitos hemos utilizado la siguiente identidad de Green para transformar una ecuación diferencial de segundo orden en una formulación fuerte de primer orden en la ecuación (4.39). En la mayoría de los textos académicos—al menos en todos con lo que me encontré—los ejemplos tratan con situaciones en las que la función $\theta(\mathbf{x})$ (que corresponde al coeficiente de difusión o la conductividad térmica por ejemplo) como uniforme. En estos casos, los autores extraen fuera de la integral este factor y continúan aplicando el teorema de la divergencia tal como todos aprendimos en nuestros primeros años universitarios. Sin embargo, la reducción del orden es posible aún para coeficientes dependientes del espacio aplicando la segunda identidad de Green, que consiste en una extensión de la integración por partes a varias dimensiones y que a falta de poder referenciar una demostración que me convenza, propongo una para el caso bidimensional en este apéndice siguiendo la metodología de desarrollar los términos vectoriales en expresiones escalares y luego volver a escribirlos en forma vectorial.

Teorema B.1 (Segunda Identidad de Green) *En un dominio conexo $U \in \mathbb{R}^2$, sean $\theta(x, y) : \mathbb{R}^2 \mapsto \mathbb{R}$ y $\psi(x, y) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ funciones continuas, entonces*

$$\int_U \theta(x, y) \cdot \operatorname{div} [\psi(x, y)] \, dx \, dy = - \int_U \operatorname{grad}[\theta(x, y)] \cdot \psi(x, y) \, dx \, dy + \int_{\partial U} \theta(x, y) \cdot [\psi(x, y) \cdot \hat{\mathbf{n}}] \, dS$$

DEMOSTRACIÓN Podemos escribir la integral que queremos calcular como

$$\int_U \theta(x, y) \cdot \operatorname{div} [\psi(x, y)] \, dx \, dy = \int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_x}{\partial x} \, dx \, dy + \int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_y}{\partial y} \, dx \, dy$$

donde las funciones $c(y)$ y $d(y)$ junto con las constantes a y b definen el dominio U en \mathbb{R}^2 (figura B.1). Prestando atención ahora a la integral sobre x del primer término del miembro derecho, podemos integrar por partes

$$\begin{aligned} \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_x}{\partial x} \, dx &= - \int_{c(y)}^{d(y)} \psi_x(x, y) \cdot \frac{\partial \theta}{\partial x} \, dx \\ &\quad + [\psi_x(d(y), y) \cdot \theta(d(y), y)] - [\psi_x(c(y), y) \cdot \theta(c(y), y)] \end{aligned}$$

Al volver a integrar esta expresión con respecto a y entre a y b obtenemos

$$\begin{aligned} \int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_x}{\partial x} \, dx \, dy &= - \int_a^b \int_{c(y)}^{d(y)} \psi_x(x, y) \cdot \frac{\partial \theta}{\partial x} \, dx \, dy \\ &\quad + \int_a^b [\psi_x(d(y), y) \cdot \theta(d(y), y)] \, dy - \int_a^b [\psi_x(c(y), y) \cdot \theta(c(y), y)] \, dy \end{aligned}$$

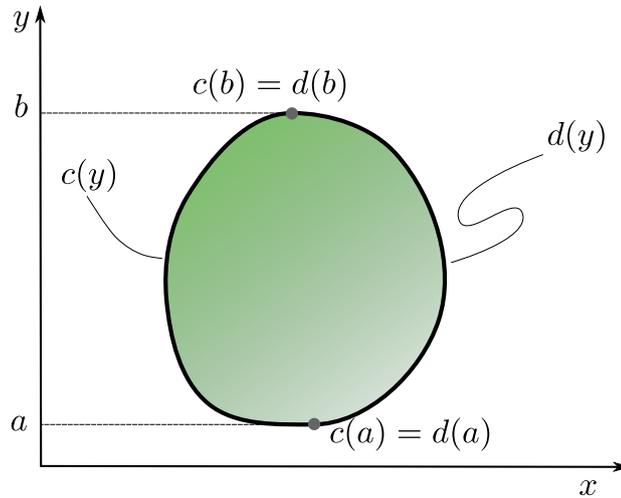


Figura B.1 – Dominio $U \in \mathbb{R}^2$ definido por las curvas $c(y)$ y $d(y)$ y por los puntos a y b .

