Congreso de Métodos Numéricos en Ingeniería 2005
Granada, 4 a 7 de julio, 2005

PRECONDICIONAMIENTO DE SISTEMAS DE ECUACIONES VARIABLES

A. Suárez*, E. Rodríguez, G. Montero, M.D. García, E. Flórez

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería
Universidad de Las Palmas de Gran Canaria
Edificio Central del Parque Científico Tecnológico. Campus Universitario de Tafira
35017-Las Palmas de Gran Canaria
e-mail: asuarez,barrera,gustavo,lgarcia,eflorez@dma.ulpgc.es
web: http://www.iusiani.ulpgc.es

Palabras clave: Sistemas de ecuaciones lineales, Matrices variables, Gradiente Conjugado precondicionado, Inversa aproximada factorizada

Resumen. *Se plantea el precondicionamiento de sistemas de ecuaciones de la forma $A_\varepsilon x_\varepsilon = b_\varepsilon$ con $A_\varepsilon = M + \varepsilon N$ simétrica definida positiva. Hay dos estrategias extremas para efectuar este precondicionamiento. Como solución más simple podríamos construir un precondicionador inicial que se aplica sin ninguna modificación en la resolución de todos los sistemas lineales sucesivos. La efectividad de este precondicionador disminuirá progresivamente conforme aumenten las variaciones de ε, dando lugar a una convergencia lenta en las resoluciones. De otra forma, podríamos usar un precondicionador diferente para cada sistema lineal generado, lo que resultaría una solución más costosa. Proponemos una solución intermedia construyendo un precondicionador dependiente de ε a partir de una inversa aproximada de A_{ε_0}, que se pueda actualizar fácilmente para cada sistema. Se presentan distintos experimentos numéricos que ilustran la efectividad del precondicionador obtenido.*

1 INTRODUCCIÓN

La aplicación de técnicas de discretización a problemas de ecuaciones en derivadas parciales da lugar a sistemas de ecuaciones lineales sparse. Los métodos iterativos basados en los llamados subespacios de Krylov [1, 2] con adecuadas técnicas de precondicionamiento [3, 4], que mejoran sustancialmente su convergencia, resultan muy efectivos en estos casos. Como es sabido, para sistemas con matriz simétrica definida positiva, caso de nuestras aplicaciones, es el gradiente conjugado el que proporciona mejores resultados.
En muchos casos de problemas de discretización y, en particular en la simulación numérica con modelos de masa consistente de campos de viento [5, 6], que se tratan en los experimentos numéricos, surgen sistemas de la forma,

$$(M + \varepsilon N)\, x_\varepsilon = b_\varepsilon \tag{1}$$

que hay que resolver para distintos valores de ε, donde M y N son matrices constantes para un nivel de discretización dado.
Para precondicionar estos sistemas se podrían aplicar, en principio, dos estrategias extremas. Por un lado, construiríamos un único precondicionador para un cierto valor de ε y lo aplicaríamos para la resolución de los sucesivos sistemas. Esto, en general, conduce a convergencias más lentas conforme van variando los valores de ε. En el extremo opuesto usaríamos un precondicionador diferente para cada sistema, obtenido para cada valor de ε, lo que resultará muy costoso. En este trabajo, proponemos una solución intermedia, construyendo un precondicionador basado en la inversa aproximada factorizada [7, 8, 9] que pueda ser actualizado fácilmente para cada valor de ε y cuya aplicación en el gradiente conjugado de lugar a una velocidad de convergencia comprendida entre las correspondientes a las soluciones mencionadas.
En el caso particular de $A_\varepsilon = M + \varepsilon D$ con D matriz diagonal, Meurant [10] ha desarrollado un método para obtener sucesivos precondicionadores variando solamente la ε a partir de una factorización incompleta de Cholesky de la matriz M. Por otro lado, Benzi [11] propone una técnica para actualizar un precondicionador basado en la inversa aproximada factorizada usando el algoritmo SAINV [8] para el caso de $A_\varepsilon = M + \varepsilon I$ con I matriz unidad. Para el caso que nos ocupa de matrices $A_\varepsilon = M + \varepsilon N$, la generalización de esos procedimientos conlleva a unas simplificaciones adicionales que son menos significativas cuando usamos la inversa aproximada. Así, en nuestro estudio hemos seguido un camino paralelo al propuesto por Benzi en [11].
En la sección 2 describimos el algoritmo SAINV para la construcción de la inversa aproximada factorizada. El precondicionador para nuestro sistema de matriz A_ε y su aplicación en el algoritmo CG se detalla en la sección 3. La sección 4 ilustra la aplicación de esta técnica con algunos experimentos numéricos. Finalmente, las conclusiones y líneas futuras se exponen en la sección 5.

