

Simulación numérica de campos de viento: implementación de un modelo tridimensional de masa consistente

G. Montero, R. Montenegro, J.M. Escobar y E. Rodríguez

Resumen

En la simulación numérica de campos de viento existen tres aspectos principales que caracterizan a un modelo y que, en definitiva, definirán la eficiencia del mismo. En primer lugar, debemos ser capaces de generar mallas tridimensionales que se adapten a dominios definidos sobre una orografía irregular. Asimismo, necesitamos más concentración de puntos en las zonas cercanas al terreno, ya que es ahí donde mayor precisión es requerida habitualmente. Para ello hemos desarrollado un generador de mallas de tetraedros que parte de una discretización de la superficie del terreno obtenida mediante un algoritmo de refinamiento-desrefinamiento, a continuación genera una nube de puntos con una función de espaciado vertical y, finalmente, construye una malla de tetraedros utilizando una triangulación de Delaunay [41]. En este proceso se produce eventualmente algún cruce de tetraedros o la construcción de elementos de baja calidad. Es necesario, por tanto, aplicar una técnica de desenredo y de suavizado. Se presenta, aquí, un algoritmo de desenredo y suavizado simultáneo de mallas que transforma elementos no válidos en aptos para la aplicación del método de elementos finitos y mejora de forma muy eficiente la calidad de los tetraedros [16; 39]. En segundo lugar, nuestro modelo deber ser competitivo computacionalmente y orientado a problemas reales. Hemos optado por un modelo del tipo de masa consistente. Estos modelos nos permiten considerar el aire como un fluido incompresible y tienen en cuenta, además, las posibles medidas experimentales de velocidades de viento obtenidas en una red de estaciones meteorológicas y las características que definen la física de la atmósfera [45; 48]. La resolución de este modelo matemático se realiza mediante el método de elementos finitos [60; 47]. Para mejorar la solución en aquellas zonas donde alguna medida del error nos indique que es necesario, se ha desarrollado un algoritmo de refinamiento de tetraedros [28]. Finalmente, todo este proceso de resolución desemboca en un sistema de ecuaciones lineales de la forma $A_\varepsilon x_\varepsilon = b_\varepsilon$ con $A_\varepsilon = M + \varepsilon N$, cuya matriz es simétrica definida positiva (SDP), de dimensión elevada y estructura hueca. Dado que la resolución de este sistema supone una gran parte del tiempo de computación de todo el modelo, es esencial aplicar técnicas muy eficientes que permitan llegar a la solución de tal forma que el modelo sea competitivo. Con las características de la matriz del sistema (SDP, grande y hueca), la elección de método está clara: el algoritmo del Gradiente Conjugado. Sin embargo, esta elección no es suficiente. Es necesario realizar un preconditionamiento del sistema para acelerar la convergencia. Esta mejora de la convergencia se ve acentuada por el efecto de una reordenación previa del sistema sobre el comportamiento

de los preconditionadores [46]. Finalmente, se propone la construcción de un preconditionador dependiente de ε a partir de una aproximada inversa de A_{ε_0} , que se pueda actualizar fácilmente para cada sistema [49].

Para ilustrar el funcionamiento del modelo completo, se ha incluido un experimento numérico localizado en la isla de Gran Canaria. Se estudia para este caso, la generación de la malla, la construcción del modelo de viento y la resolución del sistema de ecuaciones resultante.

1. Generación de mallas para dominios definidos sobre orografías irregulares

1.1. Introducción

En este capítulo nos planteamos crear 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 [73; 45], propagación de fuego [43], contaminación atmosférica [72], 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 [37; 36; 28; 27]. 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, [19] y [55], 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. Los fundamentos de este proceso se resumen en la sección 1.2.

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 que se analiza en la sección 1.5. 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.

Dedicamos la sección 1.6 a la definición de la nube de puntos en el dominio real y al análisis de 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 [18].

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 pun-

tos 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. Los detalles sobre el proceso de triangulación se presentan en la sección 1.7 de este informe, mientras que los relativos al proceso de optimización de la malla se estudian en la sección 1.8.

Finalizamos presentando diversos aspectos computacionales en la sección 1.12.

Por último, se destaca que los detalles presentados se recogen en [42; 40; 39], [44] y [16].

