UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

Instituto Universitario de Sistemas
Inteligentes y Aplicaciones Numéricas en
Ingeniería

Tesis Doctoral

PRECONDICIONAMIENTO DE SISTEMAS DE ECUACIONES DE MATRICES VARIABLES EN LA MODELIZACIÓN DE CAMPOS DE VIENTO

Héctor Sarmiento Almeida

Las Palmas de Gran Canaria, Abril de 2010

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

Instituto Universitario de Sistemas
Inteligentes y Aplicaciones Numéricas en
Ingeniería

Programa de Doctorado:
Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería

Título de la Tesis

PRECONDICIONAMIENTO DE SISTEMAS DE ECUACIONES DE MATRICES VARIABLES EN LA MODELIZACIÓN DE CAMPOS DE VIENTO

Tesis Doctoral Presentada por D. Héctor Sarmiento Almeida
Dirigida por el Dr. Antonio Suárez Sarmiento
Codirigida por el Dr. Gustavo Montero García

El Director, **El Codirector,** **El Doctorando,**

Las Palmas de Gran Canaria, a de de 2010

A mis nietos, Tomás y Sara

Agradecimientos

A Antonio Suárez Sarmiento y Gustavo Montero García, directores de esta tesis, sin cuyo aliento, asesoramiento y tutela no hubiera sido posible la realización de este trabajo.

A Eduardo Rodríguez Barrera que, pacientemente, me ha ayudado a resolver los imprevistos y dudas informáticas que fueron surgiendo durante el desarrollo del tema.

A María Dolores García León y Elizabeth Flórez Vázquez, por su inestimable colaboración, gracias a las cuales, las ideas y experimentos necesarios para el desarrollo y presentación de esta tesis, se han hecho realidad.

A todos los compañeros del Departamento de Matemáticas que de una forma u otra me han animado y ayudado, en todo momento, a la terminación de esta tesis.

A mi familia, en especial, por su amable paciencia durante los largos tiempos de ausencia que ha supuesto para ellos mi dedicación a este tema.

A todos, muchas gracias.

Resumen

En la formulación matemática de los modelos de campos de viento surgen grandes sistemas de ecuaciones lineales, caracterizados por tener matrices variables, al depender estas de un cierto parámetro, ϵ, tal que: $A_\epsilon\, x_\epsilon = b_\epsilon$ donde A_ϵ es una matriz simétrica del tipo $A_\epsilon = M + \epsilon\, N$, siendo M y N dos matrices, tipo *sparse*, diferentes, Simétricas Definidas Positivas (SDP) y el parámetro $\epsilon \geq 0$.

Los métodos basados en los subespacios de Krylov constituyen la mejor alternativa para la resolución de los sistemas de ecuaciones *sparse*. En el caso particular de sistemas cuya matriz es SDP, el método del Gradiente Conjugado, es el que presenta los mejores resultados

En esta tesis se trata de extender el uso del algoritmo del Gradiente Conjugado Precondicionado a los sistemas de ecuaciones de matrices variables, estudiando los Precondicionadores más adecuados para mejorar su convergencia.

El presente trabajo está estructurado en tres partes.En una primera parte se describen los distintos tipos de modelizaciones de campos de viento y el proceso de generación de sus sistemas de ecuaciones lineales de matrices variables. En la segunda parte se presenta el estado del arte de los métodos iterativos para la resolución de sistemas lineales basados en los subespacios de Krylov, analizando la influencia de su Precondicionamiento y Reordenación. Y en la tercera parte, se adapta la construcción de Precondicionadores al caso de los sistemas de ecuaciones de matrices variables. Se ilustra su eficacia mediante numerosos experimentos numéricos y se destaca la importancia de las técnicas presentadas en la aplicación de Algoritmos Genéticos, para la selección de los parámetros óptimos del modelo de viento.

Finalmente, se extraen las conclusiones oportunas y se exponen las posibles líneas futuras.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

INTRODUCCIÓN

1.1. ANTECEDENTES

Los modelos de campos de viento son herramientas que permiten afrontar diversos problemas relacionados con el impacto del viento en nuestro entorno, tales como el estudio de sus efectos sobre una determinada estructura, la dispersión de contaminantes en la atmósfera, la propagación de incendios o el estudio del emplazamiento y rendimiento de parques eólicos. Concretamente, en este último segmento, con los modelos de campo de viento se pueden afrontar diversos problemas surgidos en el seno de las empresas dedicadas a la explotación de este tipo de parques, tales como la evaluación de la potencia producida por un aerogenerador en función de su situación y su comparación con las curvas suministradas por el fabricante, el estudio de la ubicación óptima de la red de estaciones de medida previa a la instalación del parque y la distribución más eficaz de los aerogeneradores.

Los modelos de campos de viento son, pues, herramientas cada vez más importantes para afrontar con eficacia una amplia gama de problemas de interés social, político y económico, y cada vez se exige más de ellos.

La formulación matemática de estos modelos, al igual que la de otros muchos, correspondientes a muy diversos problemas físicos, da lugar a expresiones en derivadas parciales, de orden superior, que hacen muy difícil o imposible alcanzar su solución de forma analítica.

Ante ello, se suele recurrir a simplificar el modelo inicial, para tratar de conseguir una solución exacta, con el riesgo de alejarse demasiado de la realidad, o bien,

a buscar soluciones lo más aproximadas a las reales, pero manteniendo la expresión del modelo matemático con toda su complejidad. Este último proceso suele ser más útil y, actualmente, se puede afrontar con gran eficacia, gracias al avance del Cálculo Numérico, paralelo al desarrollo experimentado por la computación.

La solución numérica de un problema con formulación en derivadas parciales pasa por un proceso de discretización (con diferencias finitas, elementos finitos, volúmenes finitos), que permita valorar la solución en un número finito de puntos del dominio. Esta discretización, conduce a la resolución de un sistema lineal de ecuaciones, cuyas incógnitas son, precisamente, estos valores numéricos puntuales de la solución aproximada al problema físico que ha generado dicho sistema.

La mayoría de las veces, el proceso de discretización da lugar a grandes sistemas de ecuaciones lineales de matriz tipo *sparse*. Los errores de redondeo y el efecto de relleno que se produce en la aplicación de los métodos directos de factorización de la matriz del sistema, hacen más adecuados los métodos iterativos [9], teniendo especial relevancia, entre estos últimos, los métodos basados en los subespacios de Krylov, de más reciente desarrollo.

Los métodos de Krylov [46, 49, 109], utilizados para la resolución de grandes sistemas lineales se obtienen para adaptarse, en principio, a dos requerimientos básicos, esto es, minimizar una cierta norma del vector residuo sobre un subespacio de Krylov generado por la matriz del sistema, que se traduce en una convergencia suave, sin grandes fluctuaciones, y ofrecer un bajo coste computacional por iteración, sin exigir alta disponibilidad de almacenaje.

Como la eficacia de un método depende de su facilidad de implementación, la gran cantidad de métodos que existen, apropiados para aproximar la solución, se han ido desarrollando paralelamente a la evolución de los ordenadores.

Para los sistemas de ecuaciones lineales cuya matriz de coeficientes es simétrica y definida positiva, el algoritmo del Gradiente Conjugado, propuesto por Hestenes y Stiefel en 1952 [50], (desarrollado posteriormente al encontrar el soporte informático adecuado), cumple muy bien con los requisitos mencionados de minimalidad y optimalidad.

Por otro lado, debe tenerse en cuenta que, la convergencia de los métodos basados en los subespacios de Krylov, mejora notablemente con el uso de las

técnicas de Precondicionamiento y Reordenación de los sistemas, lo que hace muy importante el conocimiento y aplicación de las mismas.

Al afrontar ciertas modelizaciones, como las de los campos de viento, los sistemas lineales obtenidos como consecuencia de la discretización son de matrices simétricas definidas positivas y con coeficientes variables, dependientes de ciertos parámetros. Pues bien, el objeto principal de esta tesis, consiste en plantear propuestas que hagan posible solucionar, por métodos iterativos, no muy costosos computacionalmente, este tipo de sistemas de ecuaciones lineales de matrices variables.

1.2. OBJETIVOS

Con los antecedentes anteriormente expuestos, los objetivos específicos de esta tesis, con la finalidad de proporcionar herramientas adecuadas para obtener soluciones aproximadas, de los grandes sistemas de ecuaciones lineales de matrices variables, como las surgidas en la modelización de campos de viento, son:

- Dar una visión general de la modelización de campos de viento.
- Exponer el tipo de sistemas de ecuaciones lineales de coeficientes variables, dependientes de un parámetro, que surgen como consecuencia de la discretización, mediante elementos finitos, de los modelos de viento de masa consistente.
- Presentar las distintas consideraciones, a tener en cuenta, para conseguir la valoración óptima de los distintos parámetros que aparecen en la formulación de dichos modelos.
- Hacer un análisis de los distintos métodos, basados en los subespacios de Krylov, para la obtención iterativa de soluciones aproximadas de los sistemas de ecuaciones lineales, con mención especial de los algoritmos más apropiados para la resolución de sistemas con matrices simétricas definidas positivas.
- Estudiar el efecto del precondicionamiento de los sistemas en la convergencia de los algoritmos a utilizar en su resolución, y describir los principales

precondicionadores, tanto explícitos como implícitos.

- Exponer las técnicas de reordenación y su efecto en la resolución de sistemas precondicionados.

- Implementar los precondicionadores SAINV y los obtenidos como consecuencia de la factorización incompleta de Cholesky, para su aplicación a los sistemas de ecuaciones lineales de matrices variables.

- Realizar un estudio comparativo del efecto que producen los precondicionadores, anteriormente citados, sobre la convergencia del algoritmo del Gradiente Conjugado, al aplicarlos a los sistemas de ecuaciones lineales de matrices variables, obtenidos en la modelización práctica de diversos campos de viento de la isla de Gran Canaria.

- Comprobar la influencia de la reordenación en la solución de varios de esos sistemas lineales de coeficientes variables, obtenidos en la modelizaciones prácticas mencionadas.

- Realizar, asimismo, un estudio comparativo de la convergencia del algoritmo del Gradiente Conjugado, con la utilización de los precondicionadores propuestos, en la selección, mediante Algoritmos Genéticos, de los valores paramétricos óptimos, para diferentes campos de viento.

1.3. METODOLOGÍA

El presente trabajo se inicia con la Introducción, **Capítulo 1**, donde se presentan los antecedentes, los objetivos propuestos en el desarrollo de estas tesis, y la metodología seguida a lo largo de la misma.

El contenido básico, se estructura en tres grandes bloques. Un primer bloque, que abarca desde el capítulo 2 hasta el capítulo 4, en el que se expone la problemática de la modelización de los campos de viento, su discretización por MEF y loa procesos de estimación de los parámetros que intervienen en su formulación. Un segundo bloque, que va del capítulo 5 al 7, en el que se da una visión general del estado del arte de los métodos iterativos, para la resolución de grandes

sistemas de ecuaciones lineales, basados en los subespacios de Krylov, así como de las técnicas de precondicionamiento y reordenación de dichos sistemas. Y un tercer bloque, constituido por el capítulo 8, donde se trata de hacer una nueva aportación, extendiendo las técnicas de precondicionamiento a los sistemas de ecuaciones lineales de matrices variables. Por último, en el capítulo 9, se presentan los resultados de diversos experimentos numéricos, donde se aplican los precondicionadores propuestos a los sistemas de matrices variables, se utilizan diferentes reordenaciones y se comprueba su influencia sobre la convergencia del algoritmo del Gradiente Conjugado.

El primer bloque, dedicado a la modelización de campos de viento está constituido por:

- El **Capítulo 2**, donde se exponen el estado del arte de la modelización de los campos de viento, así como, unas breves nociones de Cálculo Variacional, dada su aplicación en la formulación matemática de los modelos en cuestión. Se presta especial atención al Modelo de Masa Consistente, por considerarse, actualmente, como el más eficiente, llegando a establecer la ecuación diferencial elíptica que lo define. Asimismo, se analiza la construcción del campo inicial, por su trascendencia en la consecución de un resultado final correcto.

- El **Capítulo 3**, en el que se afronta la discretización, por el Método de Elementos Finitos, de la ecuación elíptica, mencionada anteriormente, correspondiente a los Modelos de Masa Consistente. Se expone la problemática de la generación de mallas adaptativas, para conseguir discretizar con más eficacia, sobre todo en los terrenos de orografía compleja. Llegando finalmente a la generación del sistema de ecuaciones lineales del modelo matemático, que resulta ser de matriz Simétrica Definida Positiva (SDP), de coeficientes variables, dependientes de un parámetro, denominado parámetro de estabilidad del modelo.

- El **Capítulo 4**, dedicado al análisis y valoración, no solo, de los parámetros que intervienen an la construcción del campo inicial, sino también, del parámetro de estabilidad, por su gran protagonismo en el sistema de ecuaciones

lineales obtenido.

El segundo bloque, correspondiente a los métodos del Cálculo Numérico para la resolución de grandes sistemas de ecuaciones lineales, está formado por:

- El **Capítulo 5**, en el que se exponen los fundamentos de los métodos iterativos basados en los subespacios de Krylov, por considerarse los más adecuados para la resolución de grandes sistemas lineales. Prestando especial atención al algoritmo del Gradiente Conjugado, por tratarse del método más eficaz para resolver los sistemas de matrices SDP, que, como se ha indicado, es el tipo de sistema que se obtiene en la modelización de campos de viento de Masa Consistente.

- El **Capítulo 6**, dedicado por entero al Precondicionamiento de sistemas, por ser una herramienta que mejora sensiblemente la convergencia de los métodos de Krylov. Además de exponer el estado del arte de esta técnica, se establece el algoritmo del Gradiente Conjugado Precondicionado, pues es el método que se utilizará más adelante para afrontar los experimentos numéricos y, también, se describen los precondicionadores explícitos AINV y SAINV, así como, los implícitos, tanto los obtenidos por comparación con el método de Richardson, como los que se fundamentan en factorizaciones incompletas de la matriz inicial del sistema.

- El **Capítulo 7**, en el que se estudian los métodos más prácticos de Reordenación de sistemas, como son: el algoritmo de Cuthill-McKee Inverso, el del Mínimo Vecino y el Multicoloring, que basados en la teoría de grafos, proporcionan matrices con ancho de banda o perfil menor, lo que incide notablemente en una mayor simplificación, a la hora de construir un precondicionador más eficaz.

El tercer bloque, donde se hace la aportación novedosa de esta tesis, está constituido por:

- El **Capítulo 8**, en el que, a partir del conocimiento de los principales precondicionadores aplicables a los sistemas de matrices SDP, como son los SAINV y los basados en la factorización incompleta de Cholesky (ICHOL),

se implementa su adaptación a los sistemas de ecuaciones lineales de matrices variables, que, como ya se ha indicado, son los que se obtienen en la modelización matemática de los campos de viento.

Los resultados de los experimentos numéricos, realizados para confrontar las propuestas aportadas en esta tesis, se recogen en

- El **Capítulo 9**, que a su vez se distribuye en dos secciones.

  Una, dedicada a las aplicaciones test, donde se han utilizado ambos tipos de precondicionadores, SAINV e ICHOL, sobre sistemas de ecuaciones de matrices variables obtenidos en tres modelos de campos de viento distintos, conseguidos con tres discretizaciones diferentes, sobre una región de la isla de Gran Canaria, comprobando en la práctica, la eficacia de los distintos precondicionadores, así como, la influencia de las diferentes técnicas de Reordenación.

  Otra, orientada a la optimización del parámetro de estabilidad, donde se han aplicado los precondicionadores tipo ICHOL, dado que, con los resultados obtenidos en los experimentos anteriores, han demostrado ser los más eficaces para estos modelos. Se han utilizado sobre los sistemas de ecuaciones de matrices variables correspondientes a la modelización de dos campos de viento diferentes, usando siempre el algoritmo del Gradiente Conjugado Precondicionado. En cada uno de estos ejemplos se recogen los tiempos de computación, para dos gamas distintas de valores del parámetro, pues estos resultados influirán notablemente a la hora de seleccionar el precondicionador más adecuado para usar en la selección del valor óptimo del parámetro, mediante Algoritmos Genéticos, dado la repetitividad del proceso.

Las conclusiones obtenidas y las futuras lineas de investigación, con las que finaliza este trabajo, se recogen en el **Capítulo 10**.

Capítulo 2

CAMPOS DE VIENTO

2.1. PRELIMINARES

El viento no es otra cosa que el aire en movimiento, entendiendo por aire la masa de gases que constituyen nuestra atmósfera terrestre.

Hace unos cuatro mil seiscientos millones de años el Sistema Solar se condensó a partir de una nube de gas y polvo interestelar, la Nebulosa Solar. Las atmósferas de la Tierra, Venus y Marte se formaron a partir de materia volátil que escapó de cada planeta. La primitiva atmósfera de la Tierra estaba compuesta por dióxido de carbono (CO_2), nitrógeno (N_2) y vapor de agua (H_2O), con trazas de hidrógeno (H_2), una mezcla muy similar a la emitida hoy en día por los volcanes. La aparición del oxígeno (O_2) como componente de la atmósfera fue el resultado de su producción por la actividad fotosintética. Se estima que el nivel actual de O_2 se alcanzó hace aproximadamente cuatrocientos millones de años y se mantiene gracias a un balance entre su producción por fotosíntesis y su desaparición por la respiración de los seres vivos y gradual descenso del carbono orgánico. La atmósfera actual está compuesta principalmente por los gases $N_2(78\,\%)$, $O_2(21\,\%)$, $Ar(1\,\%)$ y una proporción muy variable de vapor de agua, que puede alcanzar hasta un $3\,\%$.

Desde la más remota antigüedad, el hombre se dió cuenta que el viento podía ser aprovechado como fuente de energía, así los egipcios navegaban ya a vela en el año 4500 a.C. Más tarde el aprovechamiento de la energía eólica continuó con la aparición de los molinos, se tienen datos de ellos desde el siglo II a.C. Los

más antiguos eran de eje vertical, pero hacia el siglo VIII aparecieron en Europa, procedentes del Este, los grandes molinos de eje horizontal con cuatro aspas. A partir de los siglos XII y XIII se generaliza el uso de los molinos de viento para la molienda de granos y para la elevación de agua, actividades que se mantienen hasta bien entrado el siglo XIX. La llegada de la revolución industrial, con la utilización masiva del vapor, la electricidad y los combustibles fósiles como fuente de energía, interrumpe su desarrollo. Sin embargo, en la segunda mitad del siglo XIX, aparece el popular molino americano multipala, utilizado para el bombeo de agua, practicamente en todo el mundo, y cuyas características habrían de sentar las bases para el diseño de los modernos aerogeneradores. Durante el siglo XX, la superficie del planeta se ha ido cubriendo paulatinamente de más y más parques eólicos, que van surgiendo como alternativa viable de las centrales térmicas. La sociedad ha ido adquiriendo conciencia de los problemas medio ambientales y valora cada vez más el uso de las energías renovables. Esta creciente inquietud social ha adquirido una gran importancia desde el punto de vista político (no hay formación política que se sustraiga a los problemas ecológicos y no los incluya en su programa electoral) y económico (las empresas invierten cada día más en estudios de impacto ambiental y en publicidad para alardear de sus valores ecológicos, sean reales o no), lo que ha producido en los últimos años un crecimiento notable de la producción de energía eléctrica de origen eólico.

La Conferencia de Madrid, marzo de 1994, consideró viable que las energías renovables contribuyeran con un 15 % a la demanda total de energía primaria en la CE, antes de 2010. España ocupa un lugar destacado en el panorama eólico comunitario, con el quinto puesto por potencia eólica instalada, detrás de Dinamarca, Alemania, Reino Unido y Holanda. Las empresas del sector necesitan herramientas cada vez más sofisticadas que les permita hacer frente a las demandas de un mercado cada vez más competitivo y exigente.

Por otra parte el desarrollo industrial ha traído como consecuencia el vertido masivo a la atmósfera de sustancias contaminantes. Es cada vez más evidente que la contaminación atmosférica tiene graves repercusiones que provocan la alteración de las condiciones medioambientales del planeta. Las consecuencias de esta contaminación van desde la lluvia ácida, hasta el aumento de los trastornos respi-

ratorios y alérgicos de la población, pasando por el preocupante efecto invernadero de graves consecuencias a largo plazo.

2.2. MODELOS DE CAMPOS DE VIENTO

Los modelos de campos de viento son herramientas que permiten afrontar diversos problemas relacionados con el impacto del viento en nuestro entorno, tales como, el estudio de sus efectos sobre una determinada estructura (especialmente puentes y edificios de gran altura), la dispersión de contaminantes en la atmósfera, la propagación de incendios o el estudio del emplazamiento y rendimiento de parques eólicos. Concretamente en este último segmento, con los modelos de campo de viento se pueden afrontar diversos problemas surgidos en el seno de las empresas dedicadas a la explotación de este tipo de parques, tales como la evaluación de la potencia producida por un aerogenerador, en función de su situación, y su comparación con las curvas suministradas por el fabricante; el estudio de la ubicación óptima de la red de estaciones de medida previa a la instalación del parque y la distribución más eficaz de los aeorgeneradores.

Los modelos de campos de viento son, pues, herramientas cada vez más importantes para afrontar con eficacia una amplia gama de problemas de interés social, político y económico, y cada vez se exige más de ellos.

Inicialmente los modelos meteorológicos se dividen en dos grandes grupos: los Modelos Físicos y los Modelos Matemáticos. Los primeros usan túneles de viento sobre reproducciones a pequeña escala del terreno en estudio. En los Modelos Matemáticos, que serán los objetos de esta tesis, se emplean técnicas algebraicas y de cálculo para resolver ecuaciones metereológicas. A su vez los Modelos Matemáticos se dividen en Analíticos y Numéricos; los primeros, por la gran complejidad de las ecuaciones que describen la atmósfera, hacen muy difícil la resolución exacta en dominios irregulares, mientras que los segundos ofrecen mejores perspectivas. Por ello el objetivo de esta tesis va a ser tratar de aportar herramientas de Cálculo Numérico que permitan la modelización matemática de campos de viento con la mayor eficacia posible.

Según la extensión del dominio a estudiar, los Modelos de Viento se pueden

clasificar en Modelos de Macroescala, cuando el área de estudio abarca desde un continente hasta el globo terráqueo completo. Modelos de Mesoescala, cuando se refieren a extensiones que van desde unos pocos kilómetros hasta alrededor de cien, y de Microescala, para regiones locales que tienen como máximo alrededor de un kilómetro.

Los Modelos Matemáticos de Campos de Viento también se pueden clasificar en Modelos de Pronóstico o Dinámicos y Modelos de Diagnóstico o Cinemáticos.

Los Modelos de Pronóstico se basan en la solución de ecuaciones hidrodinámicas y termodinámicas que depende del tiempo (llamadas también ecuaciones primitivas porque derivan directamente de los principios de conservación) modificadas para su aplicación a la atmósfera. Sin embargo la solución del conjunto completo de ecuaciones sigue siendo una tarea costosa. Además, cuanto más elaborado es el modelo, más fiables deben ser los datos de entrada para aprovechar las ventajas ofrecidas, y con frecuencia estos datos no suelen estar disponibles.

Autores como Lalas et al. [58] incluyen en los Modelos Dinámicos algunos códigos que introducen aproximaciones a las ecuaciones primitivas, a la vez que desprecian su dependencia del tiempo. Estos códigos, denominados JH, se basan en una propuesta realizada por Jackson y Hunt [51] y son ampliamente utilizados. Como ejemplo podemos citar el modelo empleado por Troen y Petersenen [110] en la confección del Atlas Europeo de Viento.

Los Modelos de Diagnóstico deben su nombre a que no se utilizan para realizar previsiones a través de la integración de las relaciones conservativas. Eliminan directamente de sus ecuaciones la dependencia del tiempo y por esta razón se les llama también Cinemáticos. Estos modelos generan un campo de viento que satisface algunas restricciones físicas. Si la única restricción que se les impone es que cumplan la ecuación de continuidad, lo que supone la conservación de la masa, el modelo se denomina de Masa Consistente. Los Modelos de Diagnóstico no requieren muchos datos de entrada y son fáciles de usar, por lo que resultan atractivos desde el punto de vista práctico. Autores como Pennel [81] han comprobado que en algunos casos los Modelos de Masa Consistente mejorados, tales como NOABL y COMPLEX, superaron los resultados de Modelos Dinámicos más complicados y costosos. Sin embargo hay que tener en cuenta que los Modelos de Diagnóstico no

consideran los efectos térmicos ni los debidos a cambios de gradientes de presión. Por ello, flujos tales como las brisa marinas, vientos en pendiente y otros tales como los de separación a favor del viento, no pueden simularse con esta modelos, a no ser que se incorporen en los datos de viento inicial, a partir de observaciones realizadas en lugares apropiados a tal efecto [54, 76].

Los Modelos de Diagnóstico están diseñados específicamente para predecir los efectos de la orografía sobre el flujo medio de viento considerado de manera estacionaria, esto es, flujos promediados en intervalos de tiempo entre 10 minutos y 1 hora.

NOABL [82] es un modelo meteorológico que proporciona una representación precisa del terreno gracias a una transformación de la coordenada vertical en la que la coordenada más baja es conforme a la superficie del terreno. Posteriormente, diversos autores [55, 56, 108] realizaron algunas modificaciones en la inicialización del mismo, con el fin de que el modelo considerara el efecto de la rugosidad del terreno sobre el perfil del viento, de forma que el modelo dispusiera de perfiles más realistas que los del código original y tuviera en cuenta el cambio debido a la fuerza de Coriolis en la dirección del viento, en la capa límite atmosférica. Los códigos resultantes fueron bautizados como NOABL* [58] y EOLOS [108], aunque actualmente es más conocido por AIOLOS, por su referencia a la etimología griega. Más tarde se introdujeron modificaciones que describen con más precisión los perfiles de viento en diferentes condiciones de estabilidad. Este nuevo código [86] se llamó WINDS(*Wind-field Interpolation by Non Divergent Squemes*). Ambos modelos, AIOLOS y WINDS, usan datos de estaciones situadas en tierra y, opcionalmente, datos observados tanto de perfiles verticales como de viento geostrófico. También utilizan coordenadas conformes al terreno. Ambos se basan en la minimización de los cuadrados de las diferencias entre las velocidades de un viento inicial, obtenido por interpolación de los datos observados, y el viento a ajustar, sujeto a la restricción de que el campo de viento ajustado ha de tener divergencia nula [99].

Además de los ya citados, existe toda una gama de Modelos de Diagnóstico que la comunidad científica ha venido utilizando en problemas relacionados con la meteorología y/o con la contaminación atmosférica. Dentro de los más conocidos

podemos citar el ATMOS-1 [18, 53], que calcula campos de viento basándose en la conservación de la masa y apoyándose en medidas experimentales de viento. Utiliza coordenadas conformes al terreno y un sistema de malla expandida verticalmente que asegura la solución cerca de la superficie. Otro modelo que se basa en la conservación de la masa es el DWM [21], capaz de generar campos de viento 3D, sobre terrenos complejos, a partir de un número limitado de observaciones, tanto a nivel del terreno como de capas altas del aire, y proporciona información de flujos de aire generados en el terreno en regiones donde no hay observaciones locales. Utiliza también un sistema de coordenadas conformes al terreno. Al igual que en los anteriores los modelos MASCON [19] y MATHEW están basados en la ecuación de la continuidad. El primero utiliza técnicas de cálculo variacional para ajustar los flujos horizontales observados y es muy similar al segundo. MATHEW [103, 88] proporciona un campo de viento medio 3D, ajustado por mínimos cuadrados. Por último, cabe mencionar MINERVE [39], modelo también de Masa Consistente, que reduce la divergencia mediante un procedimiento iterativo que puede tener en cuenta la estabilidad atmosférica.

2.3. BREVES NOCIONES DE CÁLCULO VARIACIONAL

Los modelos de Masa Consistente son actualmente los más utilizados, por su mayor economía computacional. Para su desarrollo es preciso recurrir al Cálculo Variacional, por lo que introducimos aquí este apartado, destinado a recordar ciertas nociones básicas de dicho Cálculo.

2.3.1. FUNCIONAL: SU DEFINICIÓN

Llamamos funcional a una aplicación de un cierto número de funciones sobre el conjunto de los números reales, $\Re$, es decir, se trata de un número real variable, cuyo valor se determina mediante la elección de una o más funciones de una o varias variables.

El concepto de diferencialidad se define de forma similar a como se hace para funciones de varias variables, en el supuesto que el conjunto de funciones, al que

se ha hecho referencia, tenga estructura de espacio de Banach, circunstancia que se presenta con frecuencia en numerosos casos de resolución de problemas físicos.

El área, S, de una superficie alabeada es una funcional, puesto que se trata de un número real que se determina escogiendo la superficie o función $o = z(x, y)$ en la expresión

$$S[z(x,y)] = \iint_D \sqrt{1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2}\, dx\, dy$$

Siendo D la proyección de la superficie, cuyo área queremos calcular, en el plano XY.

2.3.2. CÁLCULO VARIACIONAL

Los extremos de las funcionales se definen de forma análoga a los de las funciones de una o varias variables. El cálculo variacional estudia los métodos que permiten hallar los valores máximos y mínimos de las mismas.

Es muy conocido el problema de la braquistócrona. En él, se trata de determinar la línea que une dos puntos dados A y B situados en un mismo plano vertical, pero no en la misma recta vertical, tal que un punto material sometido a la acción de la fuerza de la gravedad, se deslice sobre dicha línea, desde A hasta B en el menor tiempo posible, suponiendo rozamiento nulo, gravedad constante y velocidad inicial cero.

Aplicando el principio de conservación de la energía, se llega a la expresión

$$t = \frac{1}{\sqrt{2g}} \int_{x_A}^{x_B} \frac{\sqrt{1 + y'(x)^2}}{\sqrt{y_A - y(x)}} dx$$

La curva buscada será aquella $y = y(x)$ que dé un valor de t mínimo. Estos tipos de problemas son los que resuelve el Cálculo Variacional.

2.3.3. ECUACIONES DE EULER

Las ecuaciones de Euler nos dan, respectivamente, las condiciones necesarias para la obtención de los extremos de distintas funcionales bajo ciertas hipótesis. Aunque establecer condiciones suficientes exigiría el estudio de la llamada variación segunda, en muchísimas ocasiones, las condiciones físicas o geométricas del

problema nos permitirán conocer la naturaleza del punto crítico, sin recurrir a ella, [23, 40].

Nos limitaremos a enunciar, a título de ejemplo, las correspondientes ecuaciones de Euler para distintas funcionales usuales, entre las que se encuentra el caso que nos ocupa, en la modelización de campos de viento:

- **Funcional de una variable**

$$v[y(x)] = \int_{x_0}^{x_1} F[x, y(x), y^{'}(x)]dx$$

siendo $y = y(x)$ continua, con derivada continua, en $[x_0, x_1]$. Para las curvas admisibles que pasen por los puntos $y(x_0) = y_0$ e $y(x_1) = y_1$. Además $F[x, y(x), y^{'}(x)]$ admite derivadas parciales continuas.

La curva $y = y(x)$ para la que la funcional considerada adopta un valor extremo es solución de la ecuación diferencial de segundo orden siguiente:

$$F_{y}^{'} - \frac{d}{dx} F_{y'}^{'} = 0 \tag{2.1}$$

o bien desarrollando, por aplicación de la regla de la cadena para funciones compuestas,

$$F_{y}^{'} - F_{y'x}^{''} - F_{y'y}^{''} \cdot y^{'} - F_{y'y'}^{''} \cdot y^{''} = 0$$

Habrá que integrar esta ED y determinar las dos constantes arbitrarias que figuran en la solución general con las condiciones de frontera $y(x_0) = y_0$ e $y(x_1) = y_1$.

- **Funcional de dos variables**

$$v[z(x, y)] = \iint_D F\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dx\, dy$$

siendo $z = z(x, y)$ continua, con derivadas parciales continuas. Todas las superficies admisibles tienen el mismo contorno alabeado. Además $F\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right)$ admite derivadas parciales continuas.

Llamando $\frac{\partial z}{\partial x} = p$, $\frac{\partial z}{\partial y} = q$

La superficie $z = z(x, y)$ para la que la funcional considerada adopta un valor extremo es solución de la ecuación diferencial en derivadas parciales:

$$F'_z - \frac{d}{dx} F'_p - \frac{d}{dy} F'_q = 0 \tag{2.2}$$

- **Funcional de tres variables** (Generalizando los casos anteriores)

$$v[u(x,y,z), v(x,y,z), w(x,y,z)] =$$

$$\iiint_\Omega F\left[u(x,y,z), v(x,y,z), w(x,y,z), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial v}{\partial z}, \frac{\partial w}{\partial x}, \frac{\partial w}{\partial y}, \frac{\partial w}{\partial z}\right] dx\, dy\, dz$$

Llamando:

$$\begin{array}{lll} \dfrac{\partial u}{\partial x} = p_1, & \dfrac{\partial u}{\partial y} = q_1, & \dfrac{\partial u}{\partial z} = r_1 \\ \dfrac{\partial v}{\partial x} = p_2, & \dfrac{\partial v}{\partial y} = q_2, & \dfrac{\partial v}{\partial z} = r_2 \\ \dfrac{\partial w}{\partial x} = p_3, & \dfrac{\partial w}{\partial y} = q_3, & \dfrac{\partial w}{\partial z} = r_3 \end{array}$$

Las funciones extremales serán solución del sistema de ecuaciones diferenciales:

$$\left\{ \begin{array}{llll} F'_u & -\frac{\partial}{\partial x}(F'_{p_1}) & -\frac{\partial}{\partial y}(F'_{q_1}) & -\frac{\partial}{\partial z}(F'_{r_1}) = 0 \\ F'_v & -\frac{\partial}{\partial x}(F'_{p_2}) & -\frac{\partial}{\partial y}(F'_{q_2}) & -\frac{\partial}{\partial z}(F'_{r_2}) = 0 \\ F'_w & -\frac{\partial}{\partial x}(F'_{p_3}) & -\frac{\partial}{\partial y}(F'_{q_3}) & -\frac{\partial}{\partial z}(F'_{r_3}) = 0 \end{array} \right. \tag{2.3}$$

2.3.4. PROBLEMAS VARIACIONALES CON LIGADURAS

En funciones de varias variables surge con frecuencia el problema de obtener los puntos críticos cuando entre las variables existen una o más condiciones de ligadura. Son los llamados extremos condicionados. Un forma de obtenerlos es

utilizando el conocido método de los multiplicadores de Lagrange. Este método consiste en formar una función auxiliar

$$F = f(x_1, x_2, ..., x_n) + \sum_{i=1}^{m} \lambda_i \varphi_i(x_1, x_2, ..., x_n)$$

Donde λ_i son unos factores constantes y $\varphi_i(x_1, x_2, ..., x_n) = 0$, son las ecuaciones de condición. Todo extremo de la función $f(x_1, x_2, ..., x_n)$ con las condiciones adicionales $\varphi_i(x_1, x_2, ..., x_n) = 0$ es un máximo o mínimo de la función auxiliar F.

En Cálculo Variacional también pueden aparecer problemas de funciones extremales con condiciones adicionales:

Sea, por ejemplo, la funcional

$$v[y(x)] = \int_{x_0}^{x_1} F[x, y(x), y^{'}(x)]dx$$

y consideremos otra funcional

$$u[y(x)] = \int_{x_0}^{x_1} G[x, y(x), y^{'}(x)]dx$$

Siendo G una función también continua y de derivadas continuas.