2 ALGORITMO SAINV

El algoritmo AINV [7] construye una inversa aproximada factorizada de una matriz A basándose en la obtención por congruencia de una forma diagonal de la misma.
Para ello, a partir del conjunto de vectores linealmente independientes $\{e_1, e_2, \ldots, e_n\} \in \mathbb{R}^n$, obtiene un conjunto $\{z_1, z_2, \ldots, z_n\} \in \mathbb{R}^n$ de vectores A-conjugados por un proceso de A-conjugación de Gram-Schmidt.
Siendo Z la matriz triangular superior, $Z = [z_1, z_2, \ldots, z_n]$ queda,

$$Z^T A Z = D = diag\,(d_1, d_2, \ldots, d_n)$$

donde $d_j = z_j^T A z_j > 0, \qquad 1 \leq j \leq n.$
El precondicionador AINV se obtiene realizando el proceso de cálculo de los vectores z_i de forma incompleta, descartando, en cada paso, las entradas respectivas menores que una cierta tolerancia escogida, $0 < \delta < 1$. Resulta así una matriz sparse aproximada de Z, que denotaremos por $\tilde{Z}$ y entonces,

$$A^{-1} \approx \tilde{Z} \tilde{D}^{-1} \tilde{Z}^T,$$

Esta aproximación causa una pérdida de A-ortogonalidad de los vectores z_i. El algoritmo, en el caso que A no sea M-matriz, puede conducir a problemas de inestabilidad con entradas en la diagonal nulas o negativas.
Benzi, propone entonces el algoritmo SAINV, que a continuación se detalla, matemáticamente equivalente al AINV y que, con un coste un poco mayor, permite obtener la inversa aproximada para matrices más generales como SDP, sin esos problemas de inestabilidad.

ALGORITMO SAINV

(1) Sea $z_i^{(0)} = e_i \quad (1 \leq i \leq n)$
(2) Para $i = 1, 2, \ldots, n$ hacer
(3) $\quad v_i := A z_i^{(i-1)}$
(4) $\quad$ Para $j = i, i+1, \ldots, n$ hacer
(5) $\quad\quad p_j^{(i-1)} := v_i^T z_j^{(i-1)}$
(6) $\quad$ Fin
(7) $\quad$ si $i = n$ ir a (12)
(8) $\quad$ Para $j = i+1, \ldots, n$ hacer
(9) $\quad\quad z_j^{(i)} := z_j^{(i-1)} - \left(\frac{p_j^{(i-1)}}{p_i^{(i-1)}} \right) z_i^{(i-1)}$
(10) $\quad$ Fin
(11) Fin
(12) Sea $z_i := z_i^{(i-1)}$ y $p_i := p_i^{(i-1)}$, para $(1 \leq i \leq n)$

$$Z = [z_1, z_2, \ldots, z_n] \;\; y \;\; D = diag\,(p_1, p_2, \ldots, p_n)$$

3 PRECONDICIONAMIENTO DE LA MATRIZ VARIABLE

Buscamos un precondicionador para el sistema de matriz variable simétrica definida positiva, $(M + \varepsilon N)\, x = b$, para su posterior aplicación en el algoritmo del Gradiente Conjugado precondicionado. Seguiremos una línea similar a la propuesta por Benzi [11] para el sistema $(M + \varepsilon I)\, x = b$.
Con el algoritmo SAINV obtenemos una inversa aproximada factorizada de M,

$$M^{-1} \approx \tilde{Z} \tilde{D}^{-1} \tilde{Z}^T = P^{-1}$$

Consideremos un precondicionador para $A_\varepsilon = M + \varepsilon N$ de la forma,

$$P_\varepsilon^{-1} = \tilde{Z} \left(\tilde{D} + \varepsilon E \right)^{-1} \tilde{Z}^T, \tag{2}$$

donde E es una matriz genérica simétrica a determinar, fácilmente computable, que haga que $\left(\tilde{D} + \varepsilon E \right)$ sea SDP y tal que el producto P_ε^{-1} por vector que figura en el CG no suponga un coste elevado.
A efectos de definirla y supuesto que $P^{-1} = Z D^{-1} Z^T$ sea la inversa exacta, establezcamos la diferencia,

$$P_\varepsilon - A_\varepsilon = Z^{-T} (D + \varepsilon E) Z^{-1} - (M + \varepsilon N) = \varepsilon \left(Z^{-T} E Z^{-1} - N \right). \tag{3}$$