Podemos ver que las últimas dos integrales son las integrales de línea de la expresión $\psi_x(\mathbf{x}) \cdot \theta(\mathbf{x})$ a lo largo de las líneas $d(y)$ y $c(y)$ respectivamente, cuya resta da por definición la integral sobre la frontera ∂U del dominio U (figura B.1). Luego

$$\int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_x}{\partial x} dx dy = - \int_a^b \int_{c(y)}^{d(y)} \psi_x(x, y) \cdot \frac{\partial \theta}{\partial x} dx dy + \int_{\partial U} \psi_x(x, y) \cdot \theta(x, y) \cdot n_x dS$$

Análogamente para las expresiones de ψ_y , tenemos

$$\int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_y}{\partial y} dx dy = - \int_a^b \int_{c(y)}^{d(y)} \psi_y(x, y) \cdot \frac{\partial \theta}{\partial y} dx dy + \int_{\partial U} \psi_y(x, y) \cdot \theta(x, y) \cdot n_y dS$$

Recordando que la integral que queremos calcular era igual a

$$\int_U \theta(x, y) \cdot \text{div} [\psi(x, y)] dx dy = \int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_x}{\partial x} dx dy + \int_a^b \int_{c(y)}^{d(y)} \theta(x, y) \cdot \frac{\partial \psi_y}{\partial y} dx dy$$

entonces

$$\begin{aligned} \int_U \theta(x, y) \cdot \text{div} [\psi(x, y)] dx dy &= - \int_a^b \int_{c(y)}^{d(y)} \left[\psi_x(x, y) \cdot \frac{\partial \theta}{\partial x} + \psi_y(x, y) \cdot \frac{\partial \theta}{\partial y} \right] dx dy \\ &\quad + \int_{\partial U} [\psi_x(x, y) \cdot \theta(x, y) \cdot n_x + \psi_y(x, y) \cdot \theta(x, y) \cdot n_y] dS \end{aligned}$$

de donde sigue la tesis del teorema. □

Cuadratura de Gauss

La integración numérica de una cierta función $f(x)$ en el intervalo $[-1 : 1]$ consiste en escribir una aproximación

$$\int_{-1}^{+1} f(x) dx \approx \sum_{i=1}^N \alpha_i \cdot f(x_i)$$

Dependiendo de si es posible elegir o no tanto la cantidad N como los puntos individuales x_i de evaluación de la función $f(x)$, existen varios métodos que proveen aproximaciones de diferentes órdenes. Por ejemplo, si aproximamos el integrando mediante monomios de orden N , los coeficientes α_i son la solución de un sistema lineal de ecuaciones (i.e. hay que invertir una matriz) pero las integrales de los términos individuales son triviales. Si interpolamos $f(x)$ con polinomios de Lagrange, podemos calcular fácilmente los coeficientes pero las expresiones para las integrales presentan cierta complejidad. En particular, si los puntos x_i son elegidos en forma equiespaciada en el intervalo, entonces el método de cuadratura tiene nombre propio y es el conocido algoritmo de Newton-Cotes.

La cuadratura de Gauss que utiliza el código milonga para construir las matrices elementales no permite elegir libremente los puntos x_i de interpolación sino que éstos son provistos por el método, a cambio de proveer una estimación de la integral que es exacta para polinomios de grado $2n - 1$ o menor. A continuación mostramos una de las tantas deducciones posibles.

C.1. Una dimensión

Consideremos la función $F(x)$ definida como

$$F(x) = \left[\sum_{i=1}^N f(x_i) \cdot \ell_i(x) \right] + P(x) \sum_{i=0}^{n-1} \beta_i \cdot x^i$$

donde $P(x)$ es el polinomio de orden N

$$P(x) = (x - x_1) \cdot (x - x_2) \dots (x - x_n)$$

y $\ell_i(x)$ es el polinomio de Lagrange de orden $N - 1$ correspondiente al punto x_i

$$\ell_i(x) = \frac{(x - x_1) \cdot (x - x_2) \dots (x - x_{i-1}) \cdot (x - x_{i+1}) \dots (x - x_N)}{(x_i - x_1) \cdot (x_i - x_2) \dots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \dots (x_i - x_N)}$$

La función $F(x)$ es una aproximación del integrando $f(x)$. En efecto, si evaluamos $F(x)$ en alguno de los puntos de integración x_i (que debemos calcular), obtenemos



$$F(x_i) = \left[\sum_{i=1}^N f(x_i) \cdot \ell_i(x_i) \right] + P(x_i) \sum_{i=0}^{n-1} \beta_i \cdot x^i \approx f(x_i) = f(x_i)$$

dado que $\ell_i(x_j) = \delta_{ij}$ y $P(x_i) = 0$. El método de Gauss propone estimar la integral de $f(x)$ como la integral de $F(x)$

$$\begin{aligned} \int_{-1}^{+1} f(x) dx \approx \int_{-1}^{+1} F(x) dx &= \int_{-1}^{+1} \left[\sum_{i=1}^N f(x_i) \cdot \ell_i(x) dx \right] + \int_{-1}^{+1} \left[P(x) \sum_{i=0}^{n-1} \beta_i \cdot x^i \right] \\ &= \sum_{i=1}^N \left[\int_{-1}^{+1} \ell_i(x) dx \right] \cdot f(x_i) + \sum_{i=0}^{n-1} \beta_i \cdot \left[\int_{-1}^{+1} P(x) \cdot x^i dx \right] \end{aligned}$$

eligiendo los N puntos de integración x_i de forma tal que se anule el último término, es decir

$$\int_{-1}^{+1} P(x) \cdot x^k dx = 0$$

para todo $k = 1, \dots, N$, donde debemos recordar que $P(x)$ depende de la elección de los puntos x_i .

