

Figura 5.13: Gráficas de iteraciones vs. pasos de tiempo para los filtros B (arriba) y C (abajo), mallas finas: (a) y (c) M_{low-up} ; (b) y (d) SSPAI.

Ahora comparamos las inversas aproximadas M_{low-up} y SSPAI con las FIC sin llenado y de umbral. Para esta última hemos elegido el parámetro $\tau = 0.005$ que produce el mejor resultado (en cuanto a coste computacional) para los problemas de convección-difusión transitorios que estamos analizando, ver capítulo 4. Las tablas 5.5, 5.6 y 5.7 resumen los resultados de los filtros A, B y C, respectivamente. Para cada una de las mallas se probaron diferentes formas de llenado variando los parámetros ϵ , n_k , s , t . En general encontramos que es adecuado seleccionar $s = 10$ índices candidatos óptimos cada que se realiza una búsqueda de entradas candidatas.

Al contrastar M_{low-up} con SSPAI nos damos cuenta que el número de iteraciones aumenta si se paraleliza la estrategia 3, y que el número de no nulos de SSPAI es igual o ligeramente mayor que los de M_{low-up} (esto también se aprecia en la figura 5.12). Si ahora comparamos los resultados de la SSPAI y las FIC sin llenado y de umbral, nos percatamos que los requerimientos de memoria siempre son

Gruesa	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
$\ A \bullet - I\ _F$	33.6	31.5	40.3	37.9		
ϵ	0.3	0.3	0.3	0.3		
n_k	30	30	30	30		
t	10	15	10	15		
$\frac{nnz(\bullet)}{nnz(A)}$	1.18	1.47	1.19	1.48	1.00	1.95
P-time	906.1	972.1	895.2	953.2	0.1	0.2
Its-time	6385.4	6460.8	8287.4	8016.6	4784.7	1617.6
T-time	7291.5	7432.9	9182.6	8969.8	4784.8	1617.8
$\frac{Its}{pasos}$	99.9	91.3	125.3	109.0	60.1	13.3
$\frac{T-time}{n_b/2}$			1311.8	1281.4		

Fina	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
$\ A \bullet - I\ _F$	47.0	40.1	47.8	41.4		
ϵ	0.3	0.3	0.3	0.3		
n_k	30	30	30	30		
t	10	15	10	15		
$\frac{nnz(\bullet)}{nnz(A)}$	0.85	1.17	0.85	1.17	1.00	1.55
P-time	22645.0	27603.1	24549.1	29425.2	0.7	1.4
Its-time	27305.1	27444.6	31755.8	34602.4	31477.8	13807.0
T-time	49950.1	55047.7	56304.9	64027.6	31478.5	13808.4
$\frac{Its}{pasos}$	47.0	41.1	54.2	52.4	32.3	10.0
$\frac{T-time}{n_b/2}$			8043.6	9146.8		

Tabla 5.5: Filtro A, mallas gruesa y fina.

menores para el preconditionador explícito. De hecho, con $nnz(SSPAI) \leq nnz(A)$ se consiguen resultados ventajosos. Además, es evidente que bajo un procesamiento secuencial las FIC(0.005) superan a las inversas aproximadas, debido a que es costosa la fase de construcción de las últimas. Sin embargo, si paralelizáramos los cálculos (el tiempo total de CPU de la SSPAI entre $n_B/2$) obtendríamos lo contrario, excepto para el filtro C malla gruesa. Recordemos que ni la FIC ni su aplicación en cada etapa de preconditionamiento se pueden paralelizar en forma eficiente.

De los experimentos presentados a lo largo de este apartado podemos destacar los siguientes aspectos:

- La fase de aplicación del preconditionador parte-simétrica a menudo es cara, sobre todo cuando se requiere aumentar el llenado. Además no se cuenta con una relación eficiente para evaluar qué tan cercano está AM_{sym-p} de la identidad I .

Gruesa	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
	$\ A \bullet - I\ _F$					
ϵ	0.30	0.18	0.30	0.18		
n_k	30	50	30	50		
t	10	10	10	10		
$\frac{nnz(\bullet)}{nnz(A)}$	0.86	1.15	0.86	1.18	1.00	1.56
P-time	6147.8	9687.0	5563.2	9841.2	0.3	0.7
Its-time	11784.0	13530.4	11453.5	11175.0	12457.6	5823.5
T-time	17931.8	23217.4	17016.7	21016.2	12457.9	5824.2
$\frac{Its}{pasos}$	58.2	60.2	58.3	50.0	40.6	12.0
$\frac{T-time}{n_b/2}$			2127.1	2627.0		

Fina	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
	$\ A \bullet - I\ _F$					
ϵ	0.3	0.3	0.3	0.3		
n_k	30	50	30	50		
t	10	10	10	10		
$\frac{nnz(\bullet)}{nnz(A)}$	0.83	0.85	0.83	0.85	1.00	1.57
P-time	20989.7	21329.4	19538.9	20953.6	0.6	1.6
Its-time	21307.7	20491.0	21344.6	21457.6	23859.9	10514.3
T-time	42297.4	41820.4	40883.5	42411.2	23860.5	10515.9
$\frac{Its}{pasos}$	55.2	52.3	54.4	52.3	33.6	10.4
$\frac{T-time}{n_b/2}$			5110.4	5301.4		

Tabla 5.6: Filtro B, mallas gruesa y fina.

- Construir una inversa aproximada *sparse* simétrica atendiendo únicamente a la información de la triangular inferior, produce un preconditionador de baja calidad.
- Al incorporar las entradas significativas de la triangular superior a través del parámetro t , el preconditionador M_{low-up} gana en calidad, pero pierde precisión.
- Se necesita paralelizar el algoritmo de la inversa aproximada *sparse* simétrica para competir con las factorizaciones incompletas de Cholesky.
- La fase de construcción de la SSPAI es mucho más costosa que generar una FIC, pero requiere un menor nivel de llenado que la mejor FIC.
- El promedio de iteraciones de la IA es mayor que el de la FIC. Sin embargo,

Gruesa	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
	$\ \mathbf{A} \bullet - \mathbf{I}\ _F$	60.5	93.3	67.6	62.1	
ϵ	0.3	0.3	0.3	0.3		
n_k	30	40	30	40		
t	10	15	10	15		
$\frac{\text{nnz}(\bullet)}{\text{nnz}(\mathbf{A})}$	1.03	1.39	1.04	1.42	1.00	1.60
P-time	3511.5	4365.2	3358.7	4290.9	0.2	0.4
Its-time	33543.8	37434.2	50032.9	48189.3	4745.4	2240.3
T-time	37055.3	41799.4	53391.6	52480.2	4745.6	2240.7
$\frac{Its}{pasos}$	443.0	434.9	657.9	554.5	45.2	14.5
$\frac{T-time}{n_b/2}$			3813.7	3748.6		

Fina	M_{low-up}		SSPAI		FIC(∞)	FIC(0.005)
	$\ \mathbf{A} \bullet - \mathbf{I}\ _F$	55.3	51.3	63.0	58.9	
ϵ	0.3	0.3	0.3	0.3		
n_k	30	30	30	30		
t	10	15	10	15		
$\frac{\text{nnz}(\bullet)}{\text{nnz}(\mathbf{A})}$	0.91	1.22	0.92	1.22	1.00	1.58
P-time	15361.9	15217.9	12607.8	14169.4	0.5	1.1
Its-time	29536.0	30341.5	72746.5	74755.0	12814.5	7260.4
T-time	44897.9	45559.4	85354.3	88924.4	12815.0	7261.9
$\frac{Its}{pasos}$	157.6	146.8	384.1	358.7	42.6	15.6
$\frac{T-time}{n_b/2}$			6096.8	6351.7		

Tabla 5.7: Filtro C, mallas gruesa y fina.

existen algunos casos en donde el tiempo de CPU de la aplicación iterativa de la SSPAI es menor que para la FIC(∞) en un procesamiento secuencial (ver tabla 5.6).

- La paralelización de la fase de construcción de la SSPAI está limitada al número de bloques. No obstante, su etapa de aplicación se puede paralelizar masivamente.
- En los problemas de convección-difusión transitorios se obtiene, comparando los costes computacionales, que la SSPAI sería más eficiente que la mejor FIC en paralelo.

5.5. Conclusiones

Hemos desarrollado una inversa aproximada *sparse* simétrica (SSPAI) basada en la SPAI de Montero et al. El preconditionador está diseñado para resolver sistemas SDP, tridiagonales por bloques con múltiples lados derechos mediante el método de gradientes conjugados preconditionado. También se establecieron fórmulas explícitas para calcular en forma eficientes y acumulativas tanto \mathbf{m}_k^J como $\|\mathbf{A}\mathbf{m}_k^J - \mathbf{v}_k\|_F^2$.

De nuestros experimentos numéricos concluimos que para generar una inversa aproximada simétrica de calidad, no solamente se requiere considerar la información de la parte triangular inferior, sino también las entradas significativas de la triangular superior. También observamos que es importante paralelizar el preconditionador explícito para poder competir con las factorizaciones incompletas de Cholesky (FIC); aún así, la fase de construcción del primero sería más costosa. De los diferentes problemas de validación inferimos que es necesario estabilizar la SSPAI. Para los problemas de convección-difusión transitorios con campo de velocidad constante y en donde la etapa de preconditionamiento es un factor relevante, se obtiene que la SSPAI sería más eficiente que la mejor FIC en cuanto a los requerimientos de memoria y tiempo de CPU. Finalmente concluimos que, aunque la paralelización tiene sus limitaciones según el número de bloques, para el número de bloques de los casos presentados y para un número de procesadores limitado, la SSPAI se presenta como una estrategia interesante.

En el siguiente capítulo se plantea otra técnica iterativa para resolver los problemas de convección-difusión transitorios usando un solo procesador. Para ello se empleará el método de descomposición de dominios solapados.

Capítulo 6

Descomposición de dominios solapados

6.1. Introducción

En este capítulo presentamos un algoritmo eficiente para encontrar la solución de los problemas de convección-difusión transitorios con convección dominante basado en el método de descomposición de dominios con solapamiento.

Estamos interesados en un algoritmo que sea eficiente para resolver problemas con mallas finas (de más de 100 000 nodos) usando un solo procesador. Hemos optado por el método multiplicativo de Schwarz (MSM), puesto que converge más rápidamente que el método aditivo (ver Cai y Sarkis 1998). Recordemos que la velocidad de convergencia del MSM depende del tamaño del solapamiento y del número de subdominios (Bramble et al. (1991) y Widlund (1988)). Además coincide con el método iterativo de Gauss-Seidel por bloques, ver Tang (1992).

Dado que es importante ordenar los dominios (y nodos) en dirección del flujo (Cai y Sarkis (1998) y Cai y Saad (1996)) se propone renumerar los nodos de la malla mediante el habitual *ordenamiento bloque-interfase* descrito en el apartado 4.5. De esa manera se obtiene una matriz global tridiagonal por bloques. Además, recordemos que se ha empleado el algoritmo de Cuthill-McKee inverso (Saad (2003)) para reducir el ancho de banda de la matriz. Puesto que teníamos en mente emplear el método multiplicativo de Schwarz, la inversa aproximada *sparse* simétrica expuesta en el capítulo anterior se ha desarrollado para matrices SDP, tridiagonales

por bloques.

El método que se propone en este capítulo se apoya en dos hechos. El primero proviene de observar que los problemas con convección dominante presentan un frente brusco en la solución que se mueve de posición conforme avanza el tiempo, es decir la solución global sólo varía en una región del dominio completo en cada paso de tiempo. Una aplicación tecnológica donde se presenta este fenómeno son los filtros de carbón activo en un proceso de carga, ya que el perfil formado por la concentración de hidrocarburo es abrupto y se mueve a medida que se saturan las partículas de carbón (ver apartado 2.2.1). El segundo hecho es utilizar la idea básica de los métodos de descomposición de dominio: "divide y vencerás". Nuestro objetivo es mostrar numéricamente que en cada paso de tiempo es preferible resolver uno o dos subproblemas en donde la solución cambia que el problema global. Por ello, proponemos activar y desactivar dominios según vaya variando la solución.