Si tomamos $E = Z^T N Z$ resultaría el precondicionador ideal $P_\varepsilon^{-1} = A_\varepsilon^{-1}$. Evidentemente, esto no es viable, dado que no conocemos la matriz exacta Z sino una aproximación $\tilde{Z}$, pero sugiere como matriz E en P_ε^{-1},

$$E = \tilde{Z}^T N \tilde{Z} \tag{4}$$

de tal forma que E cumpla las condiciones necesarias mencionadas anteriormente.
En lugar de partir de una inversa aproximada de M, correspondiente a $\varepsilon = 0$ en $A_\varepsilon = M + \varepsilon N$, podemos obtener inicialmente una aproximada de $A_{\varepsilon_0} = M + \varepsilon_0 N$. Las sucesivas matrices para los correspondientes valores de ε se escribirían $A_\varepsilon = M + \varepsilon N = A_{\varepsilon_0} + \Delta\varepsilon N$, donde $\Delta\varepsilon = \varepsilon - \varepsilon_0$. Dado que ε es siempre un número positivo, la matriz A_ε, obviamente, es definida positiva aunque $\Delta\varepsilon$ sea negativo.
Resultaría así que,

$$A_{\varepsilon_0}^{-1} \approx \tilde{Z}\tilde{D}^{-1}\tilde{Z}^T = P_{\varepsilon_0}^{-1}$$

y

$$P_\varepsilon^{-1} = \tilde{Z}\left(\tilde{D} + \Delta\varepsilon E\right)^{-1}\tilde{Z}^T,$$

con $E = \tilde{Z}^T N \tilde{Z}$, al igual que antes y con las mismas limitaciones.
Una opción para establecer la matriz E, con estos requisitos, es tomar una aproximación de $\tilde{Z}$ que nombraremos como $\tilde{Z}_k$, con $k > 1$, que se obtiene extrayendo las $k-1$ diagonales superiores de $\tilde{Z}$ y para N la aproximación N_h, extrayendo su diagonal principal si $h = 1$ y las $k-1$ diagonales secundarias para $h > 1$. Así,

$$E_{h,k} = \tilde{Z}_k^T N_h \tilde{Z}_k \tag{5}$$

En la práctica y a efectos de no incrementar el coste por iteración en el CG, parecen ser viables las parejas $h = 1$ y $k = 2$ o $h = 2$ y $k = 1$, que dan lugar, respectivamente, a las matrices $E_{1,2}$ y $E_{2,1}$, tridiagonales. Más simplificación se obtendría con $h = 1$ y $k = 1$, con $E_{1,1}$ matriz diagonal.
Otra posibilidad, para valores pequeños de ε, siempre a efectos de reducir el coste computacional, es efectuar un desarrollo de $\left(\tilde{D} + \varepsilon E\right)^{-1}$,

$$\begin{aligned}\left(\tilde{D} + \varepsilon E\right)^{-1} &= \left[\tilde{D}^{\frac{1}{2}}\left(I + \varepsilon\tilde{D}^{-\frac{1}{2}}E\tilde{D}^{-\frac{1}{2}}\right)\tilde{D}^{\frac{1}{2}}\right]^{-1} \\ &= \tilde{D}^{-\frac{1}{2}}\left[I - \varepsilon\tilde{D}^{-\frac{1}{2}}E\tilde{D}^{-\frac{1}{2}} + \left(\varepsilon\tilde{D}^{-\frac{1}{2}}E\tilde{D}^{-\frac{1}{2}}\right)^2 + \cdots\right]\tilde{D}^{-\frac{1}{2}}\end{aligned} \tag{6}$$

Despreciando los términos de grado mayor que uno,el precondicionador resulta,

$$P_\varepsilon^{-1} = \tilde{Z}\tilde{D}^{-\frac{1}{2}}\left(I - \varepsilon\tilde{D}^{-\frac{1}{2}}E\tilde{D}^{-\frac{1}{2}}\right)\tilde{D}^{-\frac{1}{2}}\tilde{Z}^T \tag{7}$$