1.2. Refinamiento y desrefinamiento en mallas bidimensionales

El proceso de generación de la malla tridimensional comienza con la determinación de los nodos situados sobre la superficie del terreno. Su distribución debe estar adaptada a las características orográficas con la finalidad de minimizar el número total de nodos necesario. El procedimiento construye inicialmente una secuencia de mallas encajadas $T = \{\tau_1 < \tau_2 < \dots < \tau_m\}$ a partir de una triangulación regular τ_1 de la zona rectangular de estudio, tal que el nivel τ_j se obtiene mediante un refinamiento global del nivel anterior τ_{j-1} aplicando el algoritmo 4-T de Rivara [58]; todos los triángulos del nivel τ_{j-1} se dividen en cuatro subtriángulos mediante la introducción de un nuevo nodo en los centros de sus lados y uniendo el nodo introducido en el lado mayor con el vértice opuesto y los otros dos nuevos nodos. Por tanto, en el nivel de malla τ_j aparecen nuevos nodos, lados y elementos que reciben el nombre de propios del nivel j . El número de niveles m de la secuencia está determinado por el grado de discretización de la digitalización del terreno, es decir, el diámetro de la triangulación τ_m debe ser del orden del paso espacial de la digitalización. De esta forma aseguramos que esta malla regular es capaz de captar toda la información orográfica mediante una interpolación de las cotas reales en los nodos de la malla. Finalmente, definimos una nueva secuencia $T' = \{\tau_1 < \tau'_2 < \dots < \tau'_{m'}\}$, $m' \leq m$, aplicando el algoritmo de desrefinamiento, [19] y [55]. En este paso se introduce como dato el parámetro de desrefinamiento ε que determina la precisión con que se desea aproximar la topografía del terreno. La diferencia en valor absoluto de las cotas resultantes en cualquier punto de la malla $\tau'_{m'}$ y su correspondiente cota real será menor que ε . Asimismo, el algoritmo de desrefinamiento (véase el algoritmo 1.1) utiliza toda la información de la genealogía de elementos y lados definida en la secuencia.

Como condición de desrefinamiento analizamos la diferencia absoluta entre la cota del nodo estudiado y el valor interpolado de las cotas correspondientes a los dos nodos extremos de su lado entorno, es decir, el lado en que ese nodo fue introducido en su punto medio durante el proceso de refinamiento. Si esa diferencia es menor que el parámetro de desrefinamiento ε , entonces el nodo podría ser eliminado, aunque en algunos casos deberá permanecer por razones de conformidad.

Destacamos que la malla bidimensional obtenida puede ser modificada al construir la triangulación de Delaunay en el dominio tridimensional, puesto que lo único que necesitamos y conservamos es la posición de sus nodos. También nos interesa tener presente el nivel en que cada nodo es propio, para proceder a la generación

Algoritmo 1.1 Algoritmo de desrefinamiento.

ENTRADA: Secuencia $T = \{\tau_1 < \tau_2 < \dots < \tau_m\}$.

Para $j = m$ hasta 2 **Hacer** {Bucle en niveles de T }

Para cada nodo propio de τ_j se evalúa la condición de desrefinamiento y se marcan nodos y lados que podrían ser eliminados mediante los vectores de desrefinamiento.

Se asegura la conformidad del nuevo nivel de malla j minimizando la zona desrefinada.

Si algún nodo propio de τ_j permanece **Entonces**

Se definen nuevas conexiones nodales para el nuevo nivel j : τ_j^j .

Se modifican los vectores de genealogía de τ_j^j y de τ_{j-1} .

Si No

El nivel actual j es eliminado de los vectores de estructura.

Se modifican los vectores de genealogía de τ_{j-1} .

Fin Si

Los cambios en la malla se heredan a las mallas siguientes.

Se comprimen los vectores de estructura.

Se obtiene una nueva secuencia de mallas encajadas T^j . Esta secuencia es la entrada en la siguiente iteración del bucle. $T^j = \{\tau_1 < \tau_2 < \dots < \tau_{j-1} < \tau_j^j < \dots < \tau_{m_j}^j\}$.

Fin Para

SALIDA: Secuencia desrefinada $T' = T^2 = \{\tau_1 < \tau_2' < \dots < \tau_{m'}'\}$.

de nodos en el interior del dominio. Este último aspecto se utiliza en las estrategias propuestas.

1.3. Algoritmo de Delaunay en 3-D

La triangulación de Delaunay es un método ampliamente utilizado en la construcción de mallas tanto en 2-D como en 3-D, para la aplicación del método de los elementos finitos por la buena calidad de las mallas resultantes. De hecho, en el caso bidimensional, de todas las triangulaciones de un conjunto de puntos del plano, la de Delaunay es la que hace máximo el mínimo ángulo de cualquier triángulo. Se sabe que el condicionamiento de las matrices asociadas al MEF depende, entre otros aspectos, del mínimo ángulo de todos los triángulos de la triangulación, lo que justifica el uso de la triangulación de Delaunay en el MEF. Sin embargo, no existe una generalización de esta propiedad para dimensión mayor que dos, al menos en términos de alguna medida angular de los símlices. A pesar de ello, los resultados experimentales revelan que las mallas tridimensionales, construidas a partir de la triangulación de Delaunay, dan buenas medidas de calidad cuando la distribución de puntos es suficientemente regular.

Para crear la triangulación de Delaunay existen algoritmos de tipo incremental, como el de Bowyer-Watson [8; 68], que construyen la triangulación de un conjunto X por adición de puntos uno a uno.