Existen dos maneras de usar los métodos de Schwarz. La primera y la más común es utilizarlos como preconditionadores para acelerar algún métodos de Krylov. La segunda es manejarlos como una técnica iterativa, ya sea de Gauss-Seidel o Jacobi por bloques. En particular, empleamos el MSM directamente como una iteración de Gauss-Seidel por bloques solapados, en donde la solución de subdominios se realiza mediante el método directo de Cholesky o la iteración de gradientes conjugados preconditionado. La aceleración de este último se efectúa con las factorizaciones incompletas de Cholesky de umbral por ser eficientes para los problemas de estudio, ver Rodríguez-Ferran y Sandoval (2005) y capítulo 4.

Nuestro algoritmo difiere de otros trabajos donde se ha estudiado los problemas de convección-difusión transitorios utilizando algún método descomposición de dominios: Lasser y Toselli (2003), Quarteroni et al. (2002), Rapin y Lube (2002), Toselli (2001), Espedal et al. (1998), Cai y Sarkis (1998) y Brakkee y Wilders (1997).

A lo largo de este capítulo se presenta el algoritmo propuesto: el método multiplicativo de Schwarz en dominios activos. Previamente, se describe el MSM en los problemas de convección-difusión transitorios y se detallan los criterios de activación y desactivación de dominios. Finalmente, se muestra la eficiencia del algoritmo propuesto a través de ejemplos numéricos 3D.

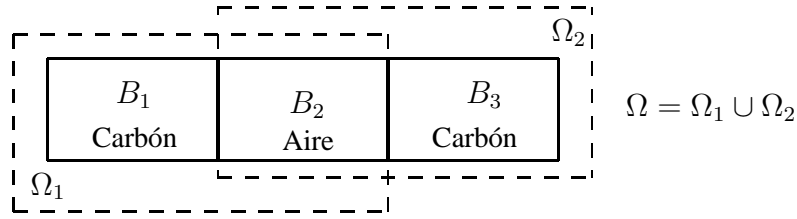


Figura 6.1: Dos subdominios solapados para un filtro con 3 cámaras.

6.2. Método multiplicativo de Schwarz en convección-difusión transitorio

En este apartado se deriva el método multiplicativo de Schwarz en forma matricial para el problema de convección-difusión en cada paso de tiempo. Para ello se considera el sistema lineal de ecuaciones (2.17) a resolver en el paso de tiempo $n + 1$

$$\mathbf{A}\mathbf{c}^{n+1} = [2\mathbf{B} - \mathbf{A}]\mathbf{c}^n = \mathbf{b} \quad (6.1)$$

donde las matrices \mathbf{A} y \mathbf{B} se definen en las relaciones (2.15) y (2.16). Se supone también que la región de estudio $\Omega(\subset \mathbb{R}^3)$ se descompone en M subdominios Ω_i de tal manera que se solapen entre sí ($\bar{\Omega} = \bigcup_{i=1}^M \bar{\Omega}_i$, con $\Omega_i \cap \Omega_{i+1} \neq \emptyset$).

Con el objeto de presentar el algoritmo del MSM a nivel matricial de una manera más clara, nos restringimos a nuestra aplicación tecnológica: los filtros de carbón activo. Por ello consideramos que la región de estudio $\Omega \subset \mathbb{R}^3$ es un filtro que se descompone en M subdominios Ω_i solapados entre sí, ver figura 6.1.

Asignamos el nombre de "bloque", en el contexto de descomposición de dominios, a los subconjuntos formados por $\Omega_i \cap \Omega_{i+1}$ (cámaras de aire) o por $\Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1})$ (cámaras de carbón). Notemos que cada bloque B_j está asociado a una cámara. Dado que los subconjuntos Ω_i son abiertos, los bloques que están en el solape $\Omega_i \cap \Omega_{i+1}$ no incluyen las fronteras Γ_{2i-1} y Γ_{2i} (interfases entre cámaras). En cambio los bloques $\Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1})$ son subconjuntos cerrados debido a que contienen las fronteras $\Gamma_{2(i-1)}$ y Γ_{2i-1} , ver figura 6.2.

Cada subdominio está formado por 2 o 3 bloques (o cámaras) según sea el caso. Los subdominios de la entrada y la salida (extremos del filtro) están constituidos

Γ_1	Γ_2	Γ_3	Γ_4	
$\Omega_1 \setminus \Omega_2$	$\Omega_1 \cap \Omega_2$	$\Omega_2 \setminus (\Omega_1 \cap \Omega_3)$	$\Omega_2 \cap \Omega_3$	$\Omega_3 \setminus \Omega_2$

Figura 6.2: Tres subdominios solapados para un filtro con 5 bloques.

por dos cámaras: una de carbón activo y otra de aire. En cambio, los dominios “interiores” están conformados por dos de aire y un bloque central, ver la figura 6.2. Observemos que el solapamiento siempre se produce en las cámaras de aire, ver figura 6.1. Además, se verifica que $\bar{\Omega} = \bigcup_{j=1}^{2M-1} \bar{B}_j$.

Se conoce que el método multiplicativo de Schwarz está relacionado con la técnica iterativa de Gauss-Seidel por bloques (ver Tang 1992). Por tanto, la forma matricial del subproblema de convección-difusión en la iteración $k + 1$ y paso de tiempo $n + 1$ restringido al subdominio Ω_1 se escribe como

$$\begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{\Omega_1 \setminus \Omega_2}^{k+1, n+1} \\ \mathbf{c}_{\Omega_1 \cap \Omega_2}^{k+1, n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\Omega_1 \setminus \Omega_2} \\ \mathbf{b}_{\Omega_1 \cap \Omega_2} - \mathbf{A}_{2,3} \mathbf{c}_{\Omega_2 \setminus (\Omega_1 \cup \Omega_3)}^{k, n+1} \end{pmatrix}.$$

Observemos que el término $\mathbf{A}_{2,3} \mathbf{c}_{\Omega_2 \setminus (\Omega_1 \cup \Omega_3)}^{k, n+1}$ está asociado a la condición de frontera Dirichlet sobre Γ_2 . Análogamente, la forma matricial del subproblema de convección-difusión restringido al subdominio Ω_2 es

$$\begin{pmatrix} \mathbf{A}_{2,2} & \mathbf{A}_{2,3} & \mathbf{0} \\ \mathbf{A}_{3,2} & \mathbf{A}_{3,3} & \mathbf{A}_{3,4} \\ \mathbf{0} & \mathbf{A}_{4,3} & \mathbf{A}_{4,4} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{\Omega_1 \cap \Omega_2}^{k+1, n+1} \\ \mathbf{c}_{\Omega_2 \setminus (\Omega_1 \cup \Omega_3)}^{k+1, n+1} \\ \mathbf{c}_{\Omega_2 \cap \Omega_3}^{k+1, n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\Omega_1 \cap \Omega_2} - \mathbf{A}_{2,1} \mathbf{c}_{\Omega_1 \setminus \Omega_2}^{k+1, n+1} \\ \mathbf{b}_{\Omega_2 \setminus (\Omega_1 \cup \Omega_3)} \\ \mathbf{b}_{\Omega_2 \cap \Omega_3} - \mathbf{A}_{4,5} \mathbf{c}_{\Omega_3 \setminus (\Omega_2 \cup \Omega_4)}^{k, n+1} \end{pmatrix}$$

donde los términos $\mathbf{A}_{2,1} \mathbf{c}_{\Omega_1 \setminus \Omega_2}^{k+1, n+1}$ y $\mathbf{A}_{4,5} \mathbf{c}_{\Omega_3 \setminus (\Omega_2 \cup \Omega_4)}^{k, n+1}$ representan las condiciones de frontera Dirichlet sobre Γ_1 y Γ_4 , respectivamente. Por consiguiente, el método multiplicativo de Schwarz con M subdominios para el problema de convección-difusión en el paso de tiempo $n + 1$ se presenta en la figura 6.3. El algoritmo inicia con $\mathbf{c}^{0, n+1} = \mathbf{c}^n$, además notemos que en nuestro caso el MSM es similar al método de Gauss-Seidel por bloques para matrices tridiagonales por bloques.

$\mathbf{c}^{0,n+1}$: aproximación inicial con la solución del paso anterior

Mientras (No convergencia & $k < kmax$)

resolver

$$\begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{\Omega_1 \setminus \Omega_2}^{k+1,n+1} \\ \mathbf{c}_{\Omega_1 \cap \Omega_2}^{k+1,n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\Omega_1 \setminus \Omega_2} \\ \mathbf{b}_{\Omega_1 \cap \Omega_2} - \mathbf{A}_{2,3} \mathbf{c}_{\Omega_2 \setminus (\Omega_1 \cup \Omega_3)}^{k,n+1} \end{pmatrix}$$

Para $i = 2, M - 1$

resolver

$$\begin{pmatrix} \mathbf{A}_{2i-2,2i-2} & \mathbf{A}_{2i-2,2i-1} & \mathbf{0} \\ \mathbf{A}_{2i-1,2i-2} & \mathbf{A}_{2i-1,2i-1} & \mathbf{A}_{2i-1,2i} \\ \mathbf{0} & \mathbf{A}_{2i,2i-1} & \mathbf{A}_{2i,2i} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{\Omega_{i-1} \cap \Omega_i}^{k+1,n+1} \\ \mathbf{c}_{\Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1})}^{k+1,n+1} \\ \mathbf{c}_{\Omega_i \cap \Omega_{i+1}}^{k+1,n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\Omega_{i-1} \cap \Omega_i} - \mathbf{A}_{2i-2,2i-3} \mathbf{c}_{\Omega_{i-1} \setminus (\Omega_{i-2} \cup \Omega_i)}^{k+1,n+1} \\ \mathbf{b}_{\Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1})} \\ \mathbf{b}_{\Omega_i \cap \Omega_{i+1}} - \mathbf{A}_{2i,2i+1} \mathbf{c}_{\Omega_{i+1} \setminus (\Omega_i \cup \Omega_{i+2})}^{k,n+1} \end{pmatrix}$$

Terminar i

resolver

$$\begin{pmatrix} \mathbf{A}_{2M-2,2M-2} & \mathbf{A}_{2M-2,2M-1} \\ \mathbf{A}_{2M-1,2M-2} & \mathbf{A}_{2M-1,2M-1} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{\Omega_{M-1} \cap \Omega_M}^{k+1,n+1} \\ \mathbf{c}_{\Omega_M \setminus \Omega_{M-1}}^{k+1,n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\Omega_{M-1} \cap \Omega_M} - \mathbf{A}_{2M-2,2M-3} \mathbf{c}_{\Omega_{M-1} \setminus (\Omega_{M-2} \cup \Omega_M)}^{k+1} \\ \mathbf{b}_{\Omega_M \setminus \Omega_{M-1}} \end{pmatrix}$$

Terminar k

Figura 6.3: Método multiplicativo de Schwarz.

Finalmente, observemos que las matrices asociadas a los subsistemas del algoritmo 6.3 son simétricas definidas positivas, dado que son submatrices principales de la matriz global \mathbf{A} que es simétrica definida positiva (ver Hackbusch 1994).

6.3. Criterios de activación y desactivación de dominios

Nuestro objetivo es mostrar que es más barato calcular la solución en uno o dos subdominios donde la solución varía, que en el dominio completo a lo largo del tiempo. Para ello proponemos activar los dominios en donde la solución cambia, y desactivar aquellos en los cuales la solución de los subdominios ya no se modifica. La activación y desactivación se realiza en cada paso de tiempo.

La principal dificultad de la propuesta radica en definir adecuadamente los criterios de activación y desactivación de dominios. Como los criterios están relacionados con las características físicas del problema nos restringimos a los filtros de carbón activo con el fin de exponer claramente los criterios propuestos. Sin embargo, estos funcionan para otro tipo de aplicaciones.