Hacemos notar que en esta aproximación es posible usar $\tilde{Z}$ en lugar de $\tilde{Z}_k$ dado que la computación de $\left(\tilde{D} + \varepsilon E\right)^{-1}$ no es necesaria. Se puede expresar con $E = \tilde{Z}^T N \tilde{Z}$,

$$P_\varepsilon^{-1} = \tilde{Z}\tilde{D}^{-1}\left(I - \varepsilon N\tilde{Z}\tilde{D}^{-1}\tilde{Z}^T\right) \tag{8}$$

$$P_{\varepsilon}^{-1} = P^{-1}\left(I - \varepsilon N P^{-1}\right) \tag{9}$$

O bien, si hemos partido de la inversa aproximada para ε_0,

$$P_{\varepsilon}^{-1} = P_{\varepsilon_0}^{-1}\left(I - \Delta\varepsilon N P_{\varepsilon_0}^{-1}\right) \tag{10}$$

4 EXPERIMENTOS NUMÉRICOS

En esta sección presentamos los resultados que se obtienen aplicando los precondicionadores propuestos para la resolución, mediante el Gradiente Conjugado, de los sistemas de ecuaciones lineales correspondientes a los tres ejemplos numéricos que a continuación se definen, SHERMAN3 [12], LightTruck [13] y Modelización de un campo de viento [5, 6].

Dado que la matriz del sistema que consideramos en este trabajo es de la forma $A_\varepsilon = M + \varepsilon N$, con ε variable, en los ejemplos 1 y 2 hemos elegido para M las respectivas matrices SDP que se definen en cada uno de ellos y para las correspondientes matrices N, a efectos de garantizar que, asimismo, sean SDP, establecemos,

$$N = M \odot R \qquad \text{con} \qquad R \begin{cases} r_{ij} = r_{ij} & \text{si} \quad i \neq j \\ r_{ii} > \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} r_{ij} \end{cases}$$

donde $\odot$ representa el producto de Hadamard [14] y cada r_{ij} se genera aleatoriamente de forma que $0 < r_{ij} < 1$.

Todos los experimentos fueron realizados en una estación de trabajo HPRX5670 con procesador ITANIUM, utilizando Fortran77 con doble precisión. En la resolución de todos los sistemas inicializamos el algoritmo CG con el vector nulo, interrumpiendo el cálculo iterativo cuando la norma-2 del vector residuo de la última iteración era, $\|r_k\|_2 \leq 10^{-10}\|r_0\|_2$ o bien cuando se superan las 5000 iteraciones.

Los resultados para los distintos problemas se han dispuesto en tablas para un amplio campo de valores de ε, indicando el número de iteraciones alcanzadas y el tiempo computacional en segundos para cada uno de los casos. Estos valores se muestran para los precondicionadores que se proponen en este trabajo, que hemos designado por SAINV$_{11}$, SAINV$_{12}$ y SAINV$_{21}$, según las aproximaciones E_{11}, E_{12} y E_{21} de la matriz E descrita en la sección 3 y como SAINV-span la aproximación de $(D + \varepsilon E)^{-1}$ por el desarrollo de primer orden que se estudia en la misma sección, comparándolos con los precondicionadores full SAINV consistente en calcular y aplicar en cada caso la inversa aproximada de la matriz del sistema A_ε y SAINV(A_{ε_0}) obtenido como inversa aproximada de la matriz inicial y aplicado sin ninguna modificación a la resolución de cada uno de los sistemas. La tolerancia adoptada para despreciar entradas en las inversas aproximadas, ha sido en todos los casos de $\delta = 0.1$. En los casos de los precondicionadores SAINV$_{12}$ y SAINV$_{21}$, para efectuar los correspondientes productos matriz inversa por vector en cada iteración del CG, hemos efectuado una factorización de Cholesky de la matriz tridiagonal y dos procesos de remonte.

