MÉTODOS ITERATIVOS EFICIENTES PARA PROBLEMAS DE CONVECCIÓN-DIFUSIÓN TRANSITORIOS

María Luisa Sandoval Solís

Tesis Doctoral
Director: Antonio Rodríguez Ferran
Barcelona, Mayo 2006

Departament de Matemàtica Aplicada III
Programa de Doctorat de Matemàtica Aplicada

A la memoria de mis padres,

a mis hermanos y hermanas.

A mi compañera de viaje:

Leticia.

Resumen

Diversos procesos naturales e industriales de interés medioambiental se modelan a través de la ecuación de convección-difusión-reacción transitoria. Dos aplicaciones tecnológicas que han motivado esta tesis son el funcionamiento de filtros de carbón activo y la dispersión de contaminantes en la atmósfera. Para que la modelización numérica de estos problemas sea eficaz es indispensable contar con un solver lineal eficiente para resolver los sistemas de ecuaciones obtenidos al discretizar la ecuación en derivadas parciales, mediante elementos finitos.

Por ello, *el objetivo de esta tesis es resolver de forma eficiente los grandes sistemas de ecuaciones, simétricos definidos positivos (SDP), tipo sparse asociados a los problemas de convección-difusión transitorios*. Con este fin se estudian los precondicionadores tanto explícitos como implícitos, así como los métodos de descomposición de dominios (DD).

La tesis se estructura en tres partes. En la primera se elabora un análisis computacional detallado del comportamiento de dos familias de factorizaciones incompletas de Cholesky (FIC): de memoria prescrita y de umbral. Estas técnicas se utilizan para precondicionar el método iterativo de gradientes conjugados (PCG). En la segunda parte se construye una inversa aproximada *sparse* simétrica (SSPAI) basada en la minimización en la norma de Frobenius. El precondicionador explícito se diseña para resolver en paralelo grandes sistemas de ecuaciones *sparse*, SDP, tridiagonales por bloques con múltiples lados derechos.

Finalmente, se desarrolla el método multiplicativo de Schwarz (MSM) en dominios activos, es decir, DD solapados con la innovación de activar y desactivar dominios. Se estudia el comportamiento de esta estrategia al resolver los subproblemas mediante: (1) el método directo de Cholesky y (2) PCG + FIC de umbral.

De los resultados numéricos presentados se concluye que es preferible utilizar el método directo de Cholesky para sistemas con menos de $30\,000$ variables. Para sistemas mayores y hasta $80\,000$ incógnitas se sugiere emplear una FIC de umbral. Y para sistemas aún más grandes, el MSM en dominios activos + PCG + FIC de umbral propuesto es el más eficiente usando un solo procesador. Por su parte, la SSPAI presentada podría superar a las FIC de umbral si se trabaja en paralelo.

Agradecimientos

Aprovecho este espacio para reconocer el apoyo de las personas e instituciones que han contribuido en la realización de este proyecto.

Agradezco a Antonio Rodríguez la dirección de esta tesis, su experiencia ha enriquecido este trabajo.

También agradezco el apoyo recibido de cada miembro del grupo de investigación LaCàN y del Departament de Matemàtica Aplicada III, así como el soporte logístico brindado por estas entidades y la Facultat de Matemàtiques i Estadística.

Una mención especial a Vanesa y Sonia por introducirme en el manejo de los programas de flujo y concentración; a Xevi por ayudarme a construir las mallas y a Rosa por facilitarme el código de dispersión de contaminantes.

Deseo agradecer a Gustavo Montero esas tardes de discusión sobre las inversas aproximadas, fueron realmente estimulantes. También agradezco a cada miembro de las divisiones GANA y DDA del IUSIANI de la ULPGC su hospitalidad. A Elizabeth y Eduardo su valiosa ayuda.

A Lety, a mi familia y a mis amigos quiero expresarles mi más sincera gratitud, su compañía y entusiasmo han sido mi fortaleza.

Finalmente agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) de México el financiamiento otorgado para realizar los estudios doctorales (número de registro: 61078). A la Universitat Politècnica de Catalunya el apoyo económico recibido para finalizar la tesis doctoral. Por último, a la Fundación Kovalévskaia y a la Sociedad Matemática Mexicana su reconocimiento.

Barcelona — María Luisa Sandoval Solís

Mayo, 2006

Índice general

Índice de tablas

Índice de figuras

Lista de símbolos

El número indica la página en que el símbolo aparece por primera vez.

Alfabeto latino

$\mathscr{S}_j$	subespacio de matrices simétricas con únicos no nulos en columna j 82
$\mathscr{S}_j^L$	subespacio de $\mathscr{S}_j$ de matrices triangulares inferiores 82
$\mathscr{S}_j^U$	subespacio de $\mathscr{S}_j$ de matrices triangulares superiores 82
$\mathbf{S}_l$	elemento de la base que genera a $\mathscr{S}_j$ 84
$\mathbf{A}\widetilde{\mathbf{S}}_l$	elemento de la base ortogonal que genera a $\mathbf{A}\mathscr{S}_j$ 85
SSPAI	inversa aproximada *sparse* simétrica 93
t	tiempo 8
t	máximo número de entradas significativas en triangular superior 91
T	tiempo final de análisis 8
tol_c	tolerancia para el criterio de la concentración 119
tol_r	tolerancia para el residuo 57
tol_x	tolerancia para las componentes de $\mathbf{x}$ 57
$\mathrm{tr}(\cdot)$	traza de una matriz 78
$\mathbf{U}$	matriz triangular superior 25
v	función de test 19
$\boldsymbol{v}(\boldsymbol{x})$	velocidad convectiva 8
$\mathbf{v}_k$	vector prefijado por simetría 89
$\mathcal{V}$	espacio de funciones de test 19
V_i	espacio de funciones de test 35
$\mathbf{V}_{\boldsymbol{d}}$	diagonal de deposición seca 17
$\mathbf{V}_{j,j}$	m. triang. superior fijada por simetría con no nulos en columna j 84
$\boldsymbol{x}$	posición en el dominio Ω 8

Alfabeto griego

α	parámetro de perturbación 45
$\boldsymbol{\alpha}$	matriz de reacción 9
Γ_i	frontera 115
$\partial\Omega$	frontera del dominio de estudio 7
$\partial\Omega_i$	frontera del subdominio Ω_i 33
δH	medida lineal de la región de solapamiento 35
Δt	incremento de tiempo 8

Otros símbolos

Siglas

Chol	método de Cholesky 122
CG	gradiente conjugados 2
DD	descomposición de dominios 4
EDP	ecuaciones en derivadas parciales 4
FI	factorización incompleta 3
FIC	factorización incompleta de Cholesky 3
HC	hidrocarburo 10
LS	mínimos cuadrados 2
ASM	método aditivo de Schwarz 34
DD-Chol	método multiplicativo de Schwarz en dominios activos con Chol 122
DD-FIC	método multiplicativo de Schwarz en dominios activos con FIC 122
MSM	método multiplicativo de Schwarz 34
PCG	gradiente conjugados precondicionado 5
RI	refinamiento iterativo 126
SDP	simétrico definido positivo 2
SPAI	inversa aproximada *sparse* 5
SSPAI	inversa aproximada *sparse* simétrica 5

Capítulo 1

Introducción

Diversos procesos naturales e industriales de interés medio ambiental son fenómenos de convección, difusión y reacción. Este es el caso, por ejemplo, del funcionamiento de los filtros de carbón activo empleados en la industria del automóvil y la dispersión de contaminantes en la atmósfera. La modelización numérica de estas aplicaciones tecnológicas deben considerar, además de los fenómenos físicos de convección, difusión y reacción, el carácter transitorio y la no linealidad del problema, el acoplamiento entre procesos, el carácter multiescala y la geometría tridimensional compleja. A estas características se añade la dificultad que presenta, por sí mismo, el tratamiento numérico de los términos convectivos.

El objetivo de la modelización numérica de problemas de convección-difusión-reacción transitorios es producir simulaciones numéricas que ayuden a predecir y analizar los procesos físicos de ensayo. Para ello se debe disponer de un acoplamiento adecuado de sofisticados algoritmos o técnicas que tomen en cuenta las características y dificultades enunciadas en el párrafo anterior. La modelización numérica de las aplicaciones tecnológicas que presentamos incluye las siguientes estrategias:

- Generar mallas 3D no estructuradas para elementos finitos que respeten las formas irregulares de los filtros de carbón y la topografía abrupta del terreno para el problema de dispersión de contaminantes.
- Evitar el carácter no lineal del problema global. Para los filtros se utiliza una estrategia de paso fraccionado para desacoplar la parte convectiva y difusiva, de la reacción. La no linealidad del término de reacción se trata a nivel local.

En el problema de dispersión se simplifican las reacciones fotoquímicas no lineales tomando en cuenta las escalas de tiempo, de esa manera se obtiene un sistema lineal de velocidades de reacción.

- Utilizar un esquema de integración temporal junto con una técnica de estabilización que sean adecuados con el fin de conseguir soluciones precisas y estables.

- *Se requiere un algoritmo eficiente para resolver los grandes sistemas lineales tipo sparse que se obtienen al discretizar los problemas 3D mediante elementos finitos. La presente tesis está enfocada a lograr este punto.*

Se conoce que el método de Galerkin en los problemas con convección dominante no funciona apropiadamente debido a las oscilaciones no físicas de su solución numérica. Por ello, en las últimas décadas se han propuesto diferentes técnicas para estabilizar el término convectivo: SUPG, GLS, SGS y LS (ver Donea y Huerta 2003). En particular, Huerta, Roig y Donea (2002) y Huerta y Donea (2002) muestran que la conjunción de Crank-Nicolson con la formulación de mínimos cuadrados (LS) funciona adecuadamente para la ecuación de convección-difusión transitoria. Esto se debe a que LS introduce más difusividad que SUPG, GLS y SGS. Otra ventaja de utilizar la formulación estabilizada de LS en la modelización numérica, y quizás la más importante, es que esta técnica produce un sistema simétrico definido positivo (SDP) en cada paso de tiempo.

Debido a la naturaleza 3D de las aplicaciones y al uso de mallas relativamente finas se hace imprescindible contar con una algoritmo eficiente y robusto para resolver los grandes sistemas de ecuaciones, que en nuestro caso son SDP, tipo *sparse*. La finalidad de esta tesis es encontrar una manera eficiente de resolver tales sistemas. De ese modo se podrá contar con simulaciones numérica eficientes que sean una herramienta predictiva y de análisis.

Existen dos formas para encontrar la solución numérica, empleando métodos directos como Cholesky, o métodos iterativos.

Las técnicas iterativas más utilizadas son los métodos basados en los subespacios de Krylov, tales como gradientes conjugados (CG), método de residuo mínimo generalizado (GMRES), método de cuasi-mínimo residuo (QMMR), entre otros;

ver por ejemplo, los libros de Saad (2003), Greenbaum (1997) y Barrett, Berry, Chan, Demmel, Donato, Dongarra, Eijkhout, Pozo, Romine y van der Vorst (1994). Observemos que para abreviar los nombres de los métodos numéricos conocidos, como CG o GMRES, se han usado sus acrónimos en inglés. En otro caso hemos utilizado siglas en español, por ejemplo SDP es el acrónimo de simétrico definido positivo. Este mecanismo se empleará a largo del texto.

La velocidad de convergencia de los métodos de Krylov dependen de las propiedades espectrales de la matriz $\mathbf{A}$. Una manera de acelerar su convergencia es multiplicando el sistema original $\mathbf{Ax} = \mathbf{b}$ por otra matriz $\mathbf{M}^{-1}$, de tal forma que mejore las propiedades espectrales de la matriz $\mathbf{M}^{-1}\mathbf{A}$, es decir, que el sistema $\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$ sea más fácil de resolver (ver Axelsson 1996). A la matriz $\mathbf{M}$ se le denomina precondicionador (implícito).

De hecho, los precondicionadores se clasifican en implícitos y explícitos. Estos últimos encuentran una matriz $\mathbf{M}^{-1}$ que se aproxime a la inversa de la matriz original $\mathbf{A}^{-1}$, tal que para hallar la solución del sistema solo se necesite una multiplicación por $\mathbf{M}^{-1}$. Ejemplos de ellos son las inversas aproximadas basados en la minimización en la norma de Frobenius, las inversas aproximadas factorizadas y aquellas que realizan primero una factorización incompleta seguida de la inversa aproximada de cada uno de sus factores incompletos, ver Benzi (2002). Como ejemplos de precondicionadores implícitos tenemos el diagonal, el SSOR y las factorizaciones incompletas (FI). En el caso de matrices SDP, los precondicionadores que han ganado mayor importancia son las factorizaciones incompletas de Cholesky (FIC), las cuales a su vez se distinguen en dos familias: de memoria prescrita y de umbral (ver Saad 2003).

Aunque la literatura señala la existencia de diversos tipos de precondicionadores, ya sea explícitos o implícitos, para cada problema es necesario buscar el precondicionador adecuado, lo que implica realizar un estudio exhaustivo de un mayor número de éstos o proponer mejoras de los mismos. De ese modo, se podrá garantizar que el precondicionador seleccionado sea el más eficiente. En particular, investigamos el comportamiento de las diferentes familias de FIC usando como referencia el método directo de Cholesky y el precondicionador diagonal.

Una característica a destacar de las inversas aproximadas basados en la minimi-

zación en la norma de Frobenius es que son altamente paralelizables. Esta clase de precondicionadores se construyen usando una sola matriz y su patrón de *sparsidad* puede ser prescrito de antemano, o en forma dinámica (ver Montero, González, Flórez, García y Suárez 2002, Grote y Huckle 1997, Gould y Scott 1998, Huckle 1998, Cosgrove, Díaz y Griewank 1992, Díaz y Macedo 1989). Es importante resaltar que en la actualidad no existe un algoritmo que genere un precondicionador explícito basado en la minimización en la norma de Frobenius usando un patrón de *sparsidad* dinámico y que tome en cuenta la simetría de una matriz. Uno de nuestros objetivos es cubrir tal vacío.

Por su parte, debido a la demanda de simulaciones 3D con mallas relativamente finas se ha recurrido en las últimas décadas a las técnicas en paralelo. Las que más auge han tenido son las técnicas de descomposición de dominios (DD). La idea básica de éstas se resume en dos palabras: "dividir y conquistar". En si, consisten en dividir el dominio asociado al problema global de ecuaciones en derivadas parciales (EDP) en subdominios finitos, de tal forma que el problema global (sistema total de ecuaciones) se pueda descomponer en tantos subproblemas (subsistemas) como subdominios existan, ver Quarteroni y Valli (1999) y Greenbaum (1997). Aunque la perspectiva moderna es utilizar éstas para construir precondicionadores que sean empleados por algún método de Krylov (Meurant 1999), nosotros manejamos su principio básico. Mostramos computacionalmente que es más eficiente resolver uno o dos subproblemas que el problema global en cada paso de tiempo. Esto es válido para aquellos en donde la solución solo varía en una región del dominio global conforme se avanza en el tiempo. Por ejemplo, los problemas de transporte de contaminantes. En particular, empleamos las simulaciones de los filtros de carbón activo en un proceso de carga para ilustrar nuestra propuesta: el método multiplicativo de Schwarz en dominios activos.

En general, con este trabajo se pretende entrar en contacto con las técnicas numéricas que están a la vanguardia en la solución numérica de grandes sistemas de ecuaciones, SDP, tipo *sparse*. La forma de llevarlo a cabo se describe en el siguiente apartado.

1.1. Objetivos

El objetivo principal de la tesis doctoral es resolver de forma eficiente los grandes sistemas de ecuaciones, simétricos definidos positivos, tipo sparse asociados a los problemas de convección-difusión transitorios. Esta clase de sistemas se obtienen al emplear el esquema de Crank-Nicolson junto con el método de mínimos cuadrados estándar.

La tesis se divide en tres partes con el fin de alcanzar nuestro objetivo principal:

1. Elaboración de un análisis computacional detallado del comportamiento de dos familias de factorizaciones incompletas de Cholesky (FIC) de: memoria prescrita y umbral. Estos precondicionadores implícitos se utilizan junto con el método de gradientes conjugados precondicionado (PCG).

2. Construcción de una inversa aproximada *sparse* simétrica (SSPAI) basada en la generalización de la SPAI de Montero et al. (2002). El precondicionador explícito se diseña para resolver numéricamente grandes sistemas de ecuaciones tipo *sparse*, SDP, tridiagonales por bloques con múltiples lados derechos.

3. Desarrollo e implementación del método multiplicativo de Schwarz en dominios activos (descomposición de dominios solapados con la innovación de activar y desactivar dominios) para resolver los problemas 3D de convección-difusión transitorios. Estudiar el comportamiento de este proceso iterativo cuando la solución a los subproblemas se realiza mediante: (1) el método directo de Cholesky y (2) PCG con la familia de FIC de umbral.

Finalmente, es importante enfatizar que con estas tres partes (u objetivos) de la tesis se ha intentado abarcar el mayor número de técnicas para proponer una forma eficiente de resolver los problemas de convección-difusión transitorios. La eficiencia computacional se determina principalmente en función de los requerimientos de memoria y los tiempos de CPU, así como del tamaño de los sistemas lineales entre otros aspectos.

1.2. Estructura de la tesis

El cuerpo de la tesis doctoral se estructura en siete capítulos. El capítulo 1 incluye la introducción al tema de estudio y los objetivos de la misma. En el capítulo 2 se exponen dos aplicaciones tecnológicas que han motivado la investigación: el funcionamiento de filtros de carbón activo y la dispersión de contaminantes en la atmósfera. Además, se presentan los sistemas de ecuaciones que se analizan a lo largo del texto. Previamente se introduce la formulación variacional de los problemas de convección-difusión transitorios, la discretización en el tiempo y el método de mínimos cuadrados estándar para estabilizar el término convectivo.

El capítulo 3 contiene el estado del arte de los métodos directos e iterativos para encontrar la solución numérica de grandes sistemas lineales. La intensión principal del mismo es ubicar, en un contexto general, cada una de las técnicas que se exploran.

La primera y segunda parte de la tesis se desarrollan en los capítulos 4 y 5, respectivamente. En el primero se analizan las FIC tanto en los filtros de carbón activo como en el problema de dispersión de contaminantes en la atmósfera. En el capítulo 5 se proponen diversas estrategias para construir un precondicionador explícito simétrico basado en la minimización en la norma de Frobenius hasta obtener el algoritmo de la inversa aproximada *sparse* simétrica (SSPAI).

En el capítulo 6 se exponen las ideas y los criterios para desarrollar el método multiplicativo de Schwarz en dominios activos. Su eficiencia se verifica a través de los filtros de carbón activo.

Por último, en el capítulo 7 se presenta la conclusión final sobre la investigación indicando la forma más eficiente de resolver los problemas de convección-difusión transitorios cuando el campo de velocidades es constante. Además, se plantean las futuras líneas de investigación.

Capítulo 2

Elementos finitos para problemas de convección-difusión transitorios. Aplicaciones tecnológicas

En este capítulo se presentan los problemas de convección-difusión transitorios que se estudian en la tesis. Particularmente, en el apartado 2.2 se exponen dos aplicaciones tecnológicas. Una de ellas está asociada a la industria del automóvil y consiste en simular el funcionamiento de los filtros de carbón activo. La otra modeliza la dispersión de contaminantes en la atmósfera. Planteadas las aplicaciones, se desarrolla en los apartados 2.3, 2.4 y 2.5 la formulación de elementos finitos, describiendo las técnicas que se usan para la integración temporal y la estabilización del término convectivo. También, en el apartado 2.5 se indica el tipo de sistemas de ecuaciones que se analizan en los capítulos 4, 5 y 6.

2.1. Problema de convección-difusión-reacción transitorio

En general, si consideramos el dominio $\Omega \subset \mathbb{R}^{n_d}$ con $n_d = 2$ o 3 y su frontera $\partial\Omega$, los problemas de convección-difusión-reacción transitorios pueden ser mode-

lados por

$$\begin{cases} \dfrac{\partial c}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla} c - \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla} c) + \sigma(c)c = f & \text{en } \Omega\times(0,T], \\ c = c_{\text{entrada}} & \text{sobre } \Gamma_E\times(0,T], \\ \boldsymbol{\nabla} c\cdot\boldsymbol{n} = 0 & \text{sobre } \partial\Omega\backslash\Gamma_E\times(0,T], \\ c(\boldsymbol{x},0) = c^0 & \text{en } \Omega, \end{cases} \tag{2.1}$$

donde $c(\boldsymbol{x},t)$ es la concentración del contaminante en el punto $\boldsymbol{x}$ e instante t, $\boldsymbol{v}(\boldsymbol{x})$ la velocidad convectiva (o advectiva), $\nu > 0$ es el coeficiente de difusividad, $\sigma(c)$ el coeficiente de reacción, $f(\boldsymbol{x},t)$ el término fuente volumétrico, $\boldsymbol{\nabla}$ el habitual operador nabla y T el tiempo final de análisis. La ecuación en derivadas parciales escalar $(2.1)_1$ se complementa con las condiciones de frontera Dirichlet y Neumann y la condición inicial, relaciones $(2.1)_2$, $(2.1)_3$ y $(2.1)_4$, respectivamente.