Pues bien, al igual que en funciones de varias variables, si $y(x)$ es una función extremal de la funcional $v[y(x)]$, con la condición de que $u[y(x)] = Cte.$, entonces $y(x)$ también es extremal de la funcional

$$V[y(x)] = \int_{x_0}^{x_1} \Big\{ F[x, y(x), y^{'}(x)] + \lambda\, G[x, y(x), y^{'}(x)] \Big\} dx. \tag{2.4}$$

2.4. MODELO DE MASA CONSISTENTE

Como ya hemos comentado se trata de un Modelo de Diagnóstico para definir campos de velocidades en tres dimensiones, a partir de un número determinado de medidas experimentales [57]. El modelo sirve tanto de herramienta generadora de mapas de viento de una zona determinada, como de preproceso para otros modelos que traten de fenómenos que tienen lugar en la atmósfera, tales como la dispersión de contaminantes o propagación de incendios. En general están gobernados por las leyes físicas definidas para un fluido incompresible.

Este modelo está basado en la ecuación de continuidad para una masa de aire incompresible que se mueve en un dominio tridimensional, Ω, con un campo de velocidades $\vec{u}(u, v, \omega)$:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \, \vec{u}) = 0$$

Considerando que la densidad del aire, ρ, sea constante en todo el dominio, la ecuación se transforma en:

$$\vec{\nabla} \cdot \vec{u} = 0 \qquad \text{en} \quad \Omega \tag{2.5}$$

que se une a la condición de impenetrabilidad sobre Γ_b (terreno y frontera superior del dominio), constituyendo así la condición de contorno:

$$\vec{n} \cdot \vec{u} = 0 \qquad \text{en} \quad \Gamma_b \tag{2.6}$$

A partir de estas condiciones, (2.5) y (2.6), los modelos de Masa Consistente plantean un problema de mínimos cuadrados en el dominio Ω con las velocidades a ajustar $\vec{u}(u, v, \omega)$ a partir de los observadas $\vec{v_0}(u_o, v_o, \omega_o)$, de acuerdo con la funcional:

$$E(u, v, \omega) = \iiint_{\Omega} [a_1^2(u - u_o)^2 + a_2^2(v - v_o)^2 + a_3^2(\omega - \omega_o)^2] \, dx \, dy \, dz \tag{2.7}$$

donde $u(x, y, z), v(x, y, z)$ y $\omega(x, y, z)$ son las componentes del viento calculadas por el modelo mediante ajuste; $u_0(x, y, z), v_0(x, y, z)$ y $\omega_0(x, y, z)$ son las componentes del campo inicial, interpolado a partir de las observaciones, y a_1, a_2, a_3 son los módulos de precisión de Gauss.

Considerando idénticos a_1 y a_2, para las direcciones horizontales, la funcional a minimizar, (2.7), se transforma en:

$$E(u, v, \omega) = \iiint_{\Omega} \Big[\alpha_1^2[(u - u_o)^2 + (v - v_o)^2] + \alpha_2^2(\omega - \omega_o)^2\Big] \, dx \, dy \, dz \tag{2.8}$$

Se trata pues, de encontrar la función extremal $\vec{u}(u, v, \omega)$ que minimice la funcional $E(\vec{u})$ con la condición $\vec{\nabla} \cdot \vec{u} = 0$.

Utilizando el método de los multiplicadores de Lagrange, la función auxiliar será:

$$F = \iiint_{\Omega} \left\{ \Big[\alpha_1^2[(u-u_o)^2+(v-v_o)^2]+\alpha_2^2(\omega-\omega_o)^2\Big] + \Phi\Big(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial \omega}{\partial z}\Big) \right\} dx \, dy \, dz \tag{2.9}$$

Utilizando el grupo de ecuaciones (2.3), que hemos visto anteriormente para las Ecuaciones de Euler, resulta que:

$$F'_{p_1} = \Phi; \quad F'_{q_1} = 0; \quad F'_{r_1} = 0$$

$$F'_{p_2} = 0; \quad F'_{q_2} = \Phi; \quad F'_{r_2} = 0$$

$$F'_{p_3} = 0; \quad F'_{q_3} = 0; \quad F'_{r_3} = \Phi$$

Y sustituyendo y derivando convenientemente tendremos:

$$\begin{cases} 2(u-u_o)\alpha_1^2 - \frac{\partial\Phi}{\partial x} = 0 \\ 2(v-v_0)\alpha_1^2 - \frac{\partial\Phi}{\partial y} = 0 \\ 2(\omega-\omega_0)\alpha_2^2 - \frac{\partial\Phi}{\partial z} = 0 \end{cases} \quad \begin{cases} u = \frac{1}{2\alpha_1^2}\frac{\partial\Phi}{\partial x} + u_0 \\ v = \frac{1}{2\alpha_1^2}\frac{\partial\Phi}{\partial y} + v_0 \\ \omega = \frac{1}{2\alpha_2^2}\frac{\partial\Phi}{\partial z} + \omega_0 \end{cases}$$

Que pueden resumirse como:

$$\vec{u} = \vec{v}_0 + T\vec{\nabla}\Phi \tag{2.10}$$

Expresión en la que $T = (T_h, T_h, T_v)$, se puede considerar como un tensor diagonal de transmisión, tal que:

$$T = diag\Big[\frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_2^2}\Big] \tag{2.11}$$

constante para un dominio dado, Ω.

Así el campo solución, $\vec{u}$, se obtendrá a partir del campo inicial, $\vec{v}_0$, y de los valores de Φ, para cuyo cálculo se recurre a la ecuación de condición

$$\vec{\nabla}\cdot\vec{u} = 0, \quad \text{o sea,} \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial z} = 0$$

resultando la ecuación diferencial siguiente:

$$\frac{1}{2\alpha_1^2}\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial u_0}{\partial x} + \frac{1}{2\alpha_1^2}\frac{\partial^2\Phi}{\partial y^2} + \frac{\partial v_0}{\partial y} + \frac{1}{2\alpha_2^2}\frac{\partial^2\Phi}{\partial z^2} + \frac{\partial \omega_0}{\partial z} = 0.$$

Teniendo en cuenta que los parámetros α_1 y α_2, se suelen considerar constantes en todo el dominio Ω, la ecuación anterior se puede simplificar multiplicando todos los términos por $2\,\alpha_1^2$ y llamando a

$$\frac{\alpha_1^2}{\alpha_2^2} = \frac{T_v}{T_h} = \alpha^2 = \epsilon. \tag{2.12}$$

se obtiene

$$\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} + \epsilon\,\frac{\partial^2\Phi}{\partial z^2} = -\frac{1}{T_h}\left(\frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y} + \frac{\partial \omega_0}{\partial z}\right) \tag{2.13}$$

ecuación elíptica en Φ, donde ϵ recibe el nombre de parámetro de estabilidad del modelo.

Teniendo en cuenta las expresiones (2.6) y (2.10), esta ecuación diferencial, en derivadas parciales, estará sujeta a una condición tipo Neuman en las fronteras impermeables (terreno y frontera superior); ya que

$$\vec{n} \cdot T\vec{\nabla}\Phi = -\vec{n} \cdot \vec{v_0} \qquad en \quad \Gamma_b \tag{2.14}$$

que se completa con la condición de Dirichlet, nula en las fronteras permeables (fronteras verticales del dominio):

$$\Phi = 0 \qquad en \quad \Gamma_a \tag{2.15}$$

Obsérvese que en la frontera superior, al ser el campo inicial $\vec{v_0}$ horizontal, la condición (2.14) se transforma en

$$\vec{n} \cdot T\vec{\nabla}\Phi = 0 \tag{2.16}$$

Por tanto, desde el punto de vista matemático, la parte esencial de la construcción de un modelo de viento de Masa Consistente, se convierte en la resolución de la ecuación diferencial en derivadas parciales (**??**), con las condiciones de contorno (2.14) y (2.15).

Obsérvese que el parámetro de estabilidad del modelo, ϵ, va a estar presente en la solución del problema, de ahí que los modelos de Masa Consistente sean criticados por su alta dependencia de parámetros. La elección acertada de sus valores es de gran importancia para la fiabilidad de los resultados. Teniendo en cuenta (2.11) y (2.12):

$$\epsilon = \frac{T_v}{T_h} \tag{2.17}$$

resultando entonces que para $\epsilon \gg 1$, predomina el ajuste del flujo en la dirección vertical, es decir, el aire tiende a sobrepasar las barreras del terreno, más que a pasar horizontalmente alrededor de ellas. Mientras que para $\epsilon \ll 1$, el ajuste del flujo ocurre primeramente en el plano horizontal, por tanto el aire pasará alrededor de las barreras del terreno más que sobre ellas. En particular, $\epsilon \to \infty$ significa ajuste vertical puro, mientras que $\epsilon \to 0$ significa ajuste horizontal puro [7].

2.5. CONSTRUCCIÓN DEL CAMPO INICIAL

Para la construcción del campo inicial partimos de los valores de la velocidad del viento y de su dirección obtenidos en las estaciones de medida. Los datos de viento se toman de estaciones ubicadas en el dominio de estudio. Cada estación de medida proporciona la velocidad (en m/s) y dirección del viento a una altura z_s sobre el nivel del terreno (típicamente 10 metros). La dirección del viento viene dada en grados sexagesimales medidos en sentido horario y tomando como referencia la dirección norte. Así el norte se corresponde a 0 grados, el sur a 180 grados, el este a 90 grados y el oeste a 270 grados. A efectos de cálculo en el modelo, es necesario obtener el ángulo medido en sentido antihorario, tomando como referencia el semieje positivo horizontal. Por otro lado, como las estaciones miden el viento en intervalos discretos de tiempo, en general es necesario interpolar las medidas para calcular el viento en un instante concreto.

El campo inicial $\vec{v}_0$ se construye en dos etapas:

2.5.1. INTERPOLACIÓN HORIZONTAL

En primer lugar, se calcula mediante *interpolación horizontal* el valor de $\vec{v}_0$ en los puntos del dominio situados a la misma altura z_s (sobre el terreno) que las estaciones de medida.

La técnica más común de interpolación se formula en términos de la inversa de la distancia al cuadrado entre el punto y la estación de medida [116]. Sin embargo, otros autores usan simplemente la altitud de los puntos de medida [79]. Aquí se propone una fórmula que tiene en cuenta ambas consideraciones,

$$\vec{v}_0(z_e) = \beta \frac{\sum_{n=1}^{N} \frac{\vec{v}_n}{d_n^2}}{\sum_{n=1}^{N} \frac{1}{d_n^2}} + (1-\beta) \frac{\sum_{n=1}^{N} \frac{\vec{v}_n}{|\Delta h_n|}}{\sum_{n=1}^{N} \frac{1}{|\Delta h_n|}} \tag{2.18}$$

El valor de $\vec{v}_n$ corresponde a la velocidad observada en la estación n, N es el número de estaciones utilizadas en la interpolación, d_n es la distancia *horizontal* desde la estación n al punto donde estamos calculando la velocidad del viento,

$|\Delta h_n|$ es la diferencia de altura entre la estación n y el punto en estudio, y β es un parámetro de peso que toma valores entre 0 y 1. Cuando $\beta \to 1$ aumenta la importancia de la distancia horizontal desde cada punto a las estaciones de medida. Esta aproximación se emplea en problemas con una orografía regular o en análisis bidimensionales. De manera análoga, si $\beta \to 0$ es entonces la diferencia de altura entre cada punto y las estaciones de medida la que resulta determinante, en detrimento de la distancia horizontal. Esta segunda aproximación es la que se usa cuando la orografía del terreno es irregular. En la práctica, las regiones geográficas estudiadas suelen combinar zonas de orografía irregular con otras de orografía mas regular, por lo que tomar un valor intermedio para β suele ser lo más apropiado.

2.5.2. EXTRAPOLACIÓN VERTICAL

Con la información obtenida en el paso anterior se realiza una *extrapolación vertical* para definir el campo de velocidades en la totalidad del dominio.

El viento se desarrolla, en primer lugar, como consecuencia de diferencias espaciales en la presión atmosférica. Estas diferencias de presión normalmente son causadas por una diferente absorción de la radiación solar. En un plano horizontal, el viento fluye de las zonas de alta presión a zonas de baja presión y verticalmente de zonas de baja presión a zonas de alta presión. La velocidad del viento es proporcional al cambio de presión por unidad de distancia o gradiente de presión. Las zonas con presiones similares se representan en los mapas meteorológicos unidas mediante líneas imaginarias denominadas isobaras. Cuanto más juntas están unas isobaras, mayor será la fuerza del viento.

Un segundo factor que afecta el movimiento del aire es la fuerza de Coriolis, debida a la rotación terrestre. El parámetro $f = 2\Theta \, sen \, \phi_l$ se denomina parámetro de Coriolis, siendo $\Theta = 7{,}292 \times 10^{-5} \; s^{-1}$ la velocidad de rotación de la Tierra y ϕ_l la latitud. Se considera positiva en el hemisferio norte, nula en el ecuador y negativa en el hemisferio sur.

En tercer lugar puede aparecer una aceleración centrípeta, cuando el viento gira en torno a un centro. Por último, aparece la fricción debida al desplazamiento del aire. Los vientos influenciados por el gradiente de presión y la fuerza de Coriolis

se denominan vientos geostróficos.

Estratificación atmosférica:

En este modelo se considera una división de la capa más baja de la atmósfera en distintas subcapas, en las que la extrapolación vertical de las velocidades de viento se realiza de forma diferente, como puede observarse en la figura 2.1.

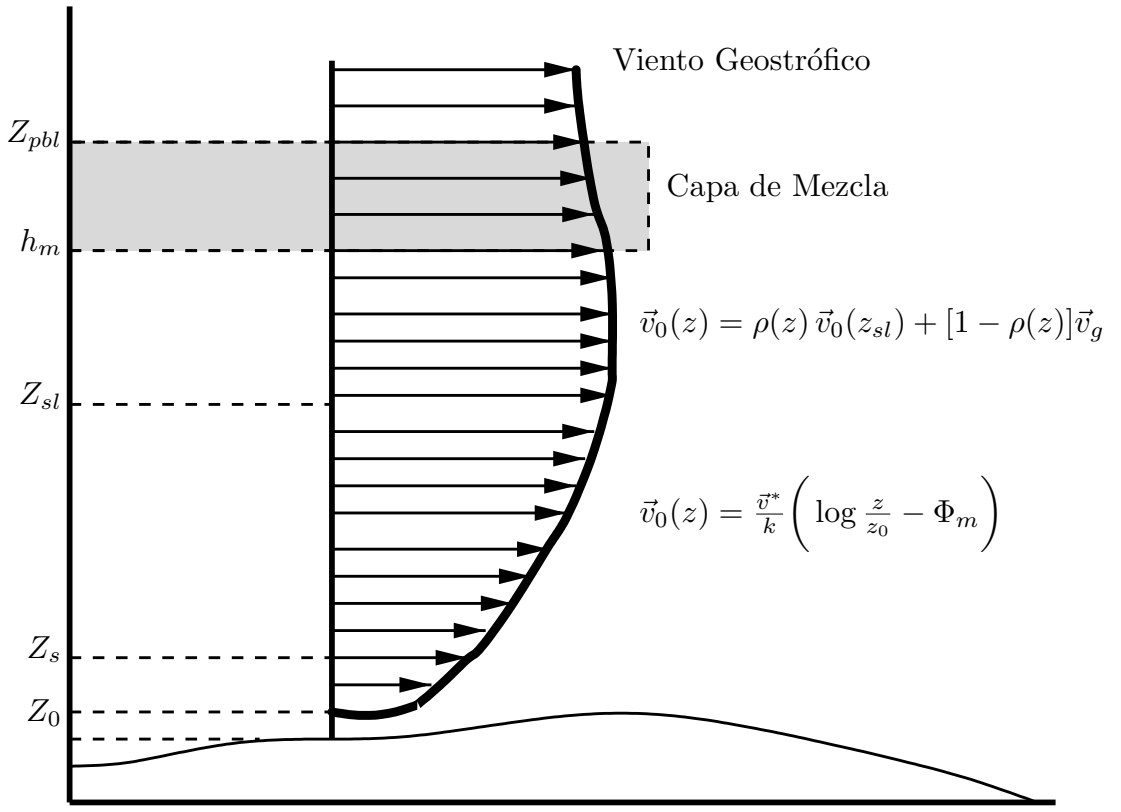


Figura 2.1: *Perfil vertical de viento definido sobre cada capa de la estratificación atmosférica.*

Así, la capa límite planetaria está situada a una altitud z_{pbl} sobre el nivel del terreno, y es la capa de la atmósfera, situada por debajo de la atmósfera libre, que está afectada directamente por la fricción de la superficie de la tierra (conocida también como capa límite atmosférica). La altitud de la capa límite planetaria z_{pbl} sobre el terreno se ha tomado tal que la dirección e intensidad del viento es constante a partir de esa altura [101]:

$$z_{pbl} = \frac{\gamma\,|\vec{v}^*|}{f} \tag{2.19}$$

siendo γ un parámetro comprendido entre 0,15 y 0,45 que depende de la estabilidad de la atmósfera y $\vec{v}^*$ la velocidad de fricción que será definida más adelante a partir

de los valores obtenidos en la interpolación horizontal.

La capa de mezcla, también llamada capa límite convectiva, es la capa límite atmosférica sujeta a fenómenos convectivos causados por el calor superficial. El aire está bien mezclado, es decir, el viento y el potencial de temperatura son prácticamente constantes con la altura. La altitud de la capa de mezcla h_m se considerará igual a z_{pbl} para condiciones neutras e inestables. En condiciones estables se aproxima por

$$h_m = \gamma' \sqrt{\frac{|\vec{v}^*|\, L}{f}} \tag{2.20}$$

donde usualmente se toma el parámetro $\gamma' = 0{,}4$ [117] y L es la longitud de Monin-Obukov, que se calcula a través de la fórmula de Liu [85],

$$\frac{1}{L} = a z_0^b \tag{2.21}$$

con a y b, definidas por la clase de estabilidad de Pasquill (Ver Tabla 2.1):

Clase de estabilidad de Pasquill	a	b
A (Extremadamente Inestable)	-0.08750	-0.1029
B (Moderadamente Inestable)	-0.03849	-0.1714
C (Ligeramente Inestable)	-0.00807	-0.3049
D (Neutra)	0.00000	0.0000
E (Ligeramente Estable)	0.00807	-0.3049
F (Moderadamente Estable)	0.03849	-0.1714

Tabla 2.1: *Coeficientes a y b para el cálculo de la longitud de Monin Obukov según la clase de estabilidad de Pasquill.*

La capa superficial, localizada a una altura z_{sl} sobre la superficie, es la capa baja, dentro de la capa límite planetaria, inmediatamente adyacente a la capa de la superficie de la tierra, en la que la fuerza de arrastre de fricción es dominante. Conocido el valor de la altura de la capa de mezcla h_m, la altitud de la capa superficial se suele fijar en [117]

$$z_{sl} = \frac{h_m}{10} \tag{2.22}$$

Estabilidad atmosférica:

El concepto de estabilidad atmosférica está relacionado tanto con la turbulencia atmosférica como con el gradiente vertical de temperatura y las situaciones de inversión térmica. La estabilidad atmosférica nos proporciona una medida cualitativa de las variaciones de la densidad del aire, debidas a los cambios de presión y temperatura y que influyen en determinados movimientos atmosféricos.

Las condiciones atmosféricas pueden clasificarse como:

Estable: Si una masa de aire sube se encontrará rodeada de aire más caliente y, por tanto, menos denso que ella, lo que la hará bajar; y si baja, se encontrará rodeada de aire más frío (más denso), y tenderá a subir. Esta tendencia que tiene el aire de permanecer en la misma capa es lo que se denomina estabilidad de la estratificación atmosférica.

Inestable: En condiciones inestables la temperatura potencial disminuye con la altura, incrementándose los movimientos verticales, es decir si el aire sube se encontrará rodeado de aire más frío y denso que él, y tenderá a seguir subiendo; y si baja se encontrará con aire más caliente y ligero, y tenderá a seguir bajando.

Neutra: Si un volumen de aire (después de un desplazamiento vertical en una capa atmosférica sin mezclar con el aire circundante) experimenta una fuerza neta vertical nula, los movimientos ascensionales no se verán perturbados por el gradiente térmico, entonces la capa atmosférica se asume neutralmente estratificada. Bajo tales condiciones, dicho volumen ni tiende a volver a su posición original (estratificación estable) ni acelera alejándose de ella (estratificación inestable).

La estabilidad atmosférica puede ser caracterizada mediante la tabla definida por Pasquill.(Ver Tabla 2.2)

Perfil vertical de velocidades de viento:

Como se muestra en la Figura 2.1, se considera un perfil logarítmico-lineal [57] en la capa límite planetaria, que tiene en cuenta la interpolación horizontal [70],

Clase de estabilidad de Pasquill

	Insolación			Noche	
Velocidad del viento en la superficie (m/s)	Fuerte	Moderada	Ligera	Cubierto ó $\geq 4/8$ nubes	$\leq 3/8$ nubes
<2	A	A-B	B	-	-
2-3	A-B	B	C	E	F
3-5	B	B-C	C	D	E
5-6	C	C-D	D	D	D
>6	C	D	D	D	D
Para A-B, tomar la media de los valores de A y B, etc.					

Tabla 2.2: *Clases de estabilidad de Pasquill según la velocidad del viento en la superficie y la insolación. Insolación fuerte corresponde al mediodía soleado de mitad de verano en Inglaterra; insolación ligera a condiciones similares en mitad del invierno. La noche se refiere al periodo que va desde una hora antes de ponerse el sol hasta una hora después de salir. La clase neutra D debería ser usada también, a pesar de la velocidad del viento, para cielos cubiertos durante el día o la noche, y para cualquier condición del cielo durante las horas precedente y siguiente de la noche definida anteriormente.*

el efecto de la rugosidad en la intensidad y dirección del viento, y la estabilidad del aire (neutra, estable o inestable) según la clasificación de Pasquill. En la capa superficial se construye un perfil logarítmico de velocidades de viento definido por,

$$\vec{v}_0(z) = \frac{\vec{v}^*}{k}\left(\log\frac{z}{z_0} - \Phi_m\right) \qquad z_0 < z \leq z_{sl} \tag{2.23}$$

donde $\vec{v}_0$ es la velocidad del viento, $k \simeq 0{,}4$ es la constante de von Karman y z es la altura sobre el terreno del punto estudiado. El término $\vec{v}^*$ representa la velocidad de fricción. En el flujo turbulento atmosférico las fuerzas que se oponen al movimiento están caracterizadas por la acción que ejercen las rugosidades o asperezas propias de la orografía del terreno. La velocidad de fricción se obtiene en cada punto a partir de las medidas interpoladas a la altura de las estaciones

(*interpolación horizontal*),

$$\vec{v}^* = \frac{k\ \vec{v}_0(z_e)}{\ln \frac{z_e}{z_0} - \Phi_m(z_e)} \tag{2.24}$$

Asimismo, z_0 corresponde a la longitud de rugosidad de la zona. El concepto de longitud de rugosidad viene a definir una altura por encima del terreno diferente de $z = 0$, donde, en teoría de la capa superficial, la velocidad del viento es cero. El valor de z_0 depende de las características del terreno. Una forma de estimarla es mediante valores estándar para diferentes tipos de terreno [64]; ver figura 2.2. Otros autores la definen como $z_0 = \frac{e}{30}$, donde e es la altura media de los obstáculos existentes en la zona de estudio.

Por último, Φ_m, es una función que depende de la estabilidad del aire [117]:

$$\begin{aligned}
\Phi_m &= 0 && \text{(neutra)} \\
\Phi_m &= -5\frac{z}{L} && \text{(estable)} \\
\Phi_m &= \log\left[\left(\frac{\theta_m^2+1}{2}\right)\left(\frac{\theta_m+1}{2}\right)^2\right] - 2\arctan\theta_m + \frac{\pi}{2} && \text{(inestable)}
\end{aligned}$$

donde

$$\theta_m = (1 - 16\frac{z}{L})^{1/4} \tag{2.25}$$

El viento geostrófico es una buena aproximación al viento real con flujo uniforme en la alta atmósfera (atmósfera libre), donde la fricción y aceleraciones no son importantes. La forma general de la expresión usada para calcular el viento geostrófico es la bien conocida ley de resistencia geostrófica (*geostrophic drag low*) [85]

$$|\vec{V}_g| = \frac{|\vec{v}^*|}{k}\sqrt{\left(\log\frac{|\vec{v}^*|}{f z_0} - A\right)^2 + B^2} \tag{2.26}$$

Los valores de los coeficientes A y B tienden a ser $\sim 1{,}8$ y $\sim 1{,}5$ respectivamente, que son los valores aceptables para condiciones neutras de estabilidad atmosférica.

El viento en la superficie se supone que gira un ángulo ϕ_g con respecto a $\vec{V}_g$, dado por la relación

$$\phi_g = \sin^{-1}\left(\frac{-B|\vec{v}^*|}{k|\vec{V}_g|}\right) \tag{2.27}$$

En nuestro modelo, desde z_{sl} hasta z_{pbl} se realiza una interpolación lineal en $\rho(z)$ con el viento geostrófico $\vec{v}_g$

$$\vec{v}_0(z) = \rho(z)\,\vec{v}_0(z_{sl}) + [1 - \rho(z)]\vec{v}_g \qquad \text{con} \qquad z_{sl} < z \leq z_{pbl} \tag{2.28}$$

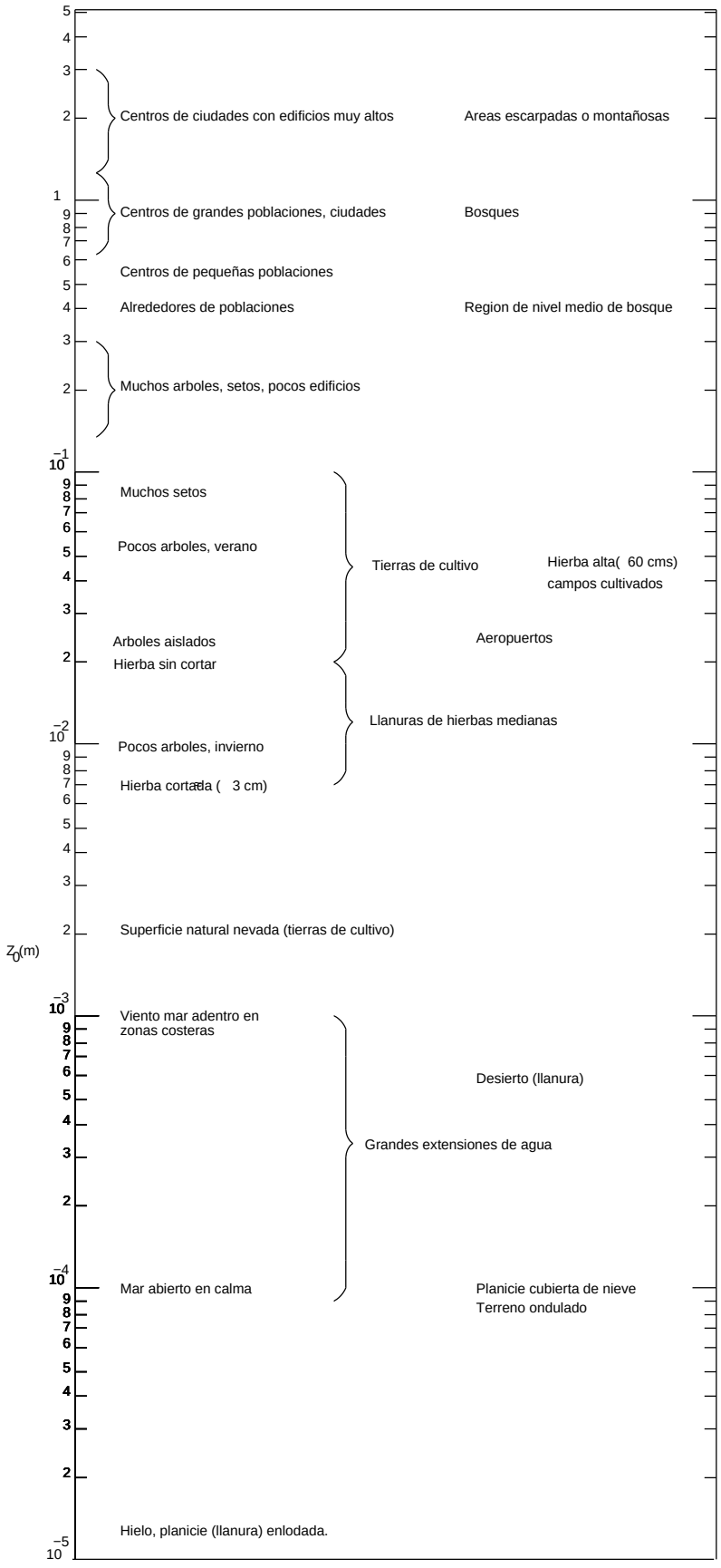


Figura 2.2: *Longitud de rugosidad: valores aproximados de z_0 para distintos tipos de terreno definidos por McRae (1982)*

donde $\rho(z)$ es

$$\rho(z) = 1 - \left(\frac{z - z_{sl}}{z_{pbl} - z_{sl}}\right)^2 \left(3 - 2\frac{z - z_{sl}}{z_{pbl} - z_{sl}}\right) \tag{2.29}$$

Finalmente, este modelo considera

$$\vec{v}_0(z) = \vec{v}_g \qquad \text{si} \qquad z > z_{pbl} \tag{2.30}$$

$$\vec{v}_0(z) = 0 \qquad \text{si} \qquad z \leq z_0 \tag{2.31}$$

Capítulo 3

DISCRETIZACIÓN MEDIANTE ELEMENTOS FINITOS

3.1. GENERALIDADES

En la actualidad los modelos de viento de Masa Consistente son los más utilizados, por su economía, pero alcanzar de forma analítica la solución de la ecuación elíptica, (13), que los define, es muy difícil ó prácticamente imposible; por ello, en lugar de buscar una solución más exacta del problema simplificado, resulta más aconsejable tratar de encontrar soluciones aproximadas mediante el Cálculo Numérico, por procedimientos iterativos, que con el notable desarrollo de la computación, se han ido perfeccionando extraordinariamente.

La solución numérica de problemas con formulación en derivadas parciales pasa por un proceso de discretización mediante el uso del Método de Elementos Finitos (MEF), Diferencias Finitas ó Volúmenes Finitos, que nos permita valorar la solución en un número finito de puntos del terreno.

Concretamente la solución de la ecuación elíptica mencionada se puede afrontar utilizando Diferencias Finitas, pero también se emplean Elementos Finitos tridimensionales, con un mallado tetraédrico, usando técnicas adaptativas en 3D, que mejoran substancialmente los resultados.

3.2. MALLAS ADAPTATIVAS

Para realizar modelos de viento en terrenos de orografía compleja, como en las Islas Canarias, resulta imprescindible disponer de mallas de alta calidad para discretizar los dominios de estudio. Los modelos de viento ya mencionados utilizan mallas regulares encajadas, pero con este tipo de mallado, en los dominios de orografía muy variada, el tamaño del elemento ha de ser excesivamente pequeño para poder captar toda la información del terreno, lo que provoca que en zonas donde no se requiere tanto detalle exista una innecesaria concentración de nodos, dando lugar, por tanto, a Sistemas de Ecuaciones de alto orden, cuya resolucuión puede suponer un coste computacional impracticable. Una propuesta razonable es el uso de mallas no estructuradas adaptativas, [89], que permiten un tamaño pequeño de elemento allí donde es necesario, manteniendo tamaños de elementos mayores en otras zonas del dominio donde no se requiera gran precisión. Con estos modelos adaptativos el mallador puede concentrar los elementos en las proximidades del terreno, que es donde se necesita mayor precisión; pudiendo recurrir al refinamiento local de la malla, en función de los requerimientos de la solución numérica, en aquellas zonas del dominio donde exista mayor error. Además de ello se debe disponer de un método de suavizado que asegure una buena calidad de la malla y su eventual desenredo si fuera necesario.

Para alcanzar el objetivo asociado a la generación de la malla tridimensional se parte de un generador de mallas bidimensional [26] que permite refinar y desrefinar triángulos utilizando el algoritmo 4T [87]. Por otra parte, se dispone de un código capaz de generar mallas de tetraedros mediante el algoritmo de triangulación de Delaunay [24]. La idea inicial para construir la malla 3D adaptada a la topografía del terreno intenta combinar estos dos códigos programados en FORTRAN. En primer lugar, se utiliza el código bidimensional para generar una malla adaptada de triángulos que aproxime la orografía del terreno con una precisión preestablecida. Esta precisión vendría determinada por el parámetro de desrefinamiento introducido en [26]. Una vez definida la distribución de puntos sobre el terreno, es necesario introducir una función de espaciado vertical para generar la nube de puntos en el resto del dominio. Es fundamental que esta función concentre una mayor densidad de puntos cerca del terreno, debido a que es en esta zona donde se

producen mayores variaciones en el campo de velocidades de viento. Esta nube de puntos sirve de base para la construcción de una malla conforme de manera que los puntos de la nube se conviertan en los nodos de la malla. Si bien existen diferentes posibilidades para abordar este objetivo, se opta por la transformación de la nube de puntos a un paralelepípedo auxiliar y construir en él la triangulación. Deshaciendo la transformación anterior y manteniendo la topología de la malla construida en el paralelepípedo, se obtiene la malla resultante en el dominio real conforme con la superficie del terreno [67, 71]. De esta forma queda resuelto el problema de la conformidad de la malla con el terreno, pero eventualmente surge un importante problema de enredo de tetraedros. Para resolverlo se desarrolla un algoritmo capaz de suavizar y desenredar la malla simultáneamente [25, 66].

En las primeras experiencias de nuestro grupo de investigación, sobre ajustes de campos de viento, se empezó con problemas bidimensionales. Dichos modelos no consideraban la orografía del terreno y simplemente construían un campo inicial mediante la interpolación de medidas de las estaciones que dependía únicamente de la distancia a los nodos. Este campo se ajustaba después mediante un modelo de masa consistente en 2D que utiliza elementos finitos adaptativos [116]. Con el desarrollo del generador de mallas de tetraedros, se pasa a un modelo más sofisticado que tiene en cuenta la orografía del terreno [70, 90]. Así, se construye una fórmula de interpolación que considera la distancia horizontal y la diferencia de cotas entre puntos. El perfil vertical de viento tiene en cuenta la rugosidad del terreno. Por otro lado, el carácter 3D del modelo permite incorporar consideraciones físicas de la atmósfera y su estratificación. Una vez construido el campo tridimensional inicial, mediante una interpolación horizontal de las medidas y la correspondiente extrapolación vertical, el problema elíptico asociado al modelo de ajuste se resuelve mediante elementos finitos, utilizando técnicas de refinamiento local basadas en la subdivisión en 8-subtetraedros [61, 62, 45] para mejorar la solución numérica del problema. Este método de refinamiento local acota la degeneración de los elementos, introduce una mínima propagación de la zona refinada por conformidad y es razonablemente rápido. Para su desarrollo se opta por el paradigma de la programación orientada a objetos y se implementa en el lenguaje de programación C++. Para determinar los elementos de la malla que deben ser

refinados se considera el uso de un simple indicador de error basado en el gradiente de la solución numérica y el diámetro de cada elemento.