Una de las principales dificultades que plantea la utilización de la triangulación de Delaunay para generar mallas es conseguir que la triangulación *respete* la frontera del dominio de definición, esto es, que no haya ningún tetraedro que corte sus super-

ficies. Cuando esto ocurre decimos que la triangulación es conforme con la frontera del dominio. Este problema se resuelve, bien imponiendo que la triangulación contenga todas las aristas que definen el contorno del dominio (*Constrained Delaunay Triangulation*), bien colocando puntos sobre las aristas del dominio en posiciones adecuadas de manera que éstas sean la unión de aristas de la triangulación (*Conforming Delaunay Triangulation*). Hay algoritmos (véase por ejemplo [25; 69]) basados en el intercambio de aristas y caras que consiguen la conformidad con la frontera del dominio. Una vez que se ha definido una malla conforme con la frontera del dominio, una buena estrategia para adaptar la malla atendiendo a la solución numérica, asegurando que en todo momento se mantiene dicha conformidad, es utilizar métodos de refinamiento/desrefinamiento de mallas encajadas. Esta estrategia se ha desarrollado para triangulaciones bidimensionales [19; 54], pero en la actualidad es una línea de investigación abierta en tres dimensiones. La tesis doctoral [27] está siendo desarrollada en esta línea y ya se tienen resultados en el refinamiento [28].

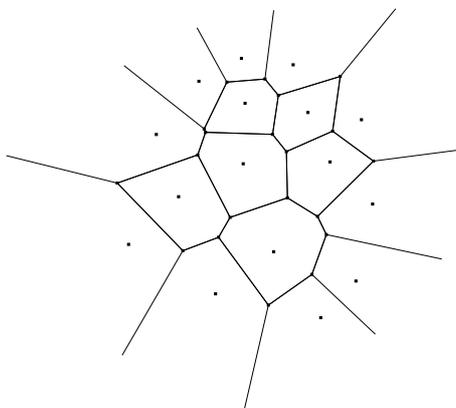
1.4. Aspectos generales

Sea X un conjunto de puntos distintos, no todos coplanarios, del espacio euclídeo tridimensional E^3 . Se define el poliedro de Voronoi, $V(x_i)$, asociado al punto $x_i \in X$ como el conjunto de puntos del espacio E^3 que están más cercanos a x_i que a cualquier otro punto de X ,

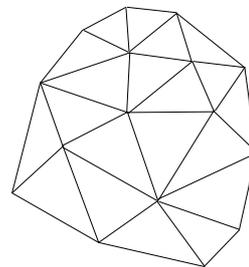
$$V(x_i) = \{x \in E^3 / d(x, x_i) \leq d(x, x_j), \forall x_j \in X, j \neq i\} \quad (1)$$

donde $d(x, y)$ es la distancia euclídea entre los puntos x e y . El poliedro de Voronoi $V(x_i)$ sería un conjunto de puntos no acotado si y solo si x_i se encuentra sobre la frontera de la envolvente convexa de X . Conforme a lo anterior, definimos el diagrama de Voronoi, $V(X)$, como el conjunto de poliedros de Voronoi asociados a los puntos de X . El diagrama de Voronoi así definido cubre todo el espacio E^3 , en el sentido de que cualquier punto de E^3 pertenece, al menos, a un poliedro de Voronoi.

Siempre es posible encontrar subconjuntos R con cuatro o más puntos de X , no todos coplanarios, tal que existe un punto $\nu_R \in E^3$, equidistante a todos los puntos de R que satisface $d(\nu_R, x_k) < d(\nu_R, x_i)$, donde $x_k \in R$ y $\forall x_i \in X - R$.



(a) Diagrama de Voronoi



(b) Triangulación de Delaunay

Figura 1: Diagrama de Voronoi y triangulación de Delaunay de 16 puntos aleatorios..

La envolvente convexa de R recibe el nombre de politopo de Delaunay (poliedro convexo acotado), $D(R)$. A partir de la definición anterior, se deduce que todos los puntos de R están sobre una misma esfera y que no hay ningún punto de X en su interior. Normalmente, todos los politopos de Delaunay contienen solamente cuatro puntos de X , en este caso, los politopos serán tetraedros y se dirá que los puntos de X están en *posición general* [20]. En el caso que el politopo $D(R)$ contenga a más de cuatro puntos de X , se dice que está degenerado.

La triangulación de Delaunay del conjunto de puntos X , $D(X)$, sería el conjunto formado por todos los politopos de Delaunay. Sólo cuando este conjunto esté compuesto por tetraedros, la triangulación de Delaunay será una triangulación propia tridimensional. Sin embargo, si $D(X)$ es una triangulación impropia, siempre será posible descomponer los politopos degenerados en tetraedros.

Se conoce como algoritmos de triangulación incrementales a aquellos que se basan en la adición de puntos uno a uno sobre triangulaciones previas. Se describe a continuación un algoritmo incremental que construye una triangulación propia de Delaunay, basado en el algoritmo de Bowyer-Watson [8; 68] que ha sido modificado por varios autores. Los principales pasos del algoritmo son:

1. Triangulación inicial:

El algoritmo se simplifica si construimos un tetraedro inicial, T_0 , o un prisma descompuesto en tetraedros, que contenga a todos los puntos de X .