Fijado N y calculados los puntos de integración x_i —llamados puntos de Gauss—los pesos α_i resultan

$$\alpha_i = \left[\int_{-1}^{+1} \ell_i(x) dx \right]$$

Luego la cuadratura de la función $f(x)$ con N puntos de Gauss es

$$\int_{-1}^{+1} f(x) dx \approx \sum_{i=1}^N \alpha_i \cdot f(x_i) \tag{C.1}$$

Para $N = 1$, la solución es

$$x_1 = 0 \qquad \alpha_1 = 2$$

Para $N = 2$,

$$\begin{aligned} x_1 &= -\sqrt{1/3} & \alpha_1 &= 1 \\ x_2 &= +\sqrt{1/3} & \alpha_2 &= 1 \end{aligned}$$

Para $N = 3$,

$$\begin{aligned} x_1 &= -\sqrt{3/5} & \alpha_1 &= 5/9 \\ x_2 &= 0 & \alpha_2 &= 8/9 \\ x_3 &= +\sqrt{3/5} & \alpha_3 &= 5/9 \end{aligned}$$



C.2. Dos dimensiones

Si queremos calcular la integral doble de una función $f(x, y)$ en el intervalo $[-1 : 1] \times [-1 : 1]$

$$\int_{-1}^{+1} \int_{-1}^{+1} f(x, y) dx dy$$

entonces primero aproximamos la integral con respecto a x usando la ecuación (C.1)

$$\int_{-1}^{+1} \left[\sum_{i=1}^N \alpha_i \cdot f(x_i, y) \right] dy$$

Podemos aproximar ahora la integral con respecto a y usando cuadratura de Gauss, utilizando ahora el subíndice j para indicar los puntos de integración sobre la variable y

$$\sum_{j=1}^N \left\{ \alpha_j \cdot \left[\sum_{i=1}^N \alpha_i \cdot f(x_i, y_j) \right] \right\}$$

con lo que la generalización a dos dimensiones queda

$$\int_{-1}^{+1} \int_{-1}^{+1} f(x, y) dx dy \approx \sum_{j=1}^N \sum_{i=1}^N (\alpha_i \alpha_j) \cdot f(x_i, y_j)$$