Consideremos que el dominio de estudio $\bar{\Omega} = \bigcup_{i=1}^M \bar{\Omega}_i$ es un filtro dividido en M subdominios solapados. Además, supongamos que el “frente” ha avanzado varios subdominios. Se tienen dos situaciones en donde el valor de la concentración no cambia. La primera es en los subdominios que anteceden al “frente” porque quedan saturados de hidrocarburo. Tales subdominios forman el conjunto Ω_{inac_s} . La segunda se encuentra en los subdominios posteriores al “frente”; como el hidrocarburo aún no ha llegado a ellos, la concentración es cero. Este conjunto de subdominios es denotado por Ω_{inac_0} . De esa manera el conjunto de subdominios inactivos Ω_{inac} está formado por $\Omega_{inac_s} \cup \Omega_{inac_0}$. Por su parte, denotamos por Ω_{act} al conjunto de subdominios activos en donde la solución varía. Observemos que la concentración de hidrocarburo sólo se modifica en un o dos subdominios en cada paso de tiempo, ver figura 6.4.

La activación y desactivación se realiza primero sobre los bloques B_j y después sobre los subdominios Ω_i . Dos condiciones se deben de satisfacer para inactivar un bloque: (1) que la solución cumpla los requisitos de precisión (criterio del residuo) y (2) que el bloque esté saturado (criterio de la concentración). La forma en cómo se llevan acabo estas ideas (al inicio de cada paso de tiempo) se describe a continuación:

1. Se efectúa la activación y desactivación de bloques:

a) Para todo $B_j \in \Omega_{act} \cup \Omega_{inac_0}$ se calcula $\mathbf{r}_j^{n+1} = \mathbf{b}_j - \mathbf{A}\mathbf{c}_j^{0,n+1}$.

b) Se activa o inactiva los bloques mediante el *criterio del residuo*:

Si $\|\mathbf{r}_j^{n+1}\|_2 < tol_r \|\mathbf{b}\|_2$, entonces $B_j \in \Omega_{inac}$
 en otro caso, $B_j \in \Omega_{act}$.

c) Se verifica que los recientes bloques activados y desactivados satisfagan el *criterio de la concentración*:

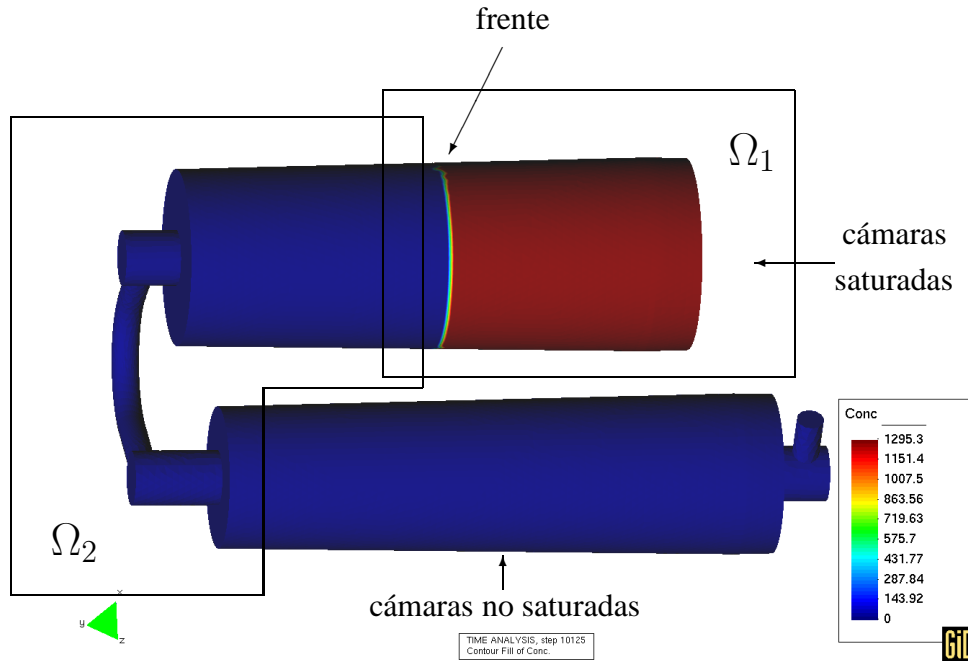


Figura 6.4: Frente de hidrocarburo en un proceso de carga.

- Si B_j es un bloque recientemente inactivo y cumple $\|c_j^{0,n+1} - c_{ext_j}\|_0 > tol_c \|c_{ext}\|_0$, entonces $B_j \in \Omega_{act}$.
 - Si el primer bloque activo B_j satisface $\|c_j^{0,n+1} - c_{ext_j}\|_0 \leq tol_c \|c_{ext}\|_0$, entonces $B_j \in \Omega_{inac}$.
2. Se efectúa la activación y desactivación de dominios:
- a) Si Ω_i contiene al menos un bloque activo, entonces Ω_i se activa.
 - Si B_j es activo y está en $\Omega_i \cap \Omega_{i+1}$, entonces Ω_i y Ω_{i+1} se activan.
 - b) En el primer paso de tiempo se activan todos los dominios.

En este caso c_{ext} es la concentración exterior, tol_c es la tolerancia asociada al criterio de la concentración y la norma $\|\cdot\|_0$ está definida en el apartado 2.3.

Es importante enfatizar que un dominio activo significa resolver el subsistema asociado a él. También notemos que Ω_{inac_s} permanece igual o crece en cada paso de tiempo. Análogamente, el conjunto Ω_{inac_\emptyset} se mantiene igual o decrece conforme se avanza en el tiempo.

6.4. Método multiplicativo de Schwarz en dominios activos

Tomando en cuenta lo señalado en el apartado anterior se propone el método multiplicativo de Schwarz en dominios activos. Para ello simplificamos la notación de algoritmo de la figura 6.3 redefiniendo: (1) las incógnitas $\mathbf{c}_{\Omega_i \cap \Omega_{i+1}}^{k,n+1}$ por \mathbf{c}_{2i}^k con $i = 1, \dots, M$; (2) las variables $\mathbf{c}_{\Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1})}^{k,n+1}$ como \mathbf{c}_{2i-1}^k para $i = 1, \dots, M$ y (3) la aproximación inicial $\mathbf{c}^{0,n+1}$ por \mathbf{c}^0 .

El algoritmo multiplicativo de Schwarz en dominios activos se presenta en la figura 6.5. Notemos que la única diferencia que existe con el de la figura 6.3 es exigir que $\Omega_i \subset \Omega_{act}$ para resolver el subproblema asociado a Ω_i . También observemos que al ser $\Omega = \Omega_{inac_s} \cup \Omega_{act} \cup \Omega_{inac_0}$ induce un desglose en la concentración: $\mathbf{c} = (\mathbf{c}_{ext}, \mathbf{c}_{act}, \mathbf{0})^T$.

\mathbf{c}^0 : aproximación inicial con la solución del paso anterior

Mientras (No convergencia & $k < kmax$)

Si $\Omega_1 \subset \Omega_{act}$, resolver

$$\begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{pmatrix} \begin{pmatrix} \mathbf{c}_1^{k+1} \\ \mathbf{c}_2^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 - \mathbf{A}_{2,3}\mathbf{c}_3^k \end{pmatrix}$$

Para cada $i = 2, M - 1$ y $\Omega_i \subset \Omega_{act}$

resolver

$$\begin{pmatrix} \mathbf{A}_{2i-2,2i-2} & \mathbf{A}_{2i-2,2i-1} & \mathbf{0} \\ \mathbf{A}_{2i-1,2i-2} & \mathbf{A}_{2i-1,2i-1} & \mathbf{A}_{2i-1,2i} \\ \mathbf{0} & \mathbf{A}_{2i,2i-1} & \mathbf{A}_{2i,2i} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{2i-2}^{k+1} \\ \mathbf{c}_{2i-1}^{k+1} \\ \mathbf{c}_{2i}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{2i-2} - \mathbf{A}_{2i-2,2i-3}\mathbf{c}_{2i-3}^{k+1} \\ \mathbf{b}_{2i-1} \\ \mathbf{b}_{2i} - \mathbf{A}_{2i,2i+1}\mathbf{c}_{2i+1}^k \end{pmatrix}$$

Terminar i

Si $\Omega_M \subset \Omega_{act}$, resolver

$$\begin{pmatrix} \mathbf{A}_{2M-2,2M-2} & \mathbf{A}_{2M-2,2M-1} \\ \mathbf{A}_{2M-1,2M-2} & \mathbf{A}_{2M-1,2M-1} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{2M-2}^{k+1} \\ \mathbf{c}_{2M-1}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{2M-2} - \mathbf{A}_{2M-2,2M-3}\mathbf{c}_{2M-3}^{k+1} \\ \mathbf{b}_{2M-1} \end{pmatrix}$$

Terminar k

Figura 6.5: Método multiplicativo de Schwarz en dominios activos.

6.4.1. Solución de subdominios: métodos directos e iterativos

Los subsistemas que aparecen en el método multiplicativo de Schwarz se pueden resolver mediante métodos directos o métodos iterativos. Existen varios trabajos donde se obtiene que los métodos iterativos son más eficientes, ver por ejemplo Cai, Gropp y Keyes (1994), Cai y Saad (1996), Brakkee et al. (1998), Brakkee et al. (1995) y Garbey y Tromeur-Dervout (2002). Sin embargo, empleamos ambas formas para encontrar la solución de los subdominios. Como método directo se utiliza la factorización de Cholesky (ver Duff et al. 1992) y como iterativo se elige a gradientes conjugados preconditionado por ser las submatrices diagonales SDP (ver Saad 2003). Además, las FIC de umbral descritas en el apartado 4.2 se emplean como preconditionadores. Esta elección está de acuerdo con los resultados numéricos obtenidos por Rodríguez-Ferran y Sandoval (2005).

6.5. Experimentos numéricos

El propósito de este apartado es mostrar la eficiencia computacional del método multiplicativo de Schwarz en dominios activos cuando se utiliza en los problemas de convección-difusión transitorios con velocidad constante. Los resultados numéricos del algoritmo propuesto se comparan con la solución del problema global empleando una FIC con umbral $\tau = 5 \cdot 10^{-3}$, ver capítulo 4. También se utiliza como referencia el método directo de Cholesky (sin iteraciones) y el método de Cholesky con refinamiento iterativo. La eficiencia computacional se mide a través de los requerimientos de memoria y tiempo de CPU.

La notación y parámetros que se van a emplear a lo largo de este apartado se condensan a continuación:

- DD-Chol, método multiplicativo de Schwarz en dominios activos con Cholesky,
- DD-FIC(τ), método multiplicativo de Schwarz en dominios activos con factorizaciones incompletas de Cholesky de umbral,
- Chol, método de Cholesky,

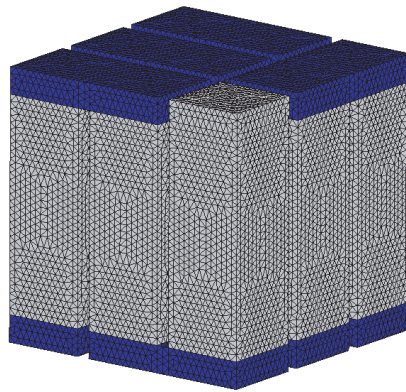
- RI, método de refinamiento iterativo,
- $\text{FIC}(\tau)$, factorización incompleta de Cholesky con umbral τ (ver Munksgaard 1980, Salvini y Shaw 1995),
- $\text{FIC}(\infty)$, factorización incompleta de Cholesky sin llenado (ver Saad 2003),
- \mathbf{A} , matriz global,
- \mathbf{A}_D , matriz diagonal por bloques formada por las submatrices de los subsistemas del algoritmo 6.5
- $\text{tol}_r = 0.5 \cdot 10^{-9}$, tolerancia para el criterio del residuo,
- $\text{tol}_c = 0.5 \cdot 10^{-7}$, tolerancia para el criterio de la concentración.

Se trabaja básicamente con dos filtros, denominados D y E. Para el filtro D se han utilizado tres mallas no estructuradas, denotadas por: “gruesa”, “fina” y “extra fina”. También se ha considerado el filtro D con el tamaño de las cámaras de aire duplicado, en este caso se usa una malla “gruesa”. Para el filtro E se emplean dos mallas: “gruesa” y “fina”. La figura 6.6 muestra la geometría de estos filtros.