2. Adición de x_{i+1} en la triangulación:

Supongamos que ya ha sido construida la triangulación $D(X_i)$, de los primeros i puntos de X . Entonces buscamos el conjunto D_1^i de tetraedros cuyas esferas circunscritas contienen al nuevo punto x_{i+1} :

$$D_1^i = \{T_j \in D(X_i) / x_{i+1} \in B(T_j)\} \quad (2)$$

Sea $C_{fD_1}^i$ el conjunto formado por las caras de la frontera de D_1^i , esto es, caras que sólo pertenecen a un tetraedro de D_1^i . Podemos definir la triangulación $D(X_{i+1})$ que incluye el punto x_{i+1} como:

$$D(X_{i+1}) = (D(X_i) - D_1^i) \cup D_2^i \quad (3)$$

donde $D_2^i = \bigcup_j \{C_j, x_{i+1}\}$ es el conjunto de tetraedros formados por el punto x_{i+1} y la cara genérica C_j de $C_{fD_1}^i$. Si las circunferencias de los tetraedros de D_1^i están evaluadas exactamente, entonces los tetraedros de D_2^i estarán bien contruidos, en el sentido de que las caras de $C_{fD_1}^i$ serán *visibles* con respecto al punto x_{i+1} , esto es lo mismo que decir que D_1^i es en forma de estrella con respecto a x_{i+1} .

Como ejemplo en la figura 2 se muestra la triangulación del conjunto $X_8 = \{x_1, \dots, x_8\}$ (figura 2(a)) y la triangulación que resulta al añadir el punto x_9 (figura 2(d)). El conjunto de caras de la frontera de D_1^8 está resaltado con líneas gruesas en la figura 2(c).

Cuando los puntos del conjunto X no están *claramente* en posición general, los problemas debidos a errores de redondeo cometidos al usar representación de números en coma flotante pueden hacer que no funcione correctamente el algoritmo

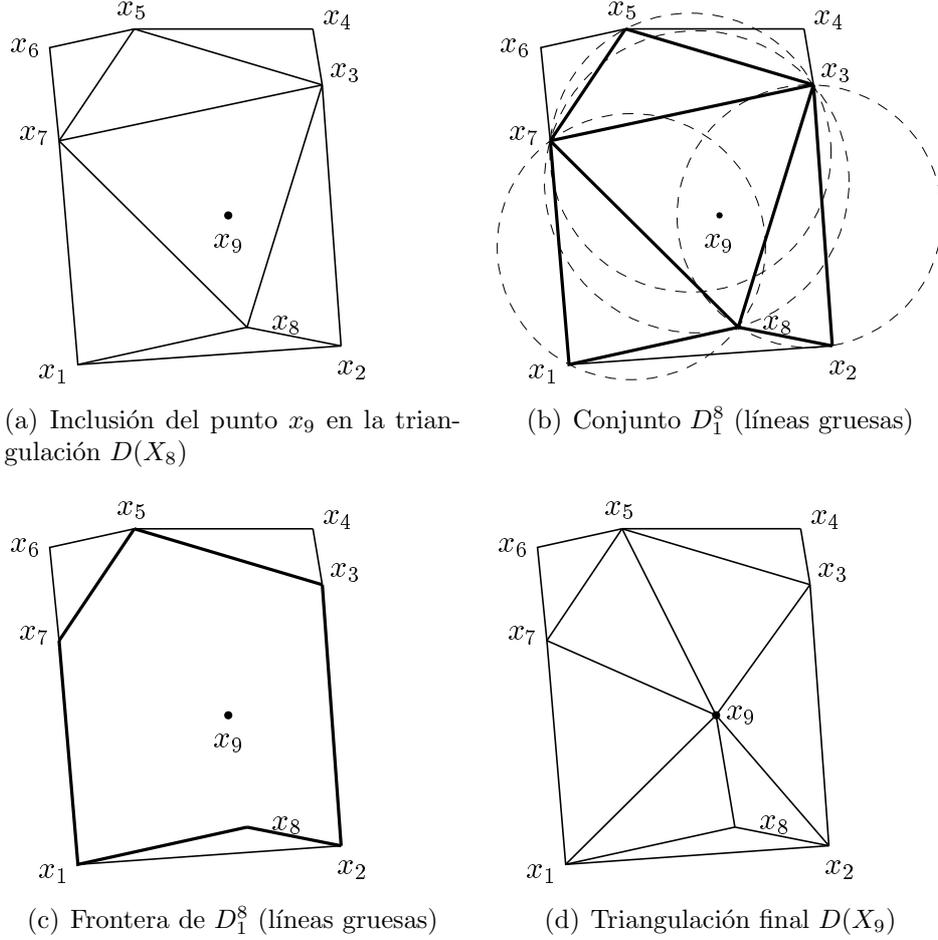


Figura 2: *Triangulación del conjunto $X_8 = \{x_1, \dots, x_8\}$ y proceso de adición de un punto x_9 .*

presentado anteriormente, pudiendo generarse tetraedros planos o cruces entre ellos. Este problema ha sido estudiado y resuelto satisfactoriamente en [17].