En la ecuación transitoria $(2.1)_1$, el término $\frac{\partial c}{\partial t}$ modela la variación de la concentración con respecto al tiempo; $\boldsymbol{v}\cdot\boldsymbol{\nabla} c$ la convección debida al movimiento del fluido ambiental, $\boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla} c)$ la difusión (dispersión de mayor a menor concentración de moléculas de contaminante), $\sigma(c)c$ la reacción (no lineal) y f la fuente externa. Además, notemos que el campo de velocidades $\boldsymbol{v}$ es constante y no uniforme.

En los problemas de convección-difusión-reacción transitorios es natural aplicar, durante la integración temporal, una separación de acuerdo a los diversos procesos físicos y químicos que están involucrados en el modelo. De hecho, es usual realizarlo en los problemas de contaminación del aire, ver Zlatev (1995). Particularmente, para la aplicación que se presenta en el apartado 2.2.1 empleamos una *separación de operadores o método de paso fraccionado*, con el fin de desacoplar la parte convectiva y difusiva, de la reacción en cada paso de tiempo (ver Quarteroni y Valli 1999, Donea y Huerta 2003). Para ello, reescribamos la ecuación $(2.1)_1$ como

$$\frac{\partial c}{\partial t} + \mathcal{L}_1 c + \mathcal{L}_2 c = 0 \tag{2.2}$$

donde $\mathcal{L}_1 c := \boldsymbol{v}\cdot\boldsymbol{\nabla} c - \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla} c)$ es el operador (lineal) de convección-difusión y $\mathcal{L}_2 c := \sigma(c)c - f$ es el operador de reacción (no lineal). Consideramos que tanto la difusividad ν como la velocidad convectiva $\boldsymbol{v}(\boldsymbol{x})$ son constantes. De esta manera, durante la integración temporal numérica, cada paso de tiempo Δt se puede subdividir en dos fases: una de convección-difusión, asociada a $\mathcal{L}_1$, y otra de reacción,

representada por $\mathcal{L}_2$. Como este último operador es no diferencial, su no linealidad se puede manejar eficientemente nodo a nodo. Por lo tanto, en nuestro estudio nos centraremos en la fase de convección-difusión, la cual conduce, mediante la discretización espacial de elementos finitos, a grandes sistemas de ecuaciones lineales.

Notemos que este tipo de separación es crucial para las aplicaciones industriales 3D: resolver el problema (2.1) sin separar los operadores significaría resolver miles de grandes sistemas de ecuaciones no lineales.

Por otro lado, la forma vectorial de la ecuación de convección-difusión-reacción transitoria se escribe como

$$\frac{\partial \mathbf{c}}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} \mathbf{c} - \nabla \cdot (\mathbf{K} \boldsymbol{\nabla} \mathbf{c}) - \boldsymbol{\alpha} \mathbf{c} = \mathbf{f} \qquad \text{en } \Omega \times (0, T], \tag{2.3}$$

donde $\mathbf{c} = [c_1(\mathbf{x}, t), c_2(\mathbf{x}, t), ..., c_q(\mathbf{x}, t)]^T$ es el vector de concentraciones para las q especies contaminantes en el punto $\mathbf{x}$ e instante t; $\mathbf{K}$ el tensor de difusividad, $\boldsymbol{\alpha}$ la matriz de reacción y $\mathbf{f}$ el vector fuente. La ecuación en derivadas parciales (EDP) vectorial (2.3) se complementa con una condición inicial y las condiciones de frontera (Dirichlet, Neumann o Robin) determinadas por el problema de estudio, ver apartado 2.2.2.

Los problemas de dispersión de contaminantes atmosféricos pueden ser modelados mediante la ecuación vectorial (2.3). En nuestro caso no es necesario emplear el método de paso fraccionado, puesto que se realizan diferentes simplificaciones en las reacciones químicas, con el fin de obtener una EDP vectorial lineal (ver apartado 2.2.2).

En esta tesis se estudian *los problemas de convección-difusión transitorios con convección dominante* (los efectos convectivos dominan a los difusivos). En el siguiente apartado se presentan, como motivación, dos aplicaciones tecnológicas a este tipo de problemas.

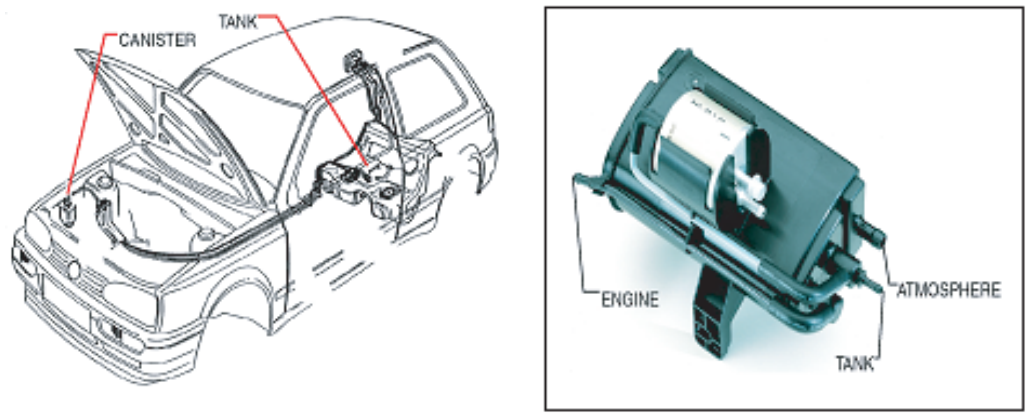


Figura 2.1: Localización del filtro de carbón activo.

2.2. Motivación: dos aplicaciones tecnológicas

2.2.1. Filtros de carbón activo

Una importante aplicación tecnológica dentro de la industria del automóvil es la simulación del comportamiento de los filtros de carbón activo, conocidos también como *canisters*. Dichos filtros contribuyen a reducir la emisión de hidrocarburos (HC) que procede del depósito de combustible y sale a la atmósfera, ver figura 2.1. Cuando el automóvil está parado el combustible del depósito se evapora. Para evitar la salida de HC a la atmósfera el *canister* se coloca entre la abertura del depósito y la conexión con el exterior como se indica en la figura 2.1. De esa forma, el carbón activo de los filtros adsorbe el HC que procede del depósito de combustible en un proceso denominado *carga*. En este proceso, las moléculas de HC en fase gaseosa pasan a estar en fase sólida y se depositan en la superficie de las partículas de carbón activo (ver Pérez-Foguet y Huerta 2006).

Por su parte, al proceso de limpieza del carbón activo del filtro se le denomina *purga* o *descarga*. En tal proceso de desorción, las moléculas de HC en fase sólida pasan a fase gaseosa a través de la circulación de aire limpio. El aire entra del exterior, pasa por el *canister* arrastrando el HC liberado y lo transporta hacia el motor. Finalmente es quemado en la combustión, ver figura 6.4.

La normativa medioambiental, cada vez más estricta, exige que la fabricación de *canisters* cumpla con determinados estándares de calidad. Entre ellos destaca la

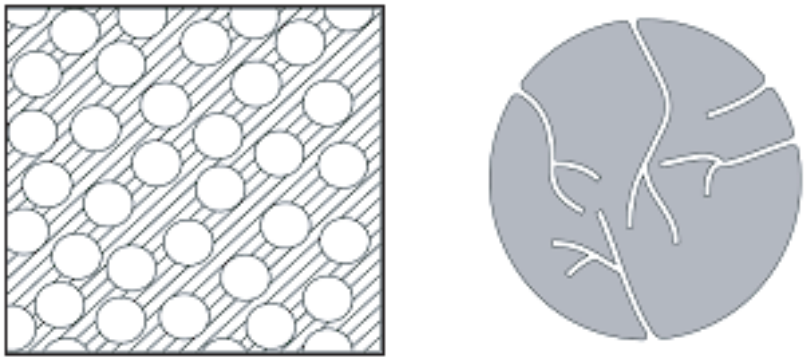


Figura 2.2: Porosidades inter e intraparticular, izquierda y derecha, respectivamente.

prueba de *capacidad de trabajo*, la cual mide la cantidad de butano (un hidrocarburo ligero) que es capaz de adsorber un filtro en una determinada secuencia de ciclos de carga-descarga. La prueba debe efectuarse en cada uno de los prototipos que se construyen a lo largo de la fase de diseño del filtro. Como la realización de la prueba es muy costosa, se plantea la posibilidad de complementar los ensayos experimentales con simulaciones numéricas. Por ello, se debe contar con un modelo adecuado, como el que se describe en los siguientes párrafos, capaz de reproducir los principales procesos físicos del ensayo.

La adsorción y desorción de hidrocarburo en los filtros están gobernadas por dos procesos con diferentes escalas: el primero, debido al transporte de hidrocarburos a través del *canister*, se realiza a macroescala, con longitudes del orden de decímetros. El segundo proceso, causado por el efecto de difusión de los hidrocarburos dentro de cada partícula, se efectúa en una microescala, con longitudes del orden de milímetros, 100 veces menor que la macroescala. El modelo planteado incorpora el segundo proceso (microescala), de tal forma que sea reflejado en la macroescala. Para ello, se propone un modelo de dos porosidades: la porosidad interparticular (entre partículas) y la porosidad intraparticular (en el interior de una partícula), ver figura 2.2. Bajo este esquema, los hidrocarburos pueden encontrarse tanto en fase gaseosa: en el fluido interparticular, o en los poros intraparticulares, como en fase sólida: adsorbidos en el carbón activo, ver Pérez-Foguet y Huerta (2006).

El transporte de los hidrocarburos a través del filtro en una macroescala, está gobernado por la ecuación de convección-difusión-reacción transitoria $(2.1)_1$. En este caso no se consideran fuentes externas y el término de reacción incorpora el

proceso a microescala de la siguiente forma

$$\frac{\partial c}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} c - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} c) = -\frac{1}{\varepsilon_e} \frac{\partial m}{\partial t} \tag{2.4}$$

con

$$m = \rho_s (1 - \varepsilon_e)(1 - \varepsilon_p) q_p + (1 - \varepsilon_e) \varepsilon_p c_p,$$

donde m es la masa de hidrocarburo por unidad de volumen, c la concentración de hidrocarburo en el fluido interparticular, q_p la masa de hidrocarburo adsorbido por unidad de masa de carbón, c_p la concentración de hidrocarburo en los poros intraparticulares, y $\boldsymbol{v}$ el vector de velocidad interparticular. En cuanto a los parámetros materiales, ν es la difusión, ε_e la porosidad interparticular, ε_p la porosidad intraparticular y ρ_s la densidad sólida del carbón. El modelo a microescala combina la difusión del soluto en fase gaseosa $(c_p = q^{1/m}/A^{1/m})$ con la difusión superficial del soluto en fase sólida $(q = A c_p^m)$ mediante una EDP con sus respectivas condiciones de frontera. En Pérez-Foguet y Huerta (2006) se detalla el modelo completo.

Por su parte, las ecuaciones generales que gobiernan el proceso de flujo y transporte están acopladas, sin embargo en una modelización se pueden desacoplar cuando la distribución de la variable a transportar no afectan a los parámetros relevantes de flujo. En nuestro caso, se supone válida la hipótesis de que la densidad del fluido es poco variable (fluido incompresible o casi incompresible). Esto conlleva a poder separar el proceso de flujo del transporte. Así, el campo de velocidades interparticular $\boldsymbol{v}$ se calcula mediante un modelo de flujo en medio poroso combinado con un flujo potencial, ver Rodríguez-Ferran, Sarrate y Huerta (2004).

Los procesos de carga y descarga, inicialmente estudiados por Hossain y Younge (1992), se han modelado mediante la ecuación de convección-difusión-reacción transitoria, como se expone en Huerta, Rodríguez-Ferran, Sarrate, Díez y Fernández-Méndez (2001). Actualmente se emplea la estrategia de paso fraccionado o separación de operadores para el modelo presentado en la ecuación (2.4). Como ya ha sido indicado, durante la integración temporal, en cada paso de tiempo Δt, primero se resuelve el problema de transporte a través de la fase de convección-difusión transitoria:

$$\frac{\partial c}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} c - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} c) = 0; \tag{2.5}$$

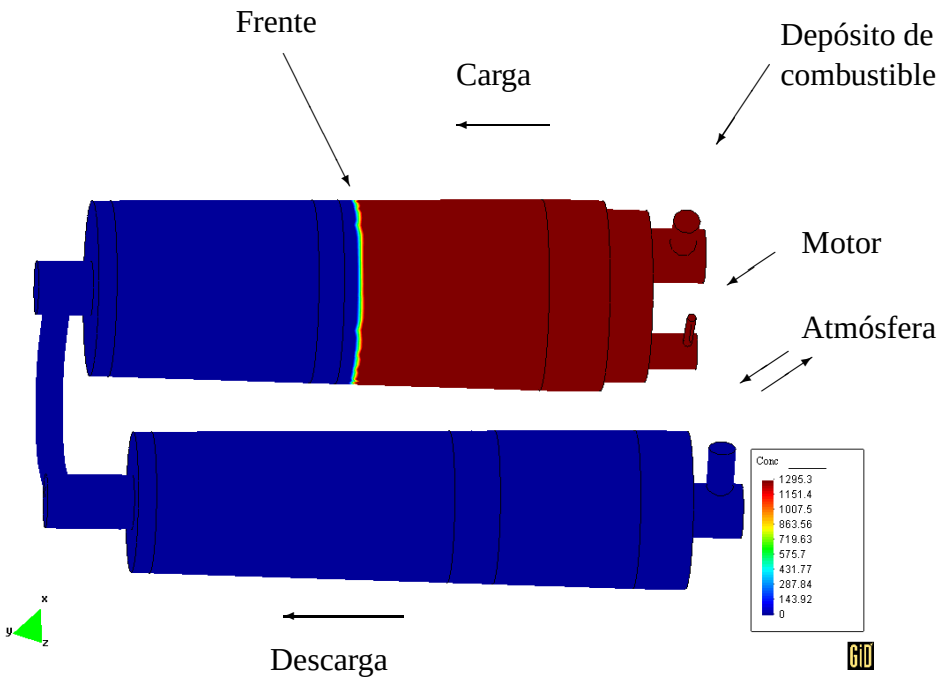


Figura 2.3: Frente abrupto en cámara de carbón en un proceso de carga.

posteriormente se realiza la transferencia de concentración de HC en el aire a la concentración del carbón, tomando en cuenta la capacidad máxima de adsorción de cada tipo de carbón activo (fase de reacción).

La modelización de los *canisters* considera, tanto los estándares de calidad para su fabricación, como los diferentes tipos de materiales que lo forman (cámaras de aire y de carbón, boquillas, espumas, diversos tipos de carbón activo, etc.). Particularmente, las cámaras de aire regulan el flujo, aunque al mismo tiempo provoca un cambio en las velocidades, ya que la velocidad del frente es más o menos 100 veces mayor en las cámaras de aire que en las regiones de carbón. De hecho, en las cámaras de carbón en un proceso de carga se crea un frente de HC con un cambio brusco de concentración, como se muestra en la figura 6.4. Esto se debe a que la velocidad en dichas cámaras es suficientemente baja, y a que un valor grande de concentración desplaza a otro más bajo, ver Pérez-Foguet y Huerta (2006).

Otros factores fundamentales a considerar en la simulación numérica son las distintas y complejas geometrías 3D del filtro, y que la ecuación (2.5) modela un

problema con convección dominante. La difusividad (ν) que se utiliza en el aire es de 10^{-5} m^2/s y para el carbón activo de 10^{-9} m^2/s.

Por otro lado, a pesar de que el modelo se ha simplificado se requiere de una combinación adecuada de sofisticados algoritmos, con el objeto de abordar cada una de las dificultades numéricas que presenta, por sí mismo, el modelo (2.5). Tal combinación de algoritmos se enumeran a continuación:

- Empleo de mallas 3D no estructuradas para elementos finitos, las cuales deben ser relativamente finas, ver ejemplos numéricos de capítulos 4, 5 y 6.
- Con el fin de obtener soluciones precisas y evitar las oscilaciones formadas principalmente en el frente del HC, se utiliza un esquema de Crank-Nicolson para discretizar el tiempo, junto con una formulación de mínimos cuadrados estándar para estabilizar el término convectivo, ver apartados 2.4 y 2.5.
- *Se precisa de un algoritmo eficiente para resolver los grandes sistemas simétricos definidos positivos sparse*, ver capítulos 4, 5 y 6.
- El uso de una estrategia de paso fraccionado para considerar la no-linealidad del término de reacción a nivel local. Por tanto, el problema global es lineal.

El objetivo de presentar esta aplicación tecnológica es que a través de la simulación numérica de los filtros de carbón activo en un proceso de carga, se pueda realizar un análisis computacional detallado empleando las diversas técnicas descritas en los capítulos 4, 5 y 6. De esa manera, se encontrará la forma más eficiente de resolver los problemas transitorios de convección-difusión. En los apartados 2.3, 2.4 y 2.5 presentamos la formulación de elementos finitos de los problemas de convección-difusión transitorios. Previamente describimos la otra aplicación tecnológica que ha motivado nuestro estudio.

2.2.2. Dispersión de contaminantes

Otra aplicación importante es la modelización de la dispersión de contaminantes en la atmósfera. Diremos que un contaminante del aire es aquella componente que está presente en la atmósfera, a niveles perjudiciales para la vida de los seres humanos, plantas y animales (Zlatev 1995, *U.S. Environmental Protection Agency* 2006).

Recordemos que la atmósfera terrestre es una mezcla gaseosa compuesta principalmente de oxígeno $(21\,\%)$ y nitrógeno $(78\,\%)$, y está constituida por varias capas. En la troposfera y estratosfera, las capas más bajas, se estudia la calidad del aire. Tales capas se extienden desde el nivel del suelo a 17 km y 50 km, respectivamente (*Wikipedia, the free encyclopedia* 2006).

Los contaminantes se clasifican en primarios y secundarios. Los primeros son emitidos directamente a la atmósfera, pueden proceder de la erupción de volcanes, incendios forestales, polvo del suelo, polen, hongos, bacterias y actividades humanas. Los secundarios se forman en la atmósfera por diversas reacciones químicas o fotoquímicas entre los contaminantes primarios. Ejemplos de ellos son el ozono (O_3), el ácido nítrico (HNO_3) y el ácido sulfúrico (H_2SO_4), entre otros (Zlatev 1995, *U.S. Environmental Protection Agency* 2006).

Algunos contaminantes primarios producidos por la actividad del hombre son: los compuestos de sulfuro (SO_2 y H_2S), de nitrógeno NO_x (conjunto de NO y NO_2), de carbón (HC, CO y CO_2), el amoniaco (NH_3), el plomo, etc. (Zlatev 1995, *U.S. Environmental Protection Agency* 2006).

Un severo problema ambiental es la deposición (lluvia) ácida, provocada principalmente por la contaminación de hidrocarburos fósiles. El carbón, el petróleo y otros combustibles fósiles contienen azufre, y debido a la combustión, el azufre se oxida a dióxido de azufre SO_2. Por otra parte, el NO se forma por reacción entre el oxígeno y nitrógeno a alta temperatura, dicha reacción se produce principalmente en los motores de coches y aviones. Las emisiones de NO se oxidan con el oxígeno atmosférico generando NO_2 (*U.S. Environmental Protection Agency* 2006).

La deposición ácida se forma por lo regular en las nubes altas, donde el SO_2 y los NO_x reaccionan con el agua y el oxígeno generando principalmente una solución diluida de ácido sulfúrico y ácido nítrico. De hecho, la radiación solar aumenta las velocidades de estas reacciones (*U.S. Environmental Protection Agency* 2006).

La lluvia, la nieve, la niebla y otras formas de precipitación arrastran estos contaminantes hacia las partes bajas de la atmósfera, depositándolos sobre la superficie de la tierra donde son absorbidos por los suelos, el agua o la vegetación. Esto se conoce como deposición húmeda. En cambio, la deposición seca es una fracción de óxidos emitidos a la atmósfera que retornan a la superficie terrestre en forma gaseosa o de

pequeñas partículas (Zlatev 1995, *U.S. Environmental Protection Agency* 2006).

Cabe resaltar que la mayor parte de la emisión del dióxido de azufre y óxidos de nitrógeno provienen de las plantas generadoras de energía eléctrica. Además, más del $50\,\%$ de la acidez en la atmósfera regresa a la superficie terrestre a través de la deposición seca. El arrastre del viento y/o los movimientos verticales determinan si la deposición seca ocurre cerca o lejos de la fuente de emisión. Por ello, se precisa de modelos numéricos eficientes que consideren estos factores y ayuden a estimar las concentraciones de los contaminantes más dañinos. De esa forma se podrá "controlar" u optimizar la emisión de estos. Además, si se cuenta con la información meteorológica adecuada, se podrán planificar nuevas fuentes de emisión donde el impacto ambiental sea mínimo.