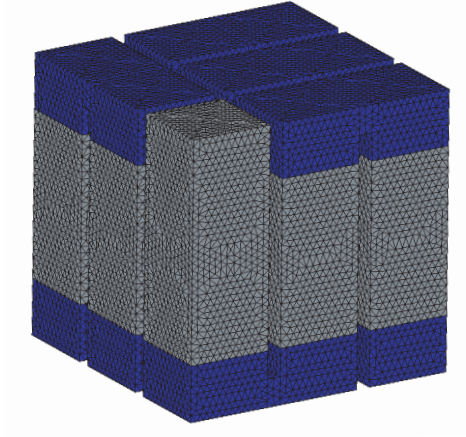
	Filtro D				Filtro E	
	Gruesa	Fina	Extra fina	C.A. doble	Gruesa	Fina
# nodos	87 703	141 706	591 857	86 624	87 970	171 802
# elem.	453 174	747 641	3 274 220	447 228	472 277	940 695
N	87 527	141 465	591 261	86 448	87 274	170 850
$\text{nnz}(\mathbf{A})$	648 577	1 059 448	4 532 527	640 569	659 253	1 301 816
$\text{nnz}(\mathbf{A}_D)$	751 912	1 235 555	5 324 074	852 365	729 294	1 435 060
# pasos	81 513	96 272	155 912	64 288	28 961	32 493
Δt	2.00	1.69	1.04	2.00	3.12	2.8
M	9	9	9	9	4	4

Tabla 6.1: Principales parámetros numéricos para la simulación de los filtros D y E.

Filtro D, malla gruesa



Filtro D con cámaras de aire de espesor doble, malla gruesa



Filtro E, malla gruesa

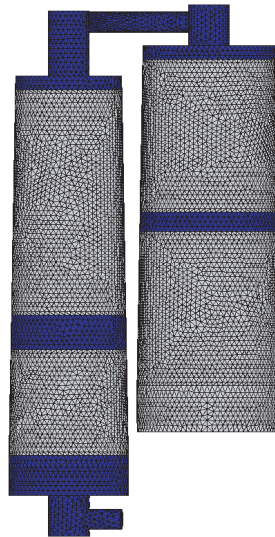


Figura 6.6: Filtros de carbón activo

La tabla 6.1 sintetiza 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) $\text{nnz}(\mathbf{A})$ representa el número de elementos no nulos de la parte triangular superior de la matriz \mathbf{A} , (5) $\text{nnz}(\mathbf{A}_D)$ es el número de no nulos asociados a la triangular superior de la matriz diagonal por bloques \mathbf{A}_D , (6) el incremento de tiempo Δt , (7) el número de pasos de tiempo y (8) el número M de subdominios.

Observemos que el número de nodos es ligeramente mayor que N , dado que las matrices no incluyen las condiciones de contorno tipo Dirichlet. Además para todas las pruebas se cumple $\text{nnz}(\mathbf{A}) < \text{nnz}(\mathbf{A}_D)$. Esto se debe a que los bloques de solapamiento aparecen duplicados y los bloques no diagonales de \mathbf{A} (asociados a las interfases) son muy vacíos. Por su parte, el incremento de tiempo Δt se elige de acuerdo al tamaño del elemento h con el fin de que el número de Courant (C) sea cercano a 1, ver apartado 2.5.

Los criterios de parada para ambos métodos iterativos son

$$|x_i^k - x_i^{k+1}| \leq \text{tol}_x (|x_i^{k+1}| + 1), \quad \forall i = 1, 2, \dots, N; \quad (6.2)$$

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}\|_2 \leq \text{tol}_r \|\mathbf{b}\|_2, \quad (6.3)$$

donde $\text{tol}_x = 0.5 \cdot 10^{-10}$. Además, se utiliza como aproximación inicial \mathbf{x}^0 , la solución final del paso de tiempo previo.

Todos los códigos están programados en Fortran 90 con aritmética en doble precisión y han sido compilados con la opción `-xP -O2 -ipo -fixed -tpp7`. Los experimentos numéricos se han realizado sobre un procesador Pentium Xeon EM64T, a 3.3 GHz con 8 Mb de cache L3.

6.5.1. Verificación de criterios de activación y desactivación

Con el objeto de verificar los criterios de activación y desactivación de dominios se utilizan los resultados numéricos de los filtros D y E, mallas gruesas. Las gráficas 6.7.a, 6.7.b y 6.8.a muestran el comportamiento del criterio de la concentración en cada bloque B_j a lo largo del tiempo. De igual forma, las figuras 6.7.c y 6.8.b presentan la activación y desactivación de los subdominios Ω_i en el transcurso del

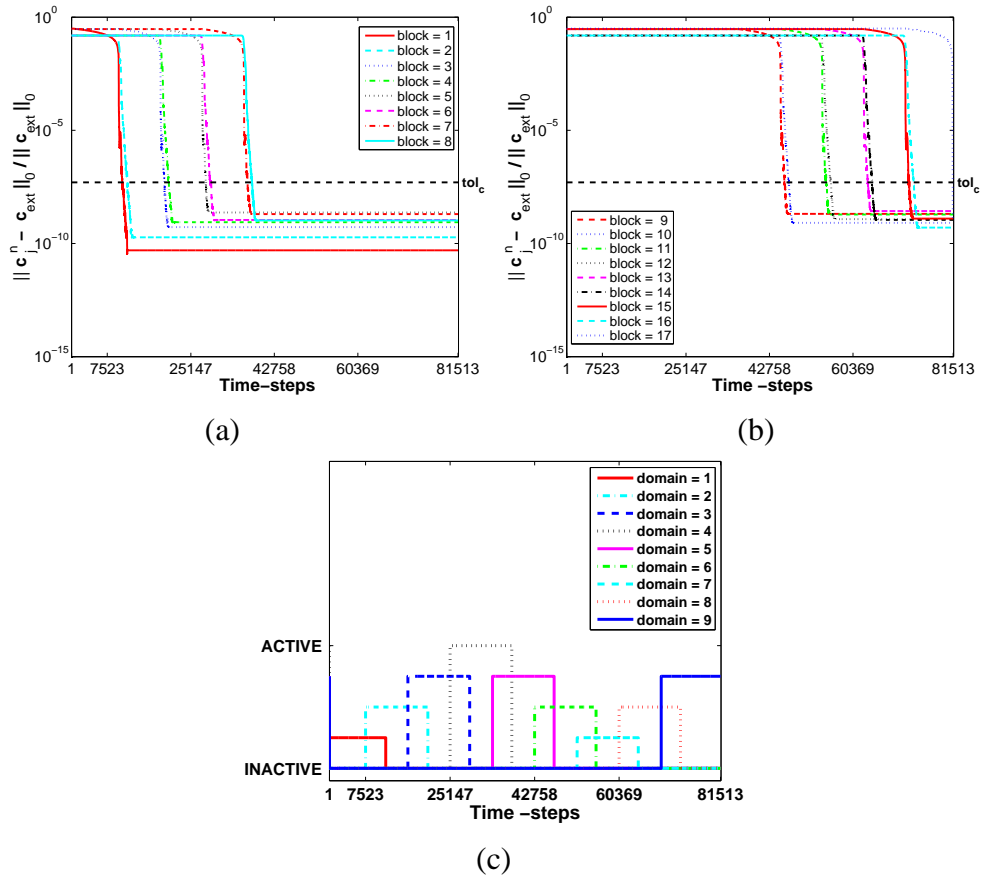


Figura 6.7: Filtro D, malla gruesa. (a) y (b) Criterio de la concentración vs. pasos de tiempo y (c) activación y desactivación de dominios vs. pasos de tiempo.

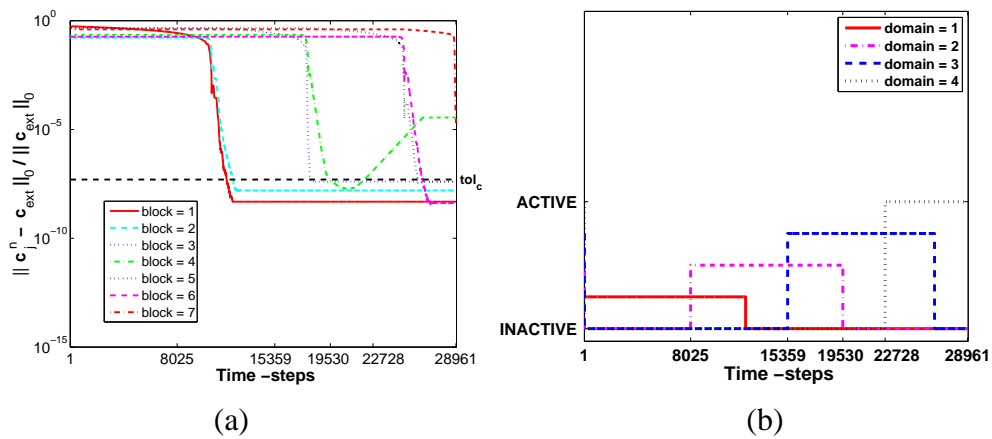


Figura 6.8: Filtro E, malla gruesa. (a) Criterio de la concentración vs. pasos de tiempo y (b) activación y desactivación de dominios vs. pasos de tiempo.

tiempo. La manera de interpretar dichas gráficas es la siguiente: los bloques 2, 3 y 4 de 6.8.a forman el dominio 2; en $n = 8\,025$ la concentración del bloque 2 es diferente de cero y se activa el dominio; para $n = 19\,530$ los tres bloques satisfacen el criterio de la concentración y por tanto se inactiva el dominio 2.

Notemos de las figuras 6.7.c y 6.8.b que la activación y desactivación de los subdominios se realiza correctamente. Además, observemos que en un porcentaje importante de pasos de tiempo se resuelven dos subdominios. En el apartado 6.5.3 se mostrará que a pesar de ello se obtienen buenos resultados numéricos.

	RI		DD-Chol		$\frac{\text{DD-Chol}-\text{RI}}{\text{RI}}$
	Tiempo (s)	%	Tiempo (s)	%	%
Cálculo de \mathbf{A} y \mathbf{A}_D	2.83	0.01	2.93	0.02	3.53
Factorización	3.98	0.02	4.44	0.03	11.56
Cálculo de \mathbf{b}	1620.29	6.43	1637.03	11.76	1.03
$\ \mathbf{b} - \mathbf{A}\mathbf{c}^k\ _2$	782.50	3.11	1138.66	8.18	45.52
Cálculo de \mathbf{b}_j	—	—	151.95	1.09	—
$\ \mathbf{b}_j - \mathbf{A}\mathbf{c}_j^0\ _2$	—	—	11.47	0.08	—
Índices de activación	—	—	5.44	0.04	—
$\ c_j^0 - c_{ext,j}\ _0$	—	—	742.07	5.33	—
Cálculos adicionales	358.74	1.42	934.88	6.72	160.60
Solución	22107.05	87.77	8973.54	64.49	−59.41
TOTAL	25188.51	100	13914.76	100	−44.76
Memoria factorización	23 074 479	100	26 014 881	99.66	12.74
Mem. vectores aux.	—	—	87 703	0.34	—
TOTAL	23 074 478	100	26 102 584	100	13.12
# pasos	81 513		81 513		0
# iteraciones	81 513		172 905		112.11
iter/pasos	1		2.12		112

Tabla 6.2: Filtro D, malla gruesa. Tiempos de CPU usando un refinamiento iterativo y el método multiplicativo de Schwarz en dominios activos.

Con el fin de verificar el desempeño del método propuesto y exigir al método directo la misma precisión que al método iterativo se emplea un refinamiento iterativo (RI), ver método en Meurant (1999). Esta técnica sirve como referencia para contrastar el coste en cuanto a memoria, cálculo de normas, de índices y de nuevos

vectores independientes, así como el número de iteraciones que genera el MSM en dominios activos.

La tabla 6.2 resume los resultados numéricos al utilizar el refinamiento iterativo y el MSM con activación de dominios para el filtro D. En ella apreciamos la relevancia del algoritmo propuesto, ya que únicamente requiere aumentar la memoria un 13 % para obtener un ahorro en tiempo de CPU total del 45 %.