El objeto o dominio Ω que deseamos mallar deberá ser definido por puntos distribuidos en su interior y en su superficie. Los puntos deben estar situados de forma que la triangulación del objeto, $T(\Omega)$, sea un subconjunto de la triangulación global, $D(X)$, en la que en el conjunto X se incluyen los puntos que definen el objeto y los puntos que pertenecen al prisma inicial descompuesto en tetraedros. Si se satisface la condición anterior, entonces diremos que la triangulación $T(\Omega)$ es conforme con la frontera de Ω . En este caso, $T(\Omega)$ resulta directamente al eliminar de la triangulación global $D(X)$ los tetraedros que no están incluidos en el objeto.

Como se mencionó anteriormente, la triangulación de Delaunay presenta problemas en lo relativo a la conformidad con la frontera, puesto que crea una malla de la envolvente convexa de los puntos que definen el dominio. En nuestro caso concreto, la conformidad con la frontera es un problema particularmente difícil de tratar debido a la complejidad de las orografías que delimitan el dominio que se pretende mallar. En la sección 1.7, se detalla la técnica que hemos utilizado y que permite el uso de la triangulación de Delaunay sin recurrir a técnicas tan complejas como la triangulación de Delaunay restringida (*Constrained Delaunay Triangulation*) o la triangulación de Delaunay conforme (*Conforming Delaunay Triangulation*), y al

mismo tiempo evita los problemas de conformidad con la frontera del dominio.

1.5. Función de espaciado vertical

Como ya se ha indicado, interesa generar una nube de puntos con mayor densidad en la zona cercana al terreno. Para ello, cada nodo va estar situado atendiendo a una función del tipo,

$$z_i = a i^\alpha + b \quad (4)$$

tal que a medida que aumenta el exponente $\alpha \geq 1$, proporciona una mayor concentración de puntos cerca de la superficie del terreno; z_i es la cota correspondiente al i -ésimo punto insertado, del tal manera que para $i = 0$ se obtiene la cota del terreno y para $i = n$, la cota del último punto introducido que debe coincidir con la altura h del plano superior que delimita el dominio a discretizar. En estas condiciones el número de puntos definidos en la vertical sería $n + 1$ y la función de espaciado vertical (ver figura 3) se puede expresar como

$$z_i = \frac{h - z_0}{n^\alpha} i^\alpha + z_0 \quad ; \quad i = 0, 1, 2, \dots, n \quad (5)$$

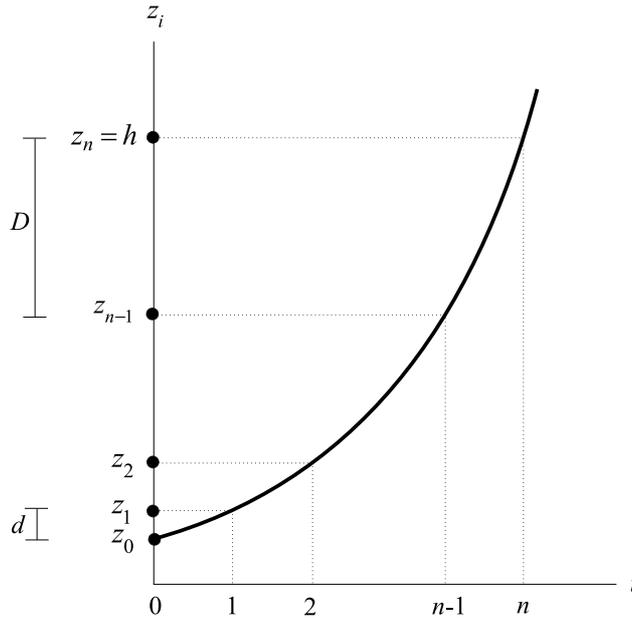


Figura 3: Distribución de $n+1$ puntos sobre el eje de ordenadas mediante la función de espaciado vertical.

En ocasiones conviene expresar la altitud de un punto en función de la del punto anterior, evitando así tener que conservar en memoria el valor de z_0 ,

$$z_i = z_{i-1} + \frac{h - z_{i-1}}{n^\alpha - (i-1)^\alpha} [i^\alpha - (i-1)^\alpha] \quad ; \quad i = 1, 2, \dots, n \quad (6)$$