Los sistemas de ecuaciones asociados resultan de matriz simétrica y definida positiva (SDP), lo que facilita su resolución mediante los métodos iterativos basados en los subespacios de Krylov, especialmente con el método del Gradiente Conjugado (GC)

Asimismo, el modelo se diseña con la intención de que pueda proporcionar un campo de velocidades a un modelo de dispersión de contaminantes en la atmósfera. La capacidad del mallador para discretizar chimeneas situadas sobre el terreno nos motivó para transformar el campo de velocidades de viento con el objeto de incluir el movimiento de los gases emitidos por las chimeneas debido a la velocidad de salida y a las diferencias de temperatura con el aire atmosférico [73]. Esta modificación se llevó a cabo incorporando un modelo de pluma gaussiana al modelo de viento [13]. Así, se comprueba que si dicha aportación al campo de velocidades se lleva a cabo en el campo interpolado, el modelo de masa consistente nos proporciona un campo ajustado que incluye el efecto de las chimeneas y sigue siendo incompresible. Un campo de estas características tiene mucho interés a la hora de resolver las ecuaciones de convección-difusión-reacción de un modelo de contaminación atmosférica [100, 115].

La implementación del modelo de viento, así como de los módulos de resolución de sistemas de ecuaciones y de precondicionadores, se realiza utilizando el lenguaje C. Es importante, dado el tamaño considerable que pueden alcanzar algunos problemas, que se haga una gestión dinámica de la memoria RAM. También conviene usar estructuras de datos adecuadas a cada aspecto del código; así, se emplea almacenamiento compacto tipo morse para las matrices y listas encadenadas para facilitar el ensamblaje del sistema de ecuaciones. En definitiva, la programación debe tener como meta un uso eficiente de la memoria sin penalizar el tiempo de ejecución. Además, la entrada de datos al modelo debe ser lo más cómoda posible para el usuario, evitando el uso de ficheros crípticos donde únicamente figuran cifras. En este sentido, se propone el uso de ficheros de texto con una sintaxis sencilla que permita al usuario introducir la información de manera natural y donde se puedan insertar comentarios descriptivos.

Después de estos antecedentes nos proponemos describir a continuación el proceso de creación de una malla de tetraedros que respete la topografía de una región rectangular con una precisión determinada, disponiendo únicamente de la información digitalizada del terreno. Este problema posee cierta dificultad debido a la fuerte irregularidad de la superficie del terreno. Por otra parte, deseamos que la malla esté adaptada, es decir, que exista una densidad de nodos mayor donde sea necesario para definir las características geométricas de nuestro dominio. La malla generada podrá utilizarse como malla base para la simulación numérica de procesos naturales en el dominio; por ejemplo, ajuste de campos de viento [116, 70], propagación de fuego [68], contaminación atmosférica [115], etc. Estos fenómenos tienen su mayor efecto en las zonas próximas al terreno, de ahí que también sea deseable que la densidad de nodos aumente al acercarnos a éste. Sobre esta malla base, adaptada a las características geométricas del dominio, se podrán aplicar posteriormente algoritmos de refinamiento y desrefinamiento de tetraedros para mejorar la solución numérica del problema [61, 62, 45, 44]. Estos algoritmos tendrán un especial interés en los problemas evolutivos.

Nuestro dominio está limitado en su parte inferior por el terreno y en su parte superior por un plano horizontal situado a una altura en la que las magnitudes objeto del estudio puedan ser consideradas estables. Las paredes laterales están formadas por cuatro planos verticales, paralelos dos a dos. Las ideas básicas para la construcción de la malla inicial combinan, por un lado, la utilización de un algoritmo de refinamiento y desrefinamiento para dominios bidimensionales y, por otro lado, un algoritmo de generación de mallas de tetraedros basado en la triangulación de Delaunay.

Es bien conocido que para construir una triangulación de Delaunay es necesario definir una nube de puntos en el dominio y su frontera. Estos nodos serán precisamente los vértices de los tetraedros que conforman la malla. La generación de puntos en nuestro dominio se realizará sobre diferentes capas, reales o ficticias, definidas desde el terreno hasta la frontera superior del dominio. En concreto, se construye una triangulación con una distribución uniforme de puntos en el plano superior del dominio. Esta malla bidimensional puede ser obtenida a partir de la

realización de un cierto número de refinamientos globales sobre una malla simple o, por ejemplo, puede también construirse realizando una triangulación de Delaunay sobre la distribución uniforme de puntos establecida. Consideraremos la malla obtenida como el nivel más bajo de la secuencia que define la distribución de los puntos en el resto de las capas. Sobre esta malla regular aplicamos a continuación el algoritmo de refinamiento y desrefinamiento, [26, 83], para definir la distribución de los nodos de la capa correspondiente a la superficie del terreno. Para ello, en primer lugar se construye una función que interpola las cotas obtenidas a partir de una digitalización de la topografía de la zona rectangular estudiada. En segundo lugar, realizamos una serie de refinamientos globales sobre la malla uniforme hasta conseguir una malla regular capaz de captar la variación topográfica del terreno. El máximo grado de discretización viene definido por el nivel de detalle de la digitalización. Posteriormente, se realizará un desrefinamiento sobre estos últimos niveles de malla utilizando como parámetro de desrefinamiento el máximo error de cotas permitido entre la superficie real del terreno y la superficie definida mediante la interpolación a trozos obtenida con la malla bidimensional resultante.

Una vez que se ha definido la distribución de nodos sobre el terreno y sobre el plano superior del dominio, comenzamos a distribuir los nodos situados entre ambas capas. Esta distribución se puede realizar mediante diferentes estrategias, en las que interviene una función de espaciado vertical . La característica fundamental de esta función es que el grado de discretización obtenido sobre la vertical debe disminuir con la altura, o a lo sumo mantenerse constante.

Esta nube de puntos será utilizada por nuestro mallador tridimensional basado en la triangulación de Delaunay. Para evitar posibles problemas de conformidad con la superficie del terreno, se propone construir la malla de tetraedros con la ayuda de un paralelepípedo auxiliar. Sobre su cara inferior se sitúan todos los nodos distribuidos sobre el terreno, proyectados sobre un plano horizontal situado a la altura definida por la cota mínima de la región de estudio, y sobre su cara superior se sitúan los puntos distribuidos en el plano superior del dominio a su altura real. Esto conlleva una transformación de coordenadas, atendiendo a la función de espaciado sobre cada vertical, para situar el resto de puntos en el

paralelepípedo auxiliar. Estos detalles nos asegurarán que la distancia máxima entre dos puntos consecutivos sobre la misma vertical del dominio real será siempre igual o inferior que la correspondiente distancia establecida en el paralelepípedo auxiliar.

Se define la nube de puntos en el dominio real y se analiza la transformación entre el dominio real y el paralelepípedo auxiliar en el que se construye la malla mediante una versión del método de triangulación de Delaunay [24]. Proponemos cuatro estrategias diferentes para determinar el número de puntos generados sobre la vertical de cada nodo de la malla bidimensional adaptada a la superficie del terreno, y analizamos las características fundamentales de cada una de ellas. Las dos primeras estrategias generan puntos sobre capas definidas entre el terreno y la frontera superior del dominio. En estos dos casos, el número de capas reales que se desea crear será introducido como dato. En la primera estrategia el grado de concentración de las capas hacia el terreno se impone, mientras que en la segunda se obtiene automáticamente en función del tamaño de los elementos existentes en la malla bidimensional adaptada a la superficie del terreno. En las dos últimas estrategias las capas generadas serán virtuales, es decir, no se define un número concreto de superficies interiores al dominio sobre las que se sitúan los puntos. Por ello, diremos que en estas dos últimas estrategias el número de capas es variable, y será calculado automáticamente en función de los tamaños de los elementos existentes en la malla bidimensional que define el terreno, o, también, en la correspondiente a la frontera superior del dominio. En concreto, la tercera estrategia concentrará los puntos hacia el terreno en función del tamaño de los elementos definidos sobre él. En cambio, la última estrategia determina automáticamente, para cada nodo del terreno, una función de espaciado vertical con el objeto de respetar las distancias desde el primer punto generado hasta el terreno, y desde el último punto generado hasta la frontera superior, en función de los tamaños de los elementos existentes sobre ambas superficies.

Una vez que se ha construido la triangulación de Delaunay de la nube de puntos en el paralelepípedo, procedemos a situar los puntos en sus posiciones reales manteniendo la topología de la malla. Hay que tener en cuenta que este proceso de compresión de la malla puede dar lugar a cruces de tetraedros que habrá que

deshacer posteriormente. Asimismo, será aconsejable aplicar una etapa de suavizado para mejorar la calidad de los elementos de la malla resultante.

3.3. GENERACIÓN DE MATRICES VARIABLES

Para la discretización mediante elementos finitos de la formulación clásica del problema elíptico que aparece en los modelos de Campos de Viento de Masa Consistente, dada por la expresión (13), con las condiciones de contorno (14) y (15) se ha utilizado una malla de tetraedros, generada mediante las técnicas adaptativas ya descritas.

Esto conduce a un conjunto de matrices elementales de dimensión 4×4 asociadas al elemento Ω_e, siendo $\hat{\psi}_i$ la función de forma correspondiente a su i-ésimo nodo, $i = 1, 2, 3, 4$, definidos en el elemento de referencia $\hat{\Omega}_e$ y $|\mathbf{J}|$ el jacobiano de la transformación de Ω_e a $\hat{\Omega}_e$,

$$\begin{aligned}\{\mathbf{A}^e\}_{ij} = \int_{\hat{\Omega}_e} \{&(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial x})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial x})+ \\ &+(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial y})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial y})+ \\ +\epsilon\,&(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial z})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial z})\} \cdot |\mathbf{J}| \;\; d\xi\, d\eta\, d\varphi\end{aligned} \tag{3.1}$$

y de vectores elementales de 4×1,

$$\begin{aligned}\{\mathbf{b}^e\}_i = \int_{\hat{\Omega}_e} -\frac{1}{T_h}\{&u_0(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial x})+ \\ &+v_0(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial y})+ \\ +w_0\,&(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial z})\} \cdot |\mathbf{J}| \;\; d\xi\, d\eta\, d\varphi\end{aligned} \tag{3.2}$$

Nótese que la matriz elemental puede escribirse como

$$\{\mathbf{A}^e\}_{ij} = \{\mathbf{M}^e\}_{ij} + \epsilon\, \{\mathbf{N}^e\}_{ij} \tag{3.3}$$

El ensamblaje de tales matrices elementales conduce a un sistema lineal de la forma:

$$A_\epsilon \, x_\epsilon = b_\epsilon \tag{3.4}$$

donde A_ϵ es una matriz simétrica variable del tipo

$$A_\epsilon = M + \epsilon \, N \tag{3.5}$$

siendo M y N dos matrices, tipo "sparse", diferentes, Simétricas Definidas Positivas,(SDP), constantes para un nivel de discretización dado y ϵ el llamado parámetro de estabilidad del modelo, ya mencionado; debiendo resolverse el sistema para cada valor diferente de ϵ.

Precisamente el objetivo principal de esta tesis consiste en plantear propuestas que hagan posible solucionar por métodos iterativos, no demasiado costosos, este tipo de sistemas de ecuaciones lineales de matrices variables.

Capítulo 4

ESTIMACIÓN DE PARÁMETROS

4.1. CONSIDERACIONES PREVIAS

La eficiencia de los modelos de masa consistente para ajuste de campos de viento depende en gran medida de ciertos parámetros que aparecen en las distintas etapas del proceso, especialmente de algunos de los que intervienen en la construcción del campo de viento inicial y de los módulos de precisión de Gauss.

En general, los valores de estos parámetros se toman usando una serie de reglas empíricas. Se puede plantear su estimación de manera automática, tal que las velocidades observadas en las estaciones de medida sean regeneradas de la forma más exacta posible por el modelo, dando lugar por tanto a un problema inverso. Existen diversos métodos de resolución de problemas inversos relacionados con la estimación de parámetros. De entre ellos, se han elegido los algoritmos genéticos, por ser una herramienta robusta y flexible, que puede ser competitiva ya que los cálculos pueden paralelizarse.

Para el cálculo automático de ciertos parámetros del modelo de viento se plantea el siguiente problema inverso: de las N estaciones de medida disponibles se toman N_r como referencia; el resto, se utilizan para el cálculo del viento. El viento así obtenido se compara con el medido en las N_r estaciones de referencia. Para la estimación de los parámetros del modelo se procede a minimizar la diferencia entre los resultados obtenidos y las medidas observadas en las estaciones de referencia.

Esta técnica supone una mejora sustancial sobre la propuesta de Barnard [8] para la estimación de uno solo de los parámetros por un procedimiento de ensayo y error. E. Rodríguez, en su Tesis: *Modelización y simulación numérica de campos de viento mediante elementos finitos adaptativos en 3-D*, [89], propone extenderlo a un total de cuatro de los parámetros que intervienen en el modelo y automatizar su cálculo; de tal forma que la función a minimizar sea:

$$F(\alpha, \beta, \gamma, \gamma') = \frac{1}{N_r} \sum_{n=1}^{N_r} \frac{|\vec{v}_n - \vec{v}(x_n, y_n, z_n)|}{|\vec{v}_n|}$$

donde $\vec{v}(x_n, y_n, z_n)$ es la velocidad del viento obtenida por el modelo en la posición de la estación n, y N_r es el número de estaciones de referencia, $1 \leq N_r \leq N$, siendo N el número total de estaciones de medida disponibles.

En primer lugar se considera el parámetro de estabilidad

$$\alpha = \frac{\alpha_1}{\alpha_2} = \sqrt{\frac{T_v}{T_h}} = \sqrt{\epsilon}$$

ya establecido como fórmula 2.12, que se deriva del funcional 2.8, y cuyo mínimo no varía si se divide por α_2^2. Hay que señalar que para $\alpha >> 1$ predomina el ajuste de viento en la dirección vertical, mientras que para $\alpha << 1$ el ajuste tiene lugar predominantemente sobre el plano horizontal. Por lo tanto la elección de α, ó ϵ, determina que el viento tienda a rodear los obstáculos o a sobrepasarlos. Diversos experimentos numéricos han hecho patente que el comportamiento de los modelos de masa consistente depende sensiblemente de la elección de los valores de ϵ, por lo que se presta particular atención a este problema. Diversos autores han estudiado cómo parametrizar la estabilidad debido a que la dificultad en la determinación de los valores de α han limitado el uso de modelos de masa consistente en terrenos de orografía compleja. En [103, 54, 14], los autores proponen tomar $\alpha = 10^{-2}$, o sea, proporcional a la magnitud de w/u. Otros, como Ross [91] y Moussiopoulos [76], relacionan α con el número de Froude, mientras que Geai, [38], Lalas, [58], y Tombrou, [108], proponen que el parámetro α varíe en la dirección vertical. Finalmente, Barnard et al., [8], proponen un procedimiento para obtener α en cada simulación del campo de viento. La idea es usar N velocidades de viento observadas para obtener el campo de viento y usar las restantes N_r como referencia. Entonces se realizarían diversas simulaciones con distintos valores de ϵ, lo que supondría

resolver la ecuación (3.4)

$$A_\epsilon x_\epsilon = b_\epsilon$$

varias veces, confirmando la idea de que es muy interesante poder resolver este tipo de ecuaciones, de matrices variables, de la forma más eficiente posible.

El valor que más acerque el viento estimado al observado en las estaciones de referencia es el que se toma como valor del parámetro de estabilidad.

Este método proporciona valores de ϵ que sólo son válidos para cada caso particular y, por tanto, no proporciona valores válidos *a priori* para otras simulaciones. E. Rodriguez, en su Tesis, [89], estudia una versión del método propuesto por Barnard et al., [8], utilizando algoritmos genéticos como herramienta de optimización que permite una selección automática de ϵ.

El segundo parámetro a estimar es el coeficiente de peso β ($0 \leq \beta \leq 1$) de la ecuación 2.18, correspondiente a la interpolación horizontal de las medidas de viento observadas. Cuando $\beta \rightarrow 1$ adquiere más importancia la *distancia horizontal* de cada punto a las estaciones de medida, mientras que para $\beta \rightarrow 0$ se da más peso a la *distancia vertical* entre cada punto y las estaciones [70]. En general, para terrenos complejos se utiliza la segunda aproximación [79]. En orografías más llanas o en análisis horizontales en 2-D, se utiliza la primera. En aplicaciones más realistas existirán zonas con orografía compleja y zonas de orografía más regular, lo que sugiere el uso de valores intermedios de β.

El siguiente parámetro objeto de estimación es γ, que aparece en la ecuación 2.19 y está relacionado con la capa límite planetaria en la estratificación atmosférica. Existen diferentes autores que proponen distintos rangos para este parámetro. Panofsky y Dutton, [80], proponen el intervalo [0.15, 0.25]. Sin embargo, en [85] se utiliza directamente el valor $\gamma = 0{,}3$ en el código de su programa *WINDS*, mientras que para Baas, [7], γ ha de estar dentro del intervalo [0.3, 0.4]. En nuestras simulaciones el espacio de búsqueda de γ incluye todas estas posibilidades.

Finalmente, también resulta de interés obtener estimaciones de los valores del parámetro γ', que interviene en el cálculo de la altura de la capa de mezcla en el caso de condiciones atmosféricas estables, véase la fórmula (2.20). Garrat propone directamente $\gamma' = 0{,}4$. También en el código de *WINDS* el valor de γ' está en

torno a 0,4. Así, hemos definido el espacio de búsqueda para el valor de γ' en el entorno de 0,4.

Desde el punto de vista de esta tesis, donde nos planteamos el precondicionamiento de los sistemas de ecuaciones variables de los modelos de campos de viento (3.4):

$$A_\epsilon x_\epsilon = b_\epsilon$$

conviene aclarar que los parámetros β, γ y γ' solo intervienen en la interpolación del viento inicial y por tanto sus valores afectan, únicamente, al cálculo del vector del segundo miembro b_ϵ, mientras que el parámetro de estabilidad α, ó ϵ, afecta al cálculo del viento resultante, ya que con él cambia la estructura de la matriz A_ϵ puesto que viene expresada como (3.5)

$$A_\epsilon = M + \epsilon\, N$$

A la hora de recurrir al precondicionamiento del sistema para mejorar su resolución por métodos iterativos los valores asignados al parámetro $\epsilon(\alpha)$ tienen un gran protagonismo, de ahí que se le dedique especial atención a su estimación óptima. Para ello se propone la utilización de Algoritmos Genéticos, como herramienta robusta, para su selección.

En el Capítulo 9.2 se presentan los resultados de numerosos experimentos numéricos, con los que se comprueba el comportamiento de los distintos precondicionadores propuestos, para una amplia gama de valores de $\alpha(\epsilon)$.

4.2. ALGORITMOS GENETICOS

Los algoritmos genéticos (en los sucesivo AG) son herramientas de optimización basadas en el mecanismo de evolución natural. Producen intentos sucesivos que tienen una probabilidad cada vez mayor de alcanzar el óptimo global. Los aspectos más importantes de los AG son la construcción de una población inicial, la evaluación de cada individuo a través de la función de aptitud o función objetivo, la selección de los padres de la siguiente generación, el cruce de esos padres para crear los hijos y la mutación, que incrementa la diversidad.

En la figura 4.1 puede verse una representación esquemática del funcionamiento de los AG. Se parte de una población inicial a la que se somete a prueba a través

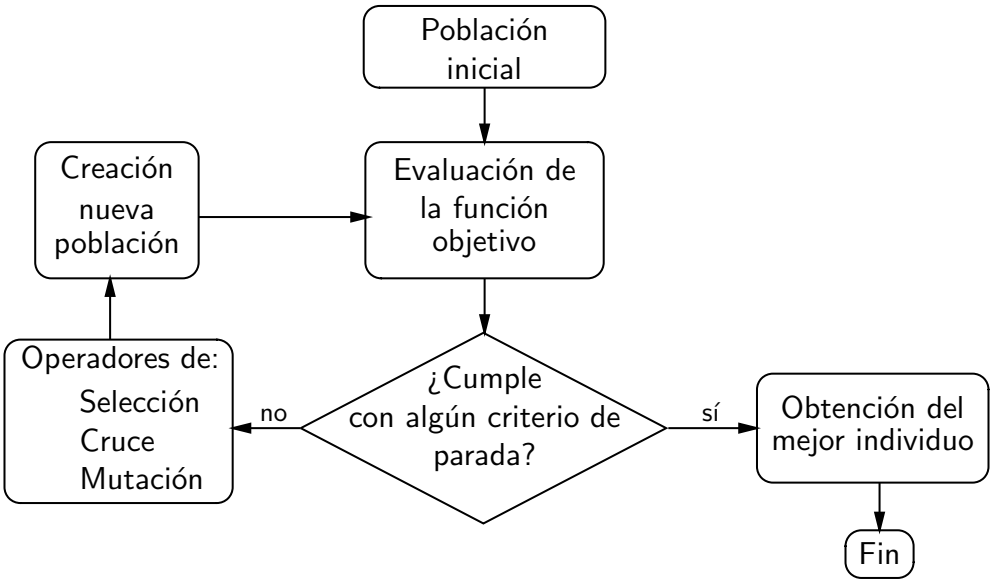


Figura 4.1: *Diagrama del funcionamiento de los AG*

de una función de aptitud, que juega el papel del medio ambiente. En función de su aptitud, una serie de individuos serán escogidos para reproducirse y dar lugar a una nueva generación, de forma que los genes de los mejor dotados se propaguen. También, como en la naturaleza, algunos individuos pueden sufrir una mutación dando lugar a variaciones que de otra forma no tendrían lugar y aumentando la diversidad de la población. Este proceso se repite hasta que se satisface un cierto criterio de parada.

Hay que decir que los AG no persiguen una simulación de los procesos naturales , sino más bien una emulación. Un algoritmo genético será tanto mejor cuanto mejores resultados proporcione al problema planteado, no cuanto más se parezca a los procesos biológicos. De hecho, desde el punto de vista biológico, la mayoría de los procesos derivados de los algoritmos genéticos resultan exageradamente simplistas, pero son suficientemente complejos como para proporcionar robustos mecanismos de búsqueda de un óptimo.

Con el fin de validar la metodología de estimación de parámetros, [72], para la tesis de E. Rodríguez, [89], se realizaron un conjunto de experimentos numéricos que además sirvieron para estudiar ciertos aspectos de gran interés, como: ¿Influye la topografía en la estimación de parámetros?. Fijada una topografía, ¿sirve la estimación realizada con una configuración de viento para otras configuraciones

diferentes?. ¿Es el refinamiento de la malla un factor importante a la hora de estimar los parámetros del modelo?.

Para contestar estas preguntas se diseñaron una serie de experimentos genéticos sobre un marco de trabajo compuesto por tres mallas diferentes y cuatro estaciones de medida. La población usada fue de 1040 individuos; el algoritmo se detenía al llegar a las 500 iteraciones. Los operadores genéticos usados fueron el de Selección Estocástica Universal (SUS), el de Cruce Uniforme (U) y el de Mutación de Reemplazo Aleatorio dentro del Rango (R).

De esta forma los rangos de variación obtenidos para los distintos parámetros fueron los siguientes:

$$10^{-6} \leq \epsilon \leq 10^{4}; \quad 0 \leq \beta \leq 1; \quad 0{,}15 \leq \gamma \leq 0{,}45 \quad y \quad 0{,}15 \leq \gamma' \leq 0{,}45.$$

Del análisis de los resultados de dichos experimentos se deduce que los valores de los parámetros depende de la topografía, pudiendo concluir lo siguiente:

• Como era de esperar, la estimación de parámetros resulta más complicada en topografías irregulares que en superficies más suaves.

• Los rangos de variación del parámetro ϵ, en cada experimento, son distintos para cada orografía. No obstante los rangos de variación obtenidos son pequeños si se comparan con los propuestos por diversos autores.

• El valor de β depende no sólo de la orografía del terreno, sino además de la dirección en que esa orografía es atacada por el viento. Por tanto, queda en parte justificada la idea de ponderar tanto el efecto de la distancia horizontal, como la diferencia de cotas en la interpolación horizontal.

• El parámetro γ alcanza el valor máximo permitido y permanece prácticamente invariable en los tres experimentos, incluso para diferentes ángulos de ataque.

• La variación de $\gamma^{'}$ sigue un cierto paralelismo con ϵ. A medida que el ajuste del viento es predominantemente horizontal (valores pequeños de ϵ), la altura de la capa de mezcla disminuye (menores valores de $\gamma^{'}$). Asimismo, cuando el ajuste empieza a dar más peso a la componente vertical del viento (aumento de ϵ), la altura de la capa de mezcla aumenta. Resultados que están de acuerdo con la definición dada previamente para la capa de la mezcla en la estratificación atmosférica descrita.

En lo que respecta a la influencia del viento en la estimación de parámetros se puede concluir lo siguiente:

• Los valores de la mejor evaluación dependen de la dirección del viento medida en las estaciones, mejorando los resultados de la estimación a medida que la dirección del viento se acerca a los 45º.

• Los valores obtenidos para ϵ, tras el ajuste, dependen de la dirección del viento.

• Tanto β como γ permanecen prácticamente constantes, independientemente de la dirección del viento.

• El paralelismo de $\gamma^{'}$ con ϵ, apuntado anteriormente, se sigue manteniendo.

Para estudiar la influencia, del grado de refinamiento de la malla, en la estimación de los parámetros del modelo, se generaron dos nuevas mallas a partir de una de las originales, τ_1. En sendos experimentos numéricos se estimaron los cuatro parámetros ϵ, β, γ y γ' para un viento con un ángulo de 45º, que es el que se corresponde con la mejor estimación de parámetros para τ_1. De los resultados obtenidos se sacaron las siguientes conclusiones:

• Los parámetros varían con el refinamiento y por tanto habrá que ajustarlos en cada malla.

• Las estimaciones realizadas sobre mallas más finas mejoran ligeramente los valores de la función objetivo. Este resultado es coherente con el hecho de que la aproximación de la solución del problema depende de la discretización del dominio. Sin embargo, al aumentar el número de elementos de la malla también crece el tiempo de cómputo necesario para estimar los parámetros. En general, parece conveniente llegar a un compromiso entre grado de refinamiento y coste computacional en relación con la posible mejora de la función objetivo.

• En los experimentos se comprobó que tras estimar adecuadamente los parámetros, si se altera uno de ellos de manera arbitraria dándole un valor razonable dentro de su rango de variación, el ajuste del modelo dista de ser bueno. La limitación impuesta por muchos programas, en el sentido de que el ajuste de los parámetros ha de hacerse de forma manual por el usuario, es bastante seria y justifica el hecho de que algunas empresas que explotan parques eólicos los usen

con muchas reservas. Esta limitación puede evitarse con el código automático de estimación, que permite mayor flexibilidad en los ajustes. Además de ser útil para crear una base de datos correspondiente a un problema concreto, con el consiguiente ahorro de tiempo de ejecución.

• Se comprobó que los parámetros obtenidos para una malla no pueden usarse para otra que resulta de su refinamiento o desrefinamiento, sino que es necesario estimarlos en cada cambio de malla.

• También se observó que existe una cierta continuidad en los parámetros, en el sentido de que el uso de unos valores de los parámetros correspondientes a la malla τ_{i+1}, en el problema con la malla τ_i, afecta menos a la función objetivo, que el uso de valores paramétricos correspondientes a mallas más alejadas, es decir:$\tau_{i+k},\ k > 1$.

• Los experimentos confirman que existen zonas del dominio que sufren mayor alteración del campo de viento que otras, cuando se cambian los parámetros estimados. Para su estudio se ha definido un parámetro de sensibilidad S_T que nos permite determinar cómo afecta la variación de los parámetros al campo de viento en general, no sólo en el entorno de las estaciones de medida:

$$S_T = \frac{\left| v_T^p - v_T^{p'} \right|}{|v_T^p|} \tag{4.1}$$

donde, v_T^p es el viento obtenido en elemento T-ésimo de la malla con el conjunto p de parámetros, y $v_T^{p'}$ el obtenido con el conjunto de parámetros p'.

Se observó que las zonas de mayor sensibilidad se producen cerca de las laderas de la montaña. Para reducir la sensibilidad y por tanto mejorar el viento obtenido en el dominio convendría situar estaciones de medida en esas zonas. De hecho, el estudio de la sensibilidad de los parámetros es una buena estrategia para determinar la distribución de nuevas estaciones de medida sobre terrenos de orografía compleja de los que se tenga información meteorológica escasa o de poca calidad.

Capítulo 5

MÉTODOS ITERATIVOS BASADOS EN LOS SUBESPACIOS DE KRYLOV

5.1. PRELIMINARES

La aplicación de Elementos Finitos o Diferencias Finitas [119] para la obtención de soluciones aproximadas de problemas de contorno en derivadas parciales, conduce a grandes sistemas de ecuaciones lineales $A\,x = b$,[9, 69] donde la matriz A es tipo *sparse* (con multitud de elementos nulos)[60].

Los métodos directos que se utilizan para la resolución de sistemas de ecuaciones lineales están basados, generalmente, en variantes de la eliminación Gaussiana [43]. Fundamentalmente consisten en conseguir una matriz triangular inferior L y otra triangular superior U, tal que $L\,U = A$ [48]. Las matrices tiangulares L y U son, asimismo *sparse*, pero más densas o llenas que la matriz original A, produciéndose el llamado efecto de relleno o "fill-in", que aumenta con el número de nodos y con la dimensión del problema, de tal forma, que, a igualdad de nodos, para una matriz obtenida de una discretización de un problema en 3-D es mayor que para uno de 2-D. Este efecto, que incrementa los requerimientos de almacenaje y el coste computacional, unido a la presencia de los errores de redondeo que influyen en los valores de la solución, teóricamente exacta, hacen más adecuados los métodos iterativos para la resolución de estos sistemas [6].

En los métodos iterativos , a partir de un vector inicial, x_0, se genera una secuencia de vectores (x_i), que converge a la solución buscada. A pesar de la convergencia relativamente lenta, el carácter *sparse* de A hace posible efectuar un elevado número de iteraciones sin un trabajo excesivo. Por otra parte, los errores de redondeo, importantes en los métodos directos, influyen, por lo general, en la velocidad de convergencia, pero no en la aproximación final.

Además de los métodos clásicos de este tipo, (Jacobi, Gauss-Seidel, SOR y SSOR) que se ajustan a estas precisiones, se han desarrollado otros que presentan mayores ventajas respecto a los mismos, sobre todo en estos sistemas *sparse*. Entre ellos han adquirido últimamente especial relevancia los algoritmos basados en los Subespacios de Krylov [94]. La rápida evolución experimentada por los sistemas informáticos ha contribuido a facilitar su implementación [41].

5.2. SUBESPACIOS DE KRYLOV

En los métodos iterativos, los sucesivos valores de la solución aproximada en la resolución de $A\,x = b$, vienen dados por la relación de recurrencia

$$x_{i+1} = x_i + B^{-1}\,(b - A\,x_i) \tag{5.1}$$

ó bien

$$B\,x_{i+1} = C\,x_i + b, \quad \textit{siendo} \quad C = B - A \tag{5.2}$$

Diferentes elecciones para las matrices B y C, en función de la matriz A, conducen a los métodos clásicos de relajación (Jacobi, Gauss-Seidel, SOR).

Pero estas expresiones también se pueden escribir en función del vector residuo,

$$r_i = b - A\,x_i$$

con lo cual

$$x_{i+1} = x_i + B^{-1}\,r_1$$

De esta forma, eligiendo una aproximación inicial, x_0, los valores de las sucesivas iteraciones se podrían calcular por las respectivas expresiones:

$x_1 = x_0 + B^{-1}\,r_0$

$$x_2 = x_1 + B^{-1}\, r_1 = x_0 + B^{-1}\, r_0 + B^{-1}(b - A\, x_1) =$$
$$= x_0 + B^{-1}\, r_0 + B^{-1}(b - A\, x_o - A\, B^{-1}\, r_0) = x_0 + B^{-1}\, r_0 + B^{-1}(r_0 - A\, B^{-1}\, r_0) =$$
$$= x_0 + 2B^{-1}\, r_0 - B^{-1}\, A\, B^{-1}\, r_0$$

$$x_3 = x_2 + B^{-1}\, r_2$$

$$x_4 = x_3 + B^{-1}\, r_3$$

.....

.....

$$x_i = x_{i-1} + B^{-1}\, r_{i-1}$$

En el caso de que $B = I$ quedaría:

$$x_1 = x_0 + r_0$$

$$x_2 = x_0 + 2\, r_0 - A\, r_0$$

$$x_3 = x_2 + r_2 = x_0 + 2\, r_0 - A\, r_0 + (b - A\, x_2) =$$
$$= x_0 + 2\, r_0 - A\, r_0 + (b - A\, x_0 + 2\, A\, r_0 - A^2\, r_0) =$$
$$= x_0 + 2\, r_0 - A\, r_0 + (r_0 + 2\, A\, r_0 - A^2\, r_0) =$$
$$= x_0 + 3\, r_0 + A\, r_0 - A^2\, r_0$$

$$x_4 = x_3 + r_3$$

.....

.....

$$x_i = x_{i-1} + r_{i-1}$$

Es decir, la i-ésima iteración de la solución aproximada se puede expresar como la suma de la aproximación inicial y una combinación lineal de i vectores:

$$x_i = x_0 + C.L.\{r_0, Ar_0, A^2 r_0, ..., A^{i-1} r_0\}$$

Por tanto

$$x_i = x_0 + [r_0, Ar_0, A^2 r_0, ..., A^{i-1} r_0] \tag{5.3}$$

El subespacio $K^i(A; r_0)$, de base $[r_0, Ar_0, A^2 r_0, ..., A^{i-1} r_0]$, es llamado subespacio de Krylov de dimensión i, correspondiente a la matriz A y residuo inicial r_0.

5.3. MÉTODO DEL GRADIENTE

En los sistemas de ecuaciones lineales $A\,x = b$, cuya matriz de coeficientes es Simétrica y Definida Positiva (SDP), el método del Gradiente ó del Máximo Descenso es una buena herramienta para su resolución.

Sea $A \in \Re^{n\times n}$ una matriz SDP y $b \in \Re^n$. Se considera como solución óptima $x \in \Re^n$ del sistema $A\,x = b$, aquella que minimiza la función de error:

$$E(x) = \Big(A\,e(x), e(x)\Big) \in \Re \quad /E(x) : \Re^n \to \Re \tag{5.4}$$

siendo $e(x)$, el error de una solución, $= x - \bar{x}, \in \Re^n$, en la que $\bar{x}$ es la solución exacta del sistema y $r(x)$, el residuo, $= b - A\,x = A\,\bar{x} - A\,x = A(\bar{x} - x)$.
Minimizar $E(x) = \Big(A(x-\bar{x}), x-\bar{x}\Big) = \Big(A\,x - A\bar{x}, x-\bar{x}\Big) = \Big(A\,x, x\Big) - 2\Big(A\bar{x}, x\Big) + \Big(A\bar{x}, \bar{x}\Big)$, implica, puesto que $\Big(A\bar{x}, \bar{x}\Big)$ es constante, obtener el mínimo para la función $\Big(A\,x, x\Big) - 2\Big(A\bar{x}, x\Big) = 2J(x)$, siendo

$$J(x) = \frac{1}{2}\Big(A\,x, x\Big) - \Big(b, x\Big) \quad /J(x) : \Re^n \to \Re \tag{5.5}$$

No cabe duda que encontrar una solución x, lo más cercana posible a $\bar{x}$ pasa por minimizar $J(x)$.