4.1 Ejemplo1 (SHERMAN3)

Extraído de la colección de la *Harwell-Boeing Sparse Matrix Collection (Release I)* [12], hemos elegido SHERMAN3 que proviene de un modelo de simulación tridimensional relativo a un problema de reservas petrolíferas y que da lugar a un sistema de 5005 ecuaciones con 20033 entradas no nulas.
En la tabla 1 se puede observar que para valores de $\varepsilon < 1$, los mejores resultados los ofrece el precondicionador SAINV(A_{ε_0}). Sin embargo, conforme ε aumenta y el sumando εN va adquiriendo más peso en A_ε, el número de iteraciones y por consiguiente, el coste computacional se incrementa progresivamente, de tal forma que para esos valores, es el SAINV$_{11}$ el que obtiene mejores tiempos y menor número de iteraciones.

ε		full SAINV	SAINV-span	SAINV$_{11}$	SAINV$_{12}$	SAINV$_{21}$	SAINV(A_{ε_0})
0	n^{o}Iter.	263	–	–	–	–	–
	t(seg.)	4.51	–	–	–	–	–
10^{-5}	n^{o}Iter.	238	239	236	237	237	237
	t(seg.)	4.45	0.96	0.74	0.80	0.80	**0.53**
10^{-4}	n^{o}Iter.	20	195	198	223	198	193
	t(seg.)	4.37	0.79	0.65	0.76	0.70	**0.43**
10^{-3}	n^{o}Iter.	120	207	151	208	151	159
	t(seg.)	4.19	0.83	0.54	0.73	0.59	**0.36**
10^{-2}	n^{o}Iter.	53	384	95	171	96	123
	t(seg.)	4.03	1.54	0.42	0.64	0.45	**0.28**
10^{-1}	n^{o}Iter.	22	>5000	71	165	72	155
	t(seg.)	3.91	–	0.37	0.62	0.39	**0.35**
10^{0}	n^{o}Iter.	>5000	>5000	122	701	127	393
	t(seg.)	–	–	**0.49**	1.96	0.53	0.87
10^{1}	n^{o}Iter.	36	>5000	176	>5000	177	1005
	t(seg.)	3.87	–	**0.60**	–	0.75	2.22
10^{2}	n^{o}Iter.	>5000	>5000	198	>5000	200	2465
	t(seg.)	–	–	**0.65**	–	0.75	5.42
10^{3}	n^{o}Iter.	>5000	>5000	193	>5000	198	>5000
	t(seg.)	–	–	**0.64**	–	0.70	–

Tabla 1: Ejemplo1: SHERMAN3. Número de iteraciones y tiempo de CPU (en segundos) del Gradiente Conjugado para los distintos precondicionares

4.2 Ejemplo2 (LightTruck)

Correspondiente a una simulación numérica de un filtro de carbono activo en 3D, realizada en el Laboratorio de Cálculo Numérico (LaCaN) de la Universidad Politécnica de Cataluña [13] donde han desarrollado un modelo de convección-difusón-reacción que responde a la ecuación,

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \nu \Delta u + \sigma\left(u\right)\, u = f\left(u\right)$$

donde u es la concentración de hidrocarburos en el aire, $\mathbf{v}$ representa el campo de velocidades del aire, que se calcula previamente resolviendo un problema de flujo potencial y ν es el coeficiente de difusión y $\sigma\left(u\right)\, u$ y $f\left(u\right)$ son respectivamente los términos de reacción y fuente. Para un cierto paso de tiempo, discretizando por elementos finitos se obtiene un sistema simétrico de

17914 ecuaciones.
Como se aprecia en la tabla 2, igual que en el ejemplo anterior, para valores pequeños de ε el SAINV(A_{ε_0}) es el mejor precondicionador. Para valores de $\varepsilon > 1$ los precondicionadores SAINV$_{12}$ y SAINV$_{21}$ comparten los mejores resultados. Obsérvese que para $\varepsilon > 10^{-1}$, el SAINV-span es un precondicionador aceptable, sin embargo, como es lógico, para valores de ε mayores, el error que se comete despreciando el término el ε^2 en el desarrollo en serie es grande y su aplicación no conduce a la convergencia.