A partir de las ecuaciones (5) o (6), los puntos quedan perfectamente definidos una vez fijados los valores de α y n . No obstante, también puede ser interesante fijar la distancia del primer punto insertado ($i = 1$) a la superficie del terreno con

el fin de mantener unos parámetros mínimos de calidad en la malla tridimensional que se pretende generar. Esto reduciría el número de grados de libertad a uno, bien sea α o bien n . Consideremos fijado y conocido el valor de esa distancia d tal que $d = z_1 - z_0$; véase figura 3. Sustituyendo en la ecuación (5),

$$d = z_1 - z_0 = \frac{h - z_0}{n^\alpha} \quad (7)$$

Si fijamos α y dejamos libre el valor de n , de (7) se obtiene,

$$n = \left(\frac{h - z_0}{d} \right)^{1/\alpha} \quad (8)$$

No obstante, en la práctica se aproximará el valor de n al número natural más cercano. En cambio, si fijamos el valor de n y dejamos libre α , resulta,

$$\alpha = \frac{\log \frac{h - z_0}{d}}{\log n} \quad (9)$$

En ambos casos, dado uno de los dos parámetros, se calcula el otro mediante las expresiones (8) o (9), respectivamente. De esta forma, la distribución de puntos en la vertical respeta la distancia d entre z_1 y z_0 .

Si además fijamos la distancia entre los dos últimos puntos introducidos, esto es $D = z_n - z_{n-1}$ (ver figura 3), entonces los parámetros α y n quedan perfectamente determinados. Supongamos que α es definido por la ecuación (9). Para $i = n - 1$, la ecuación (5) resulta

$$z_{n-1} = \frac{h - z_0}{n^\alpha} (n - 1)^\alpha + z_0 \quad (10)$$

y por tanto, usando la ecuación (9),

$$\frac{\log (n - 1)}{\log n} = \frac{\log \frac{h - z_0 - D}{d}}{\log \frac{h - z_0}{d}} \quad (11)$$

A partir de las características con que se ha definido la malla buscada *a priori* se puede afirmar que $h - z_0 > D \geq d > 0$. Por tanto, el valor de n estará acotado, $2 \leq n \leq \frac{h - z_0}{d}$, y el de α no podrá ser inferior a 1. Asimismo, para llegar a introducir al menos un punto intermedio entre la superficie del terreno y la frontera superior del dominio, se tiene que verificar que $d + D \leq h - z_0$.

Llamando $k = \frac{\log \frac{h - z_0 - D}{d}}{\log \frac{h - z_0}{d}}$, se puede comprobar fácilmente que $0 \leq k < 1$. De esta forma, la ecuación (11) se transforma en

$$n = 1 + n^k \quad (12)$$

Si denotamos $g(x) = 1 + x^k$, se puede comprobar que $g(x)$ es contractiva en $[2, \frac{h - z_0}{d}]$ con constante de Lipschitz $C = \frac{1}{2^{1-k}}$ y además está acotada,

$$2 \leq g(x) \leq 1 + \left(\frac{h - z_0}{d} \right)^k \leq \frac{h - z_0}{d} \quad (13)$$

En virtud del teorema del punto fijo, podemos asegurar que la ecuación (12) tiene solución única, y puede obtenerse numéricamente, por ejemplo, mediante el método

del punto fijo, puesto que éste converge para cualquier aproximación inicial escogida en el intervalo $[2, \frac{h-z_0}{d}]$. No obstante, en general, la solución no tomará valores enteros. Consecuentemente, si aproximamos su valor al número natural más próximo, la condición impuesta con la distancia D no se cumplirá exactamente, sino de forma aproximada.

1.6. Definición de la nube de puntos

Cualquiera que sea la estrategia a seguir, la generación de puntos se realizará en tres etapas. En la primera se define una malla regular bidimensional con la densidad de puntos que se desea obtener sobre la frontera superior del dominio. En segundo lugar, y sobre esta malla τ_1 , se lleva a cabo un proceso de refinamiento global y desrefinamiento en función de la topografía del terreno para obtener la malla $\tau'_{m'}$ que define la distribución de puntos en la superficie del terreno. Para ilustrar estas dos primeras etapas, en la figura 4 se muestra la distribución de puntos sobre ambas superficies para un problema test. Una vez definida la distribución de puntos en la superficie del terreno y en la frontera superior del dominio, se procede a la generación de la nube de puntos distribuida entre estas dos capas. Para ello, sobre la vertical de cada nodo P de la malla del terreno $\tau'_{m'}$ situaremos puntos atendiendo a la función de espaciado vertical y al nivel j en el que P es propio, siendo $1 \leq j \leq m'$. La función de espaciado vertical quedará determinada mediante la estrategia utilizada para definir los siguientes parámetros: la cota topográfica z_0 de P ; la altitud h de la frontera superior del dominio; el máximo número posible de puntos $n + 1$ en la vertical de P , incluyendo el propio P y el de la frontera superior del dominio, caso de existir; el grado de la función de espaciado α ; la distancia entre los dos primeros puntos generados $d = z_1 - z_0$; y la distancia entre los dos últimos puntos generados $D = z_n - z_{n-1}$. De esta forma, la cota del i -ésimo punto generado sobre la vertical de P viene dada por,

$$z_i = \frac{h - z_0}{n^\alpha} i^\alpha + z_0 \quad ; \quad i = 1, 2, \dots, n - 1 \quad (14)$$

Independientemente de la función de espaciado vertical definida, utilizaremos el nivel j en el que P es propio para determinar el número definitivo de puntos que se generan sobre la vertical de P excluyendo el terreno y la frontera superior. Distinguiremos:

1. Si $j = 1$, es decir, si el nodo P es propio de la malla base τ_1 , se generan a partir de la ecuación (14) para $i = 1, 2, \dots, n - 1$.
2. Si $2 \leq j \leq m' - 1$, generamos nodos para $i = 1, 2, \dots, \min(m' - j, n - 1)$.
3. Si $j = m'$, esto es, el nodo P es propio del nivel más fino $\tau'_{m'}$, entonces no se genera ningún nodo nuevo.