La condición necesaria para que una función de varias variables, diferenciable, alcance un valor mínimo en x, es que su gradiente sea cero:
$grad\,J(x) = J^{'}(x) = \frac{1}{2}\Big(A\,x, x\Big)^{'} - \Big(b, x\Big)^{'} = \frac{1}{2}\{[(A\,x)^{'}]^T x + (x^{'})^T A\,x\} - \Big((x^{'})^T b\Big) =$
$= \frac{1}{2}(A^T\,x + A\,x) - b = \frac{1}{2}(A^T + A)x - b = 0$
Si además A es simétrica:

$$J^{'}(x) = A\,x - b \tag{5.6}$$

La naturaleza del punto crítico dependerá del signo de

$$J^{''}(x) = A$$

Si A es definida positiva, no cabe duda que la solución del gradiente nulo se corresponderá con un mínimo del error, y, en este caso, con la solución exacta.

La ventaja de utilizar $J(x)$, es que el problema de optimizar $x \in \Re^n$ es reemplazado por un problema unidimensional que se puede describir como sigue:

I. Evaluar J con una aproximación inicial x_0.

II. Determinar una dirección a partir de x_0 que produzca una descenso en J.

III. Calcular cuanto debemos movernos en esa dirección para obtener una mejor solución x_1.

IV. Volver al paso I, reemplazando x_0 por x_1.

Proceso de iteración, que escribiremos como:

$$x_{i+1} = x_i + \alpha_i \, p_i \tag{5.7}$$

donde p_i es un vector dirección y α_i, un escalar, que determinaremos minimizando $J(x_i)$ a lo largo de p_i:

$$J(x_i + \alpha_i \, p_i) = min_\alpha J(x_i + \alpha \, p_i)$$

con $x_i, p_i \in \Re^n$ fijos:

$$f(\alpha) = J(x_i + \alpha \, p_i) = \frac{1}{2}(x_i + \alpha \, p_i)^T A(x_i + \alpha \, p_i) - b^T(x_i + \alpha \, p_i) =$$

$$= \frac{1}{2}\alpha^2 p_i^T A \, p_i + \alpha p_i^T (A \, x_i - b) + \frac{1}{2} x_i^T (A \, x_i - 2b)$$

El punto crítico de la función parabólica $f(\alpha)$ se puede determinar haciendo $f^{'}(\alpha) = 0$:

$$f^{'}(\alpha) = \alpha p_i^T A \, p_i + p_i^T (A \, x_i - b) = 0$$

y teniendo en cuenta que $f^{''}(\alpha) = p_i^T A \, p_i$, siendo A una matriz definida positiva, se corresponderá con un mínimo, por lo cual

$$\alpha_i = \alpha_{opt}(x_i \, p_i) = \frac{p_i^T (b - A \, x_i)}{p_i^T A \, p_i} = \frac{\Big(r_i, p_i\Big)}{\Big(A \, p_i, p_i\Big)} \tag{5.8}$$

El conocimiento del factor α_i nos permite también establecer, con facilidad, una formulación iterativa para el vector residuo:

Así, partiendo de la expresión 5.7, podemos transformarla con las operaciones siguientes:

$$A \, x_{i+1} = A \, x_i + \alpha_i \, A \, p_i$$

$$b - A \, x_{i+1} = b - A \, x_i - \alpha_i \, A \, p_i$$

$$r_{i+1} = r_i - \alpha_i \, A \, p_i \tag{5.9}$$

Formulación que nos va a permitir demostrar que cualquier vector residuo resulta siempre ortogonal a la dirección de descenso anterior. En efecto, recurriendo al producto escalar de ambos vectores, tendremos:

$$\Big(r_{i+1}, p_i\Big) = \Big(r_i, p_i\Big) - \alpha_i \Big(A\, p_i, p_i\Big) =$$

$$= \Big(r_i, p_i\Big) - \frac{\Big(r_i, p_i\Big)}{\Big(A\, p_i, p_i\Big)} \Big(A\, p_i, p_i\Big) = 0$$

resultado nulo que confirma su ortogonalidad.

Sabemos que el gradiente de una función, en un punto, define la dirección de la derivada direccional máxima y además tiene el sentido en que aumenta la función; por tanto, parece lógico, que para minimizar la función $J(x)$ tomemos como dirección de descenso, p_i, la del vector gradiente, pero en sentido contrario, o sea

$$p_i = -\nabla J(x_i) = -J^{'}(x_i)$$

y que según 5.6, se convertirá en

$$p_i = b - A\, x_i = r_i$$

por lo que la fórmula de iteración se convertirá en:

$$x_{i+1} = x_i + \alpha_i\, r_i \tag{5.10}$$

y la formulación iterativa para el vector residuo se transformará en:

$$r_{i+1} = b - A\, x_{i+1} = b - A(x_i + \alpha_i\, r_i) = r_i - \alpha_i\, A\, r_i \tag{5.11}$$

dando lugar al algoritmo siguiente [94]:

ALGORITMO DEL GRADIENTE

Valor inicial: x_0

$r_0 = b - A\, x_0$

para $i = 0, 1, 2, ...$ hasta la convergencia, hacer:

$\alpha_i = \dfrac{\Big(r_i, r_i\Big)}{\Big(A\, r_i, r_i\Big)}$

$x_{i+1} = x_i + \alpha_i\, r_i$

$r_{i+1} = r_i - \alpha_i\, A\, r_i$

5.4. MÉTODO DEL GRADIENTE CONJUGADO

Hemos visto en el método anterior, que la condición

$$J(x) \leq J(x + \alpha\, p), \quad \forall \alpha \in \Re$$

supone conseguir el valor óptimo de x en la dirección p, lo que implica que cada nuevo vector residuo es ortogonal a la dirección de descenso anterior y, a su vez, dicha dirección de descenso viene dada por un vector opuesto al gradiente de $J(x)$, que coincide con el vector residuo, con lo que resultará:

$$r_{i+1} \perp p_i \quad y \quad p_i \equiv r_i \quad \Rightarrow \quad r_{i+1} \perp r_i$$

La dificultad del Método del Gradiente reside en que esta relación de ortogonalidad no es transitiva, es decir si bien $r_{i+1} \perp r_i$ y $r_{i+2} \perp r_{i+1}$, esto no supone que $r_{i+2} \perp r_i$ y, por consiguiente, con las sucesivas iteraciones se vaya perdiendo la condición de *optimización de* x.

Para mantener esta condición, se debe trabajar con unas direcciones de descenso, que a diferencia del Método del Gradiente, conserven los requisitos de ortogonalidad con respecto a todas las direcciones anteriores y que por tanto todas las direcciones sean conjugadas.

Según hemos visto anteriormente, si $x^{'}$ es un valor óptimo de x en la dirección p resulta que:

$$x^{'} = x + \alpha\, p \quad \Rightarrow \quad r^{'} \perp p$$

Hallemos ahora un nuevo valor $x^{''}$ en una nueva dirección q:

$$x^{''} = x^{'} + \alpha\, q$$

el nuevo residuo será

$$r^{''} = b - Ax^{''} = b - A(x^{'} + \alpha\, q) = b - Ax^{'} - \alpha\, A\, q = r^{'} - \alpha\, A\, q$$

para que el nuevo valor $r^{''}$ siga siendo óptimo respecto a la dirección de p deberá cumplirse que $r^{''} \perp p$, es decir:

$$\left(r^{'} - \alpha\, A\, q, p\right) = 0$$

o sea

$$\Big(r^{'}, p\Big) - \alpha\Big(A\,q, p\Big) = 0$$

como

$$r^{'} \perp p \quad \Rightarrow \quad \Big(r^{'}, p\Big) = 0 \quad \Rightarrow \quad \Big(A\,q, p\Big) = 0$$

y, por tanto, se dice que los vectores p y q, que cumplen dicha condición, son A conjugados; como además la matriz A es simétrica y definida positiva, podemos afirmar que p y q son A-ortogonales.

En lo sucesivo usaremos direcciones de descenso $p_0, p_1, p_2, ...p_i$ que sean A-ortogonales dos a dos, es decir, tal que:

$$p_i^T \cdot A\,p_j = 0, \quad \forall i \neq j$$

.

Además, teniendo en cuenta que la matriz A es simétrica y definida positiva, podemos comprobar que el conjunto de los vectores $p_0, p_1, p_2, ...p_i$, A-ortogonales, resultan ser linealmente independientes.

En efecto, si existe una colección de coeficientes escalares α_k, tal que:

$$\alpha_0\,p_0 + \alpha_1\,p_1 + \alpha_2\,p_2 + ... + \alpha_i\,p_i = 0$$

se cumplirá también que:

$$\alpha_0\,A\,p_0 + \alpha_1\,A\,p_1 + \alpha_2\,A\,p_2 + ... + \alpha_i\,A\,p_i = 0$$

que multiplicando escalarmente por p_0, se convierte en

$$\alpha_0\,p_0^T\,A\,p_0 + \alpha_1\,p_0^T\,A\,p_1 + \alpha_2\,p_0^T\,A\,p_2 + ... + \alpha_i\,p_0^T\,A\,p_i = 0$$

como por hipótesis todos los $p_i^T \cdot A\,p_j$, $\forall i \neq j$, son nulos, resultará que $\alpha_0\,p_0^T\,A\,p_0 = 0$ y al ser A definida positiva implicará que:

$$\alpha_0 = 0$$

De forma similar se puede demostrar para cualquier

$$\alpha_k \,/\, k = 1, 2, ...i \quad \Rightarrow \quad \forall k \in \{0, 1, 2, ...i\}, \quad \alpha_k = 0$$

y por tanto todos los vectores A-ortogonales serán linealmente independientes.

El Método del Gradiente Conjugado [50, 52, 1, 84] viene a ser una variante del Método del Gradiente, en el que las sucesivas direcciones de descenso se generan como versiones conjugadas de los gradientes que se van obteniendo según progresa el método. Cada nueva dirección,p_{i+1}, se obtiene en el plano formado por las direcciones ortogonales r_{i+1} y p_i siguiendo la expresión

$$p_{i+1} = r_{i+1} + \beta_i \, p_i \tag{5.12}$$

determinándose el escalar β_i de tal forma que p_{i+1} y p_i sean A-ortogonales, es decir:

$$p_{i+1}^T \, A \, p_i = 0$$

por tanto

$$(r_{i+1} + \beta_i \, p_i)^T \, A \, p_i = (r_{i+1}^T + \beta_i \, p_i^T) \, A \, p_i = 0$$

resultando

$$\beta_i = -\frac{r_{i+1}^T \, A \, p_i}{p_i^T \, A \, p_i} = -\frac{\Big(A \, p_i, r_{i+1}\Big)}{\Big(A \, p_i, p_i\Big)} \tag{5.13}$$

lo que nos va a garantizar que los residuos sucesivos sean ortogonales, o sea que

$$\Big(r_{i+1}, r_i\Big) = 0$$

.

En efecto:

$$r_{i+1} = r_i - \alpha_i \, A \, p_i$$

por tanto

$$\Big(r_{i+1}, r_i\Big) = \Big(r_i - \alpha_i \, A \, p_i, r_i\Big) = \Big(r_i, r_i\Big) - \alpha_i \Big(A \, p_i, r_i\Big) =$$

$$= \Big(r_i, r_i\Big) - \alpha_i \Big(A \, p_i, p_i - \beta_{i-1} \, p_{i-1}\Big) = \Big(r_i, r_i\Big) - \alpha_i \Big(A \, p_i, p_i\Big) + \alpha_i \beta_{i-1} \Big(A \, p_i, p_{i-1}\Big)$$

pero teniendo en cuenta que p_i y p_{i-1} son A-ortogonales, y el valor de α_i, resultará que

$$\begin{aligned}\Big(r_{i+1}, r_i\Big) &= \Big(r_i, r_i\Big) - \Big(r_i, p_i\Big) \\ &= \Big(r_i, r_i\Big) - \Big(r_i, r_i + \beta_{i-1}\, p_{i-1}\Big) \\ &= \Big(r_i, r_i\Big) - \Big(r_i, r_i\Big) - \beta_{i-1}\Big(r_i, p_{i-1}\Big) \\ &= 0\end{aligned}$$

Como consecuencia de todo lo anterior, se puede afirmar que este Método del Gradiente Conjugado tiene dos propiedades esenciales:

A) Cada dirección de descenso es A-ortogonal a todas las direcciones anteriores, con lo cual no se pierde la condición de *óptimo* de cada nuevo vector, x, hallado: Partiendo de cada producto $\Big(A\, p_{i+1}, p_i\Big) = 0$, se puede comprobar fácilmente que

$$\Big(A\, p_{i+1}, p_k\Big) = 0, \quad para \quad 0 \leq k \leq i.$$

B) Prescindiendo de los errores de redondeo, el Método converge a lo sumo en n iteraciones [112], siendo n el orden de la matriz cuadrada A:
En efecto, tomando como base que $\Big(r_{i+1}, p_i\Big) = 0$, se puede demostrar con facilidad que

$$\Big(r_{i+1}, p_k\Big) = 0, \quad para \quad 0 \leq k \leq i.$$

o sea, que cada vector residuo es ortogonal a todas las direcciones de descenso anteriores.

Ahora bien, para $k < n-1$, puede ocurrir que $r_k = 0$, con lo cual $b - A\, x_k = 0$, habríamos encontrado ya la solución correcta y el Método convergería en k iteraciones.

Pero mientras $r_k \neq 0$, habrá un residuo no nulo, cada vez más pequeño. Así llegaríamos hasta r_n, que sería ortogonal a $p_0, p_1, p_2, ...p_{n-1}$, que son n vectores linealmente independientes y A-ortogonales , con lo cual r_n tendría que ser nulo y por tanto $b - A\, x_n = 0$, con lo que habríamos llegado a la solución exacta, y por tanto a la última y n-sima iteración.

Precisamente los n vectores, $p_0, p_1, p_2, ...p_{n-1}$, linealmente independientes, definen un subespacio de n dimensiones, A-ortogonales, que resulta ser un Subespacio de Krylov, correspondiente a la matriz $A = \Re^{n\times n}$, de vector inicial p_0, que se hace corresponder con el residuo inicial r_0.

Hay que señalar, sin embargo, que en la práctica, la aparición de errores de redondeo, hace que las direcciones de descenso no sean exactamente A-ortogonales y, por tanto, que el Gradiente Conjugado se comporte como un método iterativo cualesquiera [4].

El esquema principal del algoritmo del Gradiente Conjugado consiste en obtener

$$x_{i+1} = x_i + \alpha_i \, p_i$$

sabiendo, por la expresión (5.8), que

$$\alpha_i = \frac{\Big(r_i, p_i\Big)}{\Big(A\, p_i, p_i\Big)}$$

es el escalar óptimo, que minimiza la función error,
y tomando cada dirección de descenso, según (5.12), como

$$p_{i+1} = r_{i+1} + \beta_i \, p_i$$

siendo

$$\beta_i = -\frac{\Big(A\, p_i, r_{i+1}\Big)}{\Big(A\, p_i, p_i\Big)}$$

que nos permite mantener la ortogonalidad entre todas las direcciones óptimas, p_i.

Teniendo, además, en cuenta la expresión (5.9), de iteración de los residuos:

$$r_{i+1} = r_i - \alpha_i \, A\, p_i$$

Las condiciones de ortogonalidad del proceso, entre vectores residuo y direcciones de descenso, nos van a permitir expresar α_i y β_i de forma más sencilla, de tal manera que hagan este algoritmo más eficiente:

Así

$$\alpha_i = \frac{r_i^T \, p_i}{p_i^T \, A\, p_i} = \frac{r_i^T (r_i + \beta_{i-1} \, p_{i-1})}{p_i^T \, A\, p_i} = \frac{r_i^T \, r_i + \beta_{i-1} \, r_i^T \, p_{i-1}}{p_i^T \, A\, p_i} = \frac{r_i^T \, r_i}{p_i^T \, A\, p_i} = \frac{\Big(r_i, r_i\Big)}{\Big(A\, p_i, p_i\Big)}$$

Y de forma similar

$$\beta_i = -\frac{r_{i+1}^T\, A\, p_i}{p_i^T\, A\, p_i} = -\frac{r_{i+1}^T\, \alpha_i\, A\, p_i}{p_i^T\, \alpha_i\, A\, p_i} = -\frac{r_{i+1}^T (r_i - r_{i+1})}{p_i^T (r_i - r_{i+1})} = -\frac{r_{i+1}^T\, r_i - r_{i+1}^T\, r_{i+1}}{p_i^T r_i - p_i^T\, r_{i+1}} = \frac{r_{i+1}^T\, r_{i+1}}{p_i^T\, r_i} =$$

$$= \frac{r_{i+1}^T\, r_{i+1}}{r_i^T\, p_i} = \frac{r_{i+1}^T\, r_{i+1}}{r_i^T (r_i + \beta_{i-1}\, p_{i-1})} = \frac{r_{i+1}^T\, r_{i+1}}{r_i^T r_i + \beta_{i-1}\, r_i^T\, p_{i-1}} = \frac{r_{i+1}^T\, r_{i+1}}{r_i^T r_i} = \frac{\left(r_{i+1}, r_{i+1}\right)}{\left(r_i, r_i\right)}$$

con lo que resulta el siguiente [94]

ALGORITMO DEL GRADIENTE CONJUGADO

Valor inicial: x_0

$r_0 = b - A\, x_0 = p_0$

Para $i = 0, 1, 2, ...$ hacer:

$\alpha_i = \dfrac{\left(r_i, r_i\right)}{\left(A\, p_i, p_i\right)}$

$x_{i+1} = x_i + \alpha_i\, p_i$

$r_{i+1} = r_i - \alpha_i\, A\, p_i$

$\beta_i = \dfrac{\left(r_{i+1}, r_{i+1}\right)}{\left(r_i, r_i\right)}$

$p_{i+1} = r_{i+1} + \beta_i\, p_i$

Si $\left\| \frac{r_{i+1}}{r_0} \right\| < \epsilon \rightarrow Fin$

5.5. OTROS MÉTODOS DE KRYLOV

En general, la totalidad de los métodos de Krylov, utilizados para la resolución de grandes sistemas lineales, se obtienen para adaptarse, en principio, a dos importantes requerimientos básicos:

A) Minimizar una cierta norma del vector residuo, sobre un subespacio de Krylov generado por la matriz del sistema, que se traduce en una convergencia suave y sin grandes fluctuaciones.

B) Ofrecer un bajo coste computacional por iteración y no exigir alta disponibilidad de almacenaje.

Sin embargo, esto no siempre es posible. Sólo en sistemas de ecuaciones lineales cuya matriz de coeficientes es simétrica y definida positiva, el algoritmo del

Gradiente Conjugado (CG), propuesto por primera vez por Hestenes y Stiefel, en 1952, [50], y ampliamente desarrollado en la práctica a partir de 1970, con la introducción y desarrollo de los ordenadores en el campo de las Matemáticas, llega a alcanzar teóricamente, salvo errores de redondeo, la solución exacta en un número de iteraciones igual a la dimensión del sistema y cumple además los requisitos esenciales citados anteriormente, de minimización del residuo y óptimo coste. Por ello le hemos dedicado un apartado especial a su exposición, así como a la del Método del Gradiente o del Máximo Descenso, puesto que en su exposición aparecen ideas básicas que facilitan la comprensión del CG.

Cuando la matriz del sistema no cumple las condiciones de simetría y positividad, el método del Gradiente Conjugado no es aplicable y en general, no existen métodos que cumplan los dos requisitos anteriores simultáneamente sin añadir inconvenientes y/o desventajas. Por ello, los métodos utilizados para estos sistemas, se obtienen para adaptarse a uno de ellos, bien a la minimización, o bien métodos que ofrezcan un bajo coste computacional y de almacenamiento de datos.

Los podemos agrupar en tres grandes familias: Métodos de Ortogonalización, Métodos de Biortogonalización y Métodos basados en la Ecuación Normal.

5.5.1. MÉTODOS DE ORTOGONALIZACIÓN

Están basados en un proceso de proyección ortogonal sobre un subespacio de Krylov de dimensión menor que la del sistema. En estos métodos,las fórmulas recurrentes, relativamente largas, hacen que el coste computacional y la memoria ocupada aumenten considerablemente con el número de iteraciones. Se fundamentan, todos ellos, en el algoritmo de ortogonalización de Arnoldi, introducido en 1951 [3].

Dado el sistema lineal de ecuaciones $A\,x = b$, se comienza eligiendo una solución inicial aproximada, x_0, a partir de la cual, utilizando el algoritmo de Arnoldi, se construye una secuencia de vectores $x_k \in x_0 + K_k$, siendo $K_k \equiv C.L.\{r_0, A\,r_0, A^2\,r_0, ..., A^{k-1}\,r_0\}$ y $r_0 = b - A\,x_0$, imponiendo la condición de ortogonalidad:

$$b - A\,x_k \perp K_k$$

Basados en este procedimiento se pueden citar:

El Método FOM (*Full Orthogonalization Method*) [94, 5] que, para resolver el sistema, aplica en cada paso el algoritmo de Arnoldi, modificado Gram-Schmidt. El coste computacional de este método es del orden de $k^2\,n$, debido al proceso de ortogonalización de Gram-Schmidt, y el coste de almacenamiento es de $k\,n$, lo que resulta excesivo para grandes valores de n, puesto que la dimensión de K_k también resulta grande.

El Método GMRES (Método del Mínimo Residuo Generalizado)[94, 95, 20]. Este algoritmo aproxima la solución de forma similar al algoritmo FOM. Cumple la propiedad de minimizar el vector residuo, obteniéndose un método robusto, de convergencia rápida y con curvas de convergencia *sin picos*, pero a costa de exigir un alto coste computacional y una elevada disponibilidad de almacenaje que va incrementándose con el orden de las iteraciones. Posteriormente se han desarrollado nuevas versiones que parecen ser más eficaces como el FGMRES [93], el GMRESR [114] ó el VGMRES [35, 34].

5.5.2. MÉTODOS DE BIORTOGONALIZACIÓN

Estos métodos son también de proyección, pero intrínsicamente no ortogonales. Tienen la ventaja de que las fórmulas de recurrencia son reducidas y el almacenamiento no aumenta con el número de iteraciones, pero por contra, presentan un comportamiento irregular en la convergencia, con oscilaciones y abundantes *picos*, que con ciertos criterios de parada pueden conducir a soluciones erróneas [107].

Se fundamentan en el algoritmo de biortogonalización de Lanczos, 1952, [59, 111] y entre ellos podemos citar:

El Método Bi-CG (Método del Doble Gradiente Conjugado) [27, 52]. Este algoritmo, al igual que el del Gradiente Conjugado (CG), se deriva del algoritmo de biortogonalización de Lanczos.

Implicitamente, el algoritmo resuelve, no solo el sistema original $A\,x = b$, sino también el sistema dual ó sistema auxiliar $A^T\,x^* = b^*$, que se resuelve parale-

lamente al primero, determinando los sucesivos vectores residuos y direcciones conjugadas que figuran en el algoritmo, pero si analizar su convergencia, puesto que no influye en la solución del sistema. Este algoritmo calcula dobles vectores residuos y dobles direcciones conjugadas, además de efectuar dobles productos matriz por vector, debidos a la matriz del sistema inicial $A\,x = b$ y a la matriz del sistema auxiliar $A^T\,x^* = b^*$. El cálculo por iteración es doble que en el CG, pero presenta la ventaja, frente a otras generalizaciones del mismo, como el CGN, ó el método del Residuo Mínimo [16], de conservar el condicionamiento de la matriz del sistema original, por lo que su aplicación no empeora la convergencia de sistemas inicialmente mal condicionados.

El Método CGS(*Conjugate Gradient Squared*).Desarrollado por Sonneveld en 1989 [104], es una modificación del Bi-CG.

Sonneveld observa que en caso de convergencia del Doble Gradiente, los vectores r_i y r_j tienden a cero, pero la convergencia de los *residuos auxiliares* no es explotada y solo se calculan para la valoración de los parámetros que aparecen en el algoritmo. Propone, entonces, la siguiente modificación, donde todos los esfuerzos se concentran en la convergencia de los vectores del sistema original r_i,:

Usando las expresiones polinómicas para los vectores residuos, $r_i = P_i(A)\,r_0$ y $r_j^* = P_j(A^T)r_0^*$, queda para la relación de biortogonalidad

$$\left(r_i, r_j^*\right) = \left(P_i(A)\,r_0, P_j(A^T)r_0^*\right) = \left(P_i(A)\,P_j(A)r_0, r_0^*\right)$$

lo que sugiere trabajar con aproximaciones de $\hat{x}_i$, para las cuales

$$\hat{r_i} = b - A\,\hat{x}_i = P_i^2(A)\,r_0$$

Se obtiene así el algoritmo en base a estos nuevos vectores residuos, efectuando las transformaciones pertinentes en el algoritmo del Bi-CG que permitan generar $P_i^2(A)\,r_0$. Para ello se usan, además de las expresiones polinómicas de los vectores residuos, las correspondientes a las direcciones conjugadas de los sistemas original y auxiliar:

$$p_i = T_i(A)\,r_0 \quad y \quad p_i^* = T_i(A^T)r_0$$

Con estas relaciones, resulta el algoritmo CGS, en el que no figuran los vectores residuos ni las direcciones conjugadas del sistema auxiliar, eliminando así los

productos A^T por vector.

Cuando el Bi-CG converge, como la expresión polinómica del vector residuo es $r_i = P_i(A)\, r_0$, el operador $P_i(A)$ reduce el vector inicial a r_0, a otro menor r_i, y, dado que en el CGS se utilizan vectores $\hat{r_i} = P_i^2(A)\, r_0$, cabe esperar que al aplicarlo dos veces resultará mejorada la convergencia. Este efecto se produce con mucha frecuencia y, en general, el CGS converge más rápidamente que el Bi-CG, pero al igual que en este, las curvas de convergencia presentan bruscas oscilaciones que pueden conducir a la cancelación del algoritmo con soluciones erróneas carentes de significado. Por ello, una vez alcanzado el criterio de parada adoptado, se debe calcular el vector residuo real, $r_i = b - A\, x_i$, para proceder a una realimentación del algoritmo en caso necesario.

El Método Bi-CGSTAB (*Conjugate Gradient Stabilized*). De la misma forma que en el CGS los vectores residuos verifican $\hat{r_i} = P_i^2(A)\, r_0$, pueden construirse otros métodos iterativos en los cuales los valores aproximados de la solución x_i sean obtenidos de modo que los residuos vengan dados por $r_i = Q_i(A)\, P_i(A) r_0$, con otro polinomio *reductor*, $Q_i(A)$. Las relaciones de recurrencia del algoritmo donde intervenga este polinomio no han de ser excesivamente complicadas y los parámetros que figuren en su definición deben ser fácilmente optimizables.

Van der Vost, 1992, sugiere para $Q_i(A)$ la expresión:

$$Q_i(A) = (1 - \omega_1\, A)(1 - \omega_2\, A).....(1 - \omega_i\, A)$$

determinando ω_i, por la condición de mínimo para la norma de r_i, en la iteración i-ésima.

Para obtener el algoritmo, se toma como referencia, al igual que en el CGS, el algoritmo del Bi-CG, efectuando las transformaciones necesarias para que la relación de recurrencia de la solución sea función del nuevo vector residuo. Una vez expresados todos los vectores en función del nuevo residuo y determinadas sus relaciones de recurrencia, así como los escalares que intervienen en las mismas, se podrá escribir el algoritmo Bi-CGSTAB donde figuran dos productos matriz por vector y cuatro productos internos, mientras que en el CGS se necesitan los mismos productos matriz-vector, pero solo dos productos internos.

En la mayoría de los casos la convergencia del Bi-CGSTAB es más rápida y

uniforme que en el CGS, e incluso con menor carga computacional, para alcanzar una determinada tolerancia, pues la reducción del número de iteraciones compensa el pequeño incremento computacional por iteración que suponen estos dos productos adicionales. Sin embargo, la presencia en ocasiones, de forma similar a como ocurre en el Bi-CG y CGS, de bruscas variaciones en la norma del vector residuo, hacen aconsejable proceder, al final del algoritmo, al cálculo del vector residuo real correspondiente a la solución aproximada obtenida, para efectuar la corrección oportuna en el caso que fuese necesaria.

El Método QMR (Método del Cuasi-mínimo Residuo). Este método, propuesto por Freund y Nachtigal [32, 33], conserva la propiedad de los métodos tipo gradiente, de utilizar relaciones de recurrencia que impliquen bajo coste computacional, con la particularidad de no usar una condición estricta de minimización para generar los vectores residuos. Plantea, para ello, una técnica para cuasi-minimizar estos vectores, condición que da nombre al método, intentando suavizar las fluctuaciones que tienen lugar en la convergencia, debidas a este hecho.

Aplicando esta técnica de *suavizado de residuos* [118] al CGS y al Bi-CGSTAB, se obtienen los algoritmos llamados TFQMR y QMRCGSTAB.

El Método TFQMR (*Transpose-Free*). Derivado del algoritmo CGS y desarrollado por Freund en 1993 [31], tiene la ventaja de no requerir productos matriz por vector con A^T y puede implementarse fácilmente, simplemente cambiando unas pocas líneas en el algoritmo CGS, nombrado anteriormente.

El Método QMRCGSTAB. Desarrollado por Chan y otros, en 1994 [15], está basado en la aplicación del principio de minimización, usado en el método QMR, al algoritmo Bi-CGSTAB.

Otra variante de los métodos de cuasi-mínimo residuo son los llamados Modificados: **MQMR, MTFQMR y MQMRCGSTAB**, presentados por M.D. García en su Tesis Doctoral [37], proponiendo, de forma análoga a como se hace en la versión del GMRES expuesta por M. Galán [35, 34], la resolución directa del

problema de mínimos cuadrados que plantean estos métodos; es decir, haciendo una factorización LU, en lugar de la factorización QR tradicional [36].

5.5.3. MÉTODOS BASADOS EN LA ECUACIÓN NORMAL

La resolución del sistema de ecuaciones $A\,x = b$, donde la matriz A es no simétrica, es equivalente a resolver el sistema $A^T A\,x = A^T b$ de matriz $A^T A$, simétrica definida positiva, al que se puede aplicar el algoritmo del Gradiente Conjugado. La ecuación

$$A^T A\,x = A^T b$$

recibe el nombre de Ecuación Normal.

Al igual que el CG, los métodos desarrollados a partir de la Ecuación Normal cumplen las dos condiciones fundamentales de minimización de la norma residual y optimización del coste computacional, si embargo presentan el inconveniente de que el condicionamiento del nuevo sistema es el cuadrado del sistema inicial: $K(A^T A) = K(A)^2$, lo cual, para sistemas mal condicionados, puede resultar desastroso y además en cada iteración aparecen dos productos matriz por vector correspondientes a las matrices A y A^T aumentando el coste computacional. Resultan así métodos como:

El Método CGN(Método del Gradiente Conjugado para la Ecuación Normal). Este método construye una sucesión de vectores:

$$x_k = x_0 + \left[A^T r_0,\ (A^T A)A^T r_0,\ (A^T A)^2 A^T r_0, ..., (A^T A)^{k-1} A^T r_0\right]$$

con residuo mínimo en cada paso, sin efectuar el cálculo explícito del producto $A^T A$.

El Método LSQR(*Least-Square QR*). Propuesto por Paige y Saunders, en 1982 [78]. Este método intenta corregir el posible empeoramiento del número de condición de los sistemas al aplicar el método de la Ecuación Normal.

La idea básica del LSQR es hallar la solución del sistema simétrico:

$$\begin{pmatrix} I & A \\ A^T & -\lambda^2 I \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

minimizando,

$$\left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

donde λ es un número real arbitrario.

Capítulo 6

PRECONDICIONAMIENTO

6.1. CONSIDERACIONES PREVIAS

Aunque los métodos iterativos basados en los subespacios de Krylov están, teóricamente, bien fundamentados, casi todos ellos adolecen de lentitud en la convergencia, principalmente aquellos que se usan para resolver problemas que surgen de temas como dinámica de fluidos o simulaciones de mecanismos electrónicos. Precondicionar un sistema es un paso clave para el éxito de los métodos de Krylov que se usan en estas aplicaciones [74].

Es ampliamente reconocido que la falta de robustez es una de las debilidades de los métodos iterativos, este inconveniente ha impedido la amplia aceptación de estos métodos en aplicaciones industriales, a pesar de su intrínseco atractivo para resolver grandes sistemas de ecuaciones lineales. Tanto la eficacia, como la robustez de las técnicas iterativas pueden mejorarse con el uso de los precondicionadores. Precondicionar es simplemente un medio de transformar un sistema lineal original en otro que tenga la misma solución, pero que sea más fácil de resolver por métodos iterativos. En general la fiabilidad de estos métodos dependen mucho más de la calidad del precondicionador, que del método de Krylov elegido.

Encontrar un buen precondicionador para resolver un sistema lineal *sparse* está considerado, con frecuencia, como una combinación de arte y ciencia [94]. algunos métodos de precondicionamiento funcionan sorpresivamente bien, a pesar de sus escasas expectativas teóricas.

Nótese que en principio no hay virtualmente límites para elegir opciones que

permitan obtener buenos precondicionadores. Por ejemplo, los precondicionadores pueden derivarse del conocimiento de los problemas físicos de los que surge el sistema lineal o pueden construirse a partir de la matriz de coeficientes del sistema original. En líneas generales un precondicionador es cualquier forma explícita o implícita de modificación de un sistema lineal original que lo haga más fácil de resolver por un método iterativo dado. El sistema resultante debe poderse resolver por un método basado en los subespacios de Krylov y deben requerirse menos pasos para su convergencia, que si se aplicara el mismo método al sistema original (Aunque esto no pueda garantizarse teóricamente, sino confirmarse con la experiencia).

En definitiva, para resolver ciertos problemas es indispensable la implementación de un precondicionador adecuado para asegurar la convergencia del método de Krylov elegido.

En esta sección introduciremos ideas generales sobre precondicionamiento de sistemas y relacionaremos algunos de los precondicionadores más utilizados para resolver sistemas de matriz simétrica definida positiva, que son los que surgen en los modelos de campos de viento.

6.2. CONDICIONAMIENTO DE UN SISTEMA

Decimos que un sistema de ecuaciones, $A\,x = b$, está bien o mal condicionado, cuando pequeñas variaciones en sus coeficientes, o en sus términos independientes, producen una pequeña ó gran variación en la solución del mismo. Con objeto de dar una medida del buen o mal condicionamiento de un sistema se introduce la noción de número de condición (relativo a una norma matricial dada) [15, 32].