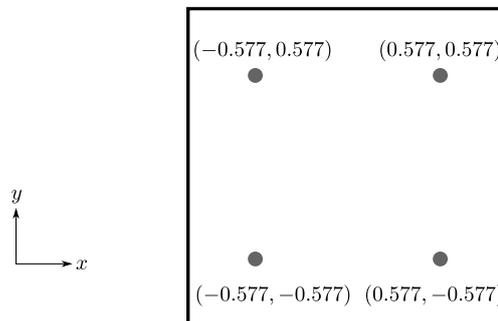


Figura C.1 – Puntos de Gauss en dos dimensiones para $N = 2$

Referencias

- [1] Klaus-Jürgen Bathe. *Finite Element Procedures*. Prentice Hall, 2006.
- [2] Eric Raymond. *The Art of Unix Programming*. Ed. by Addison-Wesley. 2003.
- [3] Eric Raymond. "The cathedral and the bazaar". In: *Knowledge, Technology & Policy* 12.3 (1999), pp. 23–49.
- [4] J. Gay and R. Stallman. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Second Edition. GNU Press, 2010. ISBN: 978-0-9831592-0-9.
- [5] R. Stallman. *What is Free Software?* URL: <http://www.gnu.org/philosophy/free-sw.html>.
- [6] Theler G., Clause A. y Bonetto F. J. «Optimización de parámetros en reactores de potencia: base de diseño del código neutrónico milonga». En: *Actas de la XXXVII Reunión Anual de la Asociación Argentina de Tecnología Nuclear*. 2010.
- [7] G. Theler. *Milonga: a free nuclear reactor core analysis code*. 2011.
- [8] S. Balay et al. *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.4. Argonne National Laboratory, 2013.
- [9] S. Balay et al. "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries". In: *Modern Software Tools in Scientific Computing*. Ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen. Birkhäuser Press, 1997, pp. 163–202.
- [10] Vicente Hernandez, Jose E. Roman, and Vicente Vidal. "SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems". In: *ACM Transactions on Mathematical Software* 31.3 (Sept. 2005), pp. 351–362.
- [11] A. F. Henry. *Nuclear reactor analysis*. Cambridge, MIT, 1975.
- [12] Samuel Glasstone and George I. Bell. *Nuclear reactor theory*. Krieger Publishing Company, 1970.
- [13] J. J. Duderstadt and L. J. Hamilton. *Nuclear reactor analysis*. Wiley, New York, 1976.
- [14] Weston M. Stacey. *Nuclear Reactor Physics*. Second. Wiley-VCH, 2007.
- [15] M.B. Chadwick et al. "ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data". In: *Nuclear Data Sheets* 112.12 (2011). Special Issue on ENDF/B-VII.1 Library, pp. 2887–2996.
- [16] Jaakko Leppänen. *Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code*. 2012.
- [17] Dinesh Kaushik et al. "Enabling high-fidelity neutron transport simulations on petascale architectures". In: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. 2009.
- [18] K. S. Smith. "Assembly homogenization techniques for light water reactor analysis". In: *Progress in Nuclear Energy* 17.3 (1986), pp. 303–335.



- [19] K. H. Beckurts and K. Wirtz. *Neutron Physics*. Springer-Verlag, 1964.
- [20] John R. Lamarsh. *Introduction to nuclear reactor theory*. Addison-Wesley, 1966.
- [21] Paul Reuss. *Neutron Physics*. EDP Sciences, 2008.
- [22] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. John Wiley and Sons, 1984.
- [23] M. Rivero. «Optimización computacional de la solución numérica de la ecuación de difusión de neutrones». Tesis de la Maestría en Simulación Numérica y Control. Univesidad de Buenos Aires, 2011.
- [24] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: its basis and fundamentals*. 6th. Vol. 1. Elsevier, 2005.
- [25] C. A. Brebbia and A. J. Ferrante, eds. *The Finite Element Technique*. Editora da urgs, Universidade Federal do Rio Grande do Sul, 1975.
- [26] C. Geuzaine and J. F. Remacle. “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.
- [27] Bruno Ruyú. *Instrucciones para escribir un texto*. 2006. URL: <http://www.talador.com.ar/jeremy/jaura/incomprensidos/instrucciones.pdf>.
- [28] Martín Silva. «Chuí: una denominación universal para las letras griegas». Instituto Bal-seiro. 2005.
- [29] Raúl Barrachina. *Uso de leyes integrales en la física*. Monografía de la Carrera de Maestría en Filosofía e Historia de la Ciencia. Facultad de Humanidades, Universidad del Comahue, 2001.
- [30] Raúl Barrachina. «De la luz corpuscular a la materia ondulatoria: estudio comparativo de dos campos controversiales». Maestría en Filosofía e Historia de la Ciencia. Facultad de Humanidades, Universidad del Comahue, 2003.
- [31] D. E. Knuth. *Selected papers on Computer Languages*. Center for Studies of Language Information, 2003.
- [32] J. N. Buxton, ed. *Simulation Programming Languages*. North Holland, 1968.
- [33] G. Theler. “Unstructured Grids and the Multigroup Neutron Diffusion Equation”. In: *Science and Technology of Nuclear Installations* 2013.641863 (2013).
- [34] Theler G. “Geometric Optimization of Nuclear Reactor Cores”. In: *Mecánica Computacional* XXXII.32 (2013), pp. 2689–2703.
- [35] Theler G. «Variaciones Numéricas Sobre un Problema Clásico de Física de Reactores». En: *Actas de la XL Reunión Anual de la Asociación Argentina de Tecnología Nuclear*. 2013.
- [36] J.A. Nelder and R. Mead. “A simplex method for function minimization”. In: *Computer Journal* 7 (1965), pp. 308–313.
- [37] M. Galassi et. al. *GNU Scientific Library Reference Manual*. 3rd. ISBN: 0954612078. URL: <http://www.gnu.org/software/gsl/>.
- [38] Computational Benchmark Problem Comitee for the Mathematics and Computation Division of the American Nuclear Society. *Argonne Code Center: Benchmark Problem Book*. Tech. rep. ANL-7416 Supplement 2. Argonne National Laboratory, June 1977.
- [39] Theler G., Clause A. y Bonetto F. J. «Solution of the 2D IAEA PWR Benchmark with the neutronic code milonga». En: XXXVIII Reunión Anual de la Asociación Argentina de Tecnología Nuclear. Buenos Aires, 2011.

15 de diciembre de 2013