Este proceso tiene su justificación, ya que la malla $\tau'_{m'}$ corresponde al nivel más fino de la secuencia de mallas encajadas $T' = \{\tau_1 < \tau_2 < \dots < \tau'_{m'}\}$, obtenida mediante el algoritmo de refinamiento y desrefinamiento, y por tanto el número de puntos introducidos decrece suavemente con la altura y además resultan distribuidos eficientemente con el fin de construir la malla tridimensional en el dominio.

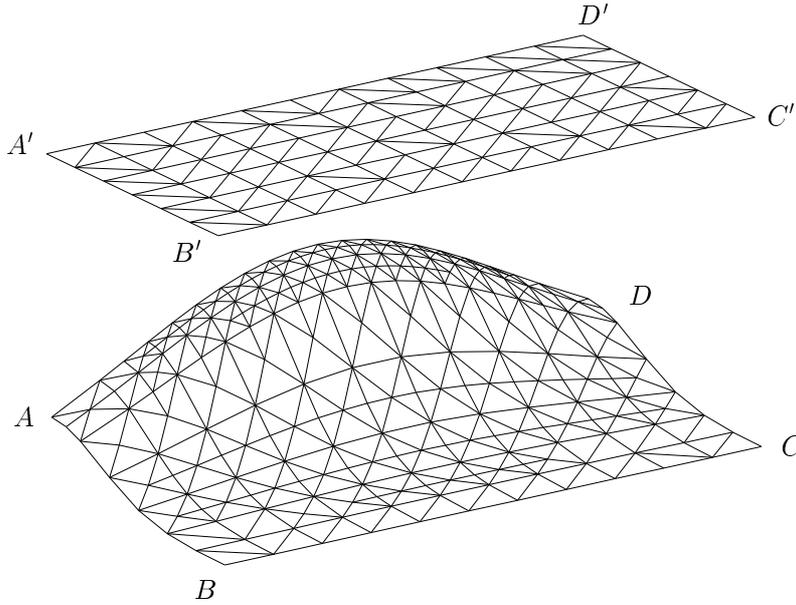


Figura 4: Representación tridimensional de una distribución de puntos sobre el terreno ($ABCD$) y la frontera superior del dominio ($A'B'C'D'$).

1.6.1. Estrategia 1: número de capas y grado de espaciado vertical fijos

En este caso, se considera impuesto el mismo valor de α y n para todo punto P de $\tau'_{m'}$, con el fin de generar los puntos a partir de la ecuación (14). Obsérvese que en la práctica, el parámetro n permite fijar el número de capas ($n + 1$) que se desea generar en el dominio, sobre las cuales se van a distribuir los puntos. Por otro lado, el valor de α determina el grado de concentración de capas hacia el terreno. En concreto, para $\alpha = 1$ la distancia entre dos capas consecutivas es constante sobre la vertical de cada punto de la superficie del terreno. En cambio, si escogemos valores de α superiores, la concentración de capas es mayor cerca del terreno. Con esta elección, los valores de α y n son introducidos como datos y, por tanto, la nube de puntos queda definida completamente. Consecuentemente, se pierde la libertad de fijar las distancias d y D . En general, esto conduce a que aquellos elementos con algún vértice sobre la superficie del terreno o sobre la frontera superior del dominio puedan tener una baja calidad. Con este procedimiento, la regularidad de las funciones que definen las capas va aumentando de una capa a otra superior, siendo la más irregular la correspondiente a la superficie del terreno y, la más regular, la capa horizontal correspondiente a la frontera superior del dominio. En base a esta regularidad, esta estrategia se ha diseñado de tal forma que, además de eliminar los nodos del nivel más fino de cada malla a medida que pasamos a una capa superior, se podría aplicar el propio algoritmo de desrefinamiento sobre la malla asociada a una capa determinada mediante un parámetro de desrefinamiento ε variable con la altura. La figura 5(a) muestra un detalle de la triangulación obtenida en la pared vertical $ADD'A'$ correspondiente al problema test de la figura 4. Al proceder a la transformación de la coordenada correspondiente a la altitud de los puntos de la nube generada al paralelepípedo auxiliar, las superficies que definen cada una de las capas pasan a ser planos horizontales.