Limitándonos al caso más sencillo, si se produce una perturbación, Δb, en el vector columna de los términos independientes, vamos a analizar la perturbación, Δx, que se producirá en la solución exacta, x, del sistema. Empleando una norma matricial arbitraria y una norma vectorial cualquiera, compatible con ella, se tendrá:

$$\left\{ \begin{array}{c} A\,x = b \\ A(x + \Delta x) = b + \Delta b \end{array} \right\} \quad \Rightarrow \quad A\,\Delta x = \Delta b$$

$$\left\{ \begin{array}{c} \Delta x = A^{-1}\,\Delta b \\ b = A\,x \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} \parallel \Delta x \parallel \leq \parallel A^{-1} \parallel \parallel \Delta b \parallel \\ \parallel b \parallel \leq \parallel A \parallel \parallel x \parallel \end{array} \right\} \Rightarrow$$

$$\parallel \Delta x \parallel \parallel b \parallel \leq \parallel A \parallel \parallel A^{-1} \parallel \parallel \Delta b \parallel \parallel x \parallel \Rightarrow$$

$$\Rightarrow \frac{\parallel \Delta x \parallel}{\parallel x \parallel} \leq \parallel A \parallel \parallel A^{-1} \parallel \frac{\parallel \Delta b \parallel}{\parallel b \parallel}$$

Al factor $\parallel A \parallel \parallel A^{-1} \parallel = K(A)$, se le denomina número de condicionamiento del sistema, relativo a la norma matricial $\parallel \cdot \parallel$; obteniéndose así la siguiente relación del error relativo de la solución, en función del error relativo en los términos independientes:

$$\frac{\parallel \Delta x \parallel}{\parallel x \parallel} \leq K(A) \frac{\parallel \Delta b \parallel}{\parallel b \parallel} \tag{6.1}$$

Evidentemente, cuanto menor sea el valor numérico, $K(A)$, menor será la variación de la solución ante las posibles fluctuaciones en los valores de los elementos de b. Dicho número será siempre mayor o igual que la unidad:

$$1 \leq \parallel I \parallel = \parallel A \cdot A^{-1} \parallel \leq \parallel A \parallel \parallel A^{-1} \parallel = K(A) \tag{6.2}$$

Cuanto más próximo a la unidad sea $K(A)$, mejor condicionado estará el sistema:

$$\lim_{K(A) \to 1} \frac{\parallel \Delta x \parallel / \parallel x \parallel}{\parallel \Delta b \parallel / \parallel b \parallel} = 1$$

Conviene resaltar que el número de condición, $K(A)$, del sistema $A\,x = b$, es una cantidad intrínseca a su matriz de coeficientes, A. En otras palabras, el condicionamiento del sistema es independiente del vector b, de términos independientes.

De forma similar, si denominamos ΔA a las variaciones introducidas en los coeficientes de la matriz, no cabe duda que el sistema se transformará en

$$(A + \Delta A)(x + \Delta x) = b$$

A partir del cual, como en el caso anterior, se puede comprobar fácilmente que:

$$\frac{\parallel \Delta x \parallel}{\parallel x + \Delta x \parallel} \leq \parallel A \parallel \parallel A^{-1} \parallel \frac{\parallel \Delta A \parallel}{\parallel A \parallel}$$

lo que nos indica que el error relativo de la nueva solución es también función del error relativo producido por las variaciones de la matriz de coeficientes y, ambos, están relacionados por el mismo número $K(A)$, ya definido anteriormente.

Por tanto, cuanto más cercano a 1 esté el número de condicionamiento, $K(A)$, tanto menor será la variación de la solución del sistema, ante las posibles fluctuaciones en los valores, tanto de los elementos de A, como de b y por tanto se dirá que el sistema está mejor condicionado.

Usando la norma espectral $\| \cdot \|_2$, $K(A) = \frac{\mu_M}{\mu_m} \geq 1$, donde μ_M y μ_m son, respectivamente, los valores singulares máximo y mínimo de la matriz del sistema, valores que se pueden calcular por $\mu_i = \sqrt{\lambda_i(AA^T)}$, siendo $\lambda_i(AA^T)$ los correspondientes valores propios de AA^T.

Cuando A es simétrica,$\mu_i(A) \equiv \lambda_i(A)$, y el número de condicionamiento sería

$$K(A) = \frac{\lambda_M}{\lambda_m} \geq 1$$

Con esta definición, dado que la *bondad* del condicionamiento de una matriz viene dada, en principio, por la proximidad de $K(A)$ a la unidad, en el caso que los valores singulares extremos coincidieran, el número de condicionamiento adquiriría este valor óptimo.

6.3. TÉCNICAS DE PRECONDICIONAMIENTO

En numerosas ocasiones, la matriz y el vector segundo miembro del sistema se calculan de forma aproximada, pudiendo existir ciertas diferencias con los valores numéricos que reflejen exactamente el problema. En estos casos, un mal condicionamiento del sistema afectaría negativamente a la convergencia.

Se hace necesario así, mejorar este condicionamiento utilizando adecuadas técnicas de precondicionamiento. Técnicas que, en general, consisten en transformar el sistema en otro de idéntica solución, pero con menor $K(A)$.

Para ello multiplicaremos la expresión $A\,x = b$ por una matriz M, llamada matriz de precondicionamiento:

$$M\,A\,x = M\,b$$

tal que

$$K(M\,A) < K(A).$$

El menor valor de $K(M\,A)$ correspondería a $M = A^{-1}$, puesto que quedaría $K(A^{-1}\,A) = 1$, que es, obviamente, el caso ideal y el sistema convergería en una sola iteración, pero el coste computacional del cálculo de A^{-1} equivaldría a resolver el sistema por un método directo.

Esta circunstancia sugiere para M una matriz lo más próxima posible a A^{-1}, sin que su determinación suponga un elevado coste. Generalmente, se opta por considerar como matriz de precondicionamiento a M^{-1}, y obtener M como aproximación de A, escribiendo,

$$M^{-1}A\,x = M^{-1}b$$

Los ordenadores, con múltiples procesadores en paralelo, ofrecen una gran versatilidad [92]. Grote y Simon [47] proponen obtener M como aproximación de A^{-1}, minimizando una norma que reduce el cálculo a n *pequeños* problemas independientes de mínimos cuadrados. Asimismo, en [102] y [77] se plantea efectuar esta aproximación por una expresión polinómica $P(A)$. El campo de posibles precondicionadores aplicables en ordenadores que no tengan estas características es también muy amplio.

Por otro lado, en los algoritmos precondicionados de los distintos métodos figurarán productos de matriz inversa por vector, que no deben exigir excesivo trabajo adicional, por ello, la matriz M debe ser fácilmente invertible. Por ejemplo una matriz diagonal, o una matriz factorizada adecuadamente para efectuar esos productos por procesos de remonte, sin necesidad de calcular M^{-1}.

Dependiendo de la forma de plantear el producto de la inversa de la matriz de precondicionamiento por la matriz del sistema, y aprovechando la descomposición en factores de aquella, se distinguen los siguientes casos:

-a) Precondicionamiento por la izquierda:

$$M^{-1}A\,x = M^{-1}b \left\{ \begin{array}{c} M^{-1}A = \tilde{A} \\ M^{-1}b = \tilde{b} \end{array} \right\} \tilde{A}\,x = \tilde{b}$$

-b) Precondicionamiento por la derecha:

$$A\,M^{-1}M\,x = b \left\{ \begin{array}{c} A\,M^{-1} = \tilde{A} \\ M\,x = \tilde{x} \end{array} \right\} \tilde{A}\,\tilde{x} = b$$

-c) Precondicionamiento por ambos lados:

Expresando M factorizada como $M = M_1 M_2$,

$$M_1^{-1} A M_2^{-1} M_2 x = M_1^{-1} b \quad \left\{ \begin{array}{l} M_1^{-1} A M_2^{-1} = \tilde{A} \\ M_2 x = \tilde{x} \\ M_1^{-1} b = \tilde{b} \end{array} \right\} \quad \tilde{A}\,\tilde{x} = \tilde{b}$$

Las características de cada problema y, en definitiva, de la matriz A, del precondicionador utilizado M e, incluso, de la tolerancia exigida para el criterio de parada adoptado, hacen más eficiente una forma u otra de precondicionamiento, sin que pueda establecerse *a priori* bases de elección que nos inclinen por una de ellas.

El precondicionamiento de un sistema tiene por finalidad mejorar la convergencia del método aplicado respecto a la convergencia del sistema sin precondicionar.

En la resolución de sistemas simétricos, la razón de convergencia del Gradiente Conjugado

$$\| x - x_i \|_A \leq 2 \left(\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} \right)^i \| x - x_0 \|_A$$

depende del número de condicionamiento $K(A)$, función de los autovalores mayor y menor de la matriz del sistema. Sin embargo, en la práctica, después de cierto número de iteraciones la convergencia se hace *superlineal*, como si el número de condicionamiento inicial fuese sustituido por otro menor, de tal forma que la razón de convergencia depende, además, de la distribución total de los valores propios de A. Las técnicas de precondicionamiento, tienen por objeto transformar el sistema original $A\,x = b$, en otro, con otra nueva matriz $\tilde{A}$, con unos nuevos autovalores que conduzcan a una convergencia más rápida, bien disminuyendo el número de condicionamiento $K(A)$, bien mejorando la distribución de los autovalores más pequeños del espectro [48, 113]. Las distintas iteraciones x_i, del sistema precondicionado verificarán:

$$\| x - x_i \|_{\tilde{A}} \leq 2 \left(\frac{\sqrt{K(\tilde{A})} - 1}{\sqrt{K(\tilde{A})} + 1} \right)^i \| x - x_0 \|_{\tilde{A}}$$

con

$$x_i \in x_0 + K^i(\tilde{A}; \tilde{r_0}).$$

En los sistemas no simétricos, es complicado probar que el sistema precondicionado resultante posee un espectro de autovalores que mejore la convergencia

respecto al sistema original. Pero, por analogía, y, aún sin demostraciones matemáticas que lo confirmen, se podría esperar un comportamiento similar. Esta conclusión, corroborada numéricamente en todas las aplicaciones, permite tratar técnicas de precondicionamiento en los métodos tipo doble-gradiente, al igual que se utiliza en el Gradiente Conjugado, con las salvedades correspondientes que contemplen la no simetría del sistema.

6.4. MÉTODO DEL GRADIENTE CONJUGADO PRECONDICIONADO

Dado que en la Modelización de los Campos de Viento el tipo de matrices que aparecen en sus Sistemas de Ecuaciones, son Simétricas Definidas Positivas (SDP), el mejor método iterativo para resolverlas es el del Gradiente Conjugado (GC), cuyo algoritmo ya ha sido expuesto anteriormente (Sección 4.4). Teniendo en cuenta, además, que mediante un Precondicionamiento adecuado se consigue mejorar la convergencia del proceso iterativo, es por ello que adoptamos, como método más adecuado para la resolución de los Sistemas de Ecuaciones originados en los Modelos de Campos de Viento, el del Gradiente Conjugado Precondicionado (GCP) [74]. De hecho, todos los ejemplos realizados, cuyos resultados se recogen en el apartado de Experimentos Numéricos, han sido afrontados con este procedimiento, por considerarlo el más idóneo, dado que el tipo de matrices sobre las que se aplica son SDP.

En los casos de Precondicionamiento expuestos anteriormente, tanto por la izquierda, como por la derecha, aunque la matriz original del sistema, A, sea Simétrica Definida Positiva, las nuevas matrices precondicionadas, $\tilde{A}$, tanto

$$\tilde{A} = M^{-1}\, A$$

como

$$\tilde{A} = A\, M^{-1}$$

en general, no tienen por que continuar siendo Simétricas, de ahí la necesidad de introducir estrategias que al Precondicionar continúen conservando la Simetría, para que el método del Gradiente Conjugado no pierda, sino que mejore, su

eficacia.

Cuando se dispone de un Precondicionador, M, factorizable, por ejemplo recurriendo a la Factorización Incompleta de Cholesky, puede expresarse como

$$M = L\,L^T$$

entonces una forma simple de conservar la Simetría es factorizar el Precondicionador para aplicarlo por ambos lados y así resolver el sistema

$$L^{-1}\,A\,L^{-T}\,L^T\,x = L^{-1}\,b$$

tal que

$$L^{-1}\,A\,L^{-T} = \tilde{A}, \qquad L^T\,x = \tilde{x} \quad y \quad L^{-1}\,b = \tilde{b}$$

o sea

$$\tilde{A}\,\tilde{x} = \tilde{b}$$

en el que $\tilde{A}$ seguirá siendo una matriz Simétrica Definida Positiva.

Sin embargo, en orden a conservar la Simetría, para aplicar el método del Gradiente Conjugado, no es necesario expresar el Precondicionador de esta manera, sino reescribir el algoritmo del GC, teniendo en cuenta que ahora

$$\tilde{r} = \tilde{b} - \tilde{A}\,\tilde{x} = L^{-1}\,b - L^{-1}\,A\,L^{-T}\,L^T\,x = L^{-1}(b - A\,x) = L^{-1}\,r$$

o sea

$$r = L\,\tilde{r}$$

y, a partir de aquí, se obtendrá la siguiente secuencia de operaciones:

$$\tilde{\alpha_i} = \frac{\left(\tilde{r_i}, \tilde{r_i}\right)}{\left(\tilde{A}\,\tilde{p_i}, \tilde{p_i}\right)} = \frac{(\tilde{r_i})^T\,\tilde{r_i}}{(\tilde{p_i})^T\,\tilde{A}\,\tilde{p_i}} = \frac{(\tilde{r_i})^T\,L^T\,L^{-T}\,\tilde{r_i}}{(\tilde{p_i})^T\,L^{-1}\,A\,L^{-T}\,\tilde{p_i}} = \frac{(L\tilde{r_i})^T\,(L^{-T}\,\tilde{r_i})}{(L^{-T}\,\tilde{p_i})^T\,A\,(L^{-T}\,\tilde{p_i})}$$

llamando a:

$$z_i = L^{-T}\,\tilde{r_i}$$

$$d_i = L^{-T}\,\tilde{p_i}$$

resulta

$$\tilde{\alpha_i} = \frac{(r_i)^T\,z_i}{(d_i)^T\,A\,d_i} = \frac{\left(r_i, z_i\right)}{\left(A\,d_i, d_i\right)} \tag{6.3}$$

La expresión:

$$\tilde{x}_{i+1} = \tilde{x}_i + \tilde{\alpha}_i \, \tilde{p}_i$$

al multiplicarla por L^{-T}, se convierte en:

$$L^{-T} \, \tilde{x}_{i+1} = L^{-T} \, \tilde{x}_i + \tilde{\alpha}_i \, L^{-T} \, \tilde{p}_i$$

resultando

$$x_{i+1} = x_i + \tilde{\alpha}_i \, d_i \tag{6.4}$$

y la expresión:

$$\tilde{r}_{i+1} = \tilde{r}_i - \tilde{\alpha}_i \, \tilde{A} \, \tilde{p}_i = \tilde{r}_i - \tilde{\alpha}_i \, L^{-1} \, A \, L^{-T} \, \tilde{p}_i$$

al multiplicarla por L, se convierte en:

$$L \, \tilde{r}_{i+1} = L \, \tilde{r}_i - \tilde{\alpha}_i \, L \, L^{-1} \, A \, L^{-T} \, \tilde{p}_i$$

dando lugar a

$$r_{i+1} = r_i - \tilde{\alpha}_i \, A \, d_i. \tag{6.5}$$

$$\tilde{\beta}_i = \frac{\left(\tilde{r}_{i+1}, \tilde{r}_{i+1}\right)}{\left(\tilde{r}_i, \tilde{r}_i\right)} = \frac{(\tilde{r}_{i+1})^T \, \tilde{r}_{i+1}}{(\tilde{r}_i)^T \, \tilde{r}_i} = \frac{(\tilde{r}_{i+1})^T \, L^T \, L^{-T} \, \tilde{r}_{i+1}}{(\tilde{r}_i)^T \, L^T \, L^{-T} \, \tilde{r}_i} = \frac{(L \, \tilde{r}_{i+1})^T \, (L^{-T} \, \tilde{r}_{i+1})}{(L \, \tilde{r}_i)^T \, (L^{-T} \, \tilde{r}_i)}$$

y así:

$$\tilde{\beta}_i = \frac{(r_{i+1})^T \, z_{i+1}}{(r_i)^T \, z_i} = \frac{\left(r_{i+1}, z_{i+1}\right)}{\left(r_i, z_i\right)} \tag{6.6}$$

$$\tilde{p}_{i+1} = \tilde{r}_{i+1} + \tilde{\beta}_i \, \tilde{p}_i$$

se transforma en

$$L^{-T} \, \tilde{p}_{i+1} = L^{-T} \, \tilde{r}_{i+1} + \tilde{\beta}_i \, L^{-T} \, \tilde{p}_i$$

o sea:

$$d_{i+1} = z_{i+1} + \tilde{\beta}_i \, d_i \tag{6.7}$$

y como

$$z_i = L^{-T} \, \tilde{r}_i = L^{-T} \, L^{-1} \, r_i = (L \, L^T)^{-1} \, r_i$$

resulta:

$$z_i = M^{-1} \, r_i \tag{6.8}$$

Reuniendo convenientemente las expresiones anteriores (de la 6.3 a la 6.8), se obtiene el siguiente algoritmo, en el que figura el precondicionador M^{-1} directamente sin necesidad de factorizarlo:

ALGORITMO DEL GRADIENTE CONJUGADO PRECONDICIONADO

Valor inicial: x_0

$r_0 = b - A\,x_0$

$z_o = M^{-1}\,r_o = d_o$

Para $i = 0, 1, 2, ...$ hacer:

$\alpha_i = \dfrac{\left(r_i, z_i\right)}{\left(A\,d_i, d_i\right)}$

$x_{i+1} = x_i + \alpha_i\,d_i$

$r_{i+1} = r_i - \alpha_i\,A\,d_i$

$z_{i+1} = M^{-1}\,r_{i+1}$

$\beta_i = \dfrac{\left(r_{i+1}, z_{i+1}\right)}{\left(r_i, z_i\right)}$

$d_{i+1} = z_{i+1} + \beta_i\,d_i$

Si $\left\| \frac{r_{i+1}}{r_0} \right\| < \epsilon \rightarrow Fin$

6.5. PRECONDICIONADORES EXPLÍCITOS E IMPLÍCITOS

El campo de posibles precondicionadores es muy amplio. En principio, los precondicionadores han de cumplir los requisitos generales:

a) Facilidad de implementación, evitando un coste computacional excesivo del producto matriz por vector.

b) Mejorar la convergencia.

Existen dos tipos básicos de precondicionadores: los explicítos y los implícitos. Los precondicionadores explícitos son aquellos que se construyen calculando directamente una matriz M, que se aproxime a la inversa A^{-1}, de la matriz dada.

Esto es, para resolver el sistema $A\,x = b$, se considera un nuevo sistema

$$M\,A\,x = M\,b = \tilde{b}$$

que se resuelve iterativamente. Como se dispone explícitamente tanto de M como de A, cada iteración requiere sólo de productos matriz-vector.

Estos precondicionadores explícitos son igualmente aplicables tanto para el caso de realizar el precondicionamiento por la izquierda, como por la derecha.

Los precondicionadores implícitos son aquellos que no se construyen directamente a partir de A, sino a partir de una factorización aproximada de A como, por ejemplo, la factorización incompleta LU y se utiliza ésta para establecer su inversa. Aquí la matriz de precondicionamiento es M^{-1}, siendo $M = LU$, y se requiere la resolución del sistema

$$M^{-1}\,A\,x = M^{-1}\,b = \tilde{b}.$$

Este tipo de precondicionadores, dado que se dispone de su factorización, resultan muy adecuados para el uso de la técnicas de precondicionamiento por ambos lados.

Por tanto, los precondicionadores explícitos puede considerarse como una inversa aproximada de A, mientras que los precondicionadores implícitos requieren previamente de una factorización de la matriz inicial del sistema.

6.6. PRECONDICIONADORES EXPLÍCITOS

6.6.1. PRECONDICIONADOR AINV

El algoritmo AINV [10] establece un método para computar una factorización incompleta de la inversa de una matriz Simétrica Definida Positiva (SDP). La inversa aproximada *sparse* factorizada resultante se usa como un precondicionador explícito.

Se basa en que si $A_{n\times n}$ es SDP, entonces se puede obtener una factorización de su inversa a partir de un conjunto de direcciones $z_1, z_2, ..., z_n$, A-conjugadas.

Si $Z = [z_1, z_2, ..., z_n]$ es la matriz cuya i-ésima columna es el vector z_i, podre-

mos diagonalizar A por congruencia, de tal forma que:

$$Z^T A\, Z = D = \begin{pmatrix} p_1 & 0 & . & . & . & 0 \\ 0 & p_2 & . & . & . & 0 \\ 0 & 0 & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & 0 \\ 0 & 0 & . & . & 0 & p_n \end{pmatrix}$$

donde

$$p_i = z_i^T A\, z_i, \tag{6.9}$$

ya que al ser los vectores z_i, A conjugados, los productos

$$z_i^T A\, z_j = 0 \quad \forall j \neq i$$

resultará por tanto que

$$A^{-1} = Z\, D^{-1} Z^T.$$

El conjunto de direcciones conjugadas z_i se puede construir por un método de A-ortogonalización de *Gram-Schmidt*, proceso aplicable a cualquier conjunto de vectores $v_1, v_2 ...., v_n$, pero elegir $v_i = e_i$ (vector unitario i-ésimo) resulta computacionalmente más conveniente.

La matriz Z estaría pues configurada por los vectores:

$$z_1 = e_1 = \begin{pmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$$

$$z_2 = e_2 - \frac{a_1^T e_2}{a_1^T e_1} z_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} - \frac{p_2^0}{p_1^0} \begin{pmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} = \begin{pmatrix} \times \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

$$z_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ . \\ 0 \end{pmatrix} - \frac{p_3^0}{p_1^0} \begin{pmatrix} 1 \\ 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} - \frac{p_3^1}{p_2^1} \begin{pmatrix} \times \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} = \begin{pmatrix} \times \\ \times \\ 1 \\ 0 \\ . \\ 0 \end{pmatrix}$$

.....

donde a_i^T representa la fila i-ésima de A, que puede considerarse como $a_i^T = e_i^T A$.

Por tanto la matriz $Z = [z_1, z_2, ..., z_n]$, viene a ser una matriz triangular superior de diagonal principal unitaria.

Resultando así el

ALGORITMO DE LA FACTORIZACIÓN INVERSA (AINV)

(1) Hacer $z_i^{(0)} = e_i \quad (1 \leq i \leq n)$

(2) Para $i = 1, 2, ..., n$

y $j = i, i+1, ..., n$

$$p_j^{(i-1)} = a_i^T z_j^{(i-1)}$$

fin.

si $i = n$ ir a (3)

para $j = i+1, ..., n$

$$z_j^{(i)} = z_j^{(i-1)} - \left(\frac{p_j^{(i-1)}}{p_i^{(i-1)}} \right) z_i^{(i-1)}$$

fin.

fin.

(3) Hacer $z_i = z_i^{(i-1)}$ y $p_i = p_i^{(i-1)}$, para $1 \leq i \leq n$.

Volver a $Z = [z_1, z_2, ..., z_n]$ y $D = \begin{pmatrix} p_1 & 0 & . & . & . & 0 \\ 0 & p_2 & . & . & . & 0 \\ 0 & 0 & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & 0 \\ 0 & 0 & . & . & 0 & p_n \end{pmatrix}$.

Nótese la simplificación que supone que, para el cálculo de los productos internos, sólo se requiere el uso de las filas de A, lo cual hace el procedimiento bastante atractivo, puesto que no es necesario almacenar explícitamente la matriz A completa.

Una vez calculados Z y D, la solución del sistema $A\,x = b$, puede computarse como

$$x^* = A^{-1}b = Z\,D^{-1}Z^T b = \sum_{i=1}^{n} \left(\frac{z_i^T b}{p_i} \right) z_i$$

Aunque A sea una matriz *sparse*, el coste de esta algoritmo, aplicado tal como se ha descrito, lo hace inviable, ya que Z tiende a ser una matriz densa. Para evitar este efecto *fill-in*, conforme se van obteniendo los vectores z_i, se efectúa la simplificación de despreciar aquellas entradas inferiores en valor absoluto a una cierta tolerancia prefijada $0 \leq \delta \leq 1$.

Llamando $\tilde{Z}$ a la matriz triangular obtenida:

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

que sería la incompleta factorizada de A.

6.6.2. PRECONDICIONADOR SAINV

Aunque no es frecuente, el algoritmo AINV puede conducir (para matrices A,SDP) a valores de p_i (entradas de la diagonal principal) nulos o negativos. En el primer caso, en cuanto se obtuviera un p_i cero, no podría proseguir el proceso,puesto que los p_i son denominadores en las fórmulas de recurrencia que se utilizan en el algoritmo AINV para calcular las z_i^j, y, si lo que ocurre, es la aparición de un $p_i < 0$,como $A^{-1} \approx Z\,D^{-1}Z^T$, daría lugar a una aproximada inversa que ya no es Definida Positiva

Evidentemente, esto no ocurre en el caso teórico de calcular exactamente los valores de z_i, ya que siendo A, Simétrica Definida Positiva, la expresión (6.9) resultará

$$p_i = z_i^T A\, z_i > 0.$$

La razón de este *breakdown* es la siguiente:

Los pivotes p_i, entradas de la diagonal principal, D, se obtienen, según el algoritmo

AINV, por la relación

$$p_i = p_i^{(i-1)} = a_i^T z_i^{(i-1)} = \sum_{l=1}^{i-1} a_{il} z_{li}^{l-1} + a_{ii} \quad (1 \leq i \leq n),$$

si hacemos cero algunas de las entradas de los vectores z_i, algunos de los productos $a_{il} z_{li}$ desaparecerían y, en el caso general de A, SDP, estos sumandos que se eliminan pueden resultar positivos con lo que realmente estamos haciendo, al prescindir de ellos, es disminuir el valor teóricamente positivo de los p_i, pudiendo llegar a hacerse nulos o negativos.

Llamando $\tilde{z}_i$ a los vectores modificados (al eliminar las entradas inferiores a δ), para algunas matrices puede ocurrir que el cálculo de los correspondientes pivotes sea:

$$\tilde{p}_i = a_i^T \tilde{z}_i \ll \tilde{z}_i^T A \, \tilde{z}_i$$

con lo cual se va perdiendo la ortogonalidad de los vectores $\tilde{z}_i$, aumentando así las probabilidades del *breakdown*.

Para robustecer el proceso, [12], el algoritmo SAINV propone que los $\tilde{p}_i$ se computen usando la expresión

$$\tilde{p}_i = \tilde{z}_i^T A \, \tilde{z}_i,$$

procedimiento algo más costoso que el del AINV, pero más seguro y que garantiza la no ruptura del mismo.

Así el cálculo de los $\tilde{p}_i$ se puede expresar como

$$\tilde{p}_i = \tilde{v}_i^T \tilde{z}_i \quad \textit{donde} \quad \tilde{v}_i^T = \tilde{z}_i^T A.$$

La diferencia de coste dependerá de cuanto más denso resulte el vector $\tilde{v}_i$ en comparación con a_i^T. Los experimentos realizados demuestran que el resultado del nuevo procedimiento da lugar a un precondicionador de más alta calidad y coste muy parecido al anterior, con lo cual compensa definitivamente el uso de una aproximación algo más costosa.

Por todo ello, el nuevo algoritmo de la factorización inversa estabilizada (SAINV) puede escribirse de la siguiente forma:

ALGORITMO SAINV

(1) Hacer $z_i^{(0)} = e_i \quad (1 \leq i \leq n)$

(2) Para $i = 1, 2, ..., n$ hacer

$$v_i = A\, z_i^{i-1}$$

(3) para $j = i, i+1, ..., n$ hacer

$$p_j^{(i-1)} = v_i^T z_j^{(i-1)}$$

fin.

si $i = n$ ir a (4)

para $j = i+1, ..., n$ hacer

$$z_j^{(i)} = z_j^{(i-1)} - \left(\frac{p_j^{(i-1)}}{p_i^{(i-1)}} \right) z_i^{(i-1)}$$

fin.

fin.

(4) Hacer $z_i = z_i^{(i-1)}$ y $p_i = p_i^{(i-1)}$, para $1 \leq i \leq n$.

Volver a $Z = [z_1, z_2, ..., z_n]$ y $D = diag(p_1, p_2, ..., p_n)$

Obviamente los algoritmos AINV y SAINV [30] son matemáticamente equivalentes; sin embargo, con el estabilizado se consigue una aproximada inversa más fiable. Aplicándolo a cualquier matriz SDP se obtiene un proceso sin rupturas.

6.7. PRECONDICIONADORES IMPLÍCITOS

6.7.1. POR COMPARACIÓN CON EL MÉTODO DE RICHARDSON

El método de Richardson, método iterativo muy simple, de relación de recurrencia

$$x_{i+1} = x_i + \alpha(b - A\, x_i)$$

con $\alpha > 0$, permite, por comparación con otros métodos iterativos, definir cierta matrices utilizables como precondicionadores.

Para ello, contrastaremos la fórmula de recurrencia para la solución que resulta de aplicar el método Richardson al sistema precondicionado por una matriz genérica M, con la fórmula correspondiente que se obtiene aplicando los métodos, teóricamente superiores, de Jacobi, SOR y SSOR al sistema sin precondicionar.

-Precondicionador de Jacobi ó Diagonal

Aplicando el método de Richardson al sistema precondicionado

$$M^{-1}A\,x = M^{-1}b$$

queda, para el cálculo de los sucesivos valores de la solución:

$$x_{i+1} = x_i + \alpha(M^{-1}b - M^{-1}A\,x_i)$$

y, multiplicando por la matriz de precondicionamiento, resulta:

$$M\,x_{i+1} = M\,x_i + \alpha(b - A\,x_i). \tag{6.10}$$

Por otro lado, considerando la descomposición de la matriz A de la forma $A = D - E - F$, (siendo D la matriz diagonal formada por los elementos de la diagonal principal de A y E, F matrices triangulares), y utilizando el método de Jacobi para la resolución del sistema $A\,x = b$, resulta,

$$x_{i+1} = D^{-1}(E + F)x_i + D^{-1}b$$

multiplicando por la matriz diagonal, y operando:

$$D\,x_{i+1} = D\,x_i + (b - A\,x_i). \tag{6.11}$$

Comparando las expresiones de recurrencia finales de ambos métodos, (6.10) y (6.11), se observa que el método de Jacobi aplicado al sistema sin precondicionar, equivale al de Richardson, con $\alpha = 1$, menos robusto y más simple, cuando se aplica al sistema precondicionado con la matriz diagonal $D = diag(A)$.

Resulta así un precondicionador elemental, que se conoce como precondicionador Diagonal, fácil de implementar y con matriz inversa que se determina con muy bajo coste computacional.

-Precondicionador SOR

Aplicando el método SOR al sistema $A\,x = b$, y con la misma descomposición anterior $A = D - E - F$, queda para la fórmula de recurrencia de la solución:

$$x_{i+1} = (D - w\,E)^{-1}[(1-w)D + w\,F]x_i + w(D - w\,E)^{-1}b,$$

donde w es el llamado parámetro de relajación y, operando convenientemente resulta:

$$(D - w\,E)x_{i+1} = (D - w\,E)x_i + w(b - A\,x_i). \tag{6.12}$$

Comparando de nuevo con el método de Richardson aplicado al sistema precondicionado, (6.10), definiríamos, en esta ocasión, la matriz de precondicionamiento como:

$$M = (D - w\,E).$$

-Precondicionador SSOR

Aplicando ahora el método SSOR al sistema sin precondicionar, se obtiene para la solución:

$$x_{i+1} = \left(\frac{D}{w} - F\right)^{-1}\left(\frac{1-w}{w}D + E\right)\left(\frac{D}{w} - E\right)^{-1}\left(\frac{1-w}{w}D + F\right)x_i +$$
$$+\left(\frac{D}{w} - F\right)^{-1}\left(\frac{2-w}{w}\right)D\left(\frac{D}{w} - E\right)^{-1}b$$

operando, para expresar esta relación de forma que se pueda comparar con la (6.10), queda

$$\frac{1}{w(2-w)}(D-wE)D^{-1}(D-wF)x_{i+1} = \frac{1}{w(2-w)}(D-wE)D^{-1}(D-wF)x_i+(b-A\,x_i) \tag{6.13}$$

con lo que resulta como matriz de precondicionamiento

$$\frac{1}{w(2-w)}(D - wE)D^{-1}(D - wF).$$

En el caso de aplicar este precondicionador a sistemas simétricos, ya que en estos casos, $(D - wF) = (D - wE)^T$, se puede expresar como un producto de dos matrices triangulares transpuestas,

$$M = \left[\frac{(D - wE)D^{-1/2}}{\sqrt{w(2-w)}}\right]\left[\frac{(D - wE)D^{-1/2}}{\sqrt{w(2-w)}}\right]^T$$

Para sistemas no simétricos, se puede escribir como producto de dos matrices triangulares, inferior y superior, respectivamente

$$M = (I - wED^{-1}) \left[\frac{D - wF}{w(2-w)} \right]$$

6.7.2. POR FACTORIZACIONES INCOMPLETAS

La propiedad que en principio, debe verificar una matriz de precondicionamiento M, de ser una aproximación, más o menos cercana, de la matriz de coeficientes del sistema, sugiere que un procedimiento para obtenerla sea el descomponer A de la forma $A \approx A_1 A_2$, de tal manera, que no suponga un excesivo esfuerzo computacional, y adoptar para la misma

$$M = A_1 A_2$$

Además de otras factorizaciones posibles destacaremos dos de ellas: la basada en la factorización en dos matrices triangulares LU, usualmente utilizada en la resolución de sistemas por métodos directos, y la factorización incompleta de Cholesky .

-Precondicionador ILU(0)

Resulta de descomponer A en dos matrices triangulares, inferior y superior, respectivamente, L y U,

$$A \approx LU = M$$

cuyos elementos, m_{ij}, sean tales que:

$$\begin{aligned} m_{ij} = 0 \quad si \quad a_{ij} = 0 \\ (A - LU)_{ij} = 0 \quad si \quad a_{ij} \neq 0 \end{aligned}$$

es decir, que los elementos nulos de la matriz del sistema, continúen siendo nulos en las posiciones respectivas de las matrices triangulares.

Si no realizáramos esta simplificación, el coste computacional se incrementaría y equivaldría a resolver el sistema por un método directo.

-Precondicionador ILU(n)

Para las matrices tridiagonales o pentadiagonales que resultan de la discretización de problemas con ecuaciones en derivadas parciales elípticas, se pueden practicar otros niveles de factorización, consistentes en *rellenar* alguna diagonal de las matrices factores de la descomposición, que en $ILU(0)$ serían nulas. La aproximación sería mayor a costa de incrementar el trabajo computacional de la descomposición.

-Factorización incompleta de Cholesky

Está especialmente indicado para factorizar matrices Simétricas Definidas Positivas, (SDP).Su objetivo consiste en descomponer A en tres matrices, una matriz central diagonal y dos matrices laterales: una triangular inferior y su transpuesta.

Sea $A = (a_{ij})$ una matriz $n \times n$, SDP. En orden a realizar su factorización podemos considerarla formada por

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & f_1^T \\ f_1 & A_2 \end{pmatrix}$$

o sea, compuesta por: su primer elemento, a_{11}; una matriz columna, $(n-1)\times 1$, f_1, formada por los elemtos de su primera columna menos el primero; su transpuesta, la matriz fila, $1 \times (n-1)$, f_1^T y la matriz A_2, $(n-1) \times (n-1)$, SDP, resultado de eliminar la primera fila y la primera columna de la matriz inicial A.