Para la tesis se ha considerado un modelo euleriano en tres dimensiones. Se estudian cuatro especies químicas significativas: NO_x, HNO_3, SO_2 y H_2SO_4, donde se emplea un modelo lineal de oxidación e hidrólisis de los óxidos de azufre (SO_2) y nitrógeno (NO_x), ver Sanín y Montero (2004). La producción de contaminantes se realiza a través de una chimenea de una planta generadora de energía, la cual se impone como una condición tipo Dirichlet. El campo de velocidades del viento se calcula mediante un modelo de masa consistente (ver Montero, Rodríguez, Montenegro, Escobar y González-Yuste 2005). El proceso de deposición seca está representado por la llamada velocidad de deposición $(\mathbf{V_d})$ y se introduce como una condición de contorno. En cambio, la deposición húmeda se incorpora al término de reacción, como se detalla más adelante. De esa forma el modelo matemático lineal está descrito por la ecuación de convección-difusión-reacción transitoria vectorial (2.3) con el vector fuente $\mathbf{f} = 0$ y las siguientes condiciones de frontera e inicial:

$$\begin{cases} \mathbf{c}\left(\mathbf{x},t\right) = \mathbf{c}_e\left(\mathbf{x}\right) & \text{sobre } \Gamma_{D1} : \textit{c. vertical con flujo entrante,} \\ \mathbf{c}\left(\mathbf{x},t\right) = \mathbf{c}_{fuente}\left(\mathbf{x}\right) & \text{sobre } \Gamma_{D2} : \textit{boca de chimenea,} \\ -\mathbf{K}\boldsymbol{\nabla}\mathbf{c}\cdot\boldsymbol{n} = \mathbf{V_d}\mathbf{c} & \text{sobre } \Gamma_R : \textit{contorno inferior,} \\ \boldsymbol{\nabla}\mathbf{c}\cdot\boldsymbol{n} = \mathbf{0} & \text{sobre } \Gamma_N = \partial\Omega\backslash\left(\Gamma_{D1}\cup\Gamma_{D2}\cup\Gamma_R\right) : \\ & \textit{c. superior y vertical con flujo saliente,} \\ \mathbf{c}\left(\mathbf{x},0\right) = \mathbf{c}^0\left(\mathbf{x}\right) & \text{en } \Omega, \end{cases} \tag{2.6}$$

donde Ω es el dominio de estudio, $\mathbf{c}^0 = \mathbf{0}$ la concentración ambiental de referencia,

$\mathbf{K}$ el tensor diagonal de difusividad, $\mathbf{V_d}$ una matriz diagonal de deposición seca, $\boldsymbol{\alpha}$ la matriz de reacción de coeficientes cinéticos; $\mathbf{c}_e$ un valor ambiental que se asume como cero, y $\mathbf{c}_{fuente}$ la fuente de contaminantes que en este caso se considera constante.

Cálculo de la matriz de coeficientes cinéticos

Las reacciones fotoquímicas elementales de las cuatro especies consideradas son doce. Tomando en cuenta aquellas altamente reactivas con un periodo de vida corto, se simplifica el conjunto de reacciones químicas eliminando los términos no lineales y las diferentes escalas de tiempo, ver detalles en Sanín (2003). Así, el sistema de velocidades de reacción resultante es

$$\begin{aligned}
\frac{dc_{NO_x}}{dt} &= \bar{\alpha}_{NO_x,NO_x} c_{NO_x}, \\
\frac{dc_{HNO_3}}{dt} &= -\bar{\alpha}_{NO_x,NO_x} c_{NO_x}, \\
\frac{dc_{SO_2}}{dt} &= \bar{\alpha}_{SO_2,SO_2} c_{SO_2}, \\
\frac{dc_{H_2SO_4}}{dt} &= -\bar{\alpha}_{SO_2,SO_2} c_{SO_2},
\end{aligned}$$

siendo $\bar{\alpha}_{NO_x,NO_x} = -2k_1k_2$ y $\bar{\alpha}_{SO_2,SO_2} = -2\frac{k_1k_3}{k_2}$, con k_1, k_2 y k_3 las constantes cinéticas asociadas, respectivamente, a las siguientes reacciones:

$$NO_2 + hv \longrightarrow NO + 0,$$

$$OH\cdot + NO_2 \longrightarrow HNO_3,$$

$$OH\cdot + SO_2 \longrightarrow HOSO_2.$$

Considerando la información previa, se define la matriz de reacción de coeficientes cinéticos ($\boldsymbol{\alpha}$) como

$$\alpha_{ij} = \bar{\alpha}_{i,j}, \text{ si } j \neq i \text{ y } \alpha_{ii} = \bar{\alpha}_{i,i} - \frac{v_{wi}}{h},$$

con h la altura de la capa de mezcla y v_{wi} la velocidad de deposición húmeda de la especie i. Esta última a su vez satisface la relación

$$\frac{v_{wi}}{h} = w_{hi} I_i,$$

siendo I_i la intensidad de precipitación y w_{hi} el coeficiente de lavado modificado para la especie i. Por lo regular, los valor de los coeficientes modificados se encuentran en tablas.

Al igual que el modelo para los filtros de carbón activo, la modelización del problema de dispersión de contaminantes en el aire debe tomar en cuenta los siguientes algoritmos, con el fin de afrontar cada una de las dificultades numéricas que presenta el modelo (2.3)-(2.6):

- Construir una malla 3D no estructurada para elementos finitos, la cual considere la orografía real de la región de estudio.
- Crear un campo de velocidades que respete la orografía real e incluya los datos meteorológicos.
- Con la finalidad de obtener soluciones precisas se emplea el esquema de Crank-Nicolson para discretizar el tiempo y una formulación de mínimos cuadrados estándar para estabilizar el término convectivo, ver apartados 2.4 y 2.5.
- *Se requieren métodos eficientes para resolver los grandes sistemas simétricos definidos positivos sparse obtenidos en cada paso de tiempo,* ver capítulos 4.
- Simplificar las reacciones fotoquímicas no lineales, tomando en cuenta las escalas de tiempo, para obtener un sistema de velocidades de reacción lineales.

2.3. Formulación variacional

En el presente apartado se introduce la formulación variacional (o formulación débil) asociada al problema escalar de convección-difusión transitorio (2.5). La formulación de elementos finitos que se desarrolla en este y los siguientes apartados es análoga a la realizada en el problema vectorial de convección-difusión-reacción transitorio (2.3)-(2.6).

Como es usual, se denota por $L_2(\Omega)$ al espacio de las funciones de cuadrado integrable sobre Ω, dotado de la norma $\|v\|_0 = \sqrt{(v,v)}$, donde $(\cdot,\cdot)$ es el producto interior estándar, definido por $(u,v) = \int_\Omega uvd\Omega$.

En general, los espacios de Sobolev son aquellos en los que tanto sus funciones como sus derivadas son de cuadrado integrable. En particular, se utiliza el espacio

$$\mathcal{H}^1(\Omega) := \{v \in L_2(\Omega) \,|\, \frac{\partial v}{\partial x_i} \in L_2(\Omega) \forall i = 1, ..., n_d\},$$

provisto de la norma $\|v\|_1 = \sqrt{(v,v)_1}$, con $(u,v)_1 = \int_\Omega \big(uv + \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v)\big) d\Omega$. A partir del espacio de Sobolev anterior, definimos los espacios

$$\begin{aligned} \mathcal{S}_t &:= \{c \in \mathcal{H}^1(\Omega) \times [0,T] \,|\, c(\mathbf{x},t) = c_{\text{entrada}} \text{ sobre } \Gamma_E\}, \\ \mathcal{V} &:= \{v \in \mathcal{H}^1(\Omega) \,|\, v(\mathbf{x}) = 0 \text{ sobre } \Gamma_E\}. \end{aligned} \tag{2.7}$$

Finalmente, para obtener la formulación débil se multiplica la ecuación (2.5) por una función de test $v \in \mathcal{V}$. Después se aplica el teorema de la divergencia (o teorema de Gauss) al término difusivo, y por último, se emplea el hecho de que $v = 0$ sobre Γ_E y $\boldsymbol{\nabla} c \cdot \boldsymbol{n} = 0$ sobre $\partial\Omega \backslash \Gamma_E$. De esa forma, la formulación variacional del problema de convección-difusión transitorio se enuncia como:

Encontrar $c \in \mathcal{S}_t$, tal que

$$\frac{\partial}{\partial t}(c,v) + a\,(v,c) = 0 \qquad \forall v \in \mathcal{V}, \tag{2.8}$$

donde

$$a\,(v,c) := (v, \boldsymbol{v} \cdot \boldsymbol{\nabla} c) + (\boldsymbol{\nabla} v, \nu \boldsymbol{\nabla} c)\,. \tag{2.9}$$

Notemos que $\boldsymbol{v}$ es constante en este caso. La existencia y unicidad de la solución se presenta en Quarteroni y Valli (1994).

2.4. Discretización en el tiempo

La precisión de la integración temporal juega un papel muy importante en los problemas no estacionarios. Donea y Huerta (2003) mencionan que los esquemas más populares para resolver problemas parabólicos son los métodos de la familia

θ. En particular, el esquema θ para la ecuación de convección-difusión transitoria (2.5) se expresa como

$$\frac{\Delta c}{\Delta t} - \theta \Delta c_t = c_t^n,$$

donde Δt es el incremento de tiempo, $\Delta c := c^{n+1} - c^n$ es el incremento en la concentración, c_t es una notación compacta para $\partial c/\partial t$ y el superíndice denota el paso de tiempo.

Cuando $\theta = 1/2$ se obtiene el método de Crank-Nicolson, que es un esquema de segundo orden. Dado que $c_t = -(\boldsymbol{v} \cdot \boldsymbol{\nabla} c - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} c))$, el esquema semidiscreto se reescribe como

$$c^{n+1} + \frac{\Delta t}{2} \mathcal{L}_1 c^{n+1} = c^n - \frac{\Delta t}{2} \mathcal{L}_1 c^n \tag{2.10}$$

donde el operador $\mathcal{L}_1$ está definido en la ecuación (2.2). Métodos de alto orden como los propuestos por Huerta y Donea (2002) y Huerta et al. (2002) también pueden ser aplicados.

2.5. Formulación de elementos finitos y mínimos cuadrados

Existen varias técnicas para estabilizar la ecuación de convección-difusión transitoria, con el objeto de evitar las oscilaciones numéricas. Entre ellas se consideran: SUPG, GLS, SGS y LS, las cuales están ampliamente descritas en el libro de Donea y Huerta (2003). En particular, se ha elegido la combinación del método de Crank-Nicolson junto con la formulación de mínimos cuadrados estándar (LS) por dos razones. La primera y la más importante es que el método de mínimos cuadrados produce sistemas simétricos definidos positivos (la matriz real $\mathbf{A}$ es definida positiva si $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^N, \mathbf{x} \neq 0$). La segunda razón, es que LS introduce más difusividad que SUPG, GLS y SGS para los problemas de convección dominante, ver Huerta y Donea (2002) y Huerta et al. (2002). Tal combinación de estrategías funciona adecuadamente (es más precisa) si el número de Courant ($C = \|\boldsymbol{v}\| \Delta t/h$, con h el tamaño del elemento) es cercano a uno .

Para desarrollar la formulación de mínimos cuadrados estándar se utiliza el operador espacial $I + \frac{\Delta t}{2} \mathcal{L}_1$, con I la identidad. De esa forma, se construye la siguiente

ecuación integral:

$$\left(v+\frac{\Delta t}{2}\mathcal{L}_1 v, c^{n+1}+\frac{\Delta t}{2}\mathcal{L}_1 c^{n+1}\right)=\left(v+\frac{\Delta t}{2}\mathcal{L}_1 v, c^{n}-\frac{\Delta t}{2}\mathcal{L}_1 c^{n}\right), \tag{2.11}$$

donde la interpolación y las funciones de forma están contenidas en un subespacio de $\mathcal{H}^{1+}$, definido por

$$\mathcal{H}^{1+}:=\{v\in\mathcal{H}^1(\Omega)\,|\,v|_{\Omega_e}\in\mathcal{H}^2(\Omega_e)\quad\forall\,\Omega_e\}. \tag{2.12}$$

Observación 2.5.1. El método de mínimos cuadrados estándar, a diferencia de las otras técnicas de estabilización, tiene la gran ventaja de producir simetría en el término de estabilización del lado izquierdo de la ecuación (2.11).

Para la discretización espacial usamos elementos finitos lineales: triángulos en 2D y tetraedros en 3D. Después de aplicar el teorema de la divergencia y las condiciones de contorno y teniendo en cuenta que $\boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla})=0$ para los elementos lineales Ω_e, se obtiene la formulación débil del problema

$$\begin{aligned}&\left(v,c^{n+1}\right)+\frac{\Delta t}{2}a\left(v,c^{n+1}\right)+\sum_e\left[\frac{\Delta t}{2}\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}v,c^{n+1}\right)_e+\frac{\Delta t^2}{4}\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}v,\boldsymbol{v}\cdot\boldsymbol{\nabla}c^{n+1}\right)_e\right]\\&=(v,c^n)-\frac{\Delta t}{2}a\left(v,c^n\right)+\sum_e\left[\frac{\Delta t}{2}\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}v,c^{n}\right)_e-\frac{\Delta t^2}{4}\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}v,\boldsymbol{v}\cdot\boldsymbol{\nabla}c^{n}\right)_e\right]\end{aligned} \tag{2.13}$$

donde la forma bilineal está definida por

$$a(v,c):=(v,\boldsymbol{v}\cdot\boldsymbol{\nabla}c)+(\boldsymbol{\nabla}v,\nu\boldsymbol{\nabla}c). \tag{2.14}$$

Finalmente, la discretización por elementos finitos de la relación (2.13) deriva en el sistema de ecuaciones lineales

$$\begin{aligned}&\left[\mathbf{M}+\frac{\Delta t}{2}\left(\mathbf{G}+\mathbf{D}\right)+\frac{\Delta t}{2}\mathbf{G}^T+\frac{\Delta t^2}{4}\widehat{\mathbf{D}}\right]\mathbf{c}^{n+1}=\\&\left[\mathbf{M}-\frac{\Delta t}{2}\left(\mathbf{G}+\mathbf{D}\right)+\frac{\Delta t}{2}\mathbf{G}^T-\frac{\Delta t^2}{4}\widehat{\mathbf{D}}\right]\mathbf{c}^{n},\end{aligned} \tag{2.15}$$

donde $\mathbf{c}$ es el vector nodal de concentraciones, y las matrices $\mathbf{M}$ (matriz de masa), $\mathbf{G}$ (matriz de convección), $\mathbf{D}$ (matriz se difusividad) y $\widehat{\mathbf{D}}$ están definidas como

$$\begin{aligned}m_{ij}&=\int_\Omega N_iN_j\mathrm{d}\Omega\qquad;\qquad g_{ij}=\int_\Omega N_i\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}N_j\right)\mathrm{d}\Omega\\d_{ij}&=\int_\Omega\nu\boldsymbol{\nabla}N_i\cdot\boldsymbol{\nabla}N_j\mathrm{d}\Omega\qquad;\qquad \widehat{d}_{ij}=\int_\Omega\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}N_i\right)\left(\boldsymbol{v}\cdot\boldsymbol{\nabla}N_j\right)\mathrm{d}\Omega,\end{aligned} \tag{2.16}$$

con N_i la función de forma en el nodo i. Con la notación compacta $\mathbf{A} := \mathbf{M} + \frac{\Delta t}{2}\left(\mathbf{D} + \mathbf{G} + \mathbf{G}^T\right) + \frac{\Delta t^2}{4}\widehat{\mathbf{D}}$ y $\mathbf{B} := \mathbf{M} + \frac{\Delta t}{2}\mathbf{G}^T$, el sistema lineal de ecuaciones (2.15) se reescribe como

$$\mathbf{A}\mathbf{c}^{n+1} = \left[2\mathbf{B} - \mathbf{A}\right]\mathbf{c}^n. \tag{2.17}$$

Observación 2.5.2. Las matrices $\mathbf{M}$, $\mathbf{D}$ y $\widehat{\mathbf{D}}$ son simétricas, mientras que $\mathbf{G}$ no lo es. Por su parte, las matrices $\widehat{\mathbf{D}}$ y $\mathbf{G}^T$ están asociadas a la estrategia de estabilización LS; la última asegura la simetría de la matrix $\mathbf{A}$.

Observación 2.5.3. La matriz de masa $\mathbf{M}$ es definida positiva, ver Johnson (1987). Por consiguiente, para Δt suficientemente pequeño, la matriz $\mathbf{A}$ es definida positiva.

Observación 2.5.4. Un forma de verificar que la matriz $\mathbf{M} + \frac{\Delta t}{2}\left(\mathbf{D} + \mathbf{G} + \mathbf{G}^T\right)$ es definida positiva, dado que el término $\frac{\Delta t^2}{4}\widehat{\mathbf{D}}$ es semidefinido positivo, es planteando el problema generalizado de valores propios $\mathbf{M}\mathbf{x} = -\frac{\Delta t}{2}\left(\mathbf{D} + \mathbf{G} + \mathbf{G}^T\right)\mathbf{x}$ y encontrar el valor propio positivo Δt más pequeño.

Observación 2.5.5. El valor de Δt que satisface las restricciones de precisión y estabilidad ($C \approx 1$) típicamente está por debajo del límite para el cual $\mathbf{A}$ es SPD (ver apartado 4.7). Para las mallas de los filtros de carbón activo Δt se elige tal que cumpla $C = \|\boldsymbol{v}\|\Delta t/\xi_e h \approx 1$, donde $\boldsymbol{v}$ es la velocidad y ξ_e la porosidad interparticular.

Observación 2.5.6. La matriz $\mathbf{A}$ del sistema (2.17) es constante en cada paso de tiempo.

Resumiendo: gracias al uso de una estabilización LS del término convectivo, *los sistemas a resolver en cada paso de tiempo* (2.17) *son simétricos definidos positivos, con matriz constante*. Esto resulta ser muy atractivo desde el punto de vista computacional, especialmente para aplicaciones industriales 3D como es el caso de las simulaciones de los filtros de carbón activo o de la dispersión de contaminantes atmosféricos.

Capítulo 3

Solución de grandes sistemas lineales: preliminares y estado del arte

En este capítulo se realiza una breve introducción a los métodos de resolución de sistemas lineales de ecuaciones que emplearemos a lo largo de este trabajo. En los primeros apartados se revisa las diferencias que existen entre las técnicas directas e iterativas, así como las características de cada uno de los algoritmos que se manejan. En el apartado 3.4 se trata el tema de precondicionadores en forma general; los precondicionadores implícitos y explícitos se estudian ampliamente en los capítulos 4 y 5, respectivamente. Finalmente, en el apartado 3.5 presentamos un breve estado del arte de las técnicas de descomposición de dominios, distinguiendo los métodos con y sin solapamiento. Referencias recientes sobre estos temas son los libros de Saad (2003), Quarteroni y Valli (1999), Meurant (1999), Greenbaum (1997), Trefethen y Bau III (1997), Axelsson (1996), Golub y Van Loan (1996), Kelley (1995), Hackbusch (1994), Barrett et al. (1994), Duff, Erisman y K. (1992).

3.1. Introducción

La solución de grandes sistemas tipo *sparse* de la forma $\mathbf{Ax} = \mathbf{b}$, donde $\mathbf{A} \in \mathscr{M}_N(\mathbb{R})$ es la matriz de coeficientes y $\mathbf{b} \in \mathbb{R}^N$ es un vector independiente dado, es central en muchas simulaciones numéricas en la ingeniería y en la ciencia, y a menudo consumen la mayor parte de tiempo de computación. Debido a esto, desde

finales del siglo XIX (métodos iterativos clásicos) y principalmente en la segunda mitad del siglo pasado se han desarrollado diversos métodos numéricos para hallar su solución.

Podemos simplificar la clasificación de los métodos de solución en directos e iterativos. Los métodos directos, basados en la descomposición de la matriz $\mathbf{A}$ en matrices fácilmente invertibles, son los más utilizados y a menudo se eligen para incluirlos en los códigos industriales, en donde la fiabilidad es una preocupación primaria. De hecho, las técnicas directas se caracterizan por su robustez, puesto que puede predecirse el tiempo (cantidad de operaciones en punto-flotante) y el almacenamiento que requieren.

Sin embargo, para la solución de problemas a gran escala los métodos directos llegan a ser poco o nada prácticos, incluso sobre plataformas en paralelo, sólo basta ver que precisan almacenar N^2 entradas y $O(N^3)$ operaciones en punto-flotante para calcular la descomposición. En este caso el uso de técnicas iterativas es obligatorio. Un ejemplo de ello son los problemas de dispersión de contaminación que requieren de cientos de miles de incógnitas para su simulación, ver apartado 4.8.

Los métodos iterativos en ocasiones pueden divergir o invertir más tiempo de CPU que los directos, como se señala en el apartado 4.7.4. Una forma de aumentar su eficiencia y fiabilidad es precondicionándolos.

Se llama *precondicionador* a la matriz no-singular $\mathbf{M}$ que aproxima a $\mathbf{A}$ (en algún sentido) y que provoca que el sistema:

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \tag{3.1}$$

tenga la misma solución de $\mathbf{A}\mathbf{x} = \mathbf{b}$ pero sea más fácil de resolver. En este caso, se dice que el sistema (3.1) está *precondicionado por la izquierda* (Benzi 2002). También existen otras formas de precondicionar, *por la derecha*:

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{u} = \mathbf{b}, \qquad \mathbf{x} = \mathbf{M}^{-1}\mathbf{u}; \tag{3.2}$$

y *por ambos lados*:

$$\mathbf{M}_{\mathrm{I}}^{-1}\mathbf{A}\mathbf{M}_{\mathrm{D}}^{-1}\mathbf{u} = \mathbf{M}_{\mathrm{I}}^{-1}\mathbf{b}, \qquad \mathbf{x} = \mathbf{M}_{\mathrm{D}}^{-1}\mathbf{u}, \qquad \text{con} \quad \mathbf{M} = \mathbf{M}_{\mathrm{I}}\mathbf{M}_{\mathrm{D}}. \tag{3.3}$$

Estas formas se aplican para obtener los métodos de Krylov precondicionados. Por ejemplo, en el apartado 3.3.1 se deduce el método de gradientes conjugados precondicionado. El tema de precondicionadores se trata en el apartado 3.4.