Cabe señalar que el coste adicional que genera aplicar los criterios de activación y desactivación son compensados por la solución de uno o dos subdominios en cada paso de tiempo.

También observemos que el número de iteraciones para DD-Chol se duplica, sin embargo no ocurre lo mismo para el tiempo del cálculo del residuo. Esto se debe a que el residuo únicamente se calcula sobre los dominios activos: $\|\mathbf{b}_{act} - \mathbf{A}\mathbf{c}_{act}^k\|_2$.

Finalmente, la tabla 6.2 indica que el refinamiento iterativo no se activa: la solución con el método directo ya satisface la precisión exigida en el método iterativo.

6.5.2. Efecto del tamaño del solapamiento versus tamaño de elemento

En este apartado se verifica numéricamente que el MSM en dominios activos conserva algunas características del método multiplicativo de Schwarz.

La figura 6.9 presenta las gráficas del número de iteraciones contra pasos de tiempo para el filtro D usando el algoritmo propuesto para diferentes mallas y distintos tamaños de solapamiento. Las gráficas (a) y (d) muestran que al aumentar el tamaño del solapamiento se acelera la convergencia; el promedio de iteraciones (número acumulado de iteraciones entre el número de pasos de tiempo) disminuye de 2.12 a 1.06. De esta forma, el MSM en dominios activos respeta la propiedad de la estima para la velocidad de convergencia (3.18) del MSM propuesta por Bramble et al. (1991).

La malla extra fina septuplica el número de elementos de la malla gruesa, aún así la iteración promedio es mayor que cuando se duplica el solapamiento, ver gráficas (c) y (d). Sin embargo el ahorro en el tiempo de CPU con respecto al RI es del 70 % para la malla extra fina y de un 64 % para el filtro con cámaras de aire de espesor doble. El MSM en dominios activos se presenta como una técnica interesante para

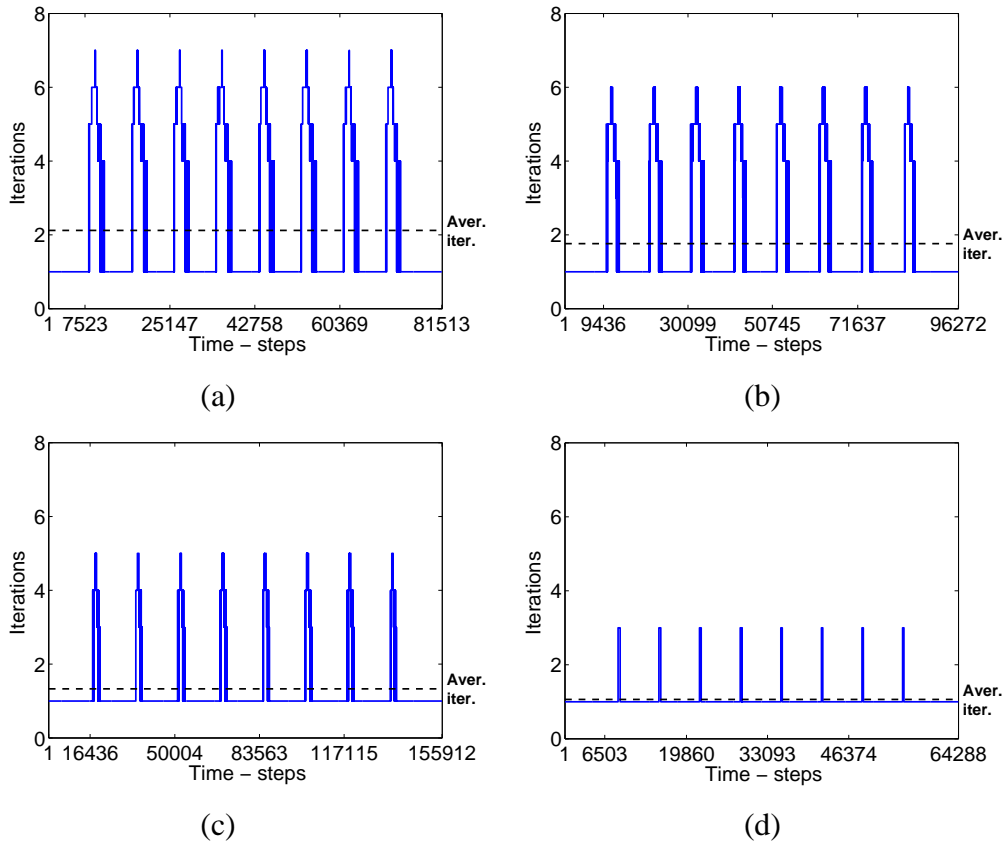


Figura 6.9: Gráficas de iteraciones vs. pasos de tiempo usando DD-Chol para la: (a) malla gruesa, (b) malla fina, (c) malla extra fina y (d) las cámaras de aire dobles.

mallas finas con pequeños solapamientos.

6.5.3. Eficiencia computacional del MSM en dominios activos

Después de verificar el buen funcionamiento del MSM en dominios activos, nos centramos en estudiar su eficiencia computacional comparándolo con otros métodos directos e iterativos.

Eficiencia computacional en la resolución de subproblemas

En este apartado se comparan dos estrategias para resolver los subproblemas, el método directo de Cholesky y la iteración de gradientes conjugados preconditionado (PCG). Para acelerar la convergencia de PCG se emplea las FIC usando

diferentes umbrales y la FIC sin llenado.

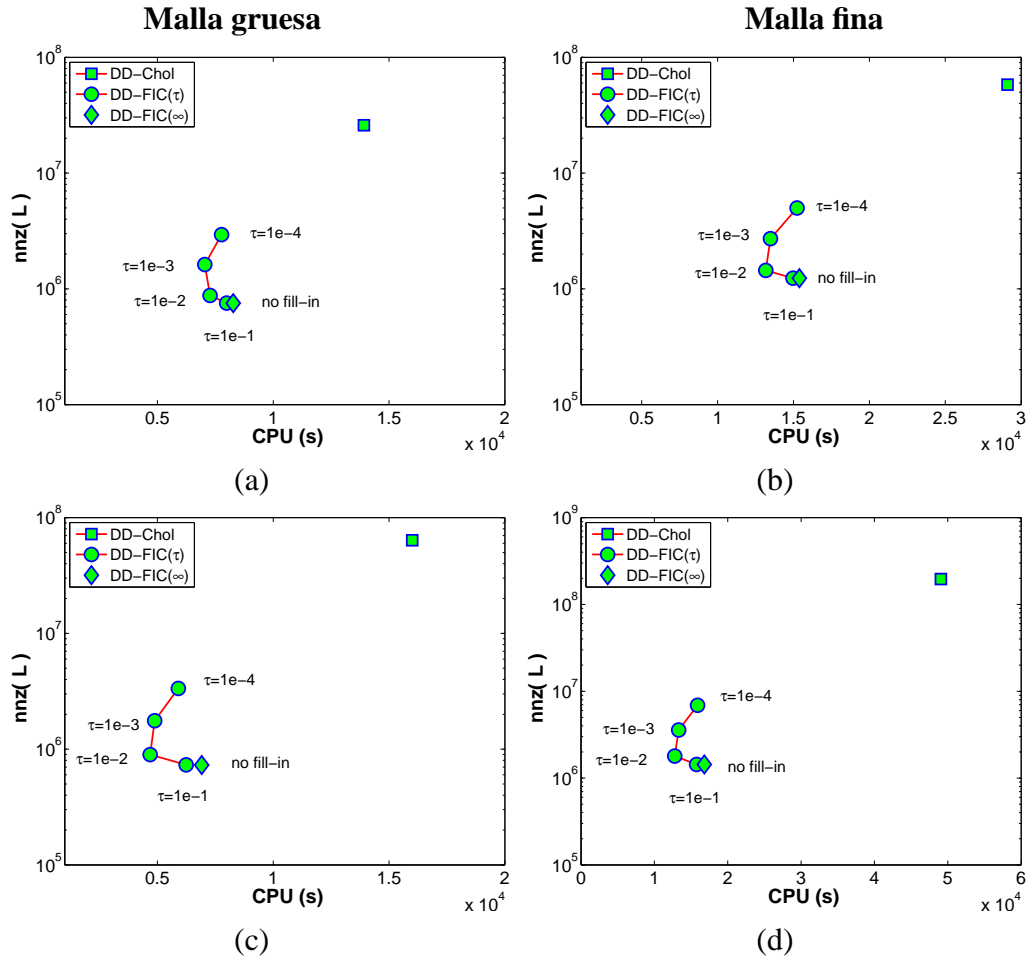


Figura 6.10: Coste computacional del MSM en dominios activos con Cholesky y PCG para la simulación del: (a) filtro D, malla gruesa, (b) filtro D, malla fina, (c) filtro E, malla gruesa, (d) filtro E, malla fina.

Las gráficas de la figura 6.10 muestran el coste computacional de las pruebas antes mencionadas para los filtros D y E con ambas mallas, gruesa y fina. Así mismo la tabla 6.3 resume los resultados para el filtro E. En ella se presenta: (1) el número de pasos de tiempo, (2) el número acumulado de iteraciones para el MSM en dominios activos, (3) el promedio de iteraciones para el MSM en dominios activos, (4) el número acumulado de iteraciones para resolver los subproblemas con PCG, (5) el promedio de iteraciones para PCG, (6) el promedio de entradas no nulas del factor L y la parte triangular superior de la matriz diagonal por bloques A_D ; y los

tiempos de CPU requeridos: (7) para calcular \mathbf{L} , (8) resolver los sistemas lineales (completos) en todos los pasos de tiempo, y (9) en total.

	DD-Chol		DD-FIC(∞)		DD-FIC(10^{-1})	
	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
# pasos	28 961	32 493	28 961	32 493	28 961	32 493
# iter MSM	46 978	50 286	42 983	46 916	42 983	46 916
iter MSM/pasos	1.62	1.55	1.48	1.44	1.48	1.44
# iter PCG	–	–	1 348 348	1 235 090	1 141 702	1 121 674
iter PCG/pasos	–	–	46.56	38.01	39.42	34.52
$\text{nnz}(\mathbf{L})/\text{nnz}(\mathbf{A}_D)$	87.47	136.77	1.00	1.00	1.00	1.00
Factorización (s)	59.1	455.8	0.50	1.22	0.52	1.29
Solución (s)	8 658.9	43 762.2	4 873.3	12 062.5	4 213.1	11 004.0
TOTAL (s)	19 867.4	49 057.1	6 913.3	16 805.3	6 242.0	15 755.7

	DD-FIC(10^{-2})		DD-FIC(10^{-3})		DD-FIC(10^{-4})	
	Gruesa	Fina	Gruesa	Fina	Gruesa	Fina
# pasos	28 961	32 493	28 961	32 493	28 961	32 493
# iter MSM	42 983	46 916	42 983	46 916	42 983	46 916
iter MSM/pasos	1.48	1.44	1.48	1.44	1.48	1.44
# iter PCG	608 552	689 475	385 252	433 599	264 385	303 310
iter PCG/pasos	21.01	21.22	13.30	13.34	9.13	9.33
$\text{nnz}(\mathbf{L})/\text{nnz}(\mathbf{A}_D)$	1.23	1.25	2.41	2.49	4.59	4.8
Factorización (s)	0.72	1.73	1.95	4.79	5.46	13.66
Solución (s)	2 648.2	8 004.6	2 841.1	8 559.9	3 848.1	11 110.6
TOTAL (s)	4 685.9	12 776.2	4 881.8	13 315.7	5 903.6	15 888.6

Tabla 6.3: Coste computacional del MSM en dominios activos con Cholesky y PCG para la simulación del filtro E.

Del análisis de las gráficas 6.10 y la tabla 6.3 se resalta los siguientes aspectos:

- El MSM en dominios activos requiere, en promedio, menos de dos iteraciones para converger.
- Se necesitan decenas de iteraciones en cada paso de tiempo para obtener la solución en todos los subdominios activos usando PCG.
- La convergencia de DD-FIC(τ) es ligeramente más rápida que la de DD-Chol: el error cometido en la resolución de los subproblemas es ligeramente

superior para Cholesky que la tolerancia exigida para los métodos iterativos.