A partir de esta estructura, como primer paso de su factorización, fácilmente puede descomponerse en un producto de tres matrices, de la siguiente forma:

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & f_1^T \\ f_1 & A_2 \end{pmatrix} = \begin{pmatrix} a_{11} & 0 \\ f_1 & I \end{pmatrix} \begin{pmatrix} a_{11}^{-1} & 0 \\ 0 & C_2 \end{pmatrix} \begin{pmatrix} a_{11} & f_1^T \\ 0 & I \end{pmatrix} = L_1\, Z_1\, L_1^T$$

Siendo I la matriz unidad y C_2 una matriz $(n-1) \times (n-1)$, SDP, obtenida a partir de A_2, tal que:

$$C_2 = A_2 - \frac{1}{a_{11}} f_1\, f_1^T = (a_{ij})^{(2)}, \qquad \forall i, j \geq 2$$

en la que, por tanto, su primer elemento será:

$$(a_{22})^{(2)} = a_{22} - \frac{a_{12}^2}{a_{11}}.$$

Esta nueva matriz C_2, así obtenida, a su vez puede estructurarse de la misma forma que se hizo con la matriz inicial A:

$$C_2 = (a_{ij})^{(2)} = \begin{pmatrix} a_{22}^{(2)} & f_2^T \\ f_2 & A_3 \end{pmatrix}$$

Siendo f_2 una matriz columna $(n-2)\times 1$ y A_3 una matriz SDP, $(n-2)\times(n-2)$. En orden a evitar el efecto *fill-in*, que aparecerá al calcular los elementos $(a_{i2})^{(2)}$, para $i \geq 3$, correspondientes a la matriz columna f_2 y su transpuesta, realizaremos una aproximación, tomando en su lugar una nueva matriz columna, l_2, que tenga las mismas entradas que f_2, pero manteniendo nulas aquellas que lo son en la matriz inicial A, y así resultará:

$$C_2 = (a_{ij})^{(2)} = \begin{pmatrix} a_{22}^{(2)} & f_2^T \\ f_2 & A_3 \end{pmatrix} \approx \begin{pmatrix} a_{22}^{(2)} & l_2^T \\ l_2 & A_3 \end{pmatrix}$$

nueva matriz, igual de *sparse* que la inicial, que volvemos a factorizar de la misma forma anterior, en una matriz central y dos triangulares a ambos lados:

$$C_2 = (a_{ij})^{(2)} \approx \begin{pmatrix} a_{22}^{(2)} & l_2^T \\ l_2 & A_3 \end{pmatrix} = \begin{pmatrix} a_{22}^{(2)} & 0 \\ l_2 & I \end{pmatrix} \begin{pmatrix} a_{22}^{(2)-1} & 0 \\ 0 & C_3 \end{pmatrix} \begin{pmatrix} a_{22}^{(2)} & l_2^T \\ 0 & I \end{pmatrix}$$

siendo ahora

$$C_3 = (a_{ij})^{(3)} = A_3 - \frac{1}{a_{22}^{(2)}} l_2\, l_2^T$$

una matriz SDP, $(n-3)\times(n-3)$, $\quad \forall i,j \geq 3$.

En dicha matriz su primera entrada ahora será:

$$a_{33}^{(3)} = a_{33}^{(2)} - \frac{a_{23}^{(2)2}}{a_{22}^{(2)}}.$$

Con las factorizaciones realizadas hasta ahora, teniendo en cuenta que $f_1 = l_1$, la matriz inicial A se podrá descomponer de la siguiente forma:

$$A \approx \begin{pmatrix} a_{11} & 0 \\ l_1 & I \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(2)} & 0 \\ 0 & l_2 & I \end{pmatrix} \begin{pmatrix} a_{11}^{-1} & 0 & 0 \\ 0 & a_{22}^{(2)-1} & 0 \\ 0 & 0 & C_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(2)} & l_2^T \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} a_{11} & l_1^T \\ 0 & I \end{pmatrix} =$$

$$= L_1\, L_2\, Z_2\, L_2^T\, L_1^T$$

producto de una matriz central y dos matrices triangulares a cada lado.

Continuando con la factorización de C_3, de forma similar a la anterior, puede descomponerse la matriz inicial como:

$$A \approx \begin{pmatrix} a_{11} & 0 \\ l_1 & I \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(2)} & 0 \\ 0 & l_2 & I \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_{33}^{(3)} & 0 \\ 0 & 0 & l_3 & I \end{pmatrix} \begin{pmatrix} a_{11}^{-1} & 0 & 0 & 0 \\ 0 & a_{22}^{(2)^{-1}} & 0 & 0 \\ 0 & 0 & a_{33}^{(3)^{-1}} & 0 \\ 0 & 0 & 0 & C_4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_{33}^{(3)} & l_3^T \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(2)} & l_2^T \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} a_{11} & l_1^T \\ 0 & I \end{pmatrix} = L_1\, L_2\, L_3\, Z_3\, (L_1\, L_2\, L_3)^T$$

Y procediendo de la misma forma, al llegar a la n-sima factorización resultará:

$$A \approx L_1\, L_2\, L_3 .....L_n\, Z_n\, (L_1\, L_2\, L_3 .....L_n)^T = L\, D\, L^T$$

tal que

$$L = \begin{pmatrix} a_{11} & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ a_{12} & a_{22}^{(2)} & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ a_{13} & a_{23}^{(2)} & a_{33}^{(3)} & 0 & \cdot & \cdot & \cdot & 0 \\ a_{14} & a_{24}^{(2)} & a_{33}^{(3)} & a_{44}^{(4)} & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1n} & a_{2n}^{(2)} & a_{3n}^{(3)} & a_{44}^{(4)} & \cdot & \cdot & \cdot & a_{nn}^{(n)} \end{pmatrix}$$

y

$$D = \begin{pmatrix} a_{11}^{-1} & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & a_{22}^{(2)^{-1}} & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & a_{33}^{(3)^{-1}} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & a_{44}^{(4)^{-1}} & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & a_{nn}^{(n)^{-1}} \end{pmatrix}$$

Con lo cual quedará la matriz A aproximadamente igual a un producto de dos matrices triangulares laterales, L y L^T, y una matriz central diagonal.

Los precondicionadores implícitos más usuales son el DIAGONAL y el $ILU(0)$.

El DIAGONAL, es con mucho, el que menos esfuerzo computacional exige. Se calcula directamente tomando los elementos de la diagonal principal de A y los productos matriz inversa por vector se efectúan, asimismo, de forma inmediata. Sin embargo su aplicación se ve reducida, puesto que en muchos sistemas mal condicionados, representativos de problemas físicos con capas límites, singularidades o condiciones de contorno especiales, no mejoran sustancialmente la convergencia.

Los precondicionadores $ILU(0)$, exigen más esfuerzo computacional inicial para su construcción y los productos de matriz inversa por vector se realizan por procesos de remonte, pero su aplicación da lugar a buenos resultados en sistemas que con el precondicionador DIAGONAL no convergen.

Capítulo 7

REORDENACIÓN

7.1. PRELIMINARES

En el capítulo anterior se comentó que en un Sistema Precondicionado $M^{-1}Ax = M^{-1}b$, la matriz de precondicionamiento ideal sería $M = A$, y así el sistema convergería en una sola iteración. Esto, aunque inviable en la práctica, ya que el coste computacional del cálculo de M equivaldría a la resolución del Sistema por un método directo, sugiere precondicionar con una matriz lo más aproximadamente posible a la inicial A, valorando el esfuerzo que su determinación suponga.

Siguiendo esta idea se ha propuesto anteriormente utilizar el Precondicionador $ILU(0)$, aproximando A por el producto de dos matrices triangulares, una inferior y otra superior, que conserven los mismos elementos nulos que la inicial, en sus mismas posiciones, sustituyendo solamente aquellos elementos distintos de cero.

Naturalmente, el coste de esta factorización será notablemente inferior al de la descomposición LU, porque en estas matrices *sparse* el número de elementos no nulos es relativamente reducido y así se evita el cálculo, en las matrices triangulares, de los valores correspondientes a las posiciones en las que los elementos de la matriz del sistema son ceros.

Para obtener una mejora en el proceso de Precondicionamiento se suele recurrir a las técnicas de Reordenación, ya establecidas, que se usan fundamentalmente en la resolución de Sistemas de Ecuaciones Lineales por métodos directos. L.C. Dutto estudia en [22] el efecto de la reordenación en la resolución de la ecuación de Navier-Stokes con el método GMRES.

Estas técnicas, basadas en la teoría de grafos [2], proporcionan una nueva matriz, equivalente a la inicial, pero disminuyendo su envoltura, consiguiendo un ancho de banda o perfil menor, con lo cual el efecto de relleno que se produce en la factorización LU queda disminuido, conservando el número máximo de sus elementos nulos en las matrices triangulares de la descomposición. Así partiendo de estas matrices reordenadas se consigue que la factorización $ILU(0)$ sea más cercana a la factorización completa LU, con lo que la matriz de Precondicionamiento se asemejará más a la matriz inicial del Sistema A.

Esto supone, teóricamente, que partiendo de una matriz reordenada se conseguirá un mejor Precondicionador; mejora que, como más adelante se comprobará, se verá reflejada en los resultados obtenidos en las aplicaciones prácticas.

Las técnicas de Reordenación, utilizadas para resolver los sistemas lineales, se basan en someter la matriz del sistema original a una permutación simétrica de sus filas y sus columnas, lo que supone cambiar el orden en que están escritas las ecuaciones del sistema inicial, así como también el de sus incógnitas.

Con una permutación simétrica se conservan los elementos de la diagonal principal, aunque colocados en orden diferente, resultando así, que el grafo asociado a una matriz, sea el mismo que le corresponde a su matriz permutada, pero con las *etiquetas* de sus vértices cambiadas. Por ello, los algoritmos desarrollados para conseguir una mejor reordenación de las matrices de un sistema lineal, se basan, fundamentalmente, en la renumeración de los vértices de sus grafos asociados.

Los algoritmos más eficientes, que utilizan estos principios, son:

1) El algoritmo Inverso de Cuthill-McKee (RCM, *Reverse Cuthill-McKee*), propuesto por George [42], para mejorar el algoritmo inicial de Cuthill-McKee [17].

2) El algoritmo del Mínimo Vecino (MN, *Minimun Neighbouring*) [63], que es una variante del algoritmo del Mínimo Grado (MDG, *Minimum Degree*), propuesto por George y Liu [42].

3) El algoritmo Multicoloring (MC), o grafo coloreado, desarrollado a partir del caso de dos colores, denominado Rerordenación Rojo-Negro (Red-Black Ordering), expuesto por Saad [94].

7.2. ALGORITMO DE CUTHILL-McKEE INVERSO (RCM)

El algoritmo de Cuthill-McKee proporciona un método sencillo para reordenar una matriz *sparse* con objeto de reducir los costes de almacenamiento (conservar la *sparsidad*, es decir reducir el efecto *fill-in*) transformando la matriz inicial en una matriz banda. La ordenación resultante al invertir el algoritmo de Cuthill-McKee resulta frecuentemente mejor que el original, en términos de reducción de perfil, aunque la anchura de la banda permanece sin mejorarse. Este algoritmo se denomina de Cuthill-McKee Inverso (RCM).

ALGORITMO RCM:

1 - Construir el grafo asociado a la matriz simétrica A, $g(x) = (V, E)$, siendo

$V = \{x = a_{ii}\}$ el conjunto de nodos y $E = \{\{a, b\}/a = a_{ii}, b = a_{jj} : i \neq j,\ a_{ij} \neq 0\}$

el conjunto de aristas.

2 - Determinar un nodo inicial (extremo y con pocas conecciones) y renumerarlo

como x_1.

3 - Renumerar los nodos conectados a x_i en orden ascendente de grado

4 - Efectuar el ordenamiento inverso.

5 - Fin.

7.3. ALGORITMO DEL MÍNIMO VECINO (MN)

Este algoritmo es una variante del algoritmo de grado mínimo, que opera eliminando los nodos de menor grado seleccionados en la estructura del grafo asociado a la matriz, pero de forma que no se define ni se inserta en los nuevos grafos, que se van originando por eliminación sucesiva de sus nodos, ninguna nueva

conexión.

Es especialmente útil cuando se busca una factorización incompleta con el mismo patrón de *sparsidad* que la matriz inicial del sistema, por ejemplo cuando se utilizan precondicionadores como el ILU(0).

ALGORITMO MN:

1 - Construir el grafo asociado a la matriz simétrica A, $g(x) = (V, E)$, siendo

$V = \{x = a_{ii}\}$ el conjunto de nodos y $E = \{\{a, b\}/a = a_{ii}, b = a_{jj} : i \neq j,\ a_{ij} \neq 0\}$

el conjunto de aristas.

2 - Mientras $V \neq \emptyset$:

2.1 - Elegir un nodo ν de grado mínimo en $g(x) = (V, E)$ y reordenar como

nodo siguiente.

2.2 - Definir:

$V_\nu = V - \nu, \quad E_\nu = \{\{a, b\} \in E/a, b \in V_\nu\}$

y hacer

$V = V_\nu,\ E = E_\nu \quad y \quad g(x) = (V, E).$

3 - Fin.

7.4. ALGORITMO DE GEORGE

La elección del nodo inicial que figura en el paso 2 de los algoritmos anteriores es muy importante para la eficiencia de los mismos. Para su determinación es muy útil el algoritmo de George [2],basado en la construcción de unas particiones del conjunto de nodos, llamadas estructuras con niveles enraizadas, que nos permite comenzar desde un nodo *pseudo-periférico*.

Si definimos la distancia $d(x, y)$ entre dos nodos x e y en un grafo $g(x) = (V, E)$, como la longitud de la trayectoria más corta que une ambos nodos, y la excentricidad de un nodo x por $\varepsilon(x) = Max\,\{d(x, y)/x, y \in V\}$, el algoritmo se escribe de la forma siguiente:

ALGORITMO DE GEORGE PARA LA BÚSQUEDA DE NODOS PSEUDO-PERIFÉRICOS

1 - Elegir un nodo arbitrario r de V.

2 - Generar una estructura con niveles enraizada en r,

$$\left\{L_0(r), L_1(r),\ \ldots, L_{\varepsilon(r)}(r)\right\}.$$

siendo $L_i(r) = \{x/d(x,r) = i\}$.

3 - Elegir un nodo x de grado mínimo en $L_{\varepsilon(r)}(r)$.

4 - Generar una estructura con niveles enraizada en x,

$$\left\{L_0(x), L_1(x),\ \ldots, L_{\varepsilon(x)}(x)\right\}.$$

5 - Si $\varepsilon(x) > \varepsilon(r)$, establecer $x \rightarrow r$ y volver al paso 3.

6 - Caso contrario tomamos x como nodo inicial.

7 - Fin.

7.5. ALGORITMO MULTICOLORING (MC)

El uso de los grafos coloreados es una técnica usual en computación que se basa en colorear los nodos de un grafo de tal forma que dos nodos adyacentes no tengan el mismo color, tratando de conseguir una coloración total del grafo usando el menor número de colores posibles.

El Método básico para obtener un *multicoloring* adecuado de un grafo [94], es el siguiente:

ALGORITMO MC

1 - Para $i = 1, ..., n$. Hacer: adjudicar $Color(i) = 0$.

2 - Para $i = 1, 2, ..., n$. Hacer:

3 - Adjudicar $Color(i)$= $\min\{k > 0/k \neq Color(j), \forall j \in Adj(i)\}$

4 - Fin.

Siendo $Adj(i)$ el conjunto de nodos adyacentes al nodo i.

En general el número de colores necesarios no excederá al del máximo grado de cada nodo +1.

Una vez asignados los colores a todos los nodos del grafo asociado a la matriz, se reordena ésta reagrupando todos los vértices, es decir, todos los elementos de la diagonal principal, del mismo color; así se consigue una nueva estructura de la matriz reordenada por bloques, en la que los bloques diagonales serán precisamente matrices diagonales y el número de bloques coincidirá con el número de colores. El resto de las entradas configuraran dos matrices triangulares *sparse* situadas a ambos lados de los bloques diagonales.

Si bien la técnica del *multicoloring* resulta muy barata, al utilizar los precondicionadores ILU(0) puede ocurrir, según Saad, que el número de iteraciones, necesarias para alcanzar la convergencia, probablemente resulte mucho más alto precondicionando la matriz con el reordenamiento multicolor, que precondicionando la matriz original directamente.

Capítulo 8

PRECONDICIONAMIENTO DE SISTEMAS DE ECUACIONES LINEALES DE MATRIZ VARIABLE

8.1. PROPUESTA DE ESTRATEGIA

El objetivo principal de esta tesis consiste en extender las técnicas de Precondicionamiento, así como las de Reordenación, a los sistemas de ecuaciones lineales de matrices variables [105, 96], que surgen de la modelización de campos de viento [75], que como ya se ha visto anteriormente, son del tipo:

$$A_\varepsilon \, x_\varepsilon = b_\varepsilon$$

donde ε representa el parámetro de estabilidad del modelo y

$$A_\varepsilon = M + \varepsilon \, N$$

siendo M y N matrices constantes para un nivel de discretización dado y, además, Simétricas Definidas Positivas (SDP), por lo que, para su resolución, utilizaremos siempre el algoritmo del Gradiente Conjugado Precondicionado.

Para precondicionar estos sistemas se podría recurrir, en principio, a dos estrategias extremas:

a) Por un lado, se podría construir un único precondicionador para un cierto valor de ε, ε_o, y lo aplicaríamos para la resolución de los distintos sistemas que surgen para cada valor de ϵ, lo que conducirá a convergencias cada vez más lentas a medida que los valores de ε se vayan alejando del valor inicial.

b) Por otro lado, yéndonos al extremo opuesto, usaríamos un precondicionador diferente para cada sistema, o sea para cada valor de ε, lo que resultaría muy costoso.

En esta tesis se propone una solución intermedia, que consiste en construir un precondicionador, que pueda ser actualizado fácilmente para cada valor de ε, y cuya aplicación al algoritmo del Gradiente Conjugado de lugar a velocidades de convergencia comprendidas entre las conseguidas al aplicar las estrategias extremas mencionadas. Con lo que en definitiva se conseguirá mejorar el grado de eficacia del algoritmo a utilizar en la resolución del sistema.

Este tipo de solución intermedia ya ha sido propuesto por Benzi [11] y Meurant [65], independientemente, usando cada uno de ellos un modelo de Precondicionador diferente que, a un bajo coste computacional, se adaptan fácilmente para cada valor del parámetro, ε; estrategia que conduce a conseguir un grado de convergencia, con el algoritmo del Gradiente Conjugado Precondicionado, intermedio entre las convergencias alcanzables usando las dos opciones extremas.

Benzi desarrolla su estudio utilizando un Precondicionador Explícito, mediante la construcción de Inversas Aproximadas, usando el algoritmo SAINV, para el caso especial de matrices variables, A_ε, del tipo

$$A_\varepsilon = M + \varepsilon\, I$$

siendo I la matriz unitaria.

Sin embargo Meurant lo hace para la matriz

$$A_\varepsilon = M + \varepsilon\, D$$

siendo D una matriz diagonal y considerando un Precondicionador Implícito, a partir de una Factorización Incompleta de Cholesky de la matriz M.

8.2. ADAPTACIÓN DEL PRECONDICIONADOR SAINV

En este estudio, proponemos seguir un camino paralelo al propuesto por Benzi [97] para el caso especial de matrices variables del tipo:

$$A_\varepsilon = M + \varepsilon\, I$$

siendo M una matriz SDP e I la matriz unitaria; pero extendiéndolo al caso más genérico de

$$A_\varepsilon = M + \varepsilon\, N$$

Con el algoritmo SAINV, ya expuesto en el Capítulo 6, (6.6.2), se puede construir una inversa aproximada factorizada de una matriz A, SDP, a partir de la obtención por congruencia de una forma diagonal de la misma:

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

mediante la matriz triangular superior $Z = [z_1, z_2, ..., z_n]$ conseguida por un proceso de A-conjugación de Grand-Schmidt, a partir del conjunto de vectores unitarios linealmente independientes $\{e_1, e_2, ..., e_n\} \in \Re^n$, donde

$$d_j = z_j^T A\, z_j > 0, \qquad 1 \leq j \leq n.$$

Si realizamos el proceso de cálculo de los vectores z_i de forma incompleta, descartando, en cada caso, las entradas respectivas menores de una cierta tolerancia escogida, δ, tal que: $0 < \delta < 1$, se obtiene una matriz *sparse* aproximada de Z, que denominamos $\tilde{Z}$, con la que se podrá construir la matriz aproximada inversa de A:

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

Pues bien, aplicando dicho algoritmo, se puede obtener una aproximada inversa de M:

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

y, a partir de ahí, se considera un precondicionador para $A_\varepsilon = M + \varepsilon\, N$ de la forma:

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

donde E sería una matriz genérica, simétrica, a determinar, fácilmente computable, que haga a $(\tilde{D} + \varepsilon\, E)$ Simétrica Definida Positiva y tal que los productos P_ε^{-1} por vector, que figuran en el Gradiente Conjugado, no supongan un coste elevado.

A efectos de definir E y, dando por supuesto que la inversa exacta de M sería $M^{-1} = Z\, D^{-1} Z^T$, se establece la diferencia

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

Si se tomara $E = Z^T N Z$, resultaría: $P_\varepsilon - A_\varepsilon = 0$, con lo cual se conseguiría el precondicionador ideal

$$P_\varepsilon^{-1} = A_\varepsilon^{-1}.$$

Evidentemente, esto no es viable, dado que no se dispone de la matriz Z, sino de su aproximación $\tilde{Z}$, pero ello sugiere como mejor expresión para la matriz E:

$$E = \tilde{Z}^T N\, \tilde{Z}$$

y así, de esta forma, E cumpliría con las condiciones necesarias mencionadas anteriormente.

En lugar de iniciar el proceso con la obtención de la aproximada inversa de M, que se corresponde con la aproximada inversa de $A_\varepsilon = M + \varepsilon\, N$, para $\varepsilon = 0$, se puede obtener inicialmente una aproximada inversa de $A_{\varepsilon_0} = M + \varepsilon_0\, N$. Y así las sucesivas matrices, para los distintos valores de ε se escribirían:

$$A_\varepsilon = M + \varepsilon\, N = A_{\varepsilon_0} - \varepsilon_0\, N + \varepsilon\, N = A_{\varepsilon_0} + \Delta\varepsilon\, N$$

donde $\Delta\varepsilon = \varepsilon - \varepsilon_0$.

Dado que ε es siempre positivo, la matriz A_ε, obviamente, es definida positiva, aunque $\Delta\varepsilon$ sea negativo.

De todo ello resulta que la aproximada inversa sería:

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

y el precondicionador para la matriz variable:

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

siendo por supuesto $E = \tilde{Z}^T N \tilde{Z}$, al igual que antes y con las mismas prestaciones.

Una opción para construir E, con los requisitos previstos, es tomar una aproximación de $\tilde{Z}$ que nombraremos como $\tilde{Z}_k$, que se obtiene extrayendo solamente su diagonal principal, si $k = 1$, y además sus $k-1$ diagonales superiores, si $k > 1$, y considerando para N la aproximación N_h, extrayendo su diagonal principal, si $h = 1$, y las $h-1$ diagonales secundarias para $h > 1$. Con lo cual denominaremos

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

En la práctica, a efectos de no incrementar el coste por iteración del Gradiente Conjugado, resulta útil considerar las parejas $h = 1$ y $k = 2$ ó $h = 2$ y $k = 1$, que dan lugar, respectivamente, a las matrices $E_{1,2}$ ó $E_{2,1}$, tridiagonales. Incluso se puede conseguir una mayor simplificación considerando $h = k = 1$, resultando así la matriz $E_{1,1}$, diagonal.

8.3. ADAPTACIÓN DE LA FACTORIZACIÓN DE CHOLESKY

Aquí optamos por generalizar la factorización incompleta de Cholesky, propuesta por Meurant [98] para el caso de matrices $A_\varepsilon = M + \varepsilon D$, siendo D una matriz diagonal, al caso más general de las matrices $A_\varepsilon = M + \varepsilon N$, siendo M y N dos matrices simétricas definidas positivas $n \times n$. Así podremos escribir A_ε como sigue:

$$A_\varepsilon = (m_i j) + \varepsilon (n_i j) = \begin{pmatrix} m_{11} + \varepsilon n_{11} & (f_{1M} + \varepsilon f_{1N})^T \\ f_{1M} + \varepsilon f_{1N} & M_2 + \varepsilon N_2 \end{pmatrix}$$

donde f_{1M}, f_{1N} representan matrices columnas $((n-1,1)$ y M_2, N_2 matrices de orden $n-1$.

Factorizando A_ε,

$$A_\varepsilon = \begin{pmatrix} m_{11} + \varepsilon n_{11} & \mathbf{0} \\ l_{1M} + \varepsilon l_{1N} & \mathbf{I} \end{pmatrix} \begin{pmatrix} (m_{11} + \varepsilon n_{11})^{-1} & \mathbf{0} \\ \mathbf{0} & C_2 \end{pmatrix} \begin{pmatrix} m_{11} + \varepsilon n_{11} & (l_{1M} + \varepsilon l_{1N})^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

con lo que nos queda,

$$A_\varepsilon = L_1 Z_1 L_1^T \tag{8.1}$$

siendo $l_{1M} = f_{1M}$ y $l_{1N} = f_{1N}$.

Identificando, término a término, se obtiene para la matriz C_2:

$$C_2 = M_2 + \varepsilon N_2 - \frac{1}{m_{11} + \varepsilon n_{11}} \left(l_{1M} + \varepsilon l_{1N}\right) \left(l_{1M} + \varepsilon l_{1N}\right)^T \tag{8.2}$$

Si, a efectos de construir el precondicionador, tomamos como primera aproximación sólo los elementos de la diagonal de N, sería $l_{1N} = 0$, quedando

$$C_2 = \varepsilon D_2 + M_2 - \frac{1}{m_{11} + \varepsilon n_{11}} l_{1M} l_{1M}^T$$

y la aproximación de orden cero,

$$C_2 = \varepsilon D_2 + M_2 - \frac{1}{m_{11}} l_{1M} l_{1M}^T$$

con lo cual, las entradas de C_2 se obtendrían fácilmente, añadiendo εD_2 a la matriz que resulta en la factorización de M.

Otra aproximación consiste en considerar, en (8.2), todas las entradas de N_2 y despreciar los productos εl_{1N}, quedando C_2, de forma similar,

$$C_2 = \varepsilon N_2 + M_2 - \frac{1}{m_{11}} l_{1M} l_{1M}^T$$

Continuando con esta aproximación,

$$C_2 = \varepsilon N_2 + \begin{pmatrix} m_{22}^{(2)} & f_{2M}^T \\ f_{2M} & M_3 \end{pmatrix} = \begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & \left(f_{2M} + \varepsilon l_{2N}\right)^T \\ f_{2M} + \varepsilon l_{2N} & M_3 + \varepsilon N_3 \end{pmatrix}$$

Haciendo ceros en f_{2M} las correspondientes entradas nulas de M, para evitar el efecto fill-in, obtenemos l_{2M}.

Factorizando C_2:

$$C_2 \approx \begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & \mathbf{0} \\ l_{2M} + \varepsilon l_{2N} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \left(m_{22}^{(2)} + \varepsilon n_{22}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & C_3 \end{pmatrix} \begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & \left(l_{2M} + \varepsilon l_{2N}\right)^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

donde,

$$C_3 = M_3 + \varepsilon N_3 - \frac{1}{m_{22}^{(2)} + \varepsilon n_{22}} \left(l_{2M} + \varepsilon l_{2N}\right) \left(l_{2M} + \varepsilon l_{2N}\right)^T .$$

Utilizando las mismas simplificaciones anteriores, se consigue

$$C_3 = M_3 + \varepsilon N_3 - \frac{1}{m_{22}^{(2)}} l_{2M} l_{2M}^T = \begin{pmatrix} m_{33} + \varepsilon n_{33} & \left(f_{3M} + \varepsilon l_{3N}\right)^T \\ f_{3M} + \varepsilon l_{3N} & M_4 + \varepsilon N_4 \end{pmatrix}$$

resultando así, para C_3, la misma ley de formación que se obtuvo para C_2.

Con estos criterios de formación de las matrices C_i , la factorización aproximada de A_ε, queda:

$$A_\varepsilon \approx L_1 Z_1 L_1^T = L_1 L_2 Z_2 L_2^T L_1^T = (L_1 L_2 \cdots L_n)\, Z_n\, (L_1 L_2 \cdots L_n)^T \tag{8.3}$$

siendo Z_n la matriz diagonal

$$\begin{pmatrix} (m_{11}+\varepsilon n_{11})^{-1} & & & . & \\ & \left(m_{22}^{(2)}+\varepsilon n_{22}\right)^{-1} & & . & \\ & & \left(m_{33}^{(3)}+\varepsilon n_{33}\right)^{-1} & . & \\ . & . & . & . & . \\ & & & . & \left(m_{nn}^{(n)}+\varepsilon n_{nn}\right)^{-1} \end{pmatrix}$$

Las entradas diagonales de la matriz triangular inferior $(L_1 L_2 \cdots L_n)$ serán $m_{ii}^{(i)} + \varepsilon n_{ii}$. Y las respectivas columnas inferiores a los elementos diagonales vendrán definidas por matrices $l_{jM} + \varepsilon l_{jN}$ de orden $(n-j) \times 1$.

Capítulo 9

EXPERIMENTOS NUMÉRICOS

9.1. APLICACIONES TEST

9.1.1. PRELIMINARES

En este Capítulo presentamos los resultados obtenidos utilizando el Gradiente Conjugado (CG), con los precondicionadores propuestos en el Capítulo anterior, para resolver los Sistemas Lineales de Ecuaciones Variables que surgen en la discretización de los modelos de Campos de Viento. Los ejemplos se han realizado considerando un Campo de Viento sobre una región de la isla de Gran Canaria, utilizando un modelo de masa consistente, según se ha descrito en el Capítulo 2(2.4), y procediendo a su discretización con un mallado inicial (Ver 3.2), que genera un Sistema Lineal Variable de 17.791 ecuaciones, procediendo luego a un proceso de refinamiento de la malla inicial que da lugar a un nuevo Sistema de 43.954 ecuaciones y terminando con otro refinamiento final que conduce a 98.999 ecuaciones, de tal forma que las matrices M y N que se obtienen en cada caso, son Simétricas Definidas Positivas.

Todos los experimentos se ejecutaron en un XEON Precision 530 con Fortran de doble precisión. En la resolución de todos los sistemas siempre iniciamos el algoritmo CG con el vector nulo e interrumpimos el cálculo iterativo cuando $\|r_k\|_2 \leq 10^{-10} \|r_0\|_2$ o bien cuando el número de iteraciones es mayor que 5000.

Los resultados se presentan en tablas para un amplio rango de valores de ε, indicando el número de iteraciones alcanzadas y los tiempos de convergencia en segundos, para cada uno de los casos. Estos valores se muestran tanto para los precondicionadores explícitos (SAINV), como para los implícitos (ICHOL).

En el caso de los explícitos, hemos designado por SAINV_{11}, SAINV_{12} y SAINV_{21} los obtenidos según las distintas aproximaciones E_{11}, E_{12} y E_{21} de la matriz E, descritas en 8.2, comparándolos con los precondicionadores full-SAINV, consistentes en calcular y aplicar en cada caso la inversa aproximada de A_ε, sin ninguna simplificación, para cada valor diferente de ε, y con el precondicionador $\text{SAINV}(A_{\varepsilon_0})$, calculado como inversa aproximada de la matriz obtenida con el valor inicial de ε y utilizado, como precondicionador único, para la resolución de los sucesivos sistemas que se obtienen para los diferentes valores de ε.

En el caso de los precondicionadores implícitos hemos designado por ICHOL_D e ICHOL_N, los obtenidos siguiendo las aproximaciones expuestas en 8.3, comparándolos con los Full-ICHOL, conseguidos con la factorización incompleta de A_ε para cada valor diferente de ε, y con el $\text{ICHOL}(A_{\varepsilon_0})$, obtenido para la matriz inicial y aplicando el mismo para todos los valores distintos de ε.

En todos los casos se empieza por considerar $\varepsilon_0 = 0$, lo que supone construir la inversa aproximada, o la factorización incompleta, de la matriz M, por tanto, todos los resultados que aparecen en las primeras filas, de cada tabla, coinciden con el obtenido por la aplicación de los precondicionadores denominados "full".

En el Ejemplo 3, correspondiente al sistema más amplio de los estudiados, 98.999 ecuaciones, los precondicionadores más eficientes se han aplicado tanto a los Sistemas Originales, como a los Sistemas Reordenados, para comprobar así la influencia de la Reordenación en el caso de su uso con los Sistemas Lineales de Ecuaciones Variables.

9.1.2. EJEMPLO 1

En este apartado se presentan los resultados obtenidos en la resolución del Sistema Lineal Variable de 17.791 ecuaciones, utilizando el método del Gradiente Conjugado Precondicionado, aplicando tanto los Precondicionadores Explícitos, con la Inversa Aproximada (SAINV), ver Tabla 9.1, como los Precondicionadores Implícitos, conseguidos con la factorización incompleta de Cholesky (ICHOL), ver Tabla 9.2.

ε		Full-SAINV	SAINV_{11}	SAINV_{12}	SAINV_{21}	$\text{SAINV}(A_{\varepsilon_0})$
0	Iter.	226	-	-	-	-
	t(seg.)	81.26	-	-	-	-
10^{-6}	Iter.	427	427	434	428	427
	t(seg.)	80.77	2.59	2.98	2.95	**2.58**
10^{-5}	Iter.	427	427	434	426	427
	t(seg.)	81.06	2.59	3.00	2.94	**2.58**
10^{-4}	Iter.	428	428	427	428	427
	t(seg.)	81.10	2.59	2.95	2.94	**2.58**
10^{-3}	Iter.	458	458	457	457	457
	t(seg.)	79.77	**2.76**	3.15	3.15	**2.76**
10^{-2}	Iter.	353	353	362	352	361
	t(seg.)	80.70	**2.15**	2.50	2.43	2.17
10^{-1}	Iter.	286	286	296	286	310
	t(seg.)	80.12	**1.73**	2.04	1.98	1.87
10^{0}	Iter.	240	240	244	236	440
	t(seg.)	79.66	**1.47**	1.68	1.63	2.66
10^{1}	Iter.	448	448	500	436	1038
	t(seg.)	80.73	**2.73**	3.44	3.01	6.25
10^{2}	Iter.	1209	1158	1238	1093	3075
	t(seg.)	85.67	**7.01**	8.51	7.50	18.47
10^{3}	Iter.	1788	1767	2170	1633	4442
	t(seg.)	89.00	**10.68**	14.91	11.20	26.69
10^{4}	Iter.	1923	1923	3249	1794	4829
	t(seg.)	89.92	**11.62**	22.31	12.30	29.01
10^{5}	Iter.	1965	1979	> 5000	1835	> 5000
	t(seg.)	89.76	**11.97**	–	12.59	–
10^{6}	Iter.	1945	1932	> 5000	4923	2430
	t(seg.)	90.14	**11.69**	–	29.60	16.61

Tabla 9.1: Ejemplo 1, 17.791 ecuaciones: *Número de iteraciones y tiempo de computación (en segundos) del Gradiente Conjugado para los distintos Precondicionadores SAINV*

La Tabla 9.1 muestra que para valores de $\varepsilon \leq 10^{-3}$ basta con utilizar el mismo

Precondicionador para los diferentes valores de ε. Sin embargo para valores de $\varepsilon \geq 1$ con los Precondicionadores SAINV_{11} se consiguen los mejores resultados, siendo la reducción del coste computacional, en varios casos, superior al 50 % del coste obtenido con el Precondicionador $\text{SAINV}(A_{\varepsilon_0})$.

El contenido de esta tabla, y las siguientes, confirman las consideraciones previas del Capítulo 8, 8.1, ya que con los precondicionadores denominados "Full"se obtienen, efectivamente, las convergencias más lentas, lo que encarece el procedimiento; mientras que con los precondicionadores del tipo (A_{ε_0}) se obtienen peores convergencias a medida que los valores de ε se alejan del valor inicial ε_0.