Cabe señalar que si $\mathbf{A}$ es simétrica definida positiva y se puede obtener un precondicionador simétrico de la forma $\mathbf{M} = \mathbf{L}\mathbf{L}^T$, entonces las formas de precondicionar (3.1) y (3.2) no preservan simetría, en cambio el sistema (3.3) es simétrico definido positivo, ver Saad (2003).

3.2. Métodos directos

Los *métodos directos* son las técnicas numéricas que producen la solución de sistemas de ecuaciones lineales en un número finito de operaciones aritméticas. El más conocido de los métodos directos es la *Eliminación Gaussiana*, ver Duff et al. (1992).

El método de Eliminación Gaussiana es equivalente a efectuar la descomposición de la matriz $\mathbf{A}$ en el producto de la matriz triangular inferior $\mathbf{L}$ con unos sobre la diagonal y la matriz triangular superior $\mathbf{U}$. La solución se obtiene resolviendo dos sistemas triangulares ($\mathbf{L}\mathbf{y} = \mathbf{b}$ y $\mathbf{U}\mathbf{x} = \mathbf{y}$), ver Duff et al. (1992) .

En general, el tipo de descomposición a realizar está determinado por las propiedades de la matriz $\mathbf{A}$. Sin embargo, como los ordenadores modernos calculan más lentamente las raíces cuadradas que las multiplicaciones y divisiones se tratan de evitar calcularlas. En la actualidad para las matrices tipo *sparse* simétricas definidas positivas no se programa $\mathbf{L}\mathbf{L}^T$, sino la forma $\mathbf{L}\mathbf{D}\mathbf{L}^T$ o $\mathbf{U}^T\mathbf{D}\mathbf{U}$, ver Meurant (1999).

Los algoritmos de las descomposiciones por lo regular cuentan con tres ciclos principales, se asigna la letra i al índice de la fila, la j al índice de la columna y k puede eventualmente ser ambos índices. Los diferentes códigos que existen para construir las descomposiciones dependen del orden de los ciclos, así podríamos tener la versión kij, o la jki, etc.; en Meurant (1999) se encuentran seis versiones diferentes para implementar la factorización $\mathbf{L}\mathbf{D}\mathbf{L}^T$. La versión más adecuada está en función del tipo de almacenamiento que se utilice para guardar $\mathbf{L}$, del lenguaje en que se implemente y de la arquitectura del ordenador. Cabe mencionar que actualmente se cuenta con diversas estructuras de datos para almacenar una matriz de tipo *sparse*, ver Saad (1994*b*) y apartado 4.6.

El método de eliminación Gaussiana falla si en el proceso de eliminación se obtiene un elemento diagonal nulo, debido a que habría una división entre cero.

Para evitar este problema se utilizan las estrategias de pivotamiento. Dichas técnicas también evitan la propagación de errores de redondeo provocadas por pivotes muy pequeños (inestabilidades) (Duff et al. 1992). En particular, el método de Cholesky no requiere una estrategia de pivotamiento, por ser $\mathbf{A}$ simétrica definida positiva, ver Trefethen y Bau III (1997).

Otra cuestión importante sobre los métodos de descomposición es la producción de llenado (*fill-in*) en los factores. La forma de reducirlo es reordenando las entradas de la matriz, ya sea antes a nivel de malla o durante la descomposición. Las diversas y complejas estrategias de reordenamiento se pueden consultar en Meurant (1999). Este tema se tratará en el apartado 4.5.

3.2.1. Método de Cholesky

Denominamos *método de Cholesky* o simplemente *Cholesky* a la descomposición $\mathbf{U}^T\mathbf{D}\mathbf{U}$ que empleamos en nuestro análisis. Dado que el algoritmo está basado en la versión jik, se utiliza un *skyline* para almacenar la matriz triangular superior, ver apartado 4.6. El algoritmo de Cholesky se presenta en la figura 3.1.

```
Para j = 1 : N
    Para i = 2 : j − 1
        temp = 0
        Para k = 1 : i − 1
            temp = temp + a_ki a_kj
        Terminar
        a_ij = a_ij − temp
    Terminar
    Para i = 1 : j − 1
        D = a_ij
        a_ij = a_ij / a_ii
        a_jj = a_jj − D a_ij
    Terminar
Terminar
```

Figura 3.1: Algoritmo del método de Cholesky, versión jik

3.3. Métodos iterativos

Un método iterativo construye una sucesión de vectores $\{\mathbf{x}^k\}$, $k = 0, 1, ...$, la cual se espera que converja a la solución $\mathbf{x}^*$ del sistema $\mathbf{Ax} = \mathbf{b}$, dado una aproximación inicial $\mathbf{x}^0$. El método se dice convergente si $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^*$.

Los métodos iterativos se dividen en métodos estacionarios y no-estacionarios, ver Barrett et al. (1994). En los estacionarios, técnicas de la forma $\mathbf{x}^k = \mathbf{Bx}^{k-1} + \mathbf{C}$, se incluyen principalmente los llamados métodos clásicos como son: Jacobi, Gauss-Seidel y los métodos de sobrerelajación sucesiva (SOR) y de sobrerelajación sucesiva simétrica (SSOR), ver Axelsson (1996). Por su parte, los métodos iterativos no-estacionarios contienen a los llamados métodos de Krylov, tales como: CG, GMRES, QMR, Bi-CGSTAB, CGN y LSQR entre otros, ver los libros de Saad (2003) y Greenbaum (1997).

Los métodos como GMRES o CG teóricamente convergen en un número finito de etapas (aritmética exacta), ver Saad (2003). Sin embargo, en aritmética finita este resultado ya no es válido. Por ejemplo, un mal condicionamiento de la matriz provoca que CG converja lentamente. La forma de acelerar la convergencia de los métodos de Krylov es precondicionandolos.

Para nuestro estudio, hemos elegido el método de gradientes conjugados precondicionado, dado que es la mejor opción a emplear dentro de las técnicas de Krylov para resolver sistemas lineales simétricos definidos positivos (Barrett et al. 1994).

3.3.1. Método de gradientes conjugados precondicionado

Uno de las técnicas iterativas más estudiadas es el *método de gradientes conjugados* (CG) desarrollado por Hestenes y Stiefel en 1952. Aunque inicialmente fue vista como método directo, no fue hasta 1959 que la estrategia de CG se considera como una técnica iterativa (Engeli, Ginsburg, Rutishauser y Stiefel 1959), ver Saad y van der Vorst (2000). Las ideas necesarias para construir tanto CG como el método de gradientes conjugados precondicionado (PCG) se presentan a continuación.

En general, en la k-ésima iteración $\mathbf{x}^k$, gradientes conjugados minimiza la función

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{b} \tag{3.4}$$

sobre $\mathbf{x}^0+\mathcal{K}_k$, donde $\mathbf{x}^0$ es la iteración inicial y $\mathcal{K}_k$ es el k-ésimo subespacio de Krylov definido por $\mathcal{K}_k = span\{\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, ..., \mathbf{A}^{k-1}\mathbf{r}^0\}$ para $k \geq 1$ con $\mathbf{r}^0 = \mathbf{b}-\mathbf{A}\mathbf{x}^0$. La existencia del mínimo únicamente se garantiza si $\mathbf{A}$ es SDP. Kelley (1995) muestra que minimizar ϕ sobre $\mathbf{x}^0 + \mathcal{K}_k$ (subconjunto de $\mathbb{R}^N$) es equivalente a minimizar $\|\mathbf{x} - \mathbf{x}^*\|_\mathbf{A}$ sobre el mismo subconjunto, donde $\|\mathbf{x}\|_\mathbf{A} = \sqrt{\mathbf{x}^T\mathbf{A}\mathbf{x}}$ se conoce como la A-norma y $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ es el mínimo. Los detalles se pueden consultar en Greenbaum (1997), Hackbusch (1994) y Kelley (1995).

Por su parte, Saad (2003) muestra que si $\mathbf{A}$ y $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ son SDP, la convergencia del método de CG es la misma (salvo errores de redondeo) en los sistemas (3.1), (3.2) y (3.3).

Saad deduce el algoritmo de PCG empleando el sistema precondicionado por la izquierda (3.1), y las siguientes ideas:

(a) El $\mathbf{M}$-*producto interior* se define como: $(\mathbf{x}, \mathbf{y})_\mathbf{M} = (\mathbf{M}\mathbf{x}, \mathbf{y})_2 = (\mathbf{x}, \mathbf{M}\mathbf{y})_2$, donde $(\cdot, \cdot)_2$ es el usual producto interior euclidiano. Observemos que $\|\mathbf{x}\|_\mathbf{A} = \sqrt{(\mathbf{x}, \mathbf{x})_\mathbf{A}}$.

(b) La matriz $\mathbf{M}^{-1}\mathbf{A}$ es auto-adjunta para el $\mathbf{M}$-producto interior:

$$(\mathbf{M}^{-1}\mathbf{A}\mathbf{x}, \mathbf{y})_\mathbf{M} = (\mathbf{A}\mathbf{x}, \mathbf{y})_2 = (\mathbf{x}, \mathbf{A}\mathbf{y})_2 = (\mathbf{x}, \mathbf{M}^{-1}\mathbf{A}\mathbf{y})_\mathbf{M}.$$

(c) La relación (3.4) se puede reescribir como

$$\phi(\mathbf{x}) = \frac{1}{2}(\mathbf{r}, \mathbf{A}^{-1}\mathbf{r})_2. \tag{3.5}$$

Ahora, si reemplazamos en la igualdad (3.5) a $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ por $\mathbf{z} = \mathbf{M}^{-1}\mathbf{r}$, a la matriz $\mathbf{A}$ por $\mathbf{M}^{-1}\mathbf{A}$ y al producto interior euclidiano por el $\mathbf{M}$-producto interior, se obtiene que el método PCG minimiza la función:

$$\phi(\mathbf{x}) = \frac{1}{2}(\mathbf{z}, (\mathbf{M}^{-1}\mathbf{A})^{-1}\mathbf{z})_\mathbf{M} \tag{3.6}$$

sobre el subespacio de Krylov $\{\mathbf{r}^0, \mathbf{M}^{-1}\mathbf{A}\mathbf{r}^0, ..., (\mathbf{M}^{-1}\mathbf{A})^{k-1}\mathbf{r}^0\}$. Observemos que el precondicionador $\mathbf{M}$ necesita ser SDP para que el $\mathbf{M}$-producto interior exista.

El algoritmo de gradientes conjugados precondicionado se presenta en la figura 3.2. Los detalles de implementación se pueden consultar en Saad (2003), Axelsson (1996) y los teóricos en Greenbaum (1997) y Hackbusch (1994).

En general existen dos resultados fundamentales sobre la convergencia de CG. El primero indica que dicho método termina en un número finito de iteraciones,

$\mathbf{x}^0$: aproximación inicial con la solución del paso anterior
Calcular $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$; $\quad k = 0$
Resolver $\mathbf{M}\mathbf{z}^0 = \mathbf{r}^0$; $\quad \mathbf{p}^0 = \mathbf{z}^0$
$\rho_0 = {\mathbf{r}^0}^T \mathbf{z}^0$
Mientras (No convergencia $\quad \& \quad k < kmax$)
$\quad \mathbf{q}^k = \mathbf{A}\mathbf{p}^k$
$\quad \alpha_k = \rho_k / {\mathbf{p}^k}^T \mathbf{q}^k$
$\quad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k$
$\quad \mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k \mathbf{q}^k$
$\quad$ resolver $\mathbf{M}\mathbf{z}^{k+1} = \mathbf{r}^{k+1}$
$\quad \rho_{k+1} = {\mathbf{r}^{k+1}}^T \mathbf{z}^{k+1}$
$\quad \beta_k = \rho_{k+1} / \rho_k$
$\quad \mathbf{p}^{k+1} = \mathbf{z}^{k+1} + \beta_k \mathbf{p}^k$
$\quad k = k + 1$
Terminar

Figura 3.2: Método de gradientes conjugados precondicionado.

ver Kelley (1995), y el segundo da una cota superior para el error que se comete al utilizar CG en términos de la A-norma:

$$\|\mathbf{x}^* - \mathbf{x}^k\|_{\mathbf{A}} \leq 2 \|\mathbf{x}^* - \mathbf{x}^0\|_{\mathbf{A}} \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right)^k, \tag{3.7}$$

donde $\kappa(\mathbf{A})$ es el número de condición de la matriz $\mathbf{A}$ definido por

$$\kappa(\mathbf{A}) = \lambda_{\text{máx}} / \lambda_{\text{mín}} \tag{3.8}$$

con $\lambda_{\text{mín}}$ y $\lambda_{\text{máx}}$ el mínimo y máximo valor propio de $\mathbf{A}$, respectivamente. Ver la demostración del resultado en Greenbaum (1997) o Golub y Van Loan (1996).

La relación (3.7) indica que la velocidad de convergencia de CG depende fuertemente del número de condición de $\mathbf{A}$, $\kappa(\mathbf{A})$. Cuánto mayor sea la distancia entre los valores propios de $\mathbf{A}$ más lenta será la convergencia. En el caso que la matriz esté muy mal condicionada (las escalas de magnitud entre $\lambda_{\text{máx}}$ y $\lambda_{\text{mín}}$ difieren considerablemente) o cuando los valores propios estén muy próximos a cero (inestabilidades) es necesario verificar que se está convergiendo a la solución, ver Sosonkina, Melson, Saad y Watson (2000).

Por último, Meurant (1999) demuestra que la velocidad de convergencia para gradientes conjugados precondicionado (por la izquierda) satisface

$$\|\mathbf{x}^* - \mathbf{x}^k\|_{\mathbf{A}} \leq 2\,\|\mathbf{x}^* - \mathbf{x}^0\|_{\mathbf{A}} \left(\frac{\sqrt{\kappa(\mathbf{K})} - 1}{\sqrt{\kappa(\mathbf{K})} + 1}\right)^k \tag{3.9}$$

donde $\mathbf{K} = \mathbf{M}^{-1}\mathbf{A}$. Observemos que si $\mathbf{M}^{-1}$ es igual a $\mathbf{A}^{-1}$ se converge en una iteración, lo que es equivalente a resolver el problema original.

3.4. Precondicionadores implícitos versus explícitos

Como ya ha sido señalado la velocidad de convergencia de PCG depende de los extremos de los valores propios de la matriz de coeficientes, ver relaciones (3.7)-(3.8). En general, los métodos de Krylov convergerán rápidamente, si la matriz $\mathbf{A}$ es cercana a la identidad, $\mathbf{I}$. Una forma de mejorar las propiedades espectrales es multiplicando la matriz original $\mathbf{A}$ por otra matriz $\mathbf{M}^{-1}$, tal que la matriz $\mathbf{M}^{-1}\mathbf{A}$ se acerque a la identidad (Greenbaum 1997). Para problemas definidos positivos, CG o MINRES obtienen una buena convergencia si la matriz precondicionada tiene valores propios agrupados con pocos que sean grandes. También la convergencia es buena si la matriz precondicionada tiene pocos valores propios que sean distintos, (ver Greenbaum 1997).

Se distinguen dos grandes grupos de precondicionadores: implícitos y explícitos. En los primeros se encuentra $\mathbf{M}$ y en los segundos se calcula directamente $\mathbf{M}^{-1}$. Como ejemplo de precondicionadores implícitos tenemos el diagonal, el SSOR y las factorizaciones incompletas (FI). Por su parte, en los explícitos están el diagonal óptimo, los polinómicos y las inversas aproximadas, ver Barrett et al. (1994).

De acuerdo a Saad y van der Vorst (2000) la idea de multiplicar un sistema de $\mathbf{Ax} = \mathbf{b}$ por un polinomio P(A) con el objeto de acelerar la convergencia del método iterativo de Richardson fue dada por Cesari en 1937. Sin embargo fue hasta 1952 que Lanczos definió claramente la noción de precondicionar.

De hecho, se puede decir que las técnicas modernas de precondicionamiento iniciaron a finales de los 60s principios de los 70s, tal es el caso del SSOR (para CG) de Axelsson (1972), así como los primeros estudios de factorizaciones incompletas para matrices estructuradas originadas por la aproximación con diferencias finitas

de los operadores elípticos (Meijerink y van der Vorst 1977). Actualmente, debido a la simulación de problemas 3D, cada vez más reales, se sigue proponiendo y empleando diversas clases de precondicionadores para matrices no estructuradas, que van desde los polinómicos (Greenbaum 1997), por bloques (Axelsson 1996), los multimallas o multinivel (Hackbusch 1994), las aproximaciones de la matriz inversa (Benzi y Tůma 1999), los creados con el método de descomposición de dominios (Quarteroni y Valli 1999) y por supuesto las factorizaciones incompletas (Saad 2003) entre otros.

Notemos de la figura 3.2 que en cada iteración de PCG se requiere resolver el sistema $\mathbf{M}\mathbf{z}^{k+1} = \mathbf{r}^{k+1}$. En caso de elegir un precondicionador implícito, como una factorización incompleta de Cholesky $\mathbf{M} = \mathbf{L}\mathbf{L}^T$, emplearíamos para resolverlo una sustitución hacia adelante y otra hacia atrás, estrategias que no son altamente paralelizables. En cambio la operación $\mathbf{z}^{k+1} = \mathbf{M}^{-1}\mathbf{r}^{k+1}$ se puede paralelizar masivamente, ya que cada multiplicación matriz-vector $\mathbf{M}^{-1}r_i^{k+1}$ es independiente. Por otro lado, las factorizaciones incompletas pueden fallar por una división entre cero, como se verá en el capítulo 4, no obstante en las inversas aproximadas basadas en la minimización en la norma de Frobenius $(\mathbf{M}^{-1})$ no ocurre esto (ver capítulo 5). En suma, en un ambiente en paralelo los precondicionadores explícitos tendrían ventajas sobre los implícitos. Sin embargo, la mayoría de los códigos industriales continúan ejecutándose secuencialmente. En el capítulo 5 se propone una inversa aproximada para resolver sistemas simétricos definidos positivos.

De entre los precondicionadores implícitos, hemos elegido las factorizaciones incompletas por la relevancia que han adquirido en los últimos años. Especialmente, en el capítulo 4 realizamos un análisis completo de dos familias de factorizaciones incompletas de Cholesky para los problemas de convección-difusión transitorios. Con el fin de poner los resultados en contexto las comparamos con el precondicionador implícito más simple: Jacobi, que se describe a continuación.

3.4.1. Precondicionador de Jacobi

El nombre de este precondicionador se origina por la comparación del método iterativo de Jacobi con la solución del sistema precondicionado empleando el método de Richardson.

Si el sistema precondicionado por la izquierda es $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$; la solución de la técnica Richardson se escribe como

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{M}^{-1}(\mathbf{A}\mathbf{x}^k - \mathbf{b}). \tag{3.10}$$

Ahora definamos al precondicionador como $\mathbf{M} = \mathbf{D}$. Separemos la matriz de coeficientes $\mathbf{A}$ por $\mathbf{A} = \mathbf{E} + \mathbf{D} + \mathbf{F}$, con $\mathbf{D}$ una matriz formada por los elementos diagonales de $\mathbf{A}$, y $\mathbf{E}$ y $\mathbf{F}$ matrices triangulares inferior y superior, respectivamente. Al sustituir las dos relaciones anteriores en (3.10) y realizando las operaciones pertinentes obtenemos el método de Jacobi:

$$\mathbf{x}^{k+1} = \mathbf{D}^{-1}\mathbf{b} - \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\mathbf{x}^k. \tag{3.11}$$

Por esta razón, a la matriz $\mathbf{M} = \mathbf{D}$ se le denomina precondicionador diagonal o de Jacobi.

3.5. Técnicas de descomposición de dominios

Existen otras técnicas iterativas útiles para la resolución numérica de ecuaciones en derivadas parciales (EDP) conocidas como los *métodos de descomposición de dominios (DD)*. La idea de estas técnicas consiste en particionar el dominio asociado al problema global de EDP en subdominios finitos, de tal forma que el problema global (sistema total de ecuaciones) se pueda descomponer en tantos subproblemas (o subsistemas) como subdominios existan. La forma de dividir el dominio de estudio y las condiciones de contorno que se impongan en las fronteras ficticias (interfases) entre los subdominios determinan el tipo de método de DD. Por su naturaleza estas estrategias son apropiadas para obtener la solución del problema global en un ordenador con múltiples procesadores. Una buena referencia para este tema es el libro de Quarteroni y Valli (1999). Para un punto de vista más algebraico consultar los libros de Saad (2003) y Meurant (1999).

Los algoritmos de descomposición de dominios pueden clasificarse en dos categorías: con y sin solapamiento. Los métodos de DD con solapamiento han sido los primeros en ser estudiados, ya que en 1869, de acuerdo con Quarteroni y Valli

(1999), Hermann Schwarz desarrolló una técnica iterativa que convergía a la solución de un problema de EDP definido sobre un dominio formado por la unión de un rectángulo con un disco. Dicha estrategia se conoce como el *método alternante de Schwarz*. Actualmente, dependiendo de la forma de la sucesión, los métodos de Schwarz se distinguen en: multiplicativo (alternante) y aditivo. Estas técnicas se detallan en el apartado 3.5.1.