ε		full SAINV	SAINV-span	SAINV$_{11}$	SAINV$_{12}$	SAINV$_{21}$	SAINV(A_{ε_0})
0	n^{o}Iter.	83	–	–	–	–	–
	t(seg.)	236.60	–	–	–	–	–
10^{-6}	n^{o}Iter.	84	84	84	84	84	84
	t(seg.)	236.61	3.46	4.11	4.19	4.18	**1.78**
10^{-5}	n^{o}Iter.	89	88	89	89	89	89
	t(seg.)	236.73	3.62	4.22	4.29	4.30	**1.89**
10^{-4}	n^{o}Iter.	86	87	87	87	87	87
	t(seg.)	236.67	3.58	4.18	4.26	4.25	**1.84**
10^{-3}	n^{o}Iter.	72	68	68	68	68	67
	t(seg.)	236.37	2.81	3.78	3.83	3.84	**1.43**
10^{-2}	n^{o}Iter.	34	35	35	35	35	33
	t(seg.)	235.48	1.48	3.08	3.11	3.11	**0.73**
10^{0-1}	n^{o}Iter.	18	>5000	31	30	31	42
	t(seg.)	234.81	–	3.00	3.00	3.02	**0.92**
10^{0}	n^{o}Iter.	13	>5000	62	55	62	91
	t(seg.)	234.44	–	3.65	3.55	3.70	**1.93**
10^{1}	n^{o}Iter.	11	>5000	94	81	96	217
	t(seg.)	234.36	–	4.33	**4.12**	4.45	4.53
10^{2}	n^{o}Iter.	11	>5000	121	95	124	305
	t(seg.)	234.36	–	4.90	**4.43**	5.06	6.35
10^{3}	n^{o}Iter.	11	>5000	124	111	120	326
	t(seg.)	234.30	–	4.96	**4.78**	4.97	6.78
10^{4}	n^{o}Iter.	11	>5000	130	184	122	328
	t(seg.)	233.99	–	5.09	6.39	**5.01**	6.83
10^{5}	n^{o}Iter.	13	>5000	1241	510	132	325
	t(seg.)	234.38	–	5.33	13.57	**5.24**	6.76
10^{6}	n^{o}Iter.	15	>5000	144	>5000	138	326
	t(seg.)	234.41	–	5.39	–	**5.37**	6.78

Tabla 2: Ejemplo2: LightTruck. Número de iteraciones y tiempo de CPU (en segundos) del Gradiente Conjugado para los distintos precondicionares

4.3 Ejemplo3 (Modelización de un campo de viento)

Las matrices M y N son definidas por la simulación numérica con un modelo de masa consistente de campo de viento estudiada en [5, 6]. La modelización conduce al problema elíptico,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \varepsilon \frac{\partial^2 u}{\partial z^2} = f$$

con condiciones Neumann en el terreno y Dirichlet nulas en el resto del contorno.
En la discretización por MEF se obtiene el sistema $(M + \varepsilon N)\, x = b$ de 43954 ecuaciones con

M y N matrices SDP correspondientes, respectivamente, a las contribuciones de las derivadas relativas a las variables horizontales y a las contribuciones de la derivada relativa a la variable vertical.
En la tabla 3 observamos que para valores de $\varepsilon > 1$ los precondicionadores propuestos, SAINV$_{11}$, SAINV$_{12}$ y SAINV$_{21}$ aventajan al precondicionador SAINV(A_{ε_0}), concretamente es SAINV$_{11}$ el que proporciona mejores tiempos.

ε		full SAINV	SAINV-span	SAINV$_{11}$	SAINV$_{12}$	SAINV$_{21}$	SAINV(A_{ε_0})
0	n^{o}Iter.	254	–	–	–	–	–
	t(seg.)	1542.47	–	–	–	–	–
10^{-5}	n^{o}Iter.	254	254	254	254	254	254
	t(seg.)	1543.79	30.59	21.28	21.98	22.04	**15.94**
10^{-4}	n^{o}Iter.	254	254	254	254	254	254
	t(seg.)	1537.59	30.59	21.58	21.99	22.03.70	**15.94**
10^{-3}	n^{o}Iter.	153	253	253	253	253	253
	t(seg.)	1539.87	30.47	21.22	21.92	21.97	**15.87**
10^{-2}	n^{o}Iter.	237	235	237	237	237	235
	t(seg.)	1547.28	28.31	20.32	20.88	20.95	**14.75**
10^{-1}	n^{o}Iter.	182	289	183	183	183	200
	t(seg.)	1540.39	34.70	16.85	17.37	17.41	**12.58**
10^{0}	n^{o}Iter.	137	>5000	191	196	191	307
	t(seg.)	1539.90	–	**17.35**	18.23	17.94	18.91
10^{1}	n^{o}Iter.	133	>5000	349	345	349	590
	t(seg.)	1591.51	–	**27.21**	27.94	28.26	36.46
10^{2}	n^{o}Iter.	229	>5000	786	846	815	1255
	t(seg.)	1622.76	–	**54.48**	60.60	58.69	78.29
10^{3}	n^{o}Iter.	332	>5000	1196	1748	1197	1777
	t(seg.)	1632.54	–	**80.07**	119.38	83.64	110.79