Analizando los resultados recogidos en la Tabla 9.2, se observa que desde el valor de $\varepsilon = 10^{-6}$ hasta $\varepsilon = 10^{-2}$, el $\text{ICHOL}(A_{\varepsilon_0})$ parace ser suficiente para alcanzar la convergencia a muy bajo coste.

Desde $\varepsilon = 10^{-1}$ a $\varepsilon = 1$, la estrategia más rápida se consigue con el Full-ICHOL.

Pero para $\varepsilon > 1$, con el ICHOL_N se obtienen los mejores resultados, dado que con el Full-ICHOL no se consigue alcanzar la convergencia. Esto se debe a que, dada la especial estructura de las matrices, para valores altos de ε el algoritmo de Factorización Incompleta no funciona adecuadamente, puesto que con él las matrices pueden perder su positividad y por tanto la aplicación del Gradiente Conjugado Precondicionado resulta inestable. También con el $\text{ICHOL}(A_{\varepsilon_0})$ se observan convergencias más lentas sobre todo a medida que los valores de ε se alejan del valor inicial ε_0. Por tanto, podemos concluir que, para altos valores de ε, con ICHOL_N se consiguen los mejores resultados.

ε		ICHOL(A_{ε_0})	ICHOL$_D$	ICHOL$_N$	Full-ICHOL
0	n^{o}Iter.	-	-	-	155
	t(s)	-	-	-	2.00
10^{-6}	n^{o}Iter.	155	157	157	155
	t(s)	**1.92**	1.94	1.95	1.99
10^{-5}	n^{o}Iter.	157	157	171	170
	t(s)	**1.93**	1.94	2.13	2.17
10^{-4}	n^{o}Iter.	157	155	155	157
	t(s)	1.93	**1.92**	1.93	2.01
10^{-3}	n^{o}Iter.	166	166	166	166
	t(s)	**2.04**	2.05	2.08	2.10
10^{-2}	n^{o}Iter.	127	128	127	127
	t(s)	**1.56**	**1.58**	1.59	1.63
10^{-1}	n^{o}Iter.	148	117	106	101
	t(s)	1.81	1.46	1.34	**1.33**
1	n^{o}Iter.	279	197	105	81
	t(s)	3.43	2.43	1.32	**1.09**
10	n^{o}Iter.	676	475	200	272
	t(s)	8.24	5.82	**2.48**	3.40
10^2	n^{o}Iter.	2004	1183	407	> 5000
	t(s)	24.53	15.51	**5.03**	–
10^3	n^{o}Iter.	3056	1794	604	> 5000
	t(s)	37.5	21.96	**7.42**	–
10^4	n^{o}Iter.	3337	1926	647	> 5000
	t(s)	40.91	23.63	**7.96**	–
10^5	n^{o}Iter.	3363	1990	655	> 5000
	t(s)	41.05	24.46	**8.07**	–
10^6	n^{o}Iter.	3473	1926	650	> 5000
	t(s)	42.42	23.71	**7.99**	–

Tabla 9.2: Ejemplo 1, 17.791 ecuaciones: *Número de iteraciones y tiempo de computación (en s.) del Gradiente Conjugado con diferentes Precondicionadores por Factorización Incompleta de Cholesky*

9.1.3. EJEMPLO 2

Aquí se recogen los resultados obtenidos con los Precondicionadores SAINV e ICHOL, pero aplicados al sistema de 43.954 ecuaciones, obtenido después del primer refinamiento.

ε		full SAINV	SAINV_{11}	SAINV_{12}	SAINV_{21}	$\text{SAINV}(A_{\varepsilon_0})$
0	n^{o}Iter.	254	-	-	-	-
	t(seg.)	1542.47	-	-	-	-
10^{-5}	n^{o}Iter.	254	254	254	254	254
	t(seg.)	1543.79	21.28	21.98	22.04	**15.94**
10^{-4}	n^{o}Iter.	254	254	254	254	254
	t(seg.)	1537.59	21.58	21.99	22.03.70	**15.94**
10^{-3}	n^{o}Iter.	153	253	253	253	253
	t(seg.)	1539.87	21.22	21.92	21.97	**15.87**
10^{-2}	n^{o}Iter.	237	237	237	237	235
	t(seg.)	1547.28	20.32	20.88	20.95	**14.75**
10^{-1}	n^{o}Iter.	182	183	183	183	200
	t(seg.)	1540.39	16.85	17.37	17.41	**12.58**
10^{0}	n^{o}Iter.	137	191	196	191	307
	t(seg.)	1539.90	**17.35**	18.23	17.94	18.91
10^{1}	n^{o}Iter.	133	349	345	349	590
	t(seg.)	1591.51	**27.21**	27.94	28.26	36.46
10^{2}	n^{o}Iter.	229	786	846	815	1255
	t(seg.)	1622.76	**54.48**	60.60	58.69	78.29
10^{3}	n^{o}Iter.	332	1196	1748	1197	1777
	t(seg.)	1632.54	**80.07**	119.38	83.64	110.79

Tabla 9.3: Ejemplo 2, 43.954 ecuaciones: *Número de iteraciones y tiempo de computación (en segundos) del Gradiente Conjugado para los distintos Precondicionares SAINV*

En la Tabla 9.3 se observa que para valores de $\varepsilon \geq 1$, se vuelve a repetir que concretamente con los SAINV_{11} se consiguen los mejores tiempos, mientras que para valores inferiores de ε bastaría con utilizar el $\text{SAINV}(A_{\varepsilon_0})$

No se recogen los resultados para valores de $\varepsilon \geq 10^4$,dado que en todos los casos la convergencia fue extremadamente lenta (más de 5000 iteraciones).

ε		ICHOL(A_{ε_0})	ICHOL$_D$	ICHOL$_N$	Full-ICHOL
0	n^{o}Iter.	-	-	-	184
	t(s)	-	-	-	6.21
10^{-6}	n^{o}Iter.	184	184	184	184
	t(s)	**5.96**	5.99	6.00	6.21
10^{-5}	n^{o}Iter.	184	184	184	184
	t(s)	**5.96**	5.99	6.00	6.23
10^{-4}	n^{o}Iter.	184	184	184	184
	t(s)	**5.98**	5.99	6.01	6.21
10^{-3}	n^{o}Iter.	181	181	181	181
	t(s)	**5.89**	5.90	5.91	6.12
10^{-2}	n^{o}Iter.	170	170	170	169
	t(s)	**5.55**	5.56	5.56	5.73
10^{-1}	n^{o}Iter.	148	135	131	126
	t(s)	4.81	4.43	**4.29**	4.35
1	n^{o}Iter.	232	149	105	78
	t(s)	7.50	4.88	3.46	**2.79**
10	n^{o}Iter.	454	303	145	76
	t(s)	14.66	9.84	4.74	**2**.73
10^2	n^{o}Iter.	995	675	261	> 5000
	t(s)	32.09	22.03	**8.46**	–
10^3	n^{o}Iter.	1452	965	354	> 5000
	t(s)	46.58	31.29	**11.50**	–
10^4	n^{o}Iter.	1583	1049	384	> 5000
	t(s)	50.73	33.98	**12.45**	–
10^5	n^{o}Iter.	1604	1059	388	> 5000
	t(s)	51.40	34.25	**12.57**	–
10^6	n^{o}Iter.	1605	1060	388	> 5000
	t(s)	51.43	34.29	**12.58**	–

Tabla 9.4: Ejemplo 2, 43.954 ecuaciones:*Número de iteraciones y tiempo de computación (en s.) del Gradiente Conjugado con diferentes Precondicionadores por Factorización Incompleta de Cholesky*

Observando los resultados de la Tabla 9.4, con los Precondicionadores por Factorización Incompleta, podemos concluir que para pequeños valores de ε no es necesario adaptar la factorización inicial puesto que con ella se consigue alcanzar la convergencia a muy bajo coste. Sin embargo, con valores altos de ε, el ICHOL$_N$ tiene el mejor comportamiento. Conviene tener en cuenta que para los valores de ε comprendidos entre 1 y 10 la re-computarización de la factorización incompleta resulta más aconsejable.

9.1.4. EJEMPLO 3

En las Tablas 9.5 y 9.7 se presentan los resultados conseguidos para el sistema de 98.999 ecuaciones, obtenido con otro refinamiento posterior, utilizando los mismos tipos de Precondicionadores.

En esta caso, tanto con los Precondicionadores SAINV como ICHOL, en todas sus variantes, la convergencia fue extremadamente lenta (más de 5000 iteraciones) para valores de $\varepsilon \geq 10^4$, por lo que a partir de $\varepsilon = 10^3$ ya no se recogen los resultados en ambas tablas.

ε		Full-SAINV	SAINV_{11}	SAINV_{12}	SAINV_{21}	$\text{SAINV}(A_{\varepsilon_0})$
0	Iter.	278	-	-	-	-
	t(seg.)	3541.68	-	-	-	-
10^{-6}	Iter.	278	278	278	278	278
	t(seg.)	3540.47	25.83	26.70	30.21	**19.04**
10^{-5}	Iter.	278	278	278	278	279
	t(seg.)	3541.55	25.78	26.71	30.24	**19.05**
10^{-4}	Iter.	280	279	278	279	279
	t(seg.)	3566.12	25.81	26.73	30.30	**19.08**
10^{-3}	Iter.	277	278	276	278	276
	t(seg.)	3540.38	25.82	26.61	30.17	**18.37**
10^{-2}	Iter.	258	257	258	257	256
	t(seg.)	3538.34	24.32	25.31	28.48	**17.39**
10^{-1}	Iter.	214	227	229	224	307
	t(seg.)	3543.85	22.34	23.21	25.79	**20.82**
10^{0}	Iter.	193	283	291	275	653
	t(seg.)	3524.28	**26.14**	27.64	30.00	44.06
10^{1}	Iter.	257	591	589	581	1724
	t(seg.)	3591.23	**47.11**	48.93	54.65	116.03
10^{2}	Iter.	548	1724	1670	1659	> 5000
	t(seg.)	3724.73	**124.27**	126.10	127.71	–
10^{3}	Iter.	1290	4234	4002	4128	> 5000
	t(seg.)	3785.34	**297.10**	304.06	305.60	–

Tabla 9.5: Ejemplo 3, 98.999 ecuaciones: *Número de iteraciones y tiempo de computación (en segundos) del Gradiente Conjugado para los distintos Precondicionadores SAINV*

Una vez más se comprueba que para valores pequeños de ε basta con un precondicionador único, el $\text{SAINV}(A_{\varepsilon_0})$; es especialmente notorio que este tipo de precondicionador empieza a fallar a partir de $\varepsilon = 10^2$.

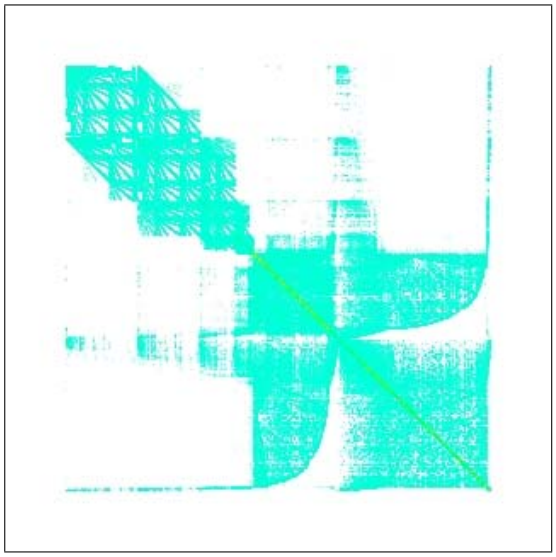
Sin embargo para valores de $\varepsilon \geq 1$ el SAINV_{11} presenta los mejores resultados.

ε		Orden Inicial	MN (69,43s)	RCM (0,70s)	MC (0,45s)
0	Iter.	278	264	273	263
	t(s)	3541.68	3092.93	**2757.27**	4626.74
10^{-6}	Iter.	278	265	274	263
	t(s)	25.83	16.35	**16.25**	20.13
10^{-5}	Iter.	278	264	274	263
	t(s)	25.78	16.29	**16.22**	20.15
10^{-4}	Iter.	279	264	273	263
	t(s)	25.81	16.27	**16.18**	20.15
10^{-3}	Iter.	278	262	272	260
	t(s)	25,82	16.16	**16.11**	19.93
10^{-2}	Iter.	257	236	247	240
	t(s)	24.32	**14.61**	14.66	28.40
10^{-1}	Iter.	227	208	212	219
	t(s)	22.34	12.89	**12.58**	16.81
1	Iter.	283	261	265	270
	t(s)	26.14	16.15	**15.71**	20.69
10	Iter.	591	520	549	553
	t(s)	47.11	**32.04**	32.32	42.24
10^2	Iter.	1724	1512	1589	1623
	t(s)	124.27	92.96	**92.23**	123.67
10^3	Iter.	4234	3701	3756	3924
	t(s)	297,10	227.79	**220.51**	294.04

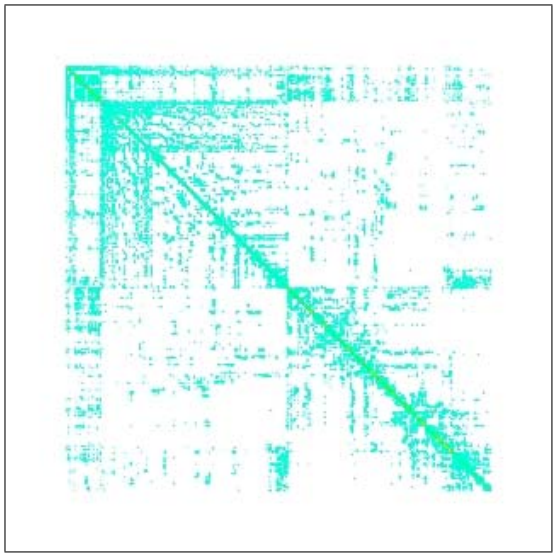
Tabla 9.6: Ejemplo 3, 98.999 ecuaciones: *Número de iteraciones y tiempo de computación (en segundos) del Gradiente Conjugado con el Precondicionador $SAINV_{11}$ para diferentes Reordenaciones*

Dado que con el precondicionador SAINV_{11} se consiguió un mejor comportamiento, fue elegido para aplicarlo también sobre los Sistemas Reordenados [28, 106] utilizando los Algoritmos citados en el Capítulo 7: Mínimo Vecino (MN), Cuthill-McKee Inverso (RCM) y Multicoloring (MC). Los resultados se recogen en la Tabla 9.6

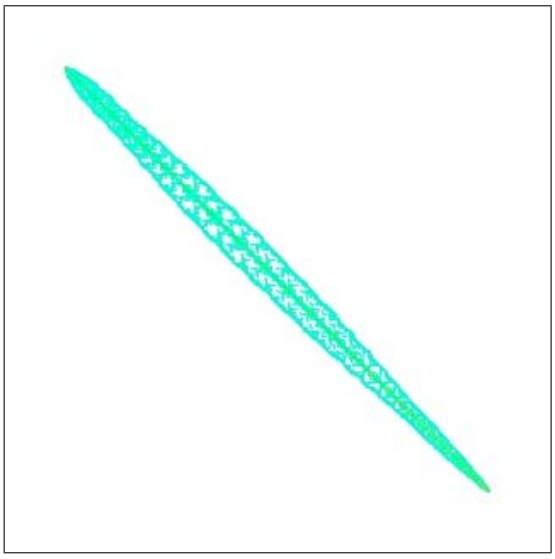
Como puede observarse, reordenando con los Algoritmos MN y RCM, mejora la convergencia del Gradiente Conjugado Precondicionado. Sin embargo el Algoritmo MC no es tan eficiente, como ya era de prever, de acuerdo con los comentarios expuestos en el apartado 6.5 sobre su falta de eficacia.



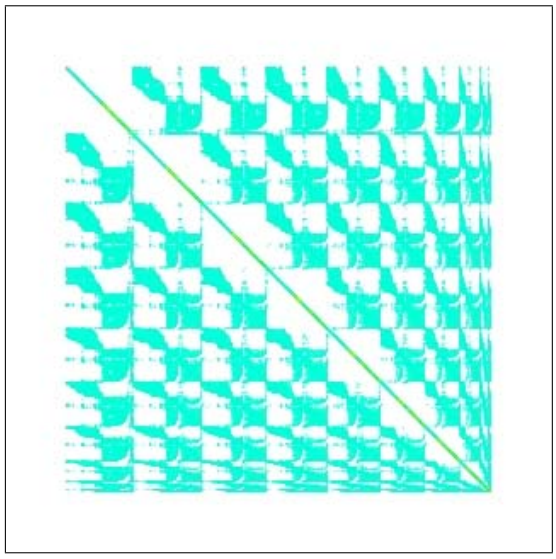
(a) Orden Inicial



(b) Mínimo Vecino



(c) Cuthill-McKee Inverso



(d) Multicoloring

Figura 9.1: *Patrones de 'sparsidad' de las matrices, con su orden inicial y una vez reordenadas*

En la figura 9.1 se muestran los patrones de *sparsidad* de la matriz inicial del sistema y de sus reordenaciones, según los Algoritmos indicados. La imagen correspondiente a la matriz con el reordenamiento Cuthill-McKee Inverso es la que más se asemeja a una matriz diagonal y en efecto es la reordenación que produce mejores resultados. Sin embargo la simple observación del patrón obtenido con el reordanamiento Multicoloring ya nos anticipa que los resultados a conseguir no van a ser mucho mejores que los del Gradiente Conjugado Precondicionado aplicado directamente sobre la matriz con su orden inicial, sobre todo para valores elevados de ε.

En la Tabla 9.7 se puede comprobar como en este ejemplo, para 98.999 ecuaciones, se obtienen resultados similares a los conseguidos con los precondicionadores ICHOL en los sistemas de 43.954 ecuaciones: en general, para pequeños valores de ε no es necesario adaptar la factorización inicial puesto que con ella se consigue alcanzar la convergencia a muy bajo coste. Sin embargo, con valores altos de ε, el ICHOL$_N$ tiene el mejor comportamiento. Teniendo en cuenta que para los valores de ε comprendidos entre 1 y 10 la re-computarización de la factorización incompleta resulta más efectiva.

De forma similar a como se hizo con los precondicionadores SAINV, aquí se ha elegido el precondicionador ICHOL$_N$, por su mejor comportamiento, para aplicarlo sobre los sistemas reordenados y comprobar así la eficacia de la Reordenación en los Sistemas de Ecuaciones Variables. Los resultados se recogen en la Tabla 9.8.

En este caso, para valores de ε comprendidos entre 0 y 10, se ha conseguido mejorar el coste de las iteraciones con la reordenación del Mínimo Vecino y a partir de valores de $\varepsilon \geq 10^2$ los mejores resultados se han conseguido con el reordenamiento de Cuthill-McKee Inverso. Pero teniendo en cuenta que el tiempo de implantación del MN(69,43s.) es muy superior al del RCM(0,7s.), en definitiva, la resolución se abarata mucho más con el Cuthill-McKee Inverso.

Por lo que respecta al Multicoloring, para ningún valos de ε, se ha conseguido mejorar la eficacia conseguida con el ICHOL$_N$ aplicado directamente a la matriz

ε		ICHOL(A_{ε_0})	ICHOL$_D$	ICHOL$_N$	Full-ICHOL
0	n^{o}Iter.	-	-	-	201
	t(s)	-	-	-	16.81
10^{-6}	n^{o}Iter.	201	201	201	201
	t(s)	**16.14**	16.16	16.19	16.82
10^{-5}	n^{o}Iter.	201	201	201	201
	t(s)	**16.15**	16.16	16.19	16.83
10^{-4}	n^{o}Iter.	201	201	201	201
	t(s)	**16.15**	16.16	16.19	16.83
10^{-3}	n^{o}Iter.	201	201	200	200
	t(s)	16.14	16.16	**16.11**	16.76
10^{-2}	n^{o}Iter.	188	191	189	189
	t(s)	**15.22**	15.35	15.24	15.87
10^{-1}	n^{o}Iter.	225	157	155	151
	t(s)	18.08	12.65	**12.52**	12.85
1	n^{o}Iter.	483	211	148	132
	t(s)	38.63	16.94	11.97	**11.33**
10	n^{o}Iter.	1350	540	259	236
	t(s)	107.71	43.14	20.91	**19.64**
10^2	n^{o}Iter.	3973	1466	593	> 5000
	t(s)	317.16	116.86	**47.62**	–
10^3	n^{o}Iter.	> 5000	3468	1269	> 5000
	t(s)	–	277.11	**101.60**	–

Tabla 9.7: Ejemplo 3: 98.999 ecuaciones. *Número de iteraciones y coste computacional (en s.) del Gradiente Conjugado con diferentes Precondicionadores por Factorización Incompleta de Cholesky*

inicial, lo cual revela la ineficacia de la reodenación MC con los precondicionadores ICHOL, confirmándose así la opinión de Yousef Saad en su obra *Iterative Methods* [94], donde comenta que *sobre todo con los precondicionadores ILU(0), puede ocurrir una pérdida de eficacia, puesto que el número de iteraciones para alcanzar la convergencia puede aumentar, resultando más alto que si se precondicionara directamente la matriz original.*

ε		Initial Ordering	MN (69.43 s)	RCM (0.70 s)	MC (0.45 s)
0	n^{o}Iter.	201	158	175	223
	t(s)	16.81	**12.86**	13.84	24.01
10^{-6}	n^{o}Iter.	201	159	175	223
	t(s)	16,19	**12.53**	13.46	22.51
10^{-5}	n^{o}Iter.	201	159	175	222
	t(s)	16.19	**12.54**	13.45	22.41
10^{-4}	n^{o}Iter.	201	158	176	223
	t(s)	16.19	**12.47**	13.53	22.51
10^{-3}	n^{o}Iter.	200	157	174	220
	t(s)	16.11	**12.37**	13.38	22.21
10^{-2}	n^{o}Iter.	189	143	160	207
	t(s)	15.24	11.31	12.30	20.90
10^{-1}	n^{o}Iter.	155	116	131	170
	t(s)	12.52	**9.19**	10.12	17.21
1	n^{o}Iter.	148	123	129	160
	t(s)	11.97	**9.73**	9.97	16.20
10	n^{o}Iter.	259	227	240	277
	t(s)	20,91	**17.88**	18.38	27.91
10^2	n^{o}Iter.	593	533	536	632
	t(s)	47.62	41.61	**40.73**	63.43
10^3	n^{o}Iter.	1269	1212	1177	1395
	t(s)	101.60	94.45	**89.23**	139.79

Tabla 9.8: Ejemplo 3, 98.999 ecuaciones: *Número de iteraciones y tiempo de computación (en segundos) del Gradiente Conjugado con el Precondicionador $ICHOL_N$ para diferentes Reordenaciones*

9.2. ELECCIÓN DEL PARÁMETRO ÓPTIMO

9.2.1. PRELIMINARES

Como ya se indicaba en el Capítulo 4, ESTIMACIÓN DE PARÁMETROS, el ajuste eficiente de un modelo de campo de viento depende, en gran medida, de la adecuada valoración de los parámetros que aparecen en las distintas etapas del proceso, especialmente de aquel que interviene de forma básica en la formulación del sistema

$$(M + \varepsilon\, N)x_\varepsilon = b_\varepsilon$$

puesto que afecta de forma directa al cálculo del viento resultante.

Dicho parámetro, ε, de estabilidad del modelo, está estrechamente relacionado con los módulos de precisión de Gauss, a través de las relaciones (2.12) y (2.17):

$$\varepsilon = \alpha^2 = \frac{T_v}{T_h} = \frac{\alpha_1^2}{\alpha_2^2}$$

E. Rodríguez en su Tesis *Modelización y simulación numérica de campos de viento mediante elementos finitos adaptativos en 3D* [89], propone la utilización de Algoritmos Genéticos, como métodos de optimización basados en un mecanismo de evolución natural, para realizar la selección automática del parámetro $\varepsilon(\alpha)$, pues se trata de una herramienta robusta, flexible, competitiva y cuyos cálculos pueden paralelizarse. Ver Capítulo 4(4.2).

Uno de los aspectos más importantes de los Algoritmos Genéticos es la construcción de una población inicial y la posterior evaluación, de cada individuo de la misma, de acuerdo con los resultados de su aplicación a una función objetivo. En el caso de la modelización de campos de viento se adopta como función objetivo la minimización de las diferencias entre el viento resultante, calculado mediante la resolución del sistema lineal $A_\varepsilon\, x_\varepsilon = b_\varepsilon$, y el observado, en las estaciones de referencia. Ello lleva como consecuencia que para cada individuo de la población inicial hay que resolver dicho sistema.

Luego, a tenor de los resultados conseguidos, se procederá a crear una nueva población, mediante operadores de Selección, Cruce y Mutación, que nuevamente hay que someterla a la evaluación de la función objetivo, lo cual supone volver a resolver el sistema tantas veces como valores seleccionados. A partir de estos

valores se vuelve a generar otra nueva población, con los mismos criterios, que se vuelve a evaluar, y así sucesivamente, hasta conseguir un individuo que cumpla con el criterio de parada, que será el valor óptimo del parámetro a considerar para ese Modelo.

Ello implica que en el proceso de selección hay que resolver el sistema $A_\varepsilon\, x_\varepsilon = b_\varepsilon$, tantas veces como posibles valores de $\varepsilon(\alpha)$ a estimar, multiplicado por el número de iteraciones a realizar, en cada paso sucesivo del Algoritmo Genético; por lo que se hace necesario disponer de métodos eficaces que permitan la implementación rápida de un precondicionador para cada valor del parámetro. De ahí la importancia de conocer el coste computacional que supone el utilizar distintos tipos de precondicionadores.

Como conclusión de los tests realizados anteriormente se estableció, que los precondicionadores basados en la Factorización Incompleta de Cholesky, parecen ser la herramienta más eficaz para mejorar la convergencia del método del Gradiente Conjugado, en la resolución de los Sistemas Lineales de Ecuaciones Variables, por tanto estos han sido los tipos de precondicionadores elegidos para comprobar su comportamiento a la hora de evaluar una población inicial de parámetros, $\varepsilon(\alpha)$, de cara a la elección de su valor óptimo mediante Algoritmos Genéticos [29].

En este apartado presentamos los resultados obtenidos al resolver los Sistemas Lineales de Ecuaciones Variables, correspondientes a la modelización de dos campos de viento diferentes, usando siempre el método del Gradiente Conjugado Precondicionado, el más eficaz cuando hay matrices Simétricas Definidas Positivas, como es el caso, y utilizando los precondicionadores basados en la Factorización Incompleta de Cholesky, ya estudiados anteriormente, en sus variantes $ICHOL(A_{\varepsilon_0})$, $ICHOL_D$, $ICHOL_N$ y $Full - ICHOL$.

Todos los experimentos han sido realizados en un equipo XENON Precisión 530 con Fortran de Doble Precisión. Los procesos de iteración siempre se han iniciado a partir de un vector nulo y finalizados si $\left\| \frac{r_i}{r_0} \right\|_2 \leq 10^{-10}$ o si el número de iteraciones se hacía superior a 10.000.

9.2.2. EJEMPLO 4

Aquí se presentan, en dos figuras, los tiempos empleados en alcanzar la convergencia con el algoritmo del Gradiente Conjugado Precondicionado, para una amplia gama de valores de $\alpha(\varepsilon)$, en la modelización del campo de viento, correspondiente a una región de la isla de Gran Canaria, que ya se utilizó en el Ejemplo 3, en el que mediante un refinamiento posterior del mallado inicial se había conseguido un sistema lineal de 98.999 ecuaciones.

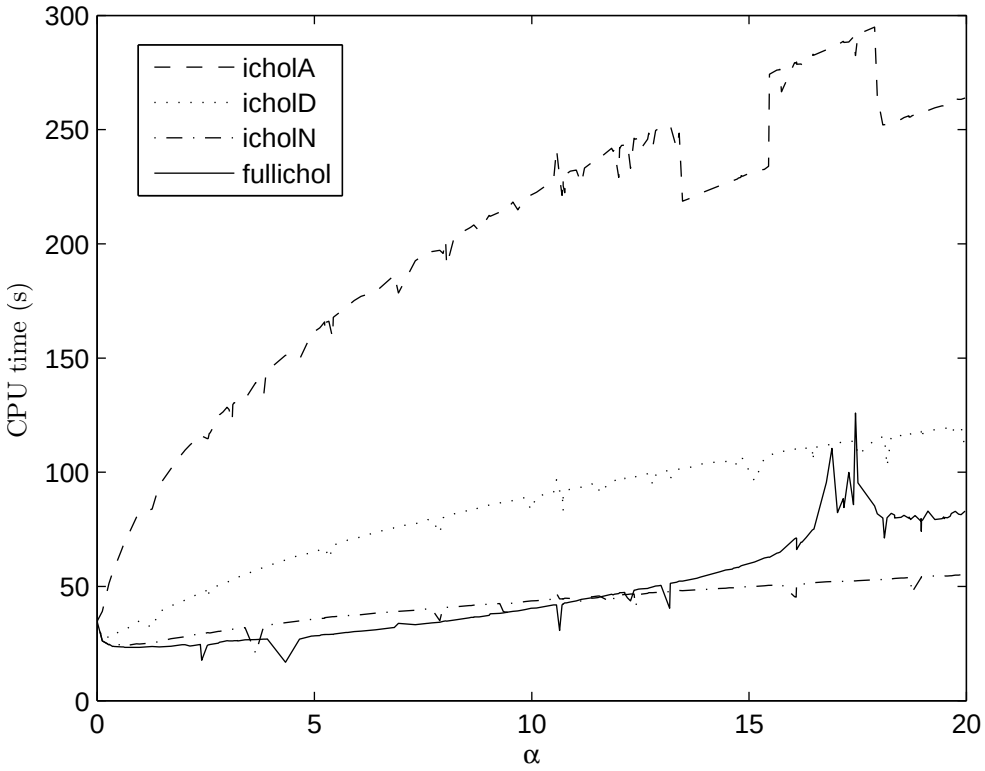


Figura 9.2: Campo de Viento de 98.999 ecuaciones: *Convergencia del Gradiente Conjugado Precondicionado, con varios Precondicionadores ICHOL, para diferentes valores del parámetro α, aleatoriamente calculados*

En la Figura 9.2 se recogen los tiempos de computación, en segundos, para una población inicial del parámetro α, formada por un conjunto muy diverso de valores, aleatoriamente elegidos, dentro de un intervalo de 0 a 20. Mediante cuatro gráficas diferentes se muestran los tiempos alcanzados utilizando los cuatro distintos precondicionadores ya citados:$ICHOL(A_{\varepsilon_0})$, $ICHOL_D$, $ICHOL_N$ y $Full - ICHOL$.

En la Figura 9.3 se muestran los tiempos de computación, en segundos, para

una población inicial, formada por un amplio conjunto de valores de α elegidos en un intervalo de 0 a 100, siguiendo una distribución normal. Así mismo, se recogen en cuatro gráficas diferentes los diversos tiempos alcanzados con la utilización de los cuatro precondicionadores ICHOL ya citados.

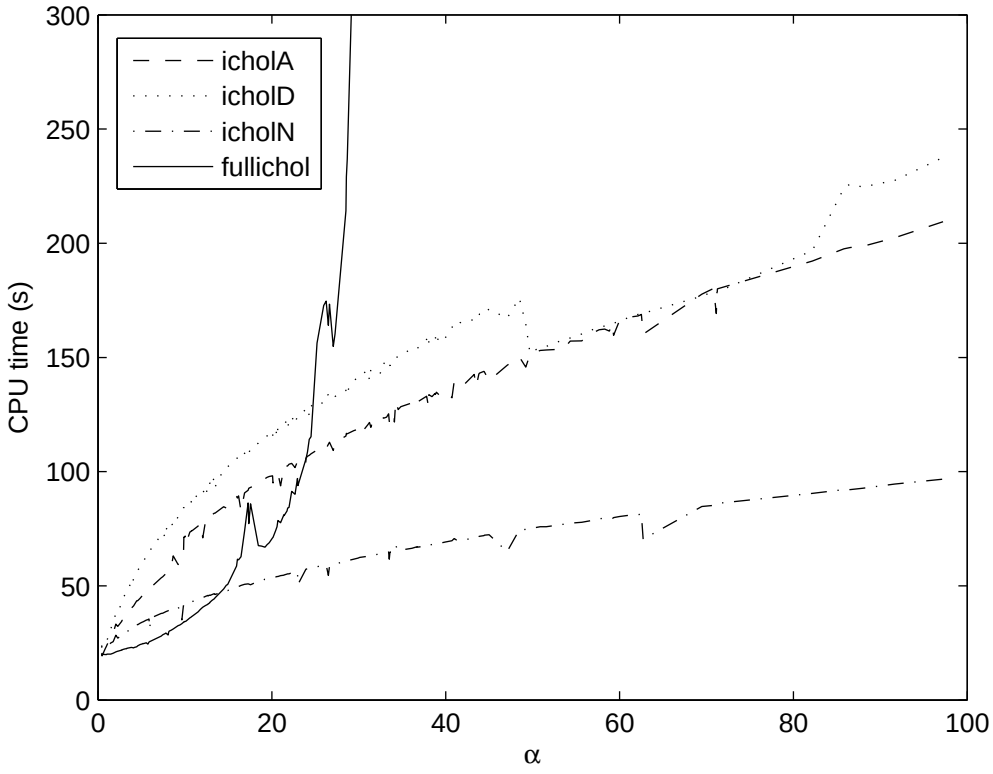


Figura 9.3: Campo de Viento de 98.999 ecuaciones: *Convergencia del Gradiente Conjugado Precondicionado, con varios Precondicionadores ICHOL, para diferentes valores del parámetro α, elegidos usando una distribución normal*

9.2.3. EJEMPLO 5

En este caso se recogen, en dos figuras también,los tiempos empleados en alcanzar la convergencia con el algoritmo del Gradiente Conjugado Precondicionado, para una amplia gama de valores de $\alpha(\varepsilon)$, pero ahora en la modelización de un campo de viento extendido a toda la isla de Gran Canaria, con un mallado en el que se consiguió un sistema lineal de 100.463 ecuaciones.