Por otro lado, los métodos de DD sin solapamiento reducen el problema original en otros de dimensión más pequeña que están relacionados mediante las variables de las interfases. Tales técnicas se conocen como *métodos del complemento de Schur o métodos subestructurados*. Las ideas básicas de estas estrategias se presentan en el apartado 3.5.2.

Las técnicas de DD también son adecuadas para resolver un problema de EDP definido sobre un dominio con frontera compleja. En este caso, se propone un domino más grande con una frontera sencilla, de tal forma que incluya el dominio original. Se resuelve el mismo problema de EDP sobre el dominio propuesto y se transmite la información de un dominio a otro por medio de multiplicadores de Lagrange, ver Glowinski, Pan y Periaux (1994) y Juárez, Glowinski y Pan (2002). A esta estrategia se le denomina *método de dominio ficticio*. Existen otros tipos y usos de estas técnicas, pero no ahondaremos en ellas. Nos centraremos en explicar, a grandes rasgos, las características de los métodos con y sin solapamiento enfatizando los primeros, debido a que utilizamos en el capítulo 6 un método de DD con solapamiento.

3.5.1. Métodos con solapamiento

En este y el siguiente apartado emplearemos el problema de Poisson para ejemplificar en qué consisten los métodos de DD. El problema de Poisson está definido por

$$\begin{cases} -\Delta u &= f & \text{en } \Omega, \\ u &= 0 & \text{sobre } \partial\Omega, \end{cases} \tag{3.12}$$

Supongamos que el dominio Ω se descompone en dos dominios solapados Ω_1 y Ω_2 $(\Omega_1 \cap \Omega_2 \neq 0)$, tal que $\Omega = \Omega_1 \cup \Omega_2$. Denotamos por $\Gamma_1 := \partial\Omega_1 \cap \Omega_2$ y $\Gamma_2 := \partial\Omega_2 \cap$

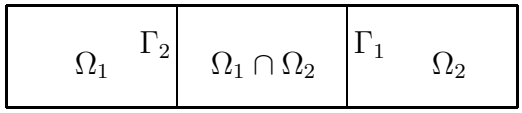


Figura 3.3: Ejemplo de dos dominios solapados.

Ω_1 como se muestra en la figura 3.3. En general, los *métodos de Schwarz* generan un sucesión $\{u^k\}$ que converge a u, y dependiendo de la forma de la sucesión se le asigna un nombre a la técnica. Por ejemplo, en el *método multiplicativo de Schwarz (MSM)* se escoge una aproximación inicial u^0 definida en Ω, con $u^0 = 0$ sobre $\partial\Omega$. Cada elemento de la sucesión (iteración) se genera en dos etapas:

(i) resolver

$$\begin{cases} -\Delta u_1^{k+1} = f & \text{en } \Omega_1, \\ u_1^{k+1} = 0 & \text{sobre } \partial\Omega \cap \partial\Omega_1, \\ u_1^{k+1} = u_2^k & \text{sobre } \Gamma_1, \end{cases} \tag{3.13}$$

para obtener

$$u^{k+1} = \begin{cases} u_1^{k+1} & \text{en } \Omega_1, \\ u_2^k & \text{en } \Omega \backslash \Omega_1; \end{cases} \tag{3.14}$$

(ii) resolver

$$\begin{cases} -\Delta u_2^{k+1} = f & \text{en } \Omega_2, \\ u_2^{k+1} = 0 & \text{sobre } \partial\Omega \cap \partial\Omega_2, \\ u_2^{k+1} = u_1^{k+1} & \text{sobre } \Gamma_2, \end{cases} \tag{3.15}$$

para obtener

$$u^{k+1} = \begin{cases} u_1^{k+1} & \text{en } \Omega \backslash \Omega_2, \\ u_2^{k+1} & \text{en } \Omega_2; \end{cases} \tag{3.16}$$

En cambio en el *método aditivo de Schwarz (ASM)* los elementos de la sucesión (iteración) respetan el suproblema de la etapa (i), pero no así las condiciones de

contorno de la etapa (ii), puesto que ahora se impone que $u_2^{k+1} = u_1^k$ sobre Γ_2, ver Dryja (1989) y Widlund (1988).

De las relaciones (3.14) y (3.16) se aprecia el por qué el algoritmo del método multiplicativo de Schwarz coincide con el método iterativo de Gauss-Seidel por bloques, donde cada bloque está solapado, ver Tang (1992). Esta idea se desarrolla con detalle en el capítulo 6. Análogamente, el ASM se asocia a la iteración de Jacobi por bloques.

La formulación variacional del problema global (3.12) es

$$a\,(u,v) = (f,v)\,, \qquad \forall v \in \mathcal{H}_0^1(\Omega), \tag{3.17}$$

donde $\mathcal{H}_0^1(\Omega) := \{v \in \mathcal{H}^1(\Omega)\,|\,v(\mathbf{x}) = 0 \text{ sobre } \partial\Omega\}$ y la forma bilineal a está determinada por

$$a\,(u,v) := \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, d\Omega.$$

Sean $V_i = \mathcal{H}_0^1(\Omega_i)$ para $i = 1, 2$. Si definimos las proyecciones $\mathscr{P}_i : V \to V$ por $a\,(\mathscr{P}_i v, \phi) = a\,(v, \phi)$, $\forall \phi \in V_i$, entonces de las etapas (i)-(ii) del MSM se obtiene la ecuación de propagación del error

$$e^{k+1} = (I - \mathscr{P}_2)\,(I - \mathscr{P}_1)\,e^k = Q_2 e^k,$$

donde $e^k = u - u^k$, ver Meurant (1999). Debido a que las proyecciones $\mathscr{P}_i$ se están multiplicando este algoritmo se conoce como *el método multiplicativo de Schwarz* (ver Quarteroni y Valli 1999). Para el caso de M subdominios el error satisface

$$e^{k+1} = \Pi_{i=1}^M\,(I - \mathscr{P}_i)\,e^k = Q_M e^k.$$

Lions (1988) muestra que para unas propiedades adecuadas de los subespacios V_1 y V_2, la sucesión $\{e^k\}$ converge a 0.

El operador Q_M juega un papel importante en la construcción de precondicionadores. Para ello, se equipara con una iteración de punto fijo para el problema precondicionado: $\mathbf{M}^{-1}\mathbf{A}\mathbf{u} = \mathbf{M}^{-1}\mathbf{b}$, donde $\mathbf{M}^{-1}\mathbf{A} = \mathbf{I} - \mathbf{Q}_M$ y $\mathbf{M}^{-1}\mathbf{b} = (\mathbf{I} - \mathbf{Q}_M)\mathbf{A}^{-1}\mathbf{b}$. Por ejemplo para dos subdominios, el MSM se escribe como

$$\mathbf{u}^{k+1} = \mathbf{u}^k + [\mathbf{I} - (\mathbf{I} - \mathbf{P}_2)(\mathbf{I} - \mathbf{P}_1)]\mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{u}^k),$$

en este caso el precondicionador es $\mathbf{M}^{-1} = (\mathbf{I} - \mathbf{Q}_2)\mathbf{A}^{-1}$. Dado que $\mathbf{M}^{-1}\mathbf{A}\mathbf{u} = (\mathbf{I} - \mathbf{Q}_M)\mathbf{u}$ y $\mathbf{g} = \mathbf{M}^{-1}\mathbf{b} = (\mathbf{I} - \mathbf{Q}_M)\mathbf{v}$, el objetivo es resolver $(\mathbf{I} - \mathbf{Q}_M)\mathbf{u} = \mathbf{g}$ usando un método iterativo como GMRES, puesto que la matriz es no simétrica; a esta estrategia se le llama *método multiplicativo de Schwarz acelerado*. También existe el *el algoritmo multiplicativo de Schwarz simetrizado*, en este caso el precondicionador que se obtiene es simétrico $\mathbf{M}^{-1} = (\mathbf{I} - \mathbf{Q}_M^T\mathbf{Q}_M)\mathbf{A}^{-1}$. Se recomienda resolver el sistema $(\mathbf{I} - \mathbf{Q}_M^T\mathbf{Q}_M)\mathbf{v} = \mathbf{g}$ mediante la iteración de gradientes conjugados; ver detalles en Saad (2003).

Los *métodos de dos niveles* introducen una malla gruesa con el fin de suministra, en cada iteración, una comunicación global entre todos los subdominios. Para construir los precondicionadores de dos niveles, ya sea simétricos o no, se inicia con los datos de la malla gruesa; ver detalles en Quarteroni y Valli (1999).

Bjørstad, Dryja y Vainikko (1996) prueban los algoritmos presentados anteriormente utilizando un procesamiento en paralelo y aplicando técnicas de coloreado para los subdominios. Sun y Tang (1996) imponen una condición tipo Robin sobre las fronteras artificiales para aumentar la eficiencia del método alternante de Schwarz. Por su parte Brakkee, Vuik y Wesseling (1995) realizaron un estudio en donde resuelven los subsistemas en forma precisa (utilizando una descomposición completa) e imprecisa (empleando FIC) para los problemas de Poisson y de convección-difusión. Encontraron que se podía reducir el tiempo de CPU del método de DD por un factor de 2 hasta 6 dependiendo del problema, ver también Brakkee, Vuik y Wesseling (1998).

Observemos que el MSM se puede paralelizar utilizando un coloreado y ordenamiento de subdominios de tal forma que dos subdominios que estén conectados entre sí no tengan el mismo color. Los experimentos numéricos muestran que al minimizar el número de colores aumenta la convergencia del método, ver Cai, Gropp y Keyes (1992) y Cai y Sarkis (1998).

Bramble, Pasciak, Wang y Xu (1991) proporcionan la siguiente estima para la velocidad de convergencia del MSM en los problemas elípticos de segundo orden:

$$1 - \frac{(\delta H)^2}{cM^2}, \tag{3.18}$$

donde c es una constante que no depende ni de H (el diámetro máximo de los subdominios), ni de h; δH es la medida lineal de la región de solapamiento entre

dos subdominios adyacentes ($0 < \delta \leq 1$), ver también Widlund (1988). Ahora bien, para que la velocidad de convergencia sea independiente de δH y h se introduce una malla gruesa (método de dos niveles), ver Quarteroni y Valli (1999).

El ASM, equivalente al método de Jacobi por bloques, por lo general, según Meurant (1999), no converge. Por ello es necesario considerarlo como un precondicionador: $\mathbf{M}^{-1} = \sum_{i=1}^{M} \mathbf{P}_i \mathbf{A}^{-1}$. Observemos que si $\mathbf{A}$ es simétrica definida positiva entonces $\mathbf{M}^{-1}$ también lo es (ver Quarteroni y Valli 1999). En *el algoritmo aditivo de Schwarz acelerado* se resuelve el sistema $\sum_{i=1}^{M} \mathbf{P}_i \mathbf{u} = \mathbf{g}$ empleando el método de gradientes conjugados. Para que la velocidad de convergencia de este método sea independiente de la medida lineal de la región de solapamiento δH se emplea una malla gruesa (método de dos niveles), ver Dryja y Widlund (1991) y Dryja y Widlund (1994).

En general, los métodos aditivos de Schwarz convergen más lentamente que los algoritmos multiplicativos de Schwarz, sin embargo, su convergencia es independiente del ordenamiento y coloreado de los subdominios. También hay ocasiones en donde los métodos de Schwarz son más eficiente sin malla gruesa, debido a que esta tiene que ser lo suficientemente "fina" para que el número de iteraciones sea menor que cuando no se incluye, ver Cai y Sarkis (1998).

Pavarino (1992) realiza una comparación de todos los algoritmos anteriores para el problema de Poisson empleando un procesador secuencial. También, Cai y Sarkis (1998) efectúan lo mismo para la ecuación de convección-difusión. Asimismo Lasser y Toselli (2003) proponen un precondicionador ASM de dos niveles para resolver los problemas de convección-difusión utilizando aproximaciones discontinuas de Galerkin. Guo y Cao (1998) calculan unas estimas para el número de condición utilizando el ASM y la versión h-p del método de elementos finitos.

3.5.2. Métodos sin solapamiento

Los métodos subestructurados o métodos del complemento de Schur reformulan el problema original con condiciones a la frontera empleando una separación de dominios disjunta, de tal forma que las soluciones de los subdominios satisfacen un adecuado acoplamiento de las condiciones en las interfases de los subdominios.

Para ejemplificar la idea previa, consideramos nuevamente el problema (3.12),

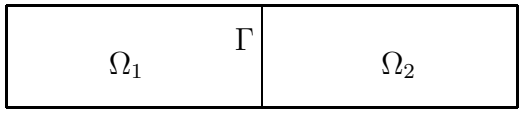


Figura 3.4: Ejemplo de dos subdominios no solapados.

pero con los dominios Ω_1 y Ω_2 disjuntos (sin solapar). Definimos por Γ la frontera ficticia entre ellos, ver figura 3.4. Por consiguiente, *el método iterativo subestructurado* se escribe como:

(i) resolver

$$\begin{cases} -\Delta u_1^{k+1} = f & \text{en } \Omega_1, \\ u_1^{k+1} = 0 & \text{sobre } \partial\Omega \cap \partial\Omega_1, \\ \alpha_1 u_1^{k+1} + \beta_1 \frac{\partial u_1^{k+1}}{\partial n} = \alpha_1 u_2^k + \beta_1 \frac{\partial u_2^k}{\partial n} & \text{sobre } \Gamma. \end{cases} \tag{3.19}$$

y

(ii) resolver

$$\begin{cases} -\Delta u_2^{k+1} = f & \text{en } \Omega_2, \\ u_2^{k+1} = 0 & \text{sobre } \partial\Omega \cap \partial\Omega_2, \\ \alpha_2 u_2^{k+1} + \beta_2 \frac{\partial u_2^{k+1}}{\partial n} = \alpha_2 u_1^k + \beta_2 \frac{\partial u_1^k}{\partial n} & \text{sobre } \Gamma. \end{cases} \tag{3.20}$$

para obtener a u^{k+1}.

Según el tipo de condiciones de transmisión $(3.19)_3$-$(3.20)_3$ que se impongan se le asigna un nombre al método subestructurado. De esa forma se obtienen los siguientes técnicas:

- método Dirichlet/Dirichlet: con $\beta_1 = \beta_2 = 0$;
- método Dirichlet/Neumann: con $\beta_1 = \alpha_2 = 0$;
- método Robin/Robin: con $\beta_1, \beta_2, \alpha_1, \alpha_2$ distintos de cero;
- método Robin/Neumann: con $\alpha_2 = 0$;

- método Dirichlet/Robin: con $\beta_1 = 0$;
- método Neumann/Neumann: con $\alpha_2 = \alpha_1 = 0$ (solución salvo una constante).

A los métodos subestructurados se les denomina también *métodos del complemento de Schur* debido a que al implementarlos, en lugar de resolver la matriz global se resuelve el sistema de complemento de Schur. De hecho, se puede interpretar al método subestructurado como un método de Richarson precondicionado para el sistema del complemento de Schur, ver Quarteroni y Valli (1999) y Saad (2003).

Por otro lado, al resolver la ecuación de convección-difusión con un método subestructurado se debe de tomar en cuenta la dirección del flujo en las condiciones de frontera de los subdominios. En la entrada del flujo se pueden imponer condiciones de tipo Dirichlet o Robin, mientras que en la salida del flujo se deben de imponer condiciones de Neumann, por lo que el signo de $\mathbf{v} \cdot \mathbf{n}$ cambia. Los algoritmos que consideran esta idea se conocen como *métodos adaptativos*, ver Gastaldi, Gastaldi y Quarteroni (1996) y Quarteroni y Valli (1999).

Algunas de estas técnicas, pero incluyendo un pequeño solapamiento se analizan en Houzeaux (2001). Por su parte, estudios relacionados con la ecuación de convección-difusión y DD sin solapamiento se pueden encontrar en Toselli (2001), Rapin y Lube (2002), Espedal, Tai y Yan (1998) y Layton (1995). Una aplicación en ingeniería biomédica usando los métodos adaptativos se plantea en Quarteroni, Veneziani y Zunino (2002).

Finalmente, cabe mencionar que ninguna de las referencias proporcionadas en este apartado y el anterior, emplean un esquema de mínimos cuadrados para elementos finitos, excepto en Bose, Carey y de Almeida (2003), donde proponen un algoritmo multi-frontal en paralelo usando el método subestructurado para fluidos viscosos incompresibles.

En el capítulo 6 se utiliza una técnica de DD con dominios solapados, concretamente proponemos una estrategia iterativa que está basada en el método multiplicativo de Schwarz.

Capítulo 4

Factorizaciones incompletas de Cholesky

En este capítulo se analiza numéricamente la eficiencia de dos familias de factorizaciones incompletas de Cholesky (FIC). La estructura del mismo de divide en dos parte. La primera incluye algunos preliminares y un breve estado del arte sobre FIC. En la segunda se presenta propiamente el análisis computacional de tales precondicionadores.

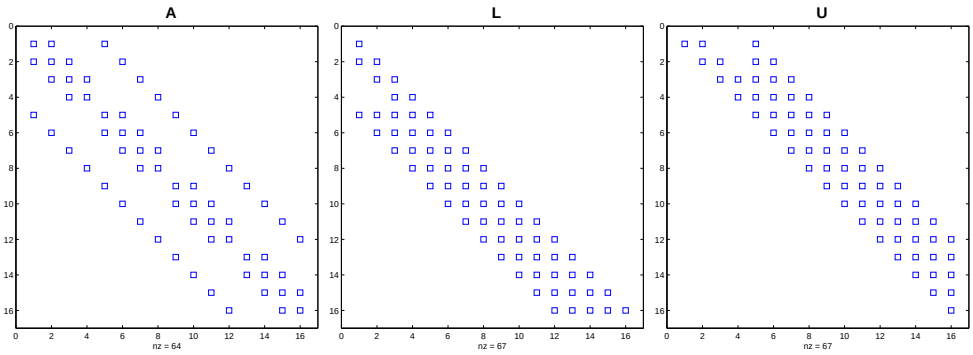
4.1. Introducción

Estamos interesados en estudiar las factorizaciones incompletas de Cholesky, aunque los preliminares y el estado de arte (apartados 4.1-4.4) se presentan con un planteamiento general para matrices no simétricas. Cuando se trabaje con el caso simétrico, se indicará.

Con el objeto de explicar todo lo referente a las factorizaciones incompletas (FI) iniciamos definiendo los siguientes conceptos.

Denominamos *patrón de sparsidad o de no nulos* de la matriz $\mathbf{A}$ al conjunto K de todas las posiciones (i, j) para las cuales $a_{ij} \neq 0$.

Entendemos por *llenado (fill-in)* a toda posición (i, j) en donde $\mathbf{A}$ tiene un nulo, pero no en la descomposición exacta, ya sea en $\mathbf{L}$ o $\mathbf{U}$. Por ejemplo, las figuras 4.1.b-4.1.c muestran los factores $\mathbf{L}$ y $\mathbf{U}$ de la descomposición completa de la matriz $\mathbf{A}$ (figura 4.1.a). Notemos que en 4.1.c existen “diagonales extra” que no están en la



(a) (b) (c)

Figura 4.1: Patrón de *sparsidad* de las matrices: (a) $\mathbf{A}$, (b) $\mathbf{L}$ y (c) $\mathbf{U}$.

parte triangular superior de la matriz $\mathbf{A}$. A tales "diagonales extras" se les nombra llenado.

El llenado de una descomposición completa $\mathbf{LU}$ implica un coste computacional alto, en cuanto a memoria y número de operaciones. La idea principal de las factorizaciones incompletas es reducirlo.

Definamos al *patrón de nulos* de la FI como el conjunto $\mathcal{P}$ de todas las posiciones (i, j) en donde $l_{ij} = 0$, o $u_{ij} = 0$ con $i \neq j$.

Dado $\mathcal{P}$, una *factorización incompleta* se define mediante un par de matrices triangulares, inferior $\mathbf{L}$ y superior $\mathbf{U}$ tales que cumple:

$$\mathbf{A} = \mathbf{LU} - \mathbf{R}, \qquad l_{ij} = 0, \quad u_{ij} = 0 \quad \text{si } (i, j) \in \mathcal{P}$$

A las matrices $\mathbf{L}$ y $\mathbf{U}$ se les denomina *factores incompletos* y la FI es usual denotarla por ILU.

Si la matriz $\mathbf{A}$ es SDP y determinamos un patrón de nulos $\mathcal{P}$, entonces la *factorización incompleta de Cholesky*, denotada por ILL^{T} , se define como

$$\mathbf{A} = \mathbf{LL^T} - \mathbf{R}, \qquad l_{ij} = 0 \quad \text{si } (i, j) \in \mathcal{P},$$

donde $\mathbf{L}$ es el *el factor incompleto de Cholesky*.