- Sin ningún incremento en el número de no-nulos, las FIC sin llenado reducen considerablemente el tiempo de CPU.
- Permitiendo un llenado apropiado en la FIC ($\tau = 10^{-2}$) la reducción en el tiempo total de CPU es significativa (por un factor de 2 o 3 según sea el caso).
- El tiempo necesario para construir la factorización completa o incompleta de A_D es *irrelevante* comparado con el tiempo requerido para calcular la solución en los subdominios activos de todos los pasos de tiempo. Este resultado está de acuerdo al obtenido por Rodríguez-Ferran y Sandoval (2005).

Eficiencia numérica de técnicas directas versus iterativas

Ahora comparamos el mejor resultado que se obtuvo en el apartado anterior, DD-FIC(10^{-2}), con el resultado óptimo ($\tau = 5 \cdot 10^{-3}$) obtenido por Rodríguez-Ferran y Sandoval (2005) cuando se utiliza PCG+FIC para resolver el problema global, ver también capítulo 4. Además, empleamos el método de Cholesky y el refinamiento iterativo para poner en contexto nuestros resultados.

Las gráficas de la figura 6.11 presentan los resultado de las técnicas antes mencionadas para todos los experimentos descritos en la tabla 6.1. De ellas podemos destacar lo siguiente:

- Se puede compensar el hecho de tener solapamientos pequeños al obtener la solución mediante DD-FIC(10^{-2}). En las gráficas (a) y (d) prácticamente el coste computacional es el mismo cuando $\tau = 10^{-2}$.
- Conforme se refina la malla el ahorro en tiempo de CPU es altamente significativo: para el filtro D malla extra fina mientras Cholesky invierte 10.8 días y la FIC($5 \cdot 10^{-3}$) 3.7 días, DD-FIC(10^{-2}) requiere únicamente 1.3 días.
- En todos los casos el MSM en dominios activos+PCG+FIC es más eficiente que resolver el problema global con una FIC de umbral. Se recomienda utilizar $\tau \in [10^{-3}, 10^{-1}]$.

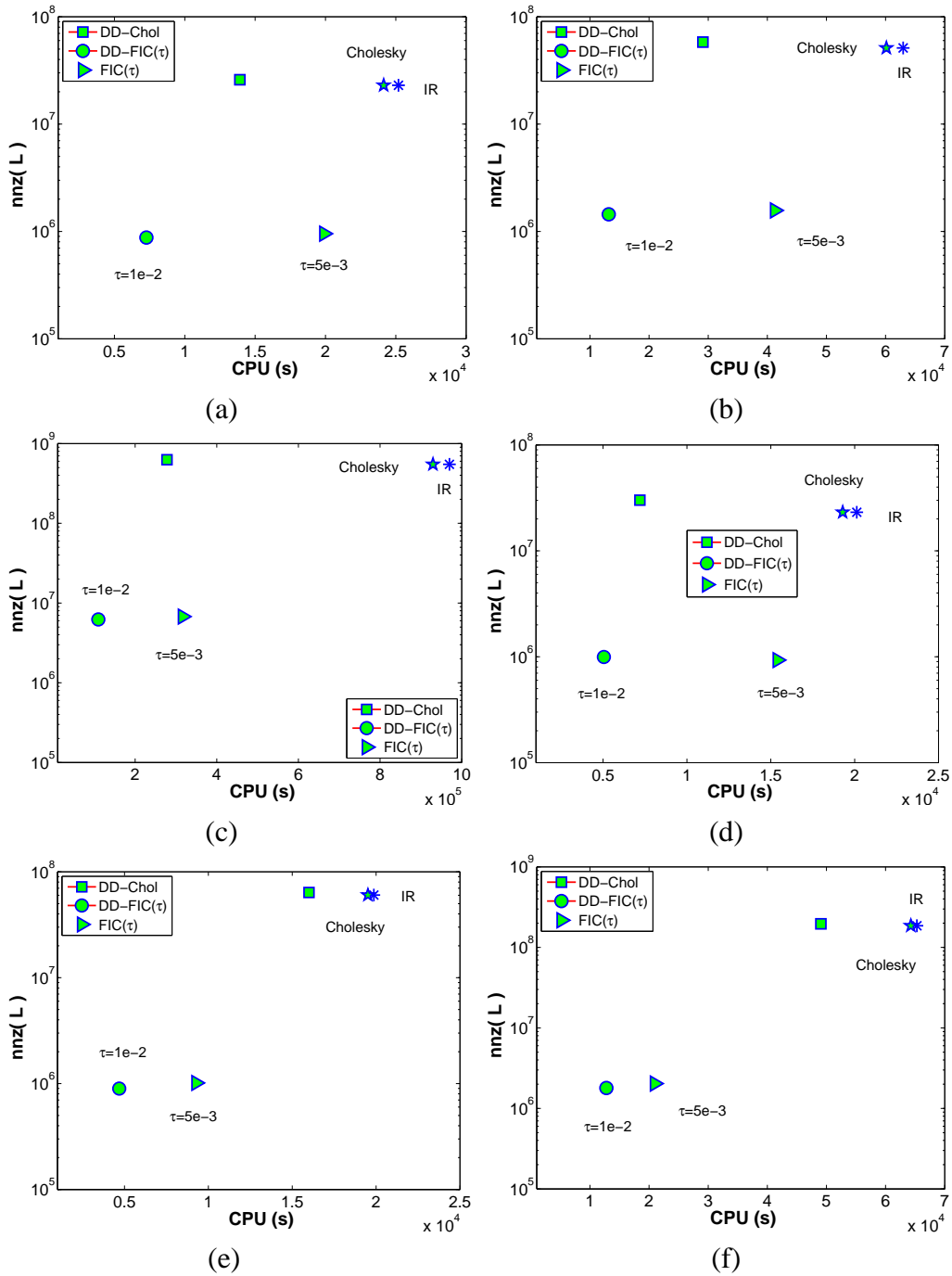


Figura 6.11: Coste computacional de Cholesky, PCG y MSM en dominios activos para la simulación del: (a) filtro D, malla gruesa, (b) filtro D, malla fina, (c) filtro D, malla extra fina, (d) filtro D, c.a. dobles, (e) filtro E, malla gruesa, (f) filtro E, malla fina.

6.6. Conclusiones

Se ha desarrollado el método multiplicativo de Schwarz en dominios activos diseñado para problemas transitorios de convección dominante que presentan un frente brusco en la solución, y donde ésta sólo cambia en una región del dominio completo en cada paso de tiempo.

Nuestros experimentos numéricos muestran que usar el MSM en dominios activos resolviendo los subproblemas con CG acelerado por una FIC de umbral o una FIC sin llenado es más eficiente que resolver el problema global empleando la iteración de CG preconditionado con la misma FIC de umbral.

Conforme a nuestros resultados se puede utilizar τ en $[10^{-3}, 10^{-1}]$, aunque se recomienda emplear un umbral igual a 10^{-2} . También los resultados señalan que el coste computacional necesario para construir las FIC asociadas a los subdominios no es un factor relevante, puesto que se puede amortizar en muchos pasos de tiempo. Ambas conclusiones están de acuerdo a lo obtenido por Rodríguez-Ferran y Sandoval (2005).

Debido a que la eficiencia numérica está completamente controlada por la fase de aplicación de cada subproblema en cada paso de tiempo, utilizar PCG+FIC(τ) compensa el hecho de realizar más iteraciones en solapamientos pequeños. Por ello, el método propuesto resulta ser muy atractivo para resolver los problemas de convección-difusión transitorios con mallas finas, en donde se tengan diferentes tamaños de solapamientos y el cálculo se realice con un solo procesador.

Finalmente, a pesar de que los criterios de activación y desactivación propuestos están relacionados con las características físicas del problema de estudio, se pueden adecuar a nuestra otra aplicación tecnológica, el problema de dispersión de contaminantes.

Capítulo 7

Conclusiones y futuras líneas de investigación

7.1. Conclusiones

En esta tesis hemos investigado diversas técnicas numéricas para resolver grandes sistemas de ecuaciones lineales, tipo *sparse*, simétricos definidos positivos. Nuestro estudio ha incluido el aprendizaje, análisis y desarrollo de diferentes algoritmos en el campo de preconditionadores, ya sean explícitos o implícitos, así como en el tema descomposición de dominios.

La clase de sistemas que hemos estudiado en general se obtienen al discretizar los problemas evolutivos. Como estamos interesados en modelizar numéricamente los fenómenos de convección-difusión-reacción, hemos limitado nuestros ejemplos a este tipo de problemas.

Se ha empleado una formulación estabilizada de mínimos cuadrados junto con el método de Crank-Nicolson para discretizar los problemas de convección-difusión transitorios. Además hemos utilizado dos aplicaciones tecnológicas para ilustrar nuestro estudio. Así, podemos asegurar que usando los resultados de nuestra investigación se pueden realizar simulaciones numéricas de problemas reales 3D en forma más eficiente.

En el capítulo 4 se ha estudiado exhaustivamente las familias de factorizaciones incompletas de Cholesky. Hemos encontrado para los sistemas estudiados que PCG con un FIC de umbral es más eficiente, en requerimientos de memoria y tiempo de

CPU, que el método directo de Cholesky a partir de 30 000 incógnitas. De hecho, con las FIC de umbral se han obtenido mejores resultados en tiempos de CPU que con las FIC de memoria prescrita. Sin embargo, para los problemas de dispersión de contaminantes en la atmósfera ha sido necesario emplear una FIC de memoria prescrita con el fin de controlar los requerimientos de memoria conforme se aumenta el número de especies a estudiar. Otro resultado relevante del análisis ha sido observar que la eficiencia numérica de un preconditionador implícito está completamente determinada por su fase de aplicación en cada paso de tiempo, y no por el tiempo de CPU invertido en su fase de construcción.

Se ha planteado en el capítulo 5, la construcción de la inversa aproximada *sparse* simétrica (SSPAI) basada en la generalización de la SPAI de Montero et al. (2002). Se han propuesto cuatro diferentes estrategias para construir una inversa aproximada simétrica hasta obtener el algoritmo de la SSPAI. Además se han establecido fórmulas explícitas para calcular en forma eficiente y acumulativa \mathbf{m}_k^L y $\|\mathbf{A}\mathbf{m}_k^L - \mathbf{v}_k\|_F^2$ para una base arbitraria de \mathcal{S}_k . Este preconditionador explícito ha sido diseñado para matrices tridiagonales por bloques, SDP con múltiples lados derechos. Aunque la paralelización de la SSPAI está limitada por el número de bloques se ha encontrado que numéricamente podría superar a las FIC de umbral. No obstante sería interesante probar tal afirmación, ya que es costosa su fase de construcción a diferencia de las FIC. Por último, se ha observado que la inversa aproximada propuesta es inestable para matrices mal condicionadas.

Se ha combinado el método de descomposición de dominios solapados con la idea de activar y desactivar los subdominios, dando lugar al método multiplicativo de Schwarz (MSM) en dominios activos. El método desarrollado ha sido diseñado para los problemas transitorios de convección dominante que presentan un frente brusco en la solución, y donde esta sólo cambia en una región del dominio completo en cada paso de tiempo. Los criterios utilizados y el algoritmo generado están detallados en el capítulo 6. Al probar la estrategia iterativa propuesta se ha encontrado que la eficiencia numérica está completamente controlada por la fase de aplicación de cada subproblema en cada paso de tiempo, y por tanto, utilizar PCG+FIC(τ) compensa el hecho de realizar más iteraciones en solapamientos pequeños. Por ello, el MSM en dominios activos resolviendo los subproblemas con CG acelerado por

una FIC de umbral es la forma más eficiente de resolver los sistemas asociados a los problemas de convección-difusión transitorios para mallas finas con diferentes tamaños de solapamientos.

Por último, a manera de resumen y conclusión final se recomienda, para los grandes sistemas de ecuaciones lineales, tipo *sparse*, SDP asociados a los problemas de convección-difusión transitorios 3D, 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 donde τ se debe seleccionar en el rango $[0.005, 0.01]$. Para sistemas aún más grandes, el MSM en dominios activos + PCG+FIC(τ) es el más eficiente en cuanto a los requerimientos de memoria y tiempo de CPU. El umbral τ se debe seleccionar en el intervalo $[10^{-3}, 10^{-1}]$. Todos los resultados numéricos se han obtenido considerando un solo procesador.