La Figura 9.4 muestra los tiempos de computación, en segundos, para una población inicial del parámetro α, formada por un amplio conjunto de valores,

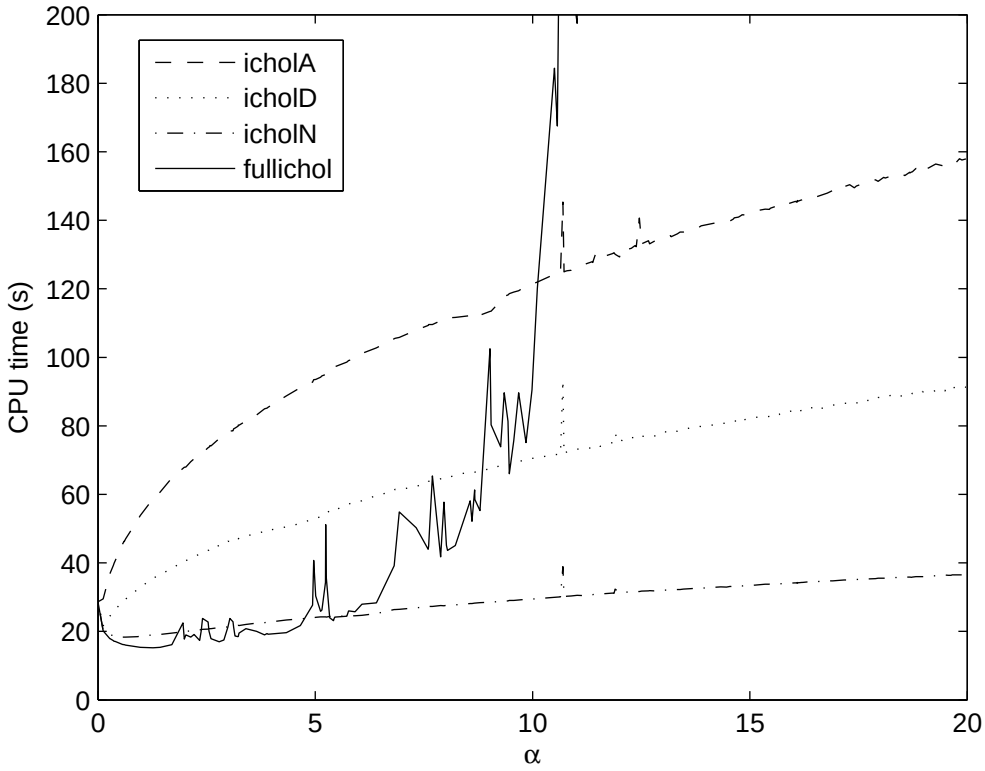


Figura 9.4: Campo de Viento de 100.643 ecuaciones: *Convergencia del Gradiente Conjugado Precondicionado, con varios Precondicionadores ICHOL, para diferentes valores del parámetro α, aleatoriamente calculados.*

aleatoriamente elegidos, dentro del intervalo de 0 a 20, mediante cuatro gráficas correspondientes a los resultados conseguidos con los cuatro distintos precondicionadores ICHOL, ya citados.

La Figura 9.5 recoge , así mismo, los tiempos de computación,en segundos, pero, para una población inicial de valores del parámetro, formada por un conjunto de valores siguiendo una distribución normal dentro del intervalo de 0 a 100. Las cuatro gráficas que aparecen se corresponden con los distintos resultados alcanzados utilizando los cuatro precondicionadores ya indicados.

9.2.4. ANÁLISIS DE RESULTADOS

Como puede comprobarse para todos los casos estudiados, recogidos en las figuras de la 9.2 a la 9.5, el precondicionador $ICHOL_N$, presentado en esta tesis, es el que conduce a la mejor convergencia.

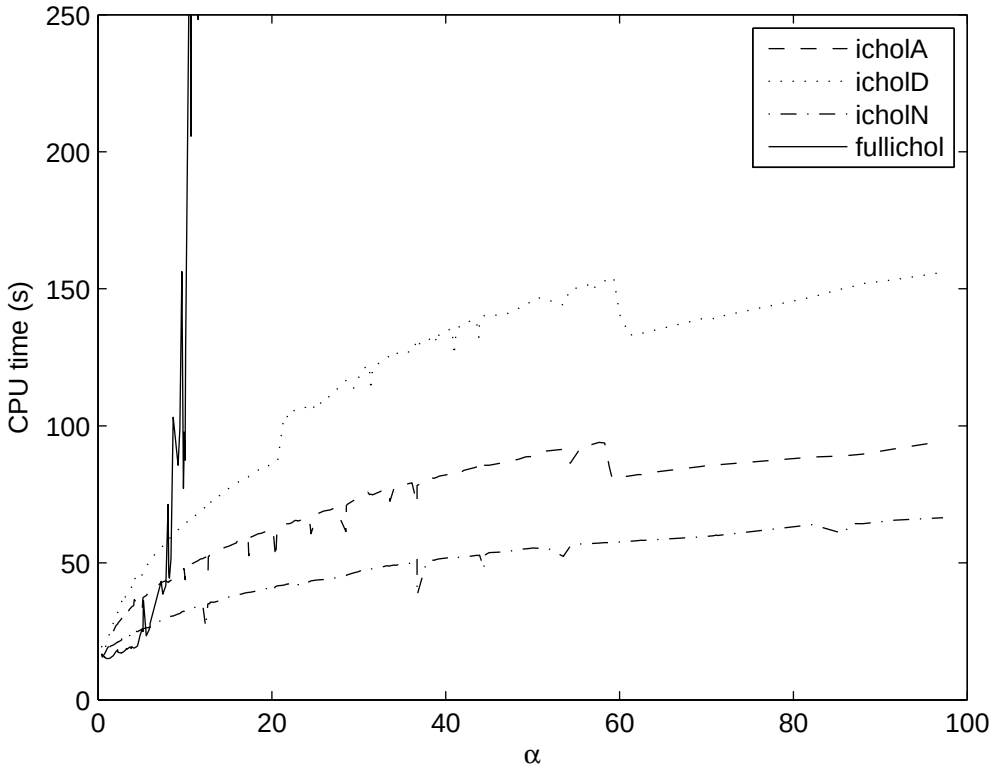


Figura 9.5: Campo de Viento de 100.643 ecuaciones: *Convergencia del Gradiente Conjugado Precondicionado, con varios Precondicionadores ICHOL, para diferentes valores del parámetro α, elegidos usando una distribución normal*

Por lo que al precondicionador $Full - ICHOL$ se refiere, si bien, tiene un mejor comportamiento para pequeños valores del parámetro, $\alpha(\varepsilon)$, resulta que a medida que este crece su convergencia empeora rápidamente, incluso llegando a alcanzar las 10.000 iteraciones sin conseguirla. Esta situación puede explicarse por el hecho de que a la hora de proceder a una Factorización Incompleta de Cholesky de la matriz Simétrica Definida Positiva, $(M + \varepsilon\, N)$, en grandes sistemas y para valores de $\varepsilon \geq 10$, no siempre puede conservarse la Positividad; por lo que puede concluirse que esta forma de precondicionar no es la más adecuada, para el uso del método del Gradiente Conjugado Precondicionado, en los sistemas lineales que aparecen en la modelización de campos de viento.

Capítulo 10

CONCLUSIONES Y LINEAS FUTURAS

10.1. CONCLUSIONES

En esta tesis se han presentado los diversos tipos de modelización de Campos de Viento, se ha comentado la importancia de poder disponer de un buen Modelo y, asimismo, se ha expuesto la metodología a seguir para construir uno de los modelos más fiables: el Modelo de Masa Consistente.

Se ha presentado la formulación de este tipo de Campos, utilizando técnicas de Cálculo Variacional, nos conduce a una Ecuación Diferencial en Derivadas Parciales, para cuya resolución se propone recurrir al Cálculo Numérico, mediante un proceso de Discretización, utilizando el Método de los Elementos Finitos con un mallado tetraédrico y con técnicas adaptativas en 3D.

Se expone como todo ello desemboca en un Sistema Lineal de Ecuaciones cuya matriz de coeficientes resulta expresada en función de un parámetro, o sea, una matriz variable, A_ε, y se comentan los métodos más adecuados para resolver este tipo de Sistemas.

Se describen los distintos Métodos Iterativos para su resolución, basados en los subespacios de Krylov, así como, las técnicas más adecuadas de Precondicionamiento y Reordenación, para conseguir la mejor eficacia de los mismos

Teniendo en cuenta que las Matrices Variables, A_ε, que se originan en la modelización de Campos de Viento de Masa Consistente, son Simétricas Definidas

Positivas, SDP, se justifica el considerar como mejor método iterativo para su resolución, por su probada eficacia, el Gradiente Conjugado Precondicionado, GCP, lo que conduce a tener que afrontar, como novedoso, el Precondicionamiento de Matrices Variables.

Dado que para cada valor del parámetro, ε, se origina una matriz diferente, el Precondicionamiento de las mismas debe conseguirse de forma rápida y eficaz, para una amplia gama de valores de su parámetro.

Se propone como estrategia, para conseguirlo, la construcción de un único Precondicionador, fácilmente adaptable a cada valor diferente del parámetro, o sea un Precondicionador Variable . Esto se plantea a través de dos modelos de Precondionamiento diferentes: uno Explícito, construyendo Inversas Aproximadas (SAINV), y otro Implícito, fundamentado en la Factorización Incompleta de Cholesky (ICHOL).

La amplia gama de experimentos numéricos realizados, con ambos tipos de Precondicionadores, nos permite obtener las siguientes conclusiones:

Por lo que al Precondicionamiento SAINV se refiere, puede afirmarse que para pequeños valores del parámetro, ε, no parece rentable la construcción de un Precondicionador Variable, bastaría con disponer de un único Precondicionador, elaborado a partir de un valor inicial de ε, al que se ha denominado SAINV(A_{ε_0}). Sin embargo, para valores de $\varepsilon \geq 1$, los llamados SAINV_{11}, SAINV_{12} SAINV_{21} presentan notables ventajas sobre el anterior y obviamente resultan más económicos que el hecho de construir un Precondicionador diferente para cada valor distinto del parámetro (nombrado como Full-SAINV). Conviene destacar que, entre todos ellos, el que proporciona los mejores resultados es el SAINV_{11}.

En lo que respecta a los Precondicionadores ICHOL, ocurre algo similar. En general, para pequeños valores de ε, bastaría con construir un único Precondicionador, para un determinado valor inicial del parámetro, el denominado ICHOL(A_{ε_0}); pero, para valores altos de ε, el ICHOL_D y el ICHOL_N tienen el mejor comportamiento. En especial el ICHOL_N presenta los mejores resultados de todos a partir de valores de $\varepsilon \geq 10^2$, para los que ni el construir un Precondicionador diferente para cada $\varepsilon(\alpha)$ (Full-ICHOL), daría resultado, puesto que así no se alcanza la convergencia. Esto se debe a que, para valores altos de ε, al realizar la factorización

incompleta de la matriz A_ε, se pierde su positividad, con lo cual la aplicación del Gradiente Conjugado resulta inestable. Sin embargo esto no ocurre con los nuevos ICHOL, el D y el N.

En los alrededores del valor unitario del parámetro, los experimentos realizados no permiten obtener una conclusión definitiva acerca de cual sería la mejor estrategia. Probablemente la re-computarización para cada valor de ε sea la elección más fiable.

Los Precondicionadores basados en al algoritmo SAINV no son tan eficientes como los que se fundamentan en la Factorización Incompleta de Cholesky, diferencia que se acentúa, a favor de estos últimos, a medida que crece el número de ecuaciones del Sistema. Por ello, puede concluirse que, de todos los experimentados, el Precondicionador denominado ICHOL_N es el que conduce a los mejores resultados.

Teniendo en cuenta lo anterior, también puede afirmarse, que el Precondicionador ICHOL_N constituye la mejor opción, a la hora de optar por seleccionar, de forma automática, el valor óptimo de un parámetro, para un Campo de Viento determinado, puesto que, hace posible utilizar para ello la potente herramienta de los Algoritmos Genéticos, al permitir afrontar con rapidez la gran cantidad de veces que se debe resolver el Sistema, para una amplia gama de valores del parámetro ε.

Por lo que respecta a la influencia de la Reordenación, previa al Precondicionamiento, podemos concluir que, para grandes sistemas de ecuaciones, un orden adecuado mejora la eficacia del método del Gradiente Conjugado Precondicionado, ya que con ello se producen Precondicionadores con mejores cualidades que permiten reducir el número de pasos para alcanzar la convergencia. Por tanto, con una adecuada Reordenación puede mejorarse la eficacia de los Precondicionamientos tanto con Inversas Aproximadas, como con Factorizaciones Incompletas de Cholesky.

Analizando conjuntamente, no sólo el nuevo coste de las iteraciones una vez Reordenado el Sistema, sino añadiendo también los tiempos requeridos para su implantación, se comprueba que el algoritmo de Reordenación más efectivo es el Cuthill-McKee Inverso y en particular asociado con la Factorización Incompleta

de Cholesky.

Sin embargo el algoritmo Multicoloring no aporta ninguna mejora a la ejecución de los métodos iterativos precondicionados.

Queda pues patente, a modo de resumen, que el uso de los Precondicionadores basados en la Factorización Incompleta de Cholesky es una herramienta eficaz para mejorar la convergencia del algoritmo del Gradiente Conjugado Precondicionado, en el caso de los Sistemas Lineales de Ecuaciones Variables, con matrices Simétricas Definidas Positivas; en especial el que se ha denominado como ICHOL$_N$, debido a su inferior coste computacional. Al menos hay una amplia gama de valores de ε para los cuales con dichos Precondicionadores se consiguen convergencias más rápidas, que con el uso de los Precondicionadores obtenidos explícitamente con la Aproximada Inversa (SAINV) y, por supuesto, mejorando los resultados que se puedan conseguir con un re-computarización completa para cada valor distinto del parámetro.

10.2. LINEAS FUTUTRAS

Con este trabajo se abren varias líneas futuras que admiten ser estudiadas en profundidad:

En primer lugar, consideramos interesante comprobar el comportamiento del algoritmo del Gradiente Conjugado Precondicionado utilizando también Precondicionadores Variables, tipos SAINV e ICHOL, similares a los aquí propuestos, pero aplicándolos sobre Sistemas Lineales de Matrices Variables Simétricas originados en la resolución de otros tipos de problemas diferentes a la Modelización de Campos de Viento.

Por otro lado, también puede ser de interés, estudiar los resultados a conseguir con dicho algoritmo al aplicarlo a Matrices Variables Simétricas en general, pero utilizando Precondicionadores Variables distintos de los SAINV e ICHOL aquí descritos.

Otras posibles líneas serían las que surjan al estudiar la eficacia de nuevos Precondicionadores Variables, distintos a los SAINV e ICHOL, apropiados para Matrices Variables No Simétricas, que se originen al abordar modelizaciones diferentes a las de los Campos de Viento y tener que recurrir a algoritmos distintos al Gradiente Conjugado, pero basados en los Subespacios de Krylov, como pueden ser otros métodos de ortogonalización, como el GMRES o métodos de biortogonalización, como el Bi-CGSTAB y los QMR, TFQMR y QMRGCSTAB.

Finalmente, debe señalarse, que incluso será interesante ampliar los estudios indicados utilizando otras Técnicas de Reordenación, no solo basadas exclusivamente en la posición de los elementos de la matriz, sino también, que tuviesen en cuenta la influencia de los valores numéricos de dichos elementos.

Bibliografía

[1] L. Adams. m-Step preconditioned conjugate gradient methods. *SIAM J. Sci. Stat. Comput.* **6,2**, 453–463 (1985).

[2] P. Almeida. "Resolución directa de sistemas sparse por grafos." Tesis Doctoral, Universidad de Las Palmas de Gran Canaria (1989).

[3] W.E. Arnoldi. The Principle of Minimized Iteration in the Solution of the Matrix Eingenvalue Problem. *Quart. Appl. Math.* **9**, 17–29 (1951).

[4] S.F. Ashby, T.A. Manteuffel y P.E. Saylor. A taxonomy for conjugate gradient methods. *SIAM J. Numer. Anal.* **27**, 1542–1568 (1990).

[5] O. Axelsson. A Restarted Version of a Generalized Preconditioned Conjugate Gradient Method. *Comunications in Applied Numerical Methods* **4**, 521–530 (1988).

[6] O. Axelsson. "Iterative Solution Methods." Cambridge University Press (1996).

[7] A.F. de Baas. "Modelling of Atmospheric Flow Fields.", capítulo Scaling Parameters and their Estimation, páginas 87–102. World Sci. Singapore (1996).

[8] J.C. Barnard, H.L. Wegley y T.R. Hiester. Improving the performance of mass consistent numerical models using optimization. *J. Climate. Appl. Meteorol.* **26**, 675–686 (1987).

[9] R. Barret, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine y H.A. Van der Vorst.

"Templates for the solution of linear systems: Building Blokcs for Iterative Methods." SIAM, Philadelphia (1994).

[10] M. Benzi. Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics* **182**, 418–477 (2002).

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

[12] M. Benzi, J.K. Cullum y M. Tuma. Robust approximate inverse preconditioning for the conjugate gradient mathod. *SIAM J. Sci. Comput.* **22**, 1318–1332 (2000).

[13] R. Boubel, D. Fox, D. Turner y A. Stern. "Fundamentals of Air Pollution". Academic Press, San Diego (1994).

[14] J.A. Businger y S.P.S. Arya. Heights of the mixed layer in the stably stratified planetary boundary layer. *Adv. Geophys* **18A**, 73–92 (1974).

[15] T.F. Chan, E. Gallopoulos, V. Simonsini, T. Szeto y C.H. Tong. A Quasi-Minimal residual variant of the BI-CGSTAB algorithm for nonsymmetric systems. *SIAM J. Sci. Comput.* **15,2**, 338–347 (1994).

[16] C. Conde y G. Winter. "Métodos y algoritmos básicos de álgebra numérica." Editorial Reverté, Barcelona (1990).

[17] E. H. Cuthill y J.M. Mckee. Reducing the Bandwidth of Sparse Symmetric Matrices. En "Proc. 24th National Conference of the Association for Computing Machinery", páginas 157–172. Brondon Press, New Jersey, U.S.A. (1969).

[18] C.G. Davis, SS. Bunker y J.P. Mutschlecner. Atmospheric Transport Models for Complex Terrain. *J. Climate. Appl. Meteorol.* **23(2)**, 235–238 (1984).

[19] M. Dikerson. A mass-consistent atmosferic flux model for regions with complex terrain. *J. Appl. Meteor.* **17**, 241–253 (1978).

[20] Q.V. Dinh, V. Mantel, J. Periaux y B. Stoufslet. "Contribution to problems T4 and T6 finit element GMRES and conjugate gradient solvers." Informe Técnico. Dassault Aviation. (1993).

[21] S. Douglas y R. Kessler. "User's guide to the Diagnostic Wind Model (Version 1.0)". System Applications, Inc., San Rafael. California (2009).

[22] L.C. Dutto. The Effect of Ordering on Preconditioned GMRES Algorithm. *Int. Jour. Num, Meth. Eng.* **36**, 457–497 (1993).

[23] L. Elsgoltz. "Ecuaciones Diferenciales y Cálculo Variacional". Editorial MIR. Moscú (1969).

[24] J.M. Escobar y R. Montenegro. Several aspects of three-dimensional Delaunay triangulation. *Advances in Engineering Software* **1/2(27)**, 27–39 (1996).

[25] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero y J.M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput. Methods Appl. Mech. Engrg.* **192**, 2775–2787 (2003).

[26] L. Ferragut, R. Montenegro y A. Plaza. Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems. *Comm. Num. Meth. Eng.* **10**, 403–412 (1994).

[27] R. Fletcher. Conjugate Gradient Methods for Indefite Systems. *Lectures Notes in Math.* **506**, 73–89 (1976).

[28] E. Flórez, M.D. Garcia, A. Suárez y H. Sarmiento. The Effect of Ordering on the Convergence of the Conjugate Gradient Method for Solving Preconditioned Shifted Linear Systems. En B.H.V. Topping, G. Montero y R. Montenegro, editores, "Proceeding of The Fifth International Conference on Engineering Computational Techonology. Las Palmas de G. C.", páginas 191–192. Civil-Comp Press, Stirlingshire, U.K. (2006).

[29] E. Flórez, H. Sarmiento, M.D. Garcia, A. Suárez y G. Montero. Incomplete factorisation for preconditioning shifted linear systems arising from a parameter estimation problem in wind modelling. En B.H.V. Topping, editor, "Proceedings of the Sixth Int. Conference on Engineering Computational Techonology. Atenas". Civil-Comp Press, Stirlingshire, U.K. (2008).

[30] E. Flórez Vázquez. "Construcción de inversas aproximadas tipo "sparse"basada en la proyección ortogonal de Frobenius para el precondicionamiento de sistemas de ecuaciones no simétricos." Tesis Doctoral, Universidad de Las Palmas de G. C. (2003).

[31] R.W. Freund. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comput.* **14**, 470–482 (1993).

[32] R.W. Freund y N.M. Nachtigal. Qmr: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Math.* **60**, 315–339 (1991).

[33] R.W. Freund y N.M. Nachtigal. An implementation of the QMR method based on coupled two-term recurrences. *SIAM J. Sci. Comp.* **15,2**, 313–337 (1994).

[34] M. Galán. "Avances en el Método de Residuo Mínimo Generalizado (algoritmo GMRES), su Desarrollo en ANSI-C con Algoritmos de Paralelización y Vectorización, y sus Aplicaciones al Método de los Elementos Finitos." Tesis Doctoral, Universidad de Las Palmas de Gran Canaria (1994).

[35] M. Galán, G. Montero y G. Winter. A direct solver for the least square problem arising from GMRES(k). *Com. Num. Meth. Eng.* **10**, 743–749 (1994).

[36] M.D. García, E. Flórez, A. Suárez, L. González y G. Montero. New implementation of QMR-type algorithms. *Computers and Structures* **83**, 2414–2422 (2005).

[37] M.D. García León. "Estrategias para la resolución de grandes sistemas de ecuaciones lineales. Métodos de Cuasi-Mínimo Residuo Modificados." Tesis Doctoral, Universidad de Las Palmas de G. C. (2003).

[38] P. Geai. "Methode d'interpolation et de reconstitution tridimensionelle d'un champ de vent: le code d'analyse objetive MINERVE". InformeTécnico. Electricitè de France (1985).

[39] P. Geai. "Reconstitution tridimensionnelle d'un champ de vent dans un domaine a'topographie complexe a partir de meseures in situ." InformeTécnico. DER/HE/34-87.05, EDF, Chatou. France (1987).

[40] I.M. Gelfand y S.V. Fomin. "Calculus of Variations". Prentice-Hall, INC (1963).

[41] A. George. Computer Implementation of the Finite Element Method. *Report Stan CS* páginas 71–208 (1971).

[42] A. George y J.W. Liu. The Evolution of the Minimum Degree Ordering Algorithms. *SIAM Rev.* **31**, 1–19 (1989).

[43] G.H. Golub y G.A. Meurant. "Résolution numérique des grands systèmes linéaires." Editions Eyrolles, París. (1983).

[44] J.M. González-Yuste. "Un Algoritmo de Refinamiento/Derefinamiento Local par Mallas de Tetraedros." Tesis Doctoral, Universidad de Las Palmas de Gran Canaria (2004).

[45] J.M. González-Yuste, R. Montenegro, J. Escobar, G. Montero y E. Rodríguez. Local refinement of 3-D triangulations using object-oriented methods. *Adv. in Eng. Softw.* (2003).

[46] A. Greenbaum. "Iterative Methods for Solving Linear Systems". SIAM, Philadelphia (1997).

[47] M. Grote y H. Simon. "Parallel Preconditioning and Approximate Inverses on the Connection Machine." Informe Técnico. NASA Contract No. NAS2-12961 (1986).

[48] I. Gustafsson. A class of first order factorization methods. *BIT* **18**, 142–156 (1978).

[49] W. Hackbusch. "Iterative Solution of Large Sparse Systems of Equations". Applied Mathematical Sciences 95, Springer-Verlag, New York. (1994).

[50] M.R. Hestenes y E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Jour. Res. Nat. Bur. Sta.* **49,6**, 409–436 (1952).

[51] P. Jackson y J. Hunt. Turbulent wind flow over a low hill. *Quart. J. Roy. Meteorol.* **101**, 833–851 (1975).

[52] A. Jennings y G.M. Malik. The solution of sparse linear equations by conjugate gradient method. *Int.Jour. Num. Meth. Eng.* **12**, 141–158 (1978).

[53] D. King y S. Bunker. Applications of atmosferic transport models for complex terrain. *J. Climate and Appl. Meteor.* **23**, 239 (1984).

[54] T. Kitada, A. Kaki, H. Veda y Peters LK. Estimation of vertical air motion from limited horizontal wind data - a numerical experiment. *Atmos. Environ* **17**, 2181–2192 (1983).

[55] D. Lalas. Modelling of the wind flow over Crete for wind energy estimation. En "EUROMECH. 173, Delphi. Greece" (1983).

[56] D. Lalas. Wind energy estimation and siting in complex terrain. *Int. J. Solar Energy* **3**, 43–71 (1985).

[57] D.P. Lalas y C. Ratto. "Modelling of Atmospheric Flow Fields". World Scientific Publishing, Singapore. (1996).

[58] D.P. Lalas, M. Trombou y M. Petrakis. Comparison of the performance of some numerical wind energy siting codes in rough terrain. En "European Community Wind Energy Conference, Herning Denmark." (1988).

[59] C. Lanczos. Solution of Systems of Linear Equations by Minimized Iterations. *Jour. Res. Nat. Bur. Sta.* **49,1** (1952).

[60] P. Lascaux y R. Théodor. "Analyse Numérique matricielle appliquée a l'art de l'ingénieur.", tomo 1,2. Edit. Masson, París. (1987).

[61] R. Löhner y J.D. Baum. Adaptive h-refinement on 3D unstructured grids for transient problems. *Int. J. Num. Meth. Fluids* **14**, 1407–1419 (1992).

[62] A. Liu y B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *Mathematics of Computations* **65(215)**, 1183–1200 (1996).

[63] G. Martin. "Méthodes de préconditionnement par factorisation incomplète." Mémoire de Maitrise, Université Laval, Quebec. Canadá (1991).

[64] G.J. McRae, W.R. Goodin y J.H. Seinfeld. Development of a second generation mathematical model for urban air pollution. *Atm. Env.* **16,4**, 679–696 (1982).

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

[66] R. Montenegro, J.M. Escobar, E. Rodríguez, G. Montero y J.M. González-Yuste. Improved objetive functions for tetrahedral mesh optimization. *Lec. Notes in Comp. Sci.* **2657**, 568–578 (2003).

[67] R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez y J.M. González-Yuste. Tetrahedral mesh generation for environmental problems over complex terrains. *Lecture Notes in Computer Science* **232**, 335–344 (2002).

[68] R. Montenegro, A. Plaza, L. Ferragut y I. Asensio. Application of a nonlinear evolution model to fire propagation. *Nonlinear Analysis, Th., Meth. and App.* **5(30)**, 2873–2882 (1997).

[69] G. Montero. "Aplicación de esquemas elemento a elemento de resolución de sistemas de ecuaciones asociados a métodos de elementos finitos adaptativos." Tesis Doctoral, Universidad de Las Palmas de Gran Canaria (1989).

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

[71] G. Montero, R. Montenegro, J.M. Escobar y E. Rodríguez. Generación automática de mallas de tetraedros adaptadas a orografías irregulares. *Rev. Int. Mét. Num. Cálc. Dis. Ing.* **19(2)**, 127–144 (2003).

[72] G. Montero, R. Montenegro, J.M. Escobar, E. Rodríguez y J.M. González-Yuste. Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model. *Adv. in Eng. Softw.* (2004).

[73] G. Montero, R. Montenegro, J.M. Escobar, E. Rodríguez y J.M. González-Yuste. Velocity field modelling for pollutant plume using 3-D adaptative finite element method. *Lect. Notes in Comp. Sci.* (2004).

[74] G. Montero y A. Suárez. "Precondicionamiento de matrices en la resolución de sistemas de ecuaciones lineales. Aplicaciones en métodos tipo-gradiente." Encuentro de Análisis matricial y aplicaciones, Vitoria (1994).

[75] G. Montero, A. Suárez, E. Flórez y M.D. García. Preconditioning shifter linear sistems arising in a wind model. En B.H.V. Topping, editor, "Proceedings of the Tenth Inter. Conf. on Civil, Structural and Enviromental Engineering Computing. Roma". Civil-Comp Press, Stirlingshire, U.K. (2005).

[76] N. Moussiopoulus, Th. Flassak y G. Knittel. A Refined Diagnostic Wind Model. *Environ. Software.* **3**, 85–94 (1988).

[77] B.Ñour-Omid y B.N. Parlett. Element preconditioning using splitting techniques. *SIAM J. Sci. Stat. Comput.* **6,3**, 761–771 (1985).

[78] C.P. Paige y M.A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software* **8,1**, 43–71 (1982).

[79] I. Palomino y F. Martín. A simple method for spatial interpolation of the wind in complex terrain. *J. Appl. Meteor.* **34**, 1678–1693 (1995).

[80] H.A. Panofsky y J.A. Dutton. "Atmosfheric Turbulence, Models and Methods for Enginnering Applications." John Wiley, New york. (1984).

[81] W. Pennel. "An evaluation of the role of numerical wind field models in wind turbine siting." InformeTécnico. Batelle Memorial Institute, Pacific Northwest Laboratory, Richland, Washington. (1983).

[82] G. Phillips y R. Traci. "A preliminary user guide for the NOABL objetive analisys code." Informe Técnico. Science Applications Inc. (1978).

[83] A. Plaza, R. Montenegro y L. Ferragut. An improved derefinement algorithm of nested meshes. *Advances in Engineering Software* **1/2(27)**, 51–57 (1996).

[84] K.V.G. Prakhya. Some Conjugate Gradient methods for symmetric and nonsymmteric systems. *Com. Appl. Num. Meth.* **4**, 531–539 (1988).

[85] C.F. Ratto. "Modelling of Atmosfeheric Flow Fields.", capítulo The AIOLOS and WINDS Codes, páginas 421–431. Word Sci. Singapore (1996).

[86] C.F. Ratto, R. Festa, O.Ñicora, R. Mosiello, A. Ricci, D.P. Lalas y O.A. Frumento. Wind field numerical simulation a new user friendly code. En W. Palz, editor, "Resource assessment. European Community Wind Energy Conference. Madrid". H.S. Stephens and Associates (1990).

[87] M.C. Rivara. A grid generator based on 4-triangles conforming. mesh-refinement algorithms. *Int. J. Numer. Methods Engrg.* **24**, 1343–1354 (1987).

[88] D.J. Rodríguez. "User's Guide to the MATHEW/ADPIC Models." UCID-20415, Lawrence Livermore National Laboratory, Atmosferic and Goephysical Sciencies Division. University of California (1985).

[89] E. Rodríguez Barrera. "Modelización y simulación numérica de campos de viento mediante elementos finitos adaptativos en 3-D." Tesis Doctoral, Universidad de Las Palmas de G. C. (2004).

[90] E. Rodriguez, G. Montero, R. Montenegro, J.M. Escobar y J.M. González-Yuste. Parameter Estimation in a Three-dimensional Wind Field Model Using Genetic Algorithms. *Lect. Notes in Computational Science.* **2329**, 950–959 (2002).

[91] D.G. Ross, I.N. Smith, P.C. Manins y D.G. Fox. Diagnostic Wind Field Modeling for Complex Terrain: Model Development and Testing. *J. Appl. Meteorol.* **27**, 785–796 (1988).

[92] Y. Saad. "Highly Hparallel preconditioners for general sparse matrices." Informe Técnico. Army High Performance Computing Research Center, Minneapolis. MN (1992).

[93] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Statist. Comput.* **14,2**, 461–469 (1993).

[94] Y. Saad. "Iterative methods for sparse linear systems." PWS Publishing Company, Boston (1996).

[95] Y. Saad y M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Statist. Comput.* **7**, 856–869 (1986).

[96] H. Sarmiento, A. Suárez, E. Flórez, M.D. Garcia y G. Montero. Implicit and explicit preconditioners for shifted linear systems. Pendiente de publicación en Advances in Engineering Software (2008).

[97] H. Sarmiento, A. Suárez y M.D. Garcia. Precondicionamiento de Sistemas de Ecuaciones Lineales Variables basado en Inversas Aproximadas Factorizadas para Modelos de Viento. En Sociedad Española de Matemática Aplicada, editor, "Resumenes del XIX Congreso de Ecuaciones Diferenciales y Aplicaciones (Cedya). Madrid.", página 215. Universidad Carlos III. Madrid. (2005).

[98] H. Sarmiento, A. Suárez y G. Montero. Incomplete Factorization for Preconditioning Shifted Linear Systems. En B.H.V. Topping, G. Montero y R. Montenegro, editores, "Proceeding of The Fifth International

Conference on Engineering Computational Techonology. Las Palmas de G. C.", páginas 187–188. Civil-Comp Press, Stirlingshire, U.K. (2006).

[99] Y. Sasaki. Some basic formalism in numerical variacional analysis. *Mon. Wea. Rew.* **98**, 875–883 (1970).

[100] J. Seinfeld. "Atmospheric Chemistry and Physics of Air Pollution". John Wiley and Sons, New York. (1998).

[101] A.M. Sempreviva. "Modelling of Atmosfeheric Flow Fields.", capítulo Roughness Changes: Response of neutral boundary layers, páginas 213–245. Word Sci. Singapore (1996).

[102] J.N. Shadid y R.S. Tuminaro. A comparison of preconditioned nonsymmetric Krylov methods on a large-scale mind machine. *SIAM J. Sci. Statist. Comput.* **15,2**, 440–459 (1994).

[103] C.A. Sherman. A mass-consistent model for wind fields over complex terrain. *J. Appl. Meteorol.* **17**, 312–319 (1978).

[104] P. Sonneveld. CGS: A fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* **10**, 36–52 (1989).

[105] A. Suárez, E. Rodríguez, G. Montero, M.D. García y E. Flórez. Precondicionamiento de Sistemas de Ecuaciones Variables. En "Resumenes del Congreso de Métodos Numéricos en Ingeniería. Granada" (2005).

[106] A. Suárez, H. Sarmiento, M.D. Garcia, E. Flórez y G. Montero. Precondicionamiento y Reordenación en Sistemas de Ecuaciones Lineales Variables. En "Resumenes del Congreso de Métodos Numéricos em Engenharia. Oporto" (2007).

[107] A. Suárez Sarmiento. "Contribución a Algoritmos de Biortogonalización para la Resolución de Sistemas de Ecuaciones Lineales." Tesis Doctoral, Universidad de Las Palmas de G. C. (1995).

[108] M. Tombrou y D.P. Lalas. A telescoping procedure for local wind energy potencial assessment. En W. Palz, editor, "Resource assessment.

European Community Wind Energy Conference. Madrid". H.S. Stephens and Associates (1990).

[109] L.N. Trefethen y D. Bau. "Numerical Linear Algebra". SIAM, Philadelphia (1997).

[110] I. Troen y E. Petersen. "European Wind Atlas". Riso National Laboratory (1989).

[111] M. Tsun-zee. Modified Lanczos method for solving large sparse linear systems. *Com. Num. Meth. Eng.* **9**, 67–79 (1993).

[112] A. Van der Sluis y H.A. Van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.* **48**, 543–560 (1986).

[113] H.A. Van der Vorst. The convergence behaviour of preconditioned CG and CG-S in the presence of rounding errors. *Lectures Notes in Math.* **1457**, 126–136 (1990).

[114] H.A. Van der Vorst y C. Vuik. "GMRESR: A family of nested GMRES methods." Informe Técnico, 91-80. Delft University of Technology, Mathematics and Informatics, Delft. The Netherlands. (1991).

[115] G. Winter, J. Betancor y G. Montero. "Ocean Circullation and Pollution Control. A mathematical and Numerical Investigation", capítulo 3D Simulation in the Lower Troposphere: Wind Field Adjustment to Observational Data and Dispersion of Air Pollutants from Combustion of Sulfur-Containing Fuel. Springer-Verlag (2004).

[116] G. Winter, G. Montero, L. Ferragut y R. Montenegro. Adaptive strategies using standard and mixed finite elements for wind field adjustment. *Solar Energy* **54(1)**, 49–56 (1995).

[117] P. Zannetti. "Air Pollution Modeling". Comput. Mech. Publ., Boston. (1990).

[118] L. Zhou y H.F. Walker. Residual smoothing techniques for iterative methods. *SIAM J. Sci. Comput.* **15,2**, 297–312 (1994).

[119] O.C. Zienkievicz. "El método de los elementos finitos." Editorial Reverté, S.A., Barcelona. (1980).