Es importante señalar que el llenado está asociado a los factores incompletos $\mathbf{L}$ y $\mathbf{U}$. Para aclarar esta idea nos referimos a la figura 4.2. Por un lado, notemos que las matrices $\mathbf{L}$ y $\mathbf{U}$ (figuras 4.2.a y 4.2.b respectivamente) tiene el mismo patrón de no nulos que las matrices triangulares inferior y superior de $\mathbf{A}$, figura 4.2.c. Por

otro lado, si realizamos la multiplicación de las matrices **L** y **U** nos percatamos que aparecen dos "diagonales extras" (figura 4.2,d). El concepto de *factorización incompleta sin llenado* está asociado a la pareja de matrices triangulares, **L** y **U**, tal que al realizar la descomposición se respeta las posiciones de las entradas no nulas de **A**, esto es, en la k-ésima etapa se satisface la siguiente relación:

$$i,j > k: \qquad a_{ij} \leftarrow \begin{cases} a_{ij} - a_{ik}a_{kk}^{-1}a_{kj} & \text{si } (i,j) \in K \\ a_{ij} & \text{en otro caso,} \end{cases}$$

tal factorización es denotada por ILU(0) (Barrett et al. 1994, Saad 2003).

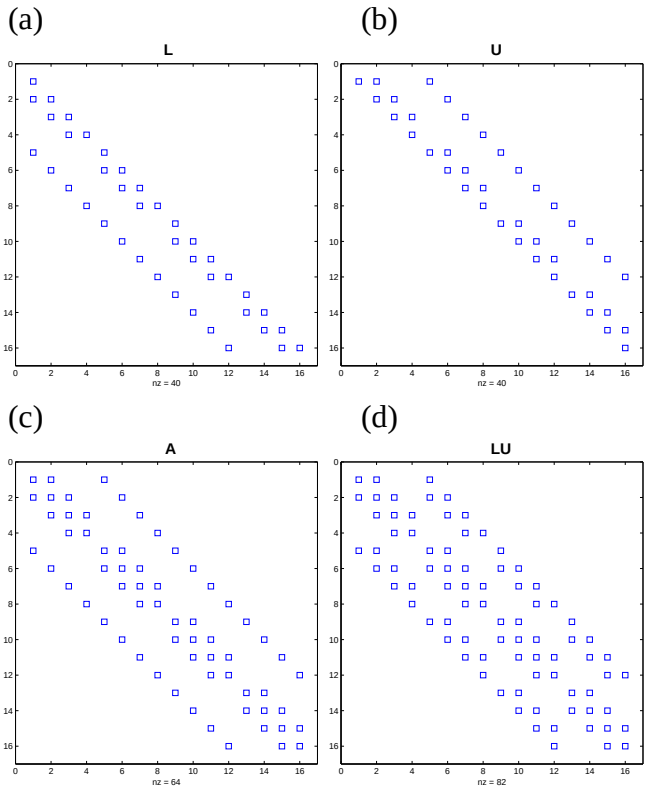


Figura 4.2: Patrón de *sparsidad* de la factorización ILU(0).

En general, dependiendo de la estrategia o criterio que se elija para determinar el llenado, las factorizaciones incompletas se clasifican en dos tipos: memoria prescrita y umbral, ver Barrett et al. (1994). Esta distinción también es válida para las FI de Cholesky como veremos en los apartados 4.2 y 4.3.

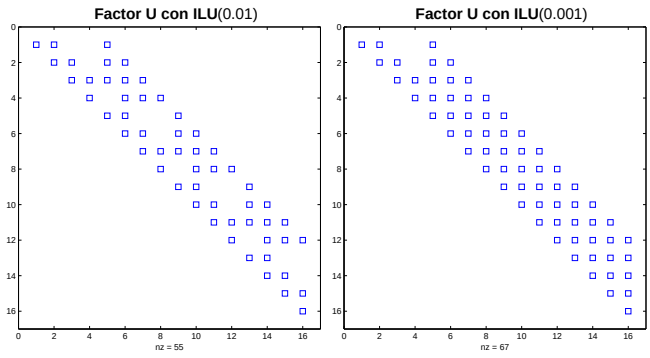
4.2. FIC de umbral

En las factorizaciones incompletas de Cholesky de umbral, los elementos no diagonales son eliminados durante la descomposición si ellos están por debajo de una cierta tolerancia o umbral.

Existen varias reglas de eliminación de las entradas "pequeñas". Munksgaard (1980), por ejemplo, elimina $a_{ij}^{(k+1)}$ durante la $k-$ésima etapa si

$$|a_{ij}^{(k+1)}| \leq \tau \sqrt{|a_{ij}^{(k)} a_{jj}^{(k)}|}, \tag{4.1}$$

donde τ es la tolerancia o umbral. En cambio, Saad (2003) y (1994*a*) realiza la eliminación si $|a_{ij}^{k}|$ es menor o igual que el producto de τ por la norma-l_2 de la $k-$ésima fila de $\mathbf{A}$. Observemos que si τ es grande, aunque habrá pocos elementos no nulos en el llenado, tenderá a ser un "pobre" precondicionador (Jacobi). En cambio, si τ tiende a cero se obtendrá prácticamente la factorización completa de Cholesky. Por ejemplo, la figura 4.3 muestra dos factores incompletos de Cholesky de la matriz $\mathbf{A}$ presentada en la figura 4.2.c. Se emplea una FIC de umbral para $\tau = 0.01$ y $\tau = 0.001$ asociadas a las gráficas (a) y (b), respectivamente.



(a) (b)

Figura 4.3: Gráficas de los factores incompletos de Cholesky de umbral para: (a) $\tau = 0.01$ y (b) $\tau = 0.001$.

Un inconveniente de las estrategias de umbral es no poder predecir la memoria para almacenar el precondicionador. Para rectificar este problema Saad (2003) y (1994*a*) propone, además de la regla de eliminación descrita en el párrafo anterior, retener los p más grandes elementos en magnitud en cada fila de $\mathbf{L}$ y $\mathbf{U}$. Esta

factorización se conoce como ILUT(p,τ). Dicha descomposición produce un precondicionador no simétrico, cuando la matriz es simétrica. Por esta razón, en esta tesis no se utiliza la ILUT de Saad y se trabaja con el algoritmo de Munksgaard (1980). Dicha estrategia se detalla a continuación.

4.2.1. Algoritmo de Munksgaard (1980)

Es prácticamente una constante el que casi cualquier artículo que mencione el tema de precondicionadores de umbral cite el algoritmo de Munksgaard (1980). En general, la estrategia construye un precondicionador de la forma $\mathbf{M} = \mathbf{LDL}^T$ usando la versión kji de la descomposición de Cholesky (Meurant 1999), como se muestra en la figura 4.4. En este caso, se almacena la matriz triangular inferior $\mathbf{L} + \mathbf{D}^{-1} - \mathbf{I}$ en coordenadas simétricas (ver apartado 4.6).

El criterio de llenado de la FIC se realiza utilizando el umbral τ y la relación 4.1. Notemos que la factorización sin llenado no puede ser obtenida como un caso particular de este tipo de FIC, puesto que la eliminación de las entradas se realiza de acuerdo a su tamaño, no a su localización en la matriz.

El algoritmo de Munksgaard (figura 4.4) es robusto debido a que incluye diversas estrategias, dos de ellas para asegurar la existencia y estabilidad de la FIC. La primera perturba globalmente la matriz original obteniendo $\widehat{\mathbf{A}} = \mathbf{A} + \alpha \mathrm{diag}(\mathbf{A})$ para asegurar que ésta sea definida positiva (ver apartado 4.4). La segunda se emplea para preservar la estabilidad si los pivotes son negativos o cercanos a cero, es decir, los pivotes se perturban en forma local y dinámicamente mediante la siguiente regla:

$$\text{Si} \quad d_{kk} \leq u \left(\sum_{j \neq k} |a_{kj}| \right) \quad \text{entonces} \quad \begin{cases} d_{kk} = \sum_{j \neq k} |a_{kj}| & \text{si} \quad \sum_{j \neq k} |a_{kj}| \neq 0 \\ d_{kk} = 1 & \text{si} \quad \sum_{j \neq k} |a_{kj}| = 0 \end{cases}, \tag{4.2}$$

donde $u = 0.01$. Notemos que si $u = 1$, el precondicionador sería diagonalmente dominante y sus múltiplos cumplirían: $|l_{ik}| < 1$ en la k-ésima etapa. Ahora, si todos los pivotes satisfacen el criterio $d_{kk} > u \left(\sum_{j \neq k} |a_{kj}| \right)$ podemos asegurar un límite razonable en el crecimiento del tamaño de los elementos en cada etapa, dado que $\max_{ij} |a_{ij}^{(k+1)}| < (1 + u^{-1}) \max_{ij} |a_{ij}^{(k)}|$, ver apartado 4.4.

Este algoritmo tiene la opción de incluir la estrategia MILU de Gustafsson (1978) mediante el parámetro `ICM=S`, ver apartado 4.3 y figura 4.4. Con el fin de disminuir el llenado, también se puede elegir permutar la matriz original para obtener $\mathbf{PLDL}^T\mathbf{P}^T$, donde $\mathbf{P}$ es una matriz de permutación.

Finalmente, notemos que si la relación (4.2) se utiliza varias veces, el precondicionador será muy pobre. En este caso se sugiere incrementar el valor de α y comenzar nuevamente la descomposición. El algoritmo inicia con $\alpha = 0$.

Resumiendo lo anterior, los resultados numéricos presentados en el apartado 4.7 se han elaborado con la subrutina F11JAF, la cual está basada en el algoritmo de Munksgaard (ver Salvini y Shaw 1995). En nuestros experimentos numéricos, $\alpha = 0$ proporcionó resultados satisfactorios y solamente se requirieron de perturbaciones locales con $u = 0.01$. Como se utiliza un ordenamiento nodal no es necesario permutar la matriz ni utilizar la estrategia MILU (`ICM=N`).

Escoger `ICM`, $u > 0$, y $\tau > 0$
Para $k = 1, ..., N-1$
  $a_{kk} \leq u\big(\sum_{j\neq k} |a_{kj}|\big) \quad \Rightarrow \quad a_{kk} = \sum_{j\neq k} |a_{kj}|$
  otro caso
    $\sum_{j\neq k} |a_{kj}| = 0 \quad \Rightarrow \quad a_{kk} = 1$
  $d_{kk} = a_{kk}$
  Para $i > k$ tal que $a_{i,k} \neq 0$
    $l_{i,k} = a_{i,k}/d_{k,k}$
    Para $j \geq i$ tal que $a_{j,k} \neq 0$
      $a_{ji} \neq 0$, o $|a_{i,k}a_{j,k}| \geq \tau\sqrt{|a_{i,i}a_{j,j}|} \quad \Rightarrow \quad a_{j,i} = a_{j,i} - l_{i,k}a_{j,k}$
      otro caso
        `ICM=S` $\Rightarrow$ $a_{i,i} = a_{i,i} - l_{i,k}a_{j,k}$
        $a_{j,j} = a_{j,j} - l_{i,k}a_{j,k}$
    Terminar
  Terminar
Terminar

Figura 4.4: Algoritmo de Munksgaard (1980).

4.3. FIC de memoria prescrita

A diferencia de las FIC de umbral, las factorizaciones incompletas de Cholesky *de memoria prescrita* fijan previamente el patrón de no nulos del FI o el número de

entradas no nulas del factor incompleto **L**.

Resulta interesante el ejemplo de Meijerink y van der Vorst (1977) quienes consideran dos formas de escoger el patrón K para las mallas estructuradas obtenidas al discretizar EDPs con diferencias finitas. Primero definen a K como el conjunto de todas las posiciones (i, j) para las cuales $a_{ij} \neq 0$, o sea la factorización incompleta sin llenado, ver figura 4.2. En la segunda estrategia K incluyen llenado, es decir, a priori determinan cuántas "diagonales extras" va a contener el factor incompleto. Ver por ejemplo las figuras 4.5.a y 4.5.b en donde aparece una diagonal extra comparada con la gráfica 4.2.c.

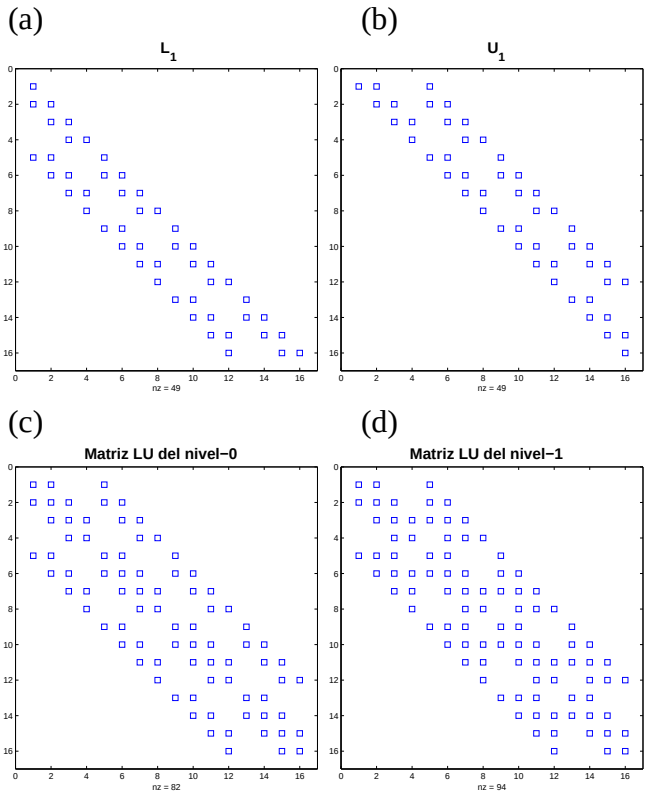


Figura 4.5: Patrón de *sparsidad* de la factorización ILU(1).

En 1978, el concepto de "diagonales extras" es cambiado a "*niveles de llenado*" por Gustafsson y generalizado para matrices no estructuradas por Watts III en 1981. Por ejemplo, el nivel-0 es equivalente a la factorización incompleta sin llenado, ILU(0) (ver figuras 4.2.a y 4.2.b). El *nivel-1 de llenado* (ver 4.5.a y 4.5.b) permite las entradas no nulas correspondientes a los no nulos de las matrices del nivel-0 (figs.

4.2.a y 4.2.b) y cualquier entrada no nula introducida por la eliminación de la matriz $\mathbf{LU}$ del nivel-0 (fig. 4.2.d), es decir, las "diagonales extras" de ILU(0). El *nivel-2 de llenado* incluye los del nivel-1 de llenado (figs. 4.5.a y 4.5.b) y cualquier no nulo producido por la eliminación de la matriz $\mathbf{LU}$ del nivel-1 de llenado (fig. 4.5.d). En la práctica dicha técnica requiere de dos fases, una simbólica para determinar la estructura de los factores $\mathbf{L}$ y $\mathbf{U}$ y otra fase numérica. Por su parte Saad (2003) une ambas fases obteniendo lo que actualmente se conoce como ILU(p), ver también Saad (1994*b*). Un inconveniente del ILU(p) es no poder predecir la cantidad de memoria para un $p > 0$; aplicaciones de esta técnica se pueden encontrar en Chow y Saad (1997), D'Azevedo, Forsyth y Tang (1992), y Dickinson y Forsyth (1994).

Otra estrategia, también considerada de memoria prescrita, es la propuesta por Jones y Plassmann (1995) para matrices simétricas definidas positivas. La técnica fija el número de entradas no nulas en cada columna de tal forma que, el número de no nulos contenidos en el factor incompleto sea igual al número de no nulos que hay en cada columna de la matriz $\mathbf{A}$. Además, el patrón de *sparsidad* K de la matriz original es ignorado y únicamente son retenidos los elementos no nulos más grandes en magnitud. Lin y Moré (1999) utilizan esta estrategia, con la salvedad de que permiten llenado controlado por el parámetro p (idea de Saad 1994*a*), es decir, para cada columna de $\mathbf{L}$, se fija el mismo número de no nulos de $\mathbf{A}$ más p entradas. Este método respeta la simetría del precondicionador y se puede predecir la memoria de $\mathbf{L}$, para $p \geq 0$, tomando en cuenta la magnitud de los elementos.

Por último se dispone de una estrategia, generalizada por Gustafsson (1978), que permite incorporar o compensar los elementos que son descartados durante el proceso de eliminación del factor incompleto $\mathbf{L}$. Consiste en sustraer todos los elementos que son descartados en la etapa k-ésima del elemento diagonal k-ésimo. La técnica conocida como *factorización incompleta modificada* y denotada por MILU, garantiza que la suma de las filas de $\mathbf{A}$ sean iguales a aquellas qué $\mathbf{LU}$ ($\mathbf{Ae} = \mathbf{LUe}$, con $\mathbf{e} = (1, 1, \ldots, 1)^T$). En Barrett et al. (1994) puede encontrarse más información sobre ella y sus variantes, y en Eijkhout (1999) sobre cuándo falla o "rompe" el proceso. Notemos que esta estrategia se puede emplear para cualquier tipo de factorización incompleta, ya sea de memoria prescrita o de umbral (apartado 4.2).

De todas las estrategias presentadas en este apartado hemos elegido el algoritmo

de Lin y Moré, debido a que se puede predecir la memoria requerida y asegurar que el precondicionador obtenido sea simétrico definido positivo, como se detalla a continuación.

4.3.1. Algoritmo de Lin y Moré (1999)

El algoritmo de Lin y Moré (1999) construye una factorización ILL^T de memoria prescrita, basada en la versión jki de la factorización de Cholesky (Meurant 1999), como se muestra en la figura 4.6.

Escoger $\alpha_s > 0$ y $p \geq 0$
Calcular $\widehat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, donde $\mathbf{D} = \mathrm{diag}(\|\mathbf{A}\mathbf{e}_i\|_2)$
$\mathrm{mín}(\hat{a}_{ii}) > 0 \implies \alpha_0 = 0$
otro caso $\alpha_0 = -\mathrm{mín}(\hat{a}_{ii}) + \alpha_s$
Para $m = 0, 1, ...,$
  Calcular $\widehat{\mathbf{A}}_m = \widehat{\mathbf{A}} + \alpha_m \mathbf{I}$
  Para $j = 1 : N$
    $\hat{a}_m(j,j) \leq 0 \implies$ `éxito = falso`
    `salir ciclo`
    $\hat{a}_m(j,j) = \sqrt{\hat{a}_m(j,j)}$
    `col_len` = `size`$(i > j : \hat{a}_m(i,j) \neq 0)$
    Para $k = 1 : j-1$ y $\hat{a}_m(j,k) \neq 0$
      Para $i = j+1 : N$ y $\hat{a}_m(i,k) \neq 0$
        $\hat{a}_m(i,j) = \hat{a}_m(i,j) - \hat{a}_m(i,k)\hat{a}_m(j,k)$
      Terminar
    Terminar
    Para $i = j+1 : N$ y $\hat{a}_m(i,j) \neq 0$
      $\hat{a}_m(i,j) = \hat{a}_m(i,j)/\hat{a}_m(j,j)$
      $\hat{a}_m(i,i) = \hat{a}_m(i,i) - \hat{a}_m(i,j)^2$
    Terminar
    Retiene `col_len` $+ p$ entradas más grandes en $\hat{a}_m(j+1 : N, j)$
  Terminar
  `éxito = falso` $\implies \alpha_{m+1} = \mathrm{máx}(2\alpha_m, \alpha_s)$
  otro caso $\alpha_F = \alpha_m$
    `salir ciclo`
Terminar

Figura 4.6: Algoritmo de Lin y Moré (1999).

El llenado se controla a través del parámetro de densidad p: en cada etapa de la

factorización se retienen las `col_len` $+ p$ entradas más grandes en magnitud (incluyendo el elemento diagonal), donde `col_len` es igual al número de elementos no nulos de la columna. Notemos que este hecho nos ayuda a predecir la memoria necesaria para construir el precondicionador; un ejemplo numérico se expone en el apartado 4.8.

Análogamente que las estrategias de umbral, una FIC sin llenado no puede ser obtenida como caso particular. Para $p = 0$, las `col_len` entradas no nulas más grandes en cada columna de $\mathbf{L}$ no coinciden, en general, en posición con los `col_len` elementos no nulos en la columna de $\mathbf{A}$. Esto significa que, $\mathbf{A}$ y $\mathbf{L}$ no tienen el mismo patrón de *sparsidad*.

La técnica inicia escalando la matriz $\mathbf{A}$ por medio de $\widehat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, con $\mathbf{D} = \mathrm{diag}(\|\mathbf{A}\mathbf{e}_i\|_2)$. Después se perturba la matriz resultante $\widehat{\mathbf{A}}_m = \widehat{\mathbf{A}} + \alpha_m\mathbf{I}$, con el objeto de asegurar que exista la factorización y continúe siendo definida positiva, ver apartado 4.4. El parámetro de perturbación comienzan con $\alpha_0 = 0$, pero si en algún momento $\hat{a}_m(j,j) \leq 0$ entonces se reinicia la factorización usando $\alpha_{m+1} = \mathrm{máx}(2\alpha_m, \alpha_s)$, donde $\alpha_s = 10^{-3}$.

Finalmente, se utiliza un almacenamiento modificado por columnas, descrito en 4.6, para guardar la información de la triangular inferior $\mathbf{L}$.

4.4. Existencia, precisión y estabilidad de las FIC

4.4.1. Existencia de las FIC

Desafortunadamente, *no toda matriz SDP tiene una factorización incompleta de Cholesky*, esto es, la técnica puede "fallar" en presencia de pivotes nulos o negativos. Un ejemplo lo presenta Kershaw (1978) al mostrar la matriz:

$$\begin{pmatrix} 3 & -2 & 0 & 2 \\ -2 & 3 & -2 & 0 \\ 0 & -2 & 3 & -2 \\ 2 & 0 & -2 & 3 \end{pmatrix},$$

la cual es simétrica definida positiva, pero $l_{44} = -5$ para la FIC sin llenado. Es importante enfatizar que al resolver los sistemas SDP mediante PCG se requiere

que ambos, la matriz $\mathbf{A}$ y el precondicionador $\mathbf{M}$, sean definidos positivos.