7.2. Futuras líneas de investigación

La experiencia adquirida en esta tesis ha motivado a plantearnos las siguientes líneas de investigación:

1. Utilizar diferentes preconditionadores, ya sean explícitos o implícitos, al avanzar en el tiempo, puesto que hemos observado que la calidad del preconditionador se va deteriorando conforme transcurre el tiempo.
2. Estabilizar la SSPAI.
3. Investigar cómo afectan las técnicas de ordenamiento a la SSPAI, ya que creemos que éstas van a beneficiar la eficiencia del preconditionador propuesto, debido a que algunos métodos de reordenación incrementaron la calidad de la inversa aproximada *sparse* de Montero et al. (ver Flórez, García, González y Montero 2002).
4. Implementar y adecuar el MSM en dominios activos para los problemas de dispersión de contaminantes en la atmósfera, considerando primero la emisión constante y después, variable.

5. Imponer diferentes condiciones de contorno para aumentar la eficiencia del MSM en dominios activos (ver Houzeaux 2001).

A lo largo de esta tesis hemos entrado en contacto con dos técnicas que son altamente paralelizables: las inversas aproximadas y los métodos de descomposición de dominios. Por ello, es natural plantearnos trabajar en paralelo. Las futuras líneas de investigación en este tema serían las siguientes:

1. Paralelizar la SSPAI estabilizada y analizar otros preconditionadores simétricos como: la FSAI de Kolotilina y Yeremin (1993) y la SAINV de Benzi et al. (2000).
2. Analizar el comportamiento de los preconditionadores construidos mediante las técnicas de DD (ver Quarteroni y Valli 1999), y a partir de eso proponer uno nuevo para los problemas de convección-difusión-reacción transitorios.
3. Estudiar los métodos de descomposición de dominios utilizando una inversa aproximada para resolver los subdominios. Por ejemplo, podríamos combinar nuestras dos propuestas: el MSM en dominios activos resolviendo en paralelo los subproblemas con la SSPAI estabilizada.

7.3. Publicaciones

Las publicaciones ya realizadas o previstas asociadas a esta tesis son las siguientes:

- A. Rodríguez-Ferran y M. Luisa Sandoval. *Incomplete Cholesky factorizations for transient convection-diffusion problems*. The Fourth International Conference on Engineering Computational Technology. Lisboa, september 2004.
- A. Rodríguez-Ferran and M.L. Sandoval (2005), 'Numerical performance of incomplete factorizations for 3D transient convection-diffusion problems'. *Advances in Engineering Software*. En prensa.

-
- M. L. Sandoval, A. Rodríguez-Ferran, A. Suárez and G. Montero, ‘A symmetric sparse approximate inverse for block tridiagonal systems with multiple right hand side’. En preparación para la revista *Numerical Linear Algebra with Applications*.
 - M. L. Sandoval and A. Rodríguez-Ferran, ‘Multiplicative Schwarz method in active domains for 3D transient convection-diffusion problems’. En preparación.

Bibliografía

- Axelsson, O. (1972), ‘A generalized SSOR method’, *BIT, Nord. Tidskr. Inf.-behandl.* **12**, 443–467.
- Axelsson, O. (1996), *Iterative Solution Methods*, second edn, Cambridge University Press, New York.
- Axelsson, O. y Kolotilina, L. (1994), ‘Diagonally compensated reduction and related preconditioning methods’, *Numer. Linear Algebra Appl.* **1**(2), 155–177.
- Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. y van der Vorst, H. (1994), *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics, Philadelphia.
- Benzi, M. (2002), ‘Preconditioning techniques for large linear systems: a survey’, *J. Comput. Phys.* **182**(2), 418–477.
- Benzi, M., Cullum, J. K. y Tũma, M. (2000), ‘Robust approximate inverse preconditioning for the conjugate gradient method’, *SIAM J. Sci. Comput.* **22**(4), 1318–1332.
- Benzi, M., Meyer, D. y Tũma, M. (1996), ‘A sparse approximate inverse preconditioner for the conjugate gradient method’, *SIAM J. Sci. Comput.* **17**(5), 1135–1149.
- Benzi, M., Szyld, D. B. y van Duin, A. (1999), ‘Ordering for incomplete factorization preconditioning of nonsymmetric problem’, *SIAM J. Sci. Comput.* **20**(5), 1652–1670.
- Benzi, M. y Tũma, M. (1998), ‘A sparse approximate inverse preconditioner for nonsymmetric linear systems’, *SIAM J. Sci. Comput.* **19**(3), 968–994.
- Benzi, M. y Tũma, M. (1999), ‘A comparative study of sparse approximate inverse preconditioners’, *Appl. Num. Math.* **30**(2-3), 305–340.

- Benzi, M. y Tuma, M. (2003), 'A robust incomplete factorization preconditioner for positive definite matrices', *Numer. Linear Algebra Appl.* **10**(5-6), 385–400.
- Bjørstad, P. E., Dryja, M. y Vainikko, E. (1996), Parallel implementation of a Schwarz domain decomposition algorithm, in J. Wasniewski, J. Dongarra, K. Madsen y D. Olesen, eds, 'Applied Parallel Computing in Industrial Problems and Optimization', Springer. Lecture Notes in Computer Science volume 1184.
- Bose, A., Carey, G. F. y de Almeida, V. F. (2003), 'A class of parallel multiple-front algorithms on subdomains', *Int. J. Numer. Meth. Engng.* **56**(11), 1569–1592.
- Brakkee, E., Vuik, C. y Wesseling, P. (1995), An investigation of Schwarz domain decomposition using accurate and inaccurate solution of subdomains, Technical Report Report 95-18, Faculty of Technical Mathematics and Informatics, Delf University of Technology.
- Brakkee, E., Vuik, C. y Wesseling, P. (1998), 'Domain decomposition for the incompressible Navier-Stokes equations: solving subdomain problems accurately and inaccurately', *Int. J. Numer. Methods Fluids* **26**(10), 1217–1238.
- Brakkee, E. y Wilders, P. (1997), 'The influence of interface conditions on convergence of Krylov-Schwarz domain decomposition for the advection-diffusion equation', *J. Sci. Comput.* **12**(1), 11–30.
- Bramble, J. H., Pasciak, J. E., Wang, J. y Xu, J. (1991), 'Convergence estimates for product iterative methods with applications to domain decomposition', *Math. Comput.* **57**(195), 1–21.
- Cai, X.-C., Gropp, D. y Keyes, D. E. (1992), A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems, in F. Tony, Chan., D. E. Keyes, G. A. Meurant, J. S. Scroggs y R. G., eds, 'Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations', SIAM, pp. 224–234.
- Cai, X.-C., Gropp, D. y Keyes, D. E. (1994), 'A comparison of some domain decomposition algorithms and ILU preconditioned iterative methods for nonsymmetric elliptic problems', *Numer. Linear Algebra Appl.* **1**(5), 477–504.
- Cai, X.-C. y Saad, Y. (1996), 'Overlapping domain decomposition algorithms for general sparse matrices', *Numer. Linear Algebra Appl.* **3**(3), 221–237.
- Cai, X.-C. y Sarkis, M. (1998), 'Local multiplicative Schwarz algorithms for steady and unsteady convection-diffusion equations', *East-West J. Numer. Math.* **6**(1), 27–41.

- Carpentieri, B., Duff, I. S. y Giraud (2000), ‘Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism’, *Numer. Linear Algebra Appl.* **7**(7-8), 667–685.
- Carpentieri, B., Duff, I. S., Giraud, L. y Magolu, M. (2004), ‘Sparse symmetric preconditioners of dense linear systems in electromagnetism’, *Numer. Linear Algebra Appl.* **11**(8-9), 753–771.
- Chow, E. y Saad, Y. (1997), ‘Experimental study of ILU preconditioners for indefinite matrices’, *J. Comput. Appl. Math.* **86**(2), 387–414.
- Chow, E. y Saad, Y. (1998), ‘Approximate inverse preconditioners via sparse-sparse iterations’, *SIAM J. Sci. Comput.* **19**(3), 995–1023.
- Cosgrove, D., Díaz, J. y Griewank, A. (1992), ‘Approximate inverse preconditioning for sparse linear systems’, *Internat. J. Comput. Math.* **44**(1-4), 91–110.
- D’Azevedo, E., Forsyth, P. y Tang, W. (1992), ‘Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems’, *SIAM J. Matrix Anal. Appl.* **13**(3), 944–961.
- Díaz, J. y Macedo, J. C. (1989), ‘Fully vectorizable block preconditionings with approximate inverses for non-symmetric systems of equations’, *Int. J. Numer. Methods Eng.* **27**(3), 501–522.
- Dickinson, J. y Forsyth, P. (1994), ‘Preconditioned conjugate gradient methods for three-dimensional linear elasticity’, *Int. J. Numer. Methods Eng.* **37**(13), 2211–2234.
- Donea, J. y Huerta, A. (2003), *Finite Element Methods for Flow Problems*, John Wiley & Sons, Chichester.
- Dryja, M. (1989), An additive Schwarz algorithm for two- and three-dimensional finite element elliptic problems., in T. Chan, R. Glowinski, J. Periaux y O. Widlund, eds, ‘Proc. 2nd Int. Symp. on Domain Decomposition Methods’, SIAM, Los Angeles, CA, pp. 168–172.
- Dryja, M. y Widlund, O. (1991), Multilevel additive methods for elliptic finite element problems, in W. Hackbusch, ed., ‘Parallel Algorithms for PDEs (Proceedings of the 6th GAMM-Seminar Kiel, January 19-21, 1990)’, Vol. 31, Notes Numer. Fluid Mech., Vieweg Verlag, Wiesbaden, pp. 58–69.
- Dryja, M. y Widlund, O. (1994), ‘Domain decomposition algorithms with small overlap’, *SIAM J. Sci. Comput.* **15**(3), 604–620.

- Duff, I. S., Erisman, A. M. y K., R. J. (1992), *Direct Methods for Sparse Matrices*, second edn, Oxford University Press, Oxford.
- Duff, I. S. y Meurant, G. A. (1989), ‘The effect of ordering on preconditioned conjugate gradients’, *BIT* **29**(4), 635–657.
- Eijkhout, V. (1999), On the existence problem of incomplete factorisation methods, Technical Report Lapack working note 144, UT-CS-99-435, Computer Science Department, University of Tennessee.
- Escobar, J. M., Rodríguez, E., Montenegro, R., Montero, G. y González-Yuste, J. M. (2003), ‘Simultaneous untangling and smoothing of tetrahedral meshes’, *Comput. Methods Appl. Mech. Eng.* **192**(25), 2775–2787.
- Espedal, M., Tai, X. y Yan, N. (1998), ‘A hybrid domain decomposition method for advection-diffusion problems’, *Numer. Algorithms* **18**(3-4), 321–336.
- Flórez, E., García, D., González, L. y Montero, G. (2002), ‘The effect of orderings on sparse approximate inverse preconditioners for non-symmetric problems’, *Adv. Eng. Softw.* **33**(7-10), 611–619.
- Gambolati, G., Pini, G. y Ferronato, M. (2001), ‘Numerical performance of projection methods in finite element consolidation models’, *Int. J. Numer. Anal. Methods Geomech.* **25**(14), 1429–1447.
- Garbey, M. y Tromeur-Dervout, D. (2002), ‘On some Aitken-like acceleration of the Schwarz method’, *Int. J. Numer. Methods Fluids* **40**(12), 1493–1513.
- Gastaldi, F., Gastaldi, L. y Quarteroni, A. (1996), ADN and ARN domain decomposition methods for advection-diffusion equations, in P. E. Bjorstad, M. S. Espedal y D. E. Keyes, eds, ‘Ninth International Conference on Domain Decomposition Methods’, Wiley and Sons, pp. 334–341.
- Glowinski, R., Pan, T.-W. y Periaux, J. (1994), ‘A fictitious domain method for external incompressible viscous flow modelled by Navier-Stokes equations’, *Comput. Methods Appl. Mech. Eng.* **112**(1-4), 133–148.
- Golub, G. H. y Van Loan, C. F. (1996), *Matrix Computations*, third edn, The Johns Hopkins University Press, Baltimore and London.
- Gould, N. y Scott, J. (1998), ‘Sparse approximate-inverse preconditioners using norm-minimization techniques’, *SIAM J. Sci. Comput.* **19**(2), 605–625.
- Greenbaum, A. (1997), *Iterative Methods for Solving Linear Systems*, Vol. 17 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia.