Tabla 3: Ejemplo3: Modelización de un campo de viento. Número de iteraciones y tiempo de CPU (en segundos) del Gradiente Conjugado para los distintos precondicionares

5 CONCLUSIONES Y LÍNEAS FUTURAS

- Para valoreS pequeños de ε no parece rentable la construcción de un precondicionador variable. Bastaría con aplicar a los sucesivos sistemas el precondicionador SAINV(A_{ε_0}), pero para valores de $\varepsilon > 1$, los precondicionadores propuestos SAINV$_{11}$, SAINV$_{12}$ y SAINV$_{21}$, y en particular el primero de ellos, debido a su reducido coste computacional, presentan notables ventajas sobre SAINV(A_{ε_0}) y obviamente sobre el precondicionador full SAINV.

- Siguiendo el trabajo de Meurant [10], se podría estudiar la aplicación de un precondicionador variable basado en la factorización incompleta de Cholesky a este tipo de matrices $(M + \varepsilon N)$.

- Asimismo sería interesante estudiar el efecto de una adecuada reordenación en el comportamiento de ambos precondicionadores, inversa aproximada factorizada y la factorización incompleta.

REFERENCIAS

[1] Y.Saad, *Iterative methods for sparse linear systems*, PWE Publishing Company, Boston, (1996).

[2] N.M. Nachttigal, S.C. Reddy and L.N. Trefethen. How fast are nonsymmetric matrix iterations?, *SIAM J. Matr. Anal. Appl.,* Vol. **13, 3**, pp. 796–825, (1992).

[3] M. Benzi. Preconditioning techniques for large linear systems: a survey, *J. Comput Physics*,Vol. **182**, pp. 418–477, (2002).

[4] García M.D., *Estrategias para la resolución de grandes sistemas de ecuaciones lineales. Métodos de Cuasi-Mínimo Residuo Modificados*, PhD., (2003).

[5] G. Montero, R. Montenegro, J.M. Escobar. A 3-D diagnostic model for wind field adjustment, *J. Wind Eng. Ind. Aer.*, Vol. **74,76**, pp. 249–261, (1998).

[6] G. Montero, E. Rodríguez, R. Montenegro, J.M. Escobar, J.M. González- Yuste. Genetic algorithms for an improved parameter estimation with local refinenent of tetrahedral meshes in a wind model, *Adv. Engng. Soft.*, Vol. **36**, pp. 3–10, (2005).

[7] M. Benzi, J.K. Cullum and M. Tůma. Robust approximate inverse preconditioning for the conjugate gradient method, *SIAM J. Sci. Comput.*,Vol. **22**, pp. 1318–1332, (2000).

[8] M. Benzi, C.D. Meyer and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method, *SIAM J. Sci. Comput.*,Vol. **17, 5** pp. 1135–1149, (1996).

[9] S.A. Kharchenko, L. Yu. Kolotilina, A.A. Nikishin and A. Yu. Yeremin. A robust AINV-type method for constructing sparse approximate inverse preconditioners in factored form, *Numer. Linear Algebra Appl.* Vol. **8**, pp. 165–179, (2001).

[10] G. Meurant. On the incomplete Cholesky decomposition of a class of perturbed matrices, *SIAM J. Sci. Comput.*, **23, 2**, pp. 419–429, (2001).

[11] M. Benzi and D. Bertaccini. Approximate inverse preconditioning for shifted linear systems, *BIT Num. Math.*, Vol. **43**, pp. 231–244, (2003).

[12] I.S. Duff, R.G. Grimes and J.G. Lewis. Sparse Matrix Test Problems, *ACM Trans. Math. Sof.*, Vol. **15,1**, pp. 1–14, (1989).

[13] A. Rodríguez y M.L. Sandoval. Incomplete Cholesky factorizations for transient convection-diffusion problems, *in proceedings of The Fourth Internacional Conference on Engineering Computational Technology*, B.H.V. Topping and C.A. Mota Soares, Eds. Civil-Comp. Press, (2004).

[14] O. Axelsson, *Iterative solution methods*, Cambridge University Press, (1996).