El artículo de Meijerink y van der Vorst (1977) es un parte-aguas en lo referente a precondicionadores, ya que comprueban la existencia de la factorización incompleta para cualquier patrón de *sparsidad* K, si $\mathbf{A}$ es M-matriz. Diremos que la matriz de coeficientes $\mathbf{A}$ es *M-matriz* si: (1) $a_{ii} > 0$, para toda i; (2) $a_{ij} \leq 0$, para $i, j = 1, ..., N$, con $j \neq i$; y (3) $\mathbf{A}$ es no singular y además $\mathbf{A}^{-1} \geq 0$ $(a_{ij}^{-1} \geq 0$, para todo $i, j)$. Se garantiza la existencia de una factorización incompleta de Cholesky cuando $\mathbf{A}$ sea una M-matriz simétrica.

Manteuffel (1980) generaliza los resultados de Meijerink y van der Vorst a un clase más grande de matrices. Él muestra que una H-matriz con elementos diagonales positivos acepta una factorización incompleta. Además, si la matriz fuera simétrica se tendría una FIC. Diremos que la matriz $\mathbf{A}$ es *H-matriz* si su matriz de comparación $\mathcal{M}(\mathbf{A})$ definida por

$$\mathcal{M}(\mathbf{A}) = \begin{cases} |a_{ij}| & \text{si } i = j \\ -|a_{ij}| & \text{si } i \neq j \end{cases}, \tag{4.3}$$

es una M-matriz. Es claro que toda M-matriz es H-matriz, pues $\mathcal{M}(\mathbf{A}) = \mathbf{A}$.

Con la finalidad de mostrar (en el apartado 4.7) si las matrices asociadas a los sistemas (2.17) son o no M-matrices o H-matrices, enunciamos los resultados del siguiente teorema. Denotamos por $\mathscr{M}_N(\mathbb{R})$ al espacio lineal de todas las matrices $N \times N$ de valores reales.

Teorema 4.1. *Sea $\mathbf{A} \in \mathscr{M}_N(\mathbb{R})$ una matriz con elementos entradas de la diagonal no positivas. Los siguientes resultados son equivalentes:*

1. $\mathbf{A}$ *en una M-matriz.*
2. $\mathbf{A}$ *es no singular y* $\mathbf{A}^{-1} \geq 0$.
3. *Todos los valores propios de* $\mathbf{A}$ *son positivos.*
4. *Todos los menores principales de* $\mathbf{A}$ *son M-matrices.*
5. $\mathbf{A}$ *puede ser factorizada en la forma* $\mathbf{A} = \mathbf{LU}$, *donde* $\mathbf{L}$ *es un matriz triangular inferior,* $\mathbf{U}$ *es triangular superior, y los elementos diagonales de cada una de ellas son positivos.*

6. *Los elementos diagonales de* $\mathbf{A}$ *son positivos, y* $\mathbf{AD}$ *es estrictamente diagonalmente positiva para alguna matriz diagonal positiva* $\mathbf{D}$.

Demostración. Ver Greenbaum (1997), página 161. □

Además de la existencia de FI, Meijerink y van der Vorst (1977) muestran que los pivotes del proceso incompleto están acotados inferiormente por los pivotes exactos, y dado que los pivotes exactos de una M-matriz son positivos (teorema 4.1) se tiene que la descomposición incompleta es definida positiva. Este resultado también se extiende para H-matrices con elementos diagonales positivos (Manteuffel 1980). Diremos que una factorización incompleta es *definida positiva* si todos sus elementos diagonales son positivos.

En especial, las matrices diagonalmente dominantes y estrictamente diagonalmente dominantes son H-matrices (ver Meurant 1999). Este resultado motivó que a finales de los 70's, se propusieran diversas formas de perturbar la diagonal con el fin de asegurar la existencia, y por ende la definición positiva de la FIC.

Por ejemplo para matrices SDP, Kershaw (1978) sustituye localmente los pivotes negativos o nulos por

$$m_{kk} = \sum_{j<k} |m_{kj}| + \sum_{j>k} |m_{jk}|,$$

y continua con el proceso. En cambio, Manteuffel (1980) perturba globalmente la diagonal antes de iniciar la factorización empleando el parámetro $\alpha > 0$: $\widehat{\mathbf{A}} = \mathbf{A} + \alpha \mathrm{diag}(\mathbf{A})$. Es claro que si α^* es el mínimo valor para el cual $\widehat{\mathbf{A}}$ es diagonalmente dominante, la factorización incompleta de $\widehat{\mathbf{A}}$ existe. Sin embargo, Manteuffel indica que los mejores resultados son usualmente obtenidos para valores de α que son mucho más pequeños que α^* y más grandes que el más pequeño α para el cual $\widehat{\mathbf{A}}$ admite una descomposición incompleta. Cabe señalar que al utilizar esta estrategia, el valor de α adecuado se elige mediante prueba y error.

Lin y Moré (1999) muestran que la matriz escalada $\widehat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ con $\mathbf{D} = \mathrm{diag}(\|\mathbf{A}\mathbf{e}_i\|_2)$, al ser perturbada globalmente $\widehat{\mathbf{A}} + \alpha\mathbf{I}$ es una H-matriz con elementos diagonales positivos. Por su parte, Munksgaard (1980) también realiza una perturbación para asegurar que $\mathbf{A} + \alpha\mathrm{diag}(\mathbf{A})$ sea definida positiva, si $\mathbf{A}$ lo es.

Existen otras estrategias para garantizar la existencia de la factorización. Una de ellas es el método de compensación de la diagonal, propuesto por Axelsson y

Kolotilina (1994). Dicho método modifica la matriz SDP $\mathbf{A}$ en una M-matriz $\widehat{\mathbf{A}}$, calcula la FIC de $\widehat{\mathbf{A}}$ y la utiliza como precondicionador del sistema original. Recientemente, Benzi y Tůma (2003) propusieron no modificar la matriz $\mathbf{A}$, pero si reformular la factorización de tal manera que se eviten los pivotes nulos o negativos utilizando un método de A-ortogonalización.

4.4.2. Precisión y estabilidad de las FIC

Diremos que la *calidad* de un factorización incompleta $\mathbf{A} \approx \mathbf{LU}$ puede ser medida, en principio, por su *precisión* y *estabilidad*. La precisión se refiere a cuán cercanos están los factores $\mathbf{L}$ y $\mathbf{U}$ de la matriz exacta $\mathbf{A}$, y se mide por $N_1 = ||\mathbf{A} - \mathbf{LU}||_F$. En cambio, la estabilidad se refiere a que tan cercana la matriz precondicionada $(\mathbf{LU})^{-1}\mathbf{A}$ está de la matriz identidad $\mathbf{I}$, y puede medirse por $N_2 = ||I - (\mathbf{LU})^{-1}\mathbf{A}||_F$.

Para una clase de problemas simétricos, Duff y Meurant (1989) muestra que el número de iteraciones de PCG depende casi directamente del valor de N_1, es decir la calidad del precondicionador está prácticamente determinada por esta norma. En cambio, para matrices indefinidas o fuertemente no-simétricas, N_2 puede ser varios órdenes de magnitud más grande que N_1, y entonces la calidad del precondicionador dependerá de ambas medidas (ver Chow y Saad 1997, Benzi, Szyld y van Duin 1999). Es importante indicar que existen casos en donde las FIC tiene pivotes negativos o cercanos a cero, lo cual conduce a inestabilidades numéricas.

Las matrices diagonalmente dominantes o bien condicionadas son estables. Debido a esto, algunas estrategias para estabilizar las FI consisten en permutar, escalar y/o perturbar la matriz, ya sea local o globalmente con el fin de aumentar la dominancia diagonal de $\mathbf{A}$. Por ejemplo van der Vorst (1981) propone, para matrices no simétricas, modificar los pivotes nulos o muy pequeños con

$$m_{kk} = \sum_{j<k} |m_{kj}| + \sum_{j>k} |m_{kj}|.$$

La descomposición resultante se llama “factorización incompleta estabilizada”.

Munksgaard (1980) estabiliza la FIC evitando los pivotes negativos o cercanos

a cero mediante la siguiente regla:

$$\text{Si} \quad m_{kk} \leq u\big(\max_{j=k+1,\ldots,N} |a_{kj}^{(k)}|\big), \quad \text{entonces} \quad m_{kk} = \max_{j=k+1,\ldots,N} |a_{kj}^{(k)}|;$$

si $m_{kk} \leq 0$ y $\max_{j=k+1,\ldots,N} |a_{kj}^{(k)}| = 0$, entonces $m_{kk} = 1$. Esta regla asegura que el tamaño de los elementos de los factores en cada etapa k estén acotados por $(1+u^{-1})$, donde $u = 0.01$. Esta idea también es utilizada para estabilizar factorizaciones completas, ver Duff et al. (1992).

Chow y Saad (1997) realizan un estudio completo sobre este tópico para matrices indefinidas. Además proponen una perturbación dinámica para estabilizar el precondicionador. Recientemente, Wang y Zhang (2003) han presentado una estrategia en donde se construye dos veces la descomposición de una matriz perturbada, obteniendo así una factorización incompleta ILU estable y precisa para las matrices indefinidas originadas por las aplicaciones en dinámica de fluidos computacional.

4.5. Ordenamiento nodal

Es preciso que en las descomposiciones, ya sean completas o incompletas, se minimice el llenado, puesto que a su vez se minimizará los requerimientos de memoria, así como el número de operaciones. La manera de reducir el llenado es realizando un reordenamiento de las entradas no nulas de la matriz. Dependiendo de las propiedades de la matriz (después de ensamblarla) el ordenamiento se puede realizar de dos formas: la primera es determinando el número de índices del llenado antes de realizar la factorización, cuando las propiedades sólo dependen de la estructura y no de los valores de las entradas, como es el caso de las matrices simétricas definidas positivas; esto se hace en una etapa de preproceso llamada *factorización simbólica*. En la segunda forma, el ordenamiento se efectúa al mismo tiempo que la factorización, ver por ejemplo Meurant (1999), D'Azevedo et al. (1992), y Saint-Georges, Warzee, Beauwens y Notay (1996).

Una tercera forma de reordenar se lleva a cabo a nivel de malla, en este caso se denomina *ordenamiento nodal* (Gambolati, Pini y Ferronato 2001). La idea de esta técnica es reordenar la numeración global de los nodos, de tal manera que reduzca el ancho de banda de la matriz asociada. En este caso, son válidas las estrategias

diseñadas para las factorizaciones simbólicas, las cuales se pueden consultar en Meurant (1999) y Saad (2003).

En este estudio hemos trabajado con un ordenamiento nodal utilizando el algoritmo de Cuthill-Mckee inverso (Saad 2003), con el fin de reducir el llenado del método directo. Como es discutido en Duff y Meurant (1989), esta estrategia también se recomienda emplearla cuando se usa el método de CG precondicionado con factorizaciones incompletas de Cholesky, especialmente para el caso de mallas no estructuradas de elementos finitos.

Teniendo en cuenta el método de descomposición de dominios, ver capítulo 6, hemos restringido el reordenamiento para respetar la típica estructura *bloque-interfase* de muchos filtros de carbón activo (una de nuestras aplicaciones tecnológicas).

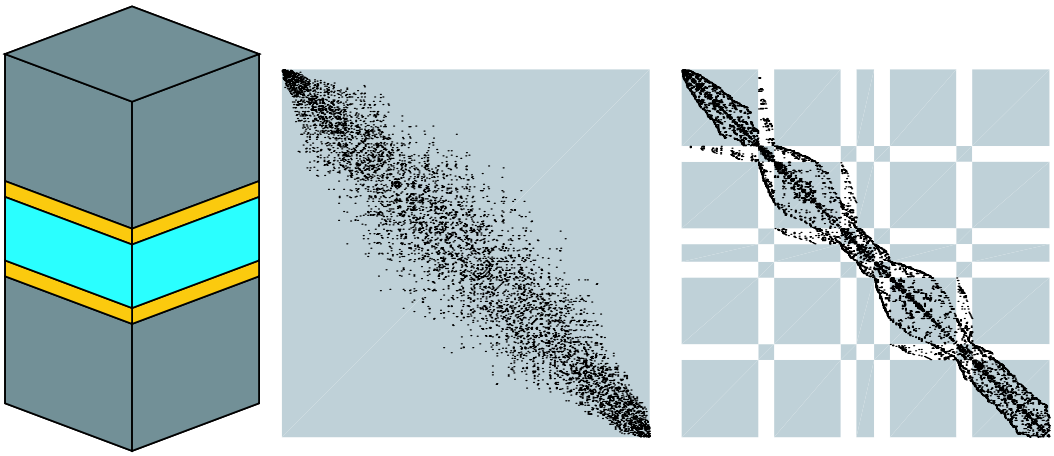


Figura 4.7: Reordenación nodal: (a) típico domino bloque-interfase, (b) matriz antes del reodenamiento y (c) matriz después de ordenar.

La figura 4.7.a muestra un filtro de carbón activo formado por 5 elementos: dos cámaras de carbón activo (extremos) con una cámara de aire intermedia y dos espumas. Las gráficas de las matrices sin y con ordenamiento se presentan en 4.7.b y 4.7.c, respectivamente. En el inciso (c) se visualiza un cuadrícula en gris y blanco como fondo a la gráfica de la matriz. Notemos que las 5 columnas y 5 filas de cuadriláteros grises están asociadas a los 5 bloques del filtro. Análogamente, las columnas y filas en blanco se relacionan con las 4 interfases del *canister*.

En el apartado 4.7.1 se muestra mediante ejemplos numéricos, cómo influye este ordenamiento en el ahorro de memoria y tiempo de CPU.

4.6. Esquemas de almacenamiento

En la literatura se puede encontrar una gran variedad de esquemas de almacenamientos para matrices, así como algunas subrutinas para cambiar dichos esquemas. Una buena referencia es la librería SPARSKIT (Saad 1994*b*).

En nuestro caso se han utilizado diversos esquemas. En el método de Cholesky (versión jik, figura 3.1), se ha empleado un *skyline* para almacenar la matriz triangular superior $\mathbf{DU}$. En cambio, para los algoritmos de Munksgaard (versión kji, figura 4.4) y de Lin y Moré (versión jki, figura 4.6) se ha usado, para guardar la matriz triangular inferior, almacenamientos en *coordenadas simétricas* y *modificado por columnas*, respectivamente.

El *almacenamiento en Skyline* (por columnas) emplea dos vectores; el primero, `AMAT_A`, guarda las columnas de la triangular superior de $\mathbf{A}$ comenzando en cada columna desde el primer elemento no nulo hasta la diagonal. El segundo vector, `DIAG_A` contiene las posiciones de los elementos diagonales dentro de `AMAT_A`.

El *almacenamiento en coordenadas simétricas (SCS)* consiste en tres vectores de igual tamaño: `A_vec` contiene las entradas no nulas de la triangular inferior $\mathbf{A}$, y los otros dos vectores, `ifil_vec` y `icol_vec`, incluyen los índices de las filas y columnas de los elementos en `A_vec`. Este esquema tiene el inconveniente de realizar operaciones en aritmética de enteros para calcular los índices de las columnas y filas; además puede conducir a un error de "overflow" para matrices grandes.

El *almacenamiento modificado por columnas (MCS)* lo conforman cuatro vectores: `AMAT_A` guarda por columnas los elementos no nulos de la matriz $\mathbf{A}$; `DIAG_A` almacena las entradas diagonales; `Afil_vec` contiene los índices de las filas de cada componente de `AMAT_A` y `Acol_vec` acumula los punteros que indican la posición en `AMAT_A` del primer elemento de cada columna.

4.7. Aplicación: filtros de carbón activo

El objetivo principal de este apartado es comparar la eficiencia de dos familias de factorizaciones incompletas de Cholesky (de umbral y de memoria prescrita) como precondicionadores del método de gradientes conjugados. Para poner en contexto y como referencia, incluiremos en el análisis al precondicionador de Jacobi y el esquema estándar de CG. También probaremos el método directo de Cholesky: desde el punto de vista de aplicaciones industriales, los métodos iterativos (precondicionados o no) serán preferidos a los directos, si son realmente mucho más eficientes. En las comparaciones del coste computacional se considerarán los requerimientos de memoria (número de entradas no nulas en el factor incompleto $\mathbf{L}$) y el tiempo de CPU.

Los criterios de parada para los métodos iterativos (figura 3.2) son:

$$\left|x_i^k - x_i^{k+1}\right| \leq tol_x\big(\left|x_i^{k+1}\right| + 1\big), \qquad \forall i = 1, 2, ..., N; \tag{4.4}$$

$$\left\|\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}\right\|_2 \leq tol_r \left\|\mathbf{b}\right\|_2, \tag{4.5}$$

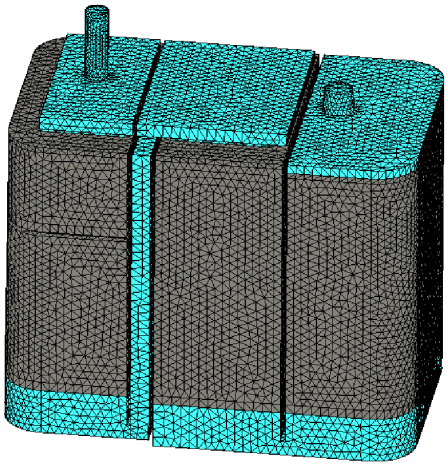
con $tol_x = 0.5 \cdot 10^{-10}$ y $tol_r = 0.5 \cdot 10^{-9}$. Además, la solución final del paso de tiempo previo se utiliza como solución inicial $\mathbf{x}^0$.

Todos los algoritmos están implementados en Fortran 90 usando aritmética en doble precisión y han sido compilados con la opción de optimización -Ofast=ip35. Las pruebas se han ejecutado sobre un procesador de una SGI Origin 3000 con ocho procesadores a 600 Mhz y 8 Gb de RAM. Para el postproceso se ha utilizado Matlab, versión 6.1.

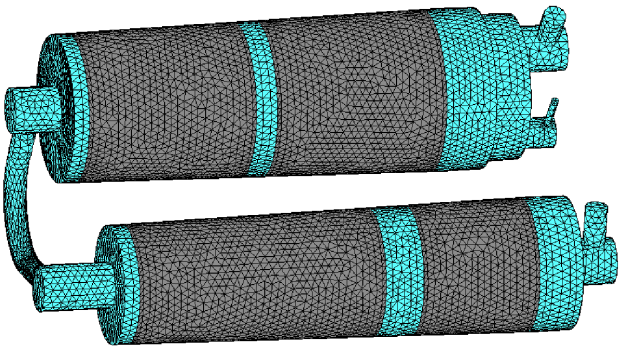
En nuestro análisis computacional trabajamos con los filtros de carbón activo de la figura 4.8. Para cada uno de esos filtros hemos empleado dos mallas no estructuradas de elementos finitos, denotadas por: “gruesa” y “fina”. Para el filtro A también usamos una tercera malla “más gruesa”.

La tabla 4.1 resume los principales parámetros numéricos. Para cada filtro y malla se presenta: (1) el número de nodos, (2) el número de elementos finitos, (3) N corresponde al tamaño de la matriz $\mathbf{A}$, (4) $\mathrm{nnz}(\mathbf{A})$ representa el número de elementos no nulos de la parte triangular superior de la matriz $\mathbf{A}$, (5) el incremento de tiempo Δt y (6) el número de pasos de tiempo.

Filtro A, malla fina



Filtro B, malla fina



Filter C, malla gruesa

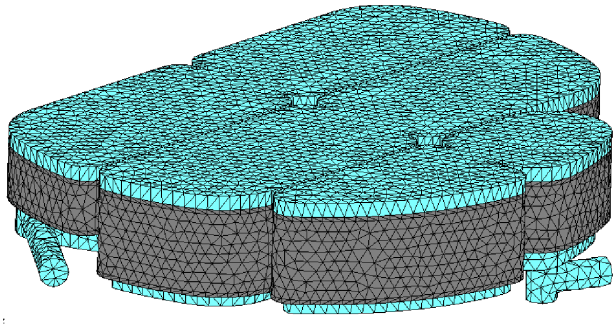


Figura 4.8: Filtros de carbón activo

	Filtro A			**Filtro B**		**Filtro C**	
	Más gruesa	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
# nodos	5 186	17 943	87 502	48 138	84 087	33 041	65 941
# elem.	19 081	74 139	460 765	249 645	454 815	167 558	346 907
N	5 171	17 914	87 473	48 125	84 072	33 028	65 925
$\mathrm{nnz}(\mathbf{A})$	32 928	117 181	653 435	357 251	636 669	242 793	493 278
Δt	5	8	4.31	4	3.6	8.59	6.82
# pasos	23 490	17 961	32 867	20 719	23 152	11 135	13 974

Tabla 4.1: Principales parámetros numéricos en las simulaciones de los filtros.

Notemos que el número de nodos es mayor que N, debido a que las matrices no incluyen las condiciones de contorno tipo Dirichlet. También observemos que el incremento de tiempo Δt es seleccionado en concordancia con el tamaño del elemento h, esto se debe a que el número de Courant (C) debe ser cercano a 1, ver apartado 2.5.