- Grote, M. y Huckle, T. (1997), 'Parallel preconditioning with sparse approximate inverses', *SIAM J. Sci. Comput* **18**(3), 838–853.
- Guo, B. y Cao, W. (1998), 'Domain decomposition method for the h - p version finite element method', *Comput. Methods Appl. Mech. Eng.* **157**(3-4), 425–440.
- Gustafsson, I. (1978), 'A class of first order factorization methods', *BIT, Nord. Tidskr. Inf.-behandl.* **18**, 142–156.
- Hackbusch, W. (1994), *Iterative Solution of Large Sparse Systems of Equations*, Vol. 95 of *Applied Mathematical Sciences*, Springer-Verlag, New York, NY.
- Hossain, M. A. y Younge, D. R. (1992), 'Finite element modeling of single-solute activated-carbon adsorption', *Journal of Environmental Engineering* **118**, 238–252.
- Houzeaux, G. (2001), A geometrical domain decomposition method in computational fluid dynamics, PhD thesis, Universitat Politècnica de Catalunya, Barcelona, España.
- Huckle, T. K. (1998), 'Efficient computation of sparse approximate inverses', *Numer. Linear Algebra Appl.* **5**(1), 57–71.
- Huerta, A. y Donea, J. (2002), 'Time-accurate solution of stabilized convection-diffusion-reaction equations: I — Time and space discretization', *Commun. Numer. Methods Eng.* **18**(8), 565–573.
- Huerta, A., Rodríguez-Ferran, A., Sarrate, J., Díez, P. y Fernández-Méndez, S. (2001), Numerical modelling, a tool to improve the design of active carbon canisters, in 'Abstracts of the Sixth U.S. National Congress on Computational Mechanics', Dearborn, Michigan.
- Huerta, A., Roig, B. y Donea, J. (2002), 'Time-accurate solution of stabilized convection-diffusion-reaction equations: II — Accuracy analysis and examples', *Commun. Numer. Methods Eng.* **18**(8), 575–584.
- Johnson, C. (1987), *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge.
- Jones, M. y Plassmann, P. E. (1995), 'An improved incomplete Cholesky factorization', *ACM Trans. Math. Softw.* **21**(1), 5–17.
- Juárez, L., Glowinski, R. y Pan, T. (2002), 'Numerical simulation of the sedimentation of rigid bodies in an incompressible viscous fluid by Lagrange multiplier/fictitious domain methods combined with the Taylor-Hood finite element approximation', *J. Sci. Comput.* **17**(1-4), 683–694.

- Kelley, C. (1995), *Iterative Methods for Linear and Nonlinear Equations*, Vol. 16 of *Frontiers in applied mathematics*, SIAM, Philadelphia, PA.
- Kershaw, D. S. (1978), ‘The incomplete Cholesky-conjugate gradient method for iterative solution of systems of linear equations’, *J. Comput. Phys.* **26**(1), 43–65.
- Kolotilina, L. Y. y Yeremin, A. Y. (1993), ‘Factorized sparse approximate inverse preconditioning. I. theory.’, *SIAM J. Matrix Anal. Appl.* **14**(1), 45–58.
- Lasser, C. y Toselli, A. (2003), ‘An overlapping domain decomposition preconditioner for a class of discontinuous Galerkin approximations of advection-diffusion problems’, *Math. Comput.* **72**(243), 1215–1238.
- Layton, W. J. (1995), ‘Peaceman-Rachford procedure and domain decomposition for finite element problems’, *Numer. Linear Algebra Appl.* **2**(4), 363–393.
- Lin, C.-J. y Moré, J. (1999), ‘Incomplete Cholesky factorizations with limited memory’, *SIAM J. Sci. Comput.* **21**(1), 24–45.
- Lions, P. L. (1988), On the Schwarz alternating method I, in R. Glowinski, G. H. Golub, G. Meurant y J. Periaux, eds, ‘Domain decomposition methods for partial differential equations’, SIAM Publications, pp. 1–42.
- Manteuffel, T. (1980), ‘An incomplete factorization technique for positive definite linear systems’, *Math. Comput.* **34**(150), 473–497.
- Meijerink, J. y van der Vorst, H. (1977), ‘An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix’, *Math. Comput.* **31**(137), 148–162.
- Meurant, G. (1999), *Computer Solution of Large Linear Systems*, Vol. 28 of *Studies in Mathematics and its Applications*, Elsevier Science B.V., Amsterdam; The Netherlands.
- Montenegro, R., Montero, G., Escobar, J. M. y Rodríguez, E. (2002), ‘Efficient strategies for adaptive 3-D mesh generation over complex orography’, *Neural, Parallel & Scientific Comp.* **10**(1), 57–76.
- Montero, G., González, L., Flórez, E., García, M. D. y Suárez, A. (2002), ‘Approximate inverse computation using Frobenius inner product’, *Numer. Linear Algebra Appl.* **9**(3), 239–247.
- Montero, G., Rodríguez, E., Montenegro, R., Escobar, J. M. y González-Yuste, J. M. (2005), ‘Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model’, *Adv. Eng. Softw.* **36**(1), 3–10.

- Munksgaard, N. (1980), 'Solving sparse symmetric sets of linear equations by pre-conditioned conjugate gradients', *ACM Trans. Math. Softw.* **6**(2), 206–219.
- Pavarino, L. F. (1992), Domain decomposition algorithms for the p-version finite element method for elliptic problems, PhD thesis, Courant Institute of Mathematical Science, New York University, USA.
- Pérez-Foguet, A. y Huerta, A. (2006), Mathematical modelling of mass transport equations in fixed-bed absorbers, in M. P. A. Di Bucchianico, R.M.M. Matthéij, ed., 'Progress in industrial mathematics to ECMI 2004', Mathematics in industry ; 8. European consortium for mathematics in industry, Springer, Berlin.
- Quarteroni, A. y Valli, A. (1994), *Numerical Approximation of Partial Differential Equations*, Vol. 23 of *Springer Series in Computational Mathematics*, Springer-Verlag, Berlin.
- Quarteroni, A. y Valli, A. (1999), *Domain Decomposition Methods for Partial Differential Equations*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York.
- Quarteroni, A., Veneziani, A. y Zunino, P. (2002), 'A domain decomposition method for advection-diffusion processes with application to blood solutes', *SIAM J. Sci. Comput.* **23**(6), 1959–1980.
- Rapin, G. y Lube, G. (2002), Comparison of two iterative substructuring methods for advection-diffusion problems, in N. Debit, M. Garbey, R. Hoppe, J. Périaux y Y. Kuznetsov, eds, 'Papers of the thirteenth international conference on domain decomposition methods, Lyon, France, October 9-12, 2000.', Theory Eng. Appl. Comput. Methods., pp. 437–444.
- Rodríguez, E. (2004), Modelización y simulación numérica de campos de viento mediante elementos finitos adaptativos en 3-D, PhD thesis, Universidad de Las Palmas de Gran Canaria, España.
- Rodríguez, E., Montero, G., Montenegro, R., Escobar, J. M. y González-Yuste, J. M. (2002), Parameter estimation in a three-dimensional wind field model using genetic algorithms, in P. Sloot, C. Tan, J. Dongarra y A. Hoekstra, eds, 'Computational Science - ICCS 2002. International Conference, Amsterdam, The Netherlands, April 21-24, 2002. Proceedings, Part I', Vol. 2329 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 950–959.
- Rodríguez-Ferran, A. y Sandoval, M. L. (2005), 'Numerical performance of incomplete factorizations for 3D transient convection-diffusion problems', *Adv. Eng. Softw.* . En prensa.

- Rodríguez-Ferran, A., Sarrate, J. y Huerta, A. (2004), 'Numerical modelling of void inclusions in porous media', *Int. J. Numer. Methods Eng.* **59**(4), 577–596.
- Saad, Y. (1994a), 'ILUT: a dual threshold incomplete LU factorization', *Numer. Linear Algebra Appl.* **1**(4), 387–402.
- Saad, Y. (1994b), SPARSKIT: a basic tool kit for sparse matrix computations, Technical Report Version 2, CSRD, university of Illinois and IRACS, NASA Ames Research Center.
- Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia.
- Saad, Y. y van der Vorst, H. A. (2000), 'Iterative solution of linear systems in the 20-th century', *J. Comput. Appl. Math* **123**(1-2), 1–33.
- Saint-Georges, P., Warzee, G., Beauwens, R. y Notay, Y. (1996), 'High-performance PCG solvers for FEM structural analysis', *Int. J. Numer. Methods Eng.* **39**(8), 1313–1340.
- Salvini, S. A. y Shaw, G. J. (1995), An evaluation of new NAG Library solvers for large sparse symmetric linear systems, Technical Report TR1/95, NAG Ltd, Oxford.
- Sanín, N. (2003), Construcción de un modelo tridimensional para ajuste de campos de viento y dispersión de contaminantes en la atmósfera, PhD thesis, Universidad de Las Palmas de Gran Canaria, España.
- Sanín, N. y Montero, G. (2004), Air pollution modelling using finite differences in a terrain conformal coordinate system, in B. Topping y C. M. Soares, eds, 'Proceedings of The Fourth International Conference on Engineering Computational Technology. Lisboa, septiember 7-9', Civil-Comp Press, Stirling, Scotland.
- Sosonkina, M., Melson, J. T., Saad, Y. y Watson, L. (2000), 'Preconditioning strategies for linear systems arising in tire design', *Numer. Linear Algebra Appl.* **7**(8), 743–757.
- Suárez, A. (1995), Contribución a algoritmos de biortogonalización para la resolución de sistemas de ecuaciones lineales, PhD thesis, Universidad de Las Palmas de Gran Canaria, España.
- Sun, H. y Tang, W.-P. (1996), 'An overdetermined Schwarz alternating method', *SIAM J. Sci. Comput.* **17**(4), 884–905.

- Tang, W.-P. (1992), 'Generalized Schwarz splittings', *SIAM J. Sci. Comput.* **13**(2), 573–595.
- Toselli, A. (2001), 'FETI domain decomposition methods for scalar advection-diffusion problems', *Comput. Methods. Appl. Mech. Eng.* **190**(43-44), 5759–5776.
- Trefethen, L. y Bau III, D. (1997), *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, SIAM, Philadelphia.
- U.S. Environmental Protection Agency (2006),
<http://www.epa.gov/ebtpages/airairpollutants.html>.
(Consultada el 10-04-2006).
- van der Vorst, H. A. (1981), 'Iterative solution for certain sparse linear systems with a non-symmetric matrix arising from PDE-problems', *J. Comput. Phys.* **44**(1), 1–19.
- Vázquez, E. F. (2003), Construcción de inversas aproximadas tipo sparse basada en la proyección ortogonal de Frobenius para el preconditionamiento de sistemas de ecuaciones no simétricos, PhD thesis, Universidad de Las Palmas de Gran Canaria, España.
- Wang, L. y Zhang, J. (2003), 'A new stabilization strategy for incomplete LU preconditioning of indefinite matrices', *Applied Mathematics and Computation* **144**, 75–87.
- Widlund, O. (1988), A comparison of some domain decomposition and iterative refinement algorithms for elliptic finite element problems, Technical Report BSC 88/15, IBM Bergen Scientific Centre, Bergen, Norway.
- Wikipedia, *the free encyclopedia* (2006),
http://en.wikipedia.org/wiki/Earth's_atmosphere.
(Consultada el 10-04-2006).
- Zlatev, Z. (1995), *Computer Treatment of Large Air Pollution Models*, Kluwer Academic Publishers, Dordrecht.