	Filtro A		**Filtro B**		**Filtro C**	
	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
$\lambda_{\text{máx}}$	$2.68 \cdot 10^{-4}$	$5.11 \cdot 10^{-5}$	$1.02 \cdot 10^{-4}$	$4.39 \cdot 10^{-5}$	$2.42 \cdot 10^{-4}$	$1.15 \cdot 10^{-4}$
$\lambda_{\text{mín}}$	$8.87 \cdot 10^{-10}$	$5.65 \cdot 10^{-10}$	$4.11 \cdot 10^{-9}$	$4.08 \cdot 10^{-9}$	$1.30 \cdot 10^{-9}$	$1.35 \cdot 10^{-9}$
$\kappa(\mathbf{A})$	$3.02 \cdot 10^{5}$	$9.04 \cdot 10^{4}$	$2.48 \cdot 10^{4}$	$1.08 \cdot 10^{4}$	$1.86 \cdot 10^{5}$	$8.52 \cdot 10^{4}$

Tabla 4.2: Matrices definidas positivas: $\lambda_{\text{máx}}$ es el máximo valor propio, $\lambda_{\text{mín}}$ es el mínimo valor propio y $\kappa(\mathbf{A}) = \lambda_{\text{máx}}/\lambda_{\text{mín}}$ es el número de condición

La tabla 4.2 muestra el máximo y mínimo valor propio para cada matriz resultante $\mathbf{A}$, así como el número de condición definido en la ecuación (3.8). Dado que todos los mínimos valores propios son positivos concluimos que los sistemas en todos los ejemplos son definidos positivos. De hecho, nuestros experimentos numéricos indican que la condición de Courant es más restrictiva en la selección de Δt que el requerimiento de ser SDP.

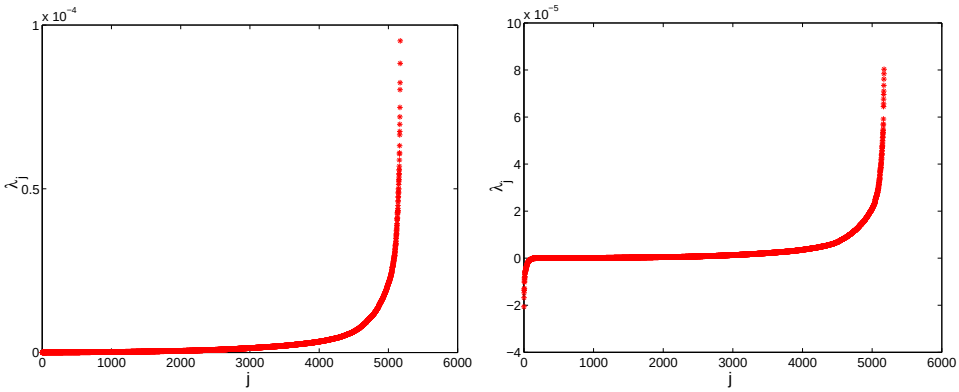
También advertimos, por los valores de $\kappa(\mathbf{A})$, que las matrices no están bien condicionadas, ver tabla 4.2. De hecho, tales valores indican que la velocidad de convergencia de CG será lenta, ver relación (3.7).

Antes de continuar con el análisis computacional, será interesante mostrar que

las matrices en estudio no son ni M-matrices, ni tampoco H-matrices. Para ello utilizamos el filtro A, malla más gruesa. Siendo así, las primeras entradas no diagonales de la matriz $\mathbf{A}$ son:

$$a_{1,2} = 2.19 \cdot 10^{-06} \quad ; \quad a_{2,1} = 2.19 \cdot 10^{-06} \quad ; \quad a_{1,7} = 8.46 \cdot 10^{-07} \quad ; \quad a_{7,1} = 8.46 \cdot 10^{-07} \quad ;$$
$$a_{1,3} = 2.11 \cdot 10^{-06} \quad ; \quad a_{3,1} = 2.11 \cdot 10^{-06} \quad ; \quad a_{1,9} = -4.57 \cdot 10^{-06} \quad ; \quad a_{9,1} = -4.57 \cdot 10^{-06} \quad ;$$
$$a_{1,4} = 1.95 \cdot 10^{-06} \quad ; \quad a_{4,1} = 1.95 \cdot 10^{-06} \quad ; \quad a_{1,10} = -3.00 \cdot 10^{-06} \quad ; \quad a_{10,1} = -3.00 \cdot 10^{-06} \quad .$$

Claramente, no todas las $a_{ij} \leq 0$; por definición $\mathbf{A}$ no es M-matriz, ver apartado 4.4.



(a) (b)

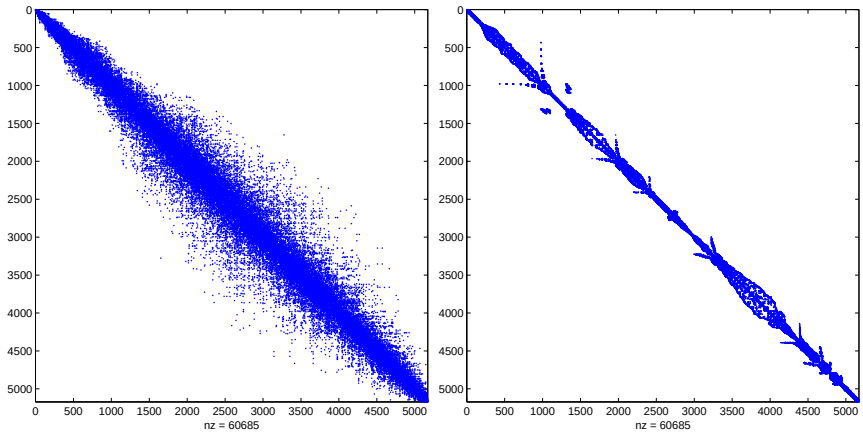
Figura 4.9: Filtro A, malla más gruesa. Gráficas de los valores propios de las matrices: (a) $\mathbf{A}$ y (b) $\mathcal{M}(\mathbf{A})$.

La figura 4.9.b presenta el espectro de la matriz de comparación definida en (4.3). Como existen algunos valores propios negativos por el teorema 4.1, $\mathcal{M}(\mathbf{A})$ no es M-matriz. Por tanto $\mathbf{A}$ tampoco es H-matriz.

En consecuencia, no podemos asegurar teóricamente que para las matrices en estudio exista una FIC. En el apartado 4.7.3 se muestra que hay casos en donde es necesario perturbar globalmente la matriz A para obtener una FIC de memoria prescrita.

4.7.1. Efecto del ordenamiento nodal

El efecto del ordenamiento nodal se evalúa mediante la simulación del filtro A malla más gruesa. En la figura 4.10 se muestra el patrón de *sparsidad* de la matriz $\mathbf{A}$: (a) antes de la reordenación y (b) después del ordenamiento nodal. Notemos



(a) (b)

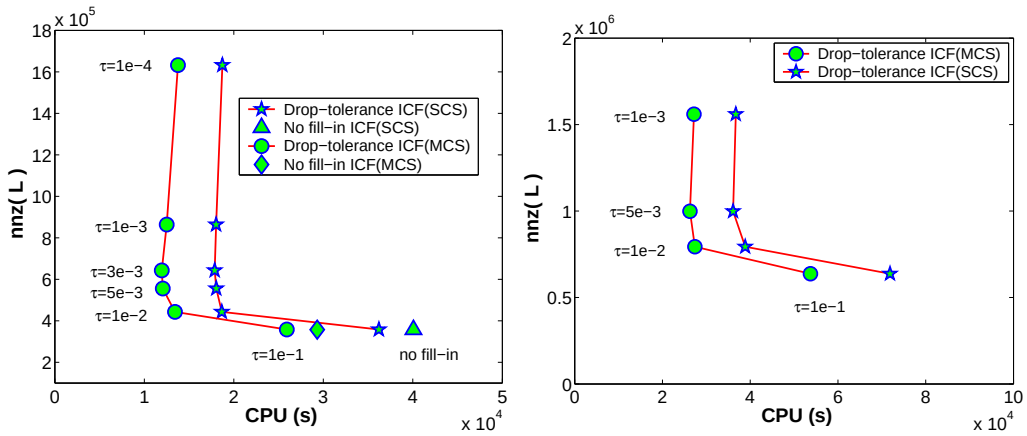
Figura 4.10: Patrón de *sparsidad* para la matriz del filtro A malla más gruesa: (a) antes de reordenar y (b) después del ordenamiento nodal

	Cholesky		
	Antes (s)	Después (s)	Reducción (%)
Factorización	2.3	0.6	73.9
Solución	896.6	213.8	76.2
TOTAL	898.9	214.4	76.2

	FIC de umbral					
	$\tau = 10^{-2}$			$\tau = 10^{-4}$		
	Antes (s)	Después (s)	Red. (%)	Antes (s)	Después (s)	Red. (%)
Precon.	0.4	0.4	0.0	1.5	1.6	-6.7
Solución	723.5	726.1	-3.6	568.2	608.6	-7.1
TOTAL	723.9	726.5	-3.6	569.7	610.2	-7.1

	FIC de memoria prescrita					
	$p = 2$			$p = 7$		
	Antes (s)	Después (s)	Red. (%)	Antes (s)	Después (s)	Red. (%)
Precon.	0.0	0.0	0.0	0.0	0.0	0.0
Solución	919.9	920.1	0.0	776.1	796.8	-2.7
TOTAL	919.9	920.1	0.0	776.1	796.8	-2.7

Tabla 4.3: Efecto de la reordenación en la simulación del filtro A malla más gruesa. Tiempos de CPU antes y después de reordenar, y el porcentaje de reducción.



(a) (b)

Figura 4.11: Efecto del esquema de almacenamiento en la simulación del filtro B: (a) malla gruesa y (b) malla fina.

que el perfil de la matriz cambia; el ancho de banda para el primer caso es de 3 249 y con ordenamiento es de 1 093, es decir, se reduce un 66 %.

Para examinar cómo repercute el ordenamiento nodal hemos realizado diversas pruebas, antes y después de reordenar. Como método directo, hemos empleado la descomposición completa de Cholesky, y como método iterativo PCG con varios precondicionadores: FIC de umbral (figura 4.4) con $\tau = 10^{-2}$ y $\tau = 10^{-4}$, y FIC de memoria prescrita (figura 4.6) con $p = 2$ y 7. La tabla 4.3 resume los tiempos de CPU para las pruebas antes mencionadas. Notemos que el ordenamiento nodal repercute muy favorablemente en el método directo (reduce más del $75\,\%$ el tiempo de CPU), mientras que afecta ligeramente la eficiencia de las FIC. En todos los siguientes experimentos numéricos se contempla tal reordenación.

4.7.2. Efectos del cambio de almacenamiento

Con el objeto de detectar cómo afecta el cambio de almacenamiento se considera el filtro B con una malla gruesa y una fina, y el algoritmo de umbral (figura 4.4). En la figura 4.11 comparamos el coste computacional del esquema SCS originalmente usado por Munksgaard (1980) y el esquema MCS, ver apartado 4.6. En ésta y las siguientes figuras, el coste computacional es representado por el tiempo de CPU (en segundos) sobre el eje x y el número de entradas no nulas del factor $\mathbf{L}$ sobre el eje

	A		**B**		**C**	
	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
$p = 0$	0	0	0	0	0.001	0
$p = 1$	0	0	0	0	0	0
$p = 2$	0.016	0.004	0	0	0	0
$p = 3$	0	0	0	0	0	0
$p = 4$	0	0	0.016	0	0	0
$p = 5$	0.002	0	0.008	0	0	0
$p = 6$	0.004	0	0.016	0	0	0
$p = 7$	0	0	0	0	0	0
$p = 12$	0	0	0	0	0	0
$p = 26$	0	0	0	0	0	0

Tabla 4.4: Valores de la perturbación α para los tres filtros con ambas mallas

y.

En ambas gráficas de la figura 4.11 se advierte inmediatamente una reducción significativa del tiempo de CPU (más del $24\,\%$) al emplear el esquema MCS. Esto se debe a que las sustituciones hacia arriba y hacia abajo se resuelven más rápidamente con MCS que empleando coordenadas simétricas. Por esta razón, en los siguientes apartados usaremos el formato MCS para ambas FIC (de umbral y memoria prescrita).

4.7.3. Perturbación global de la matriz

Como se explica en los apartados 4.3 y 4.2, ambas familias de FIC incluyen la opción de perturbar globalmente la matriz con el fin de asegurar la existencia de una factorización incompleta. Sin embargo, esta opción no se ha utilizado en la construcción de las FIC tipo umbral.

La tabla 4.4 contiene una recopilación de los valores del parámetro de perturbación α requeridos en cada una de las pruebas al emplear una FIC de memoria prescrita. Notemos que al aumentar el llenado (aumentando p) no se asegura la existencia de la FIC y es necesario perturbar la matriz original. En cambio, para las malla finas (excepto un caso) no se requiere perturbar globalmente $\mathbf{A}$. Una explicación a este fenómeno podría ser, que la matriz $\mathbf{A}$ comienza a ser diagonalmente

	A		B		C	
	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
%fdd	58.0	56.1	45.8	36.8	44.5	48.1

Tabla 4.5: Porcentajes de filas diagonalmente dominantes (%fdd) para los tres filtros con ambas mallas

dominante a medida que se refina la malla. Lo cual es falso, basta con calcular el *porcentaje de filas diagonalmente dominantes* (%fdd) contenidos en $\mathbf{A}$ (este parámetro se utiliza en Sosonkina et al. (2000)). La tabla 4.5 muestra los porcentajes para los tres filtros con ambas mallas. Observemos en los filtros A y B que el porcentaje de filas diagonalmente dominantes disminuye cuando se refina la malla. En consecuencia queda abierta la explicación del por qué no se necesita perturbar la matriz original cuando se refina la malla.

4.7.4. Eficiencia computacional de los métodos directos e iterativos

En este subapartado se evalúa la eficiencia numérica de los métodos directos e iterativos presentados en este trabajo. Para ello, usamos las simulaciones de los filtros descritos en la tabla 4.1.

Gradientes conjugados estándar versus PCG

Comencemos con un análisis global. Para ello, se estudia el filtro B con todos los métodos discutidos: Cholesky completo, CG estándar y PCG con los tres diferentes precondicionadores: el diagonal, la FIC de umbral y la de memoria prescrita.

Los resultados se presentan en la figura 4.12. Claramente se aprecia la necesidad de precondicionar el método iterativo. Basta con reparar en el precondicionador diagonal, que provoca una reducción significativa en el tiempo de CPU (por un factor mayor que 4) con respecto al CG estándar. Además, no requiere un almacenamiento adicional.

Al reparar nuevamente en la figura 4.12, también nos percatamos del relevante

ahorro de memoria de las FIC con respecto a Cholesky completo. Aspecto que tratamos con mayor detalle a continuación.

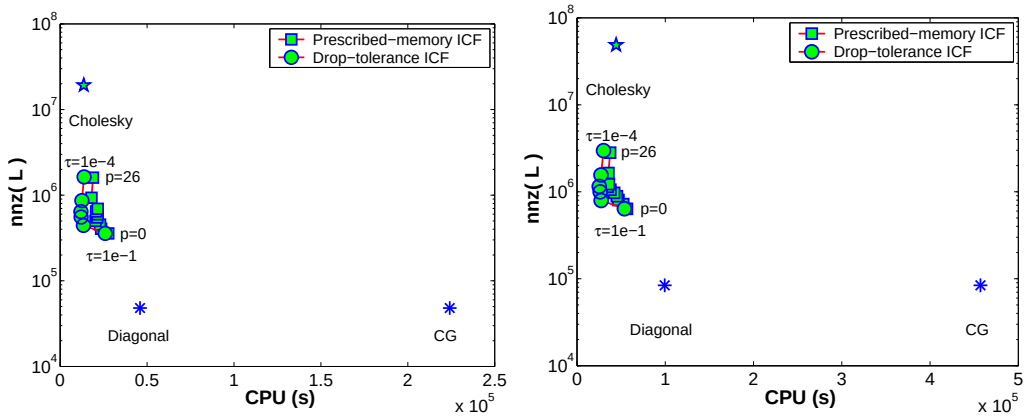


Figura 4.12: Coste computacional de todos las estrategias para la simulación del filtro B: (a) malla gruesa y (b) malla fina.

Estrategias directas versus iterativas

Ahora se comparan el método directo de Cholesky y el método iterativo PCG con diversos precondicionadores: diagonal, FIC sin llenado, y las familias de FIC de umbral y memoria prescrita. Estas pruebas se realizan para los tres filtros y los resultados se muestran en la figura 4.13.

Al examinar las gráficas apreciamos los siguientes aspectos:

- Si las restricciones de memoria admiten su uso, el método de Cholesky es más rápido que la iteración de CG con un sencillo precondicionador, es decir, el diagonal o una FIC sin llenado. Se requiere de un mejor precondicionador para superar al método directo.

- Permitiendo algo de llenado en las familias de FIC, se reduce considerablemente el tiempo de CPU de la FIC sin llenado a cambio de un modesto incremento en las entradas no nulas.

- *Las mejores FIC superan al método directo tanto en requerimientos de memoria como en tiempo de CPU.*

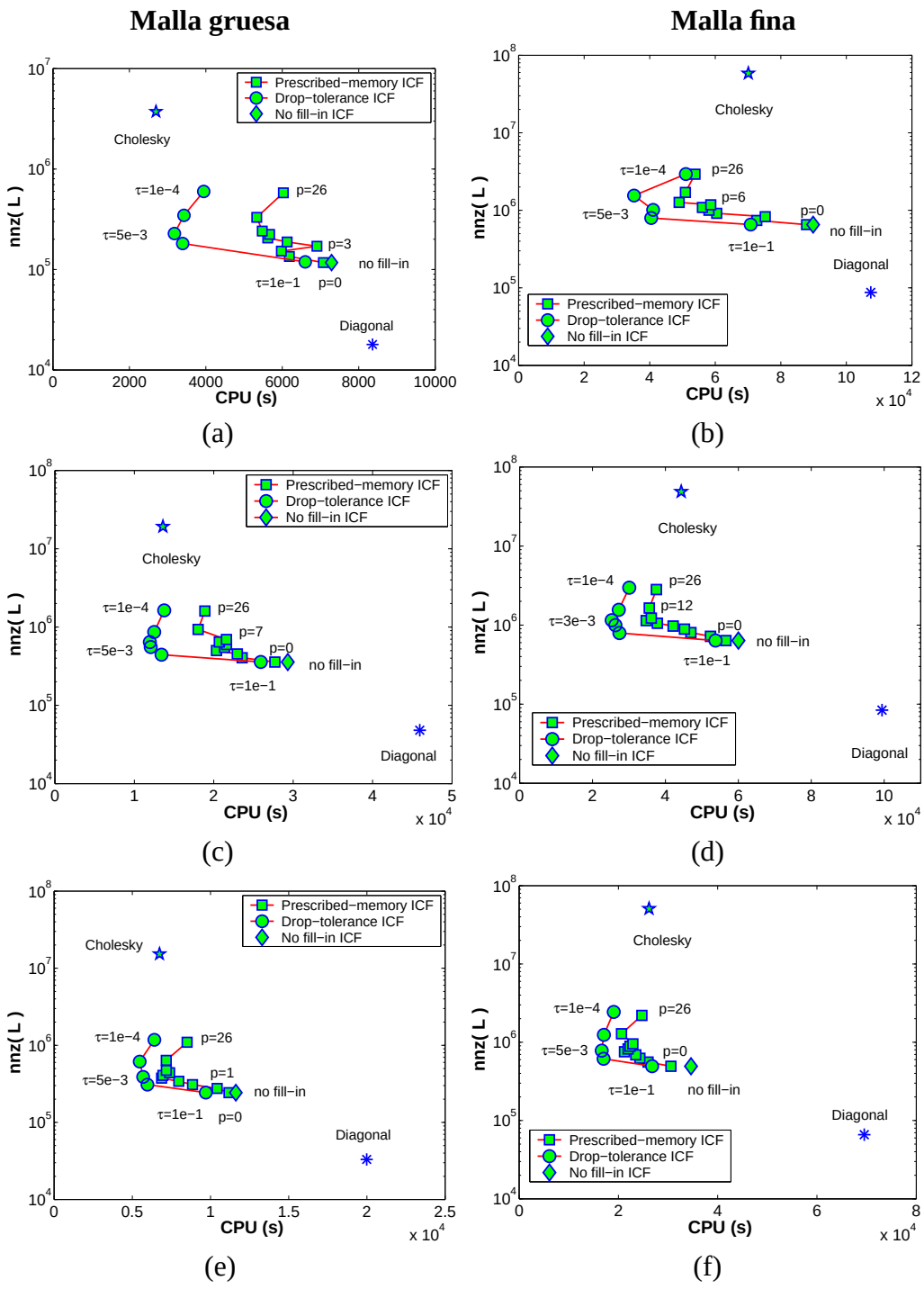


Figura 4.13: Coste computacional de Cholesky y PCG para la simulación de los tres filtros: (a) filtro A, malla gruesa, (b) filtro A, malla fina, (c) filtro B, malla gruesa, (d) filtro B, malla fina, (e) filtro C, malla gruesa, y (f) filtro C, malla fina.