1.6.2. Estrategia 2: número de capas fijo y grado de espaciado vertical variable

En esta estrategia imponemos también el mismo valor de n para todo punto P de $\tau'_{m'}$, fijando, por tanto, el número de capas ($n + 1$) que se desea generar en el dominio. En cambio, el valor de α se determina automáticamente en función del tamaño de los elementos más próximos al terreno, mediante la ecuación 9. En particular, hemos definido el valor de d para cada punto P como el promedio de las longitudes de las aristas de los triángulos que contienen a P en la malla $\tau'_{m'}$ sobre la superficie del terreno. Una vez determinada la función de espaciado vertical para cada punto P , con esta estrategia, la nube de puntos respeta las distancias exigidas entre la superficie del terreno y la primera capa generada. De esta forma, los posibles elementos de baja calidad se podrían dar en la parte superior del dominio, ya que el valor de D resulta impuesto por la propia función de espaciado vertical. Las capas resultantes con este procedimiento son menos regulares que las obtenidas con la estrategia 1. No obstante, estas capas se suavizan a partir de una altitud determinada, en función de la irregularidad topográfica, si hemos fijado un número total de capas suficiente. Igualmente, esta estrategia permite aplicar el propio algoritmo de desrefinamiento sobre la malla asociada a una capa determinada mediante un parámetro de desrefinamiento ε variable con la altura. La figura 5(b) muestra un detalle de la triangulación obtenida en la pared vertical $ADD'A'$ correspondiente al problema test representado en la figura 4. Por último, el paso de la nube de puntos al paralelepípedo auxiliar, se realiza de forma similar a la estrategia 1, tal que las superficies que definen las diferentes capas se transformen en planos horizontales. Con este fin, hemos utilizado en la transformación un valor fijo de $\alpha = 1$ para todos los puntos P de $\tau'_{m'}$. Por tanto, la distribución de los correspondientes planos en el paralelepípedo es uniforme.

1.6.3. Estrategia 3: número de capas variable y grado de espaciado vertical fijo

En esta estrategia imponemos el mismo valor de α para todo punto P de $\tau'_{m'}$. En cambio, el valor de n se determina automáticamente en función del tamaño de los elementos más próximos al terreno, a partir del valor natural más cercano al real obtenido con la ecuación 8, donde el valor de d para cada punto P se ha determinado de igual forma que en la estrategia 2. Esta estrategia posee por tanto características similares a la anterior en cuanto a las distancias exigidas entre la superficie del terreno y la primera capa generada. Puesto que con esta estrategia el número de capas obtenido es diferente en función del punto P considerado, no podemos definir superficies asociadas a cada capa. Por ello, hablaremos de la existencia de capas virtuales para cada punto. La figura 5(c) muestra un detalle de la triangulación obtenida en la pared vertical $ADD'A'$ del problema test representado en la figura 4. Por último, el paso de la nube de puntos al paralelepípedo auxiliar, se realiza de forma similar a la estrategia 1. Puesto que tenemos capas virtuales, la transformación de los puntos se realiza según la función de espaciado vertical asociada a cada punto P de $\tau'_{m'}$.

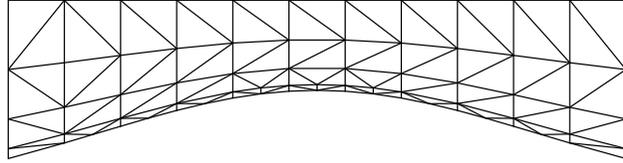
1.6.4. Estrategia 4: número de capas y grado de espaciado vertical variables

Finalmente, definimos una estrategia en la que los valores de α y n se determinan automáticamente para cada punto P de $\tau'_{m'}$, en función del tamaño de los elementos más próximos al terreno y a la parte superior del dominio. En primer lugar, comenzamos determinando el valor de d de igual manera que en las estrategias 2 y 3. En segundo lugar, fijamos un único valor de D en función de la distancia deseada entre el último punto que teóricamente sería generado sobre las diferentes verticales y el plano superior del dominio. Esta distancia se determina fácilmente en función del tamaño de los elementos de la malla regular τ_1 . Una vez obtenido d y D , para todo punto P de $\tau'_{m'}$, calculamos su correspondiente valor de n resolviendo la ecuación (12). Finalmente, la función de espaciado vertical queda determinada al obtener el valor de α mediante la ecuación (9). Esta estrategia posee, por tanto, como características principales que respeta las distancias exigidas entre la superficie del terreno y la primera capa generada, así como la distancia impuesta entre la última capa virtual generada y la frontera superior del dominio. La figura 5(d) muestra un detalle de la triangulación obtenida en la pared vertical $ADD'A'$ del problema test de la figura 4. Asimismo, el paso de la nube de puntos al paralelepípedo auxiliar, se realiza de forma similar a la estrategia 1.

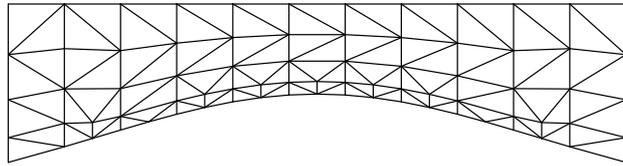
1.7. Construcción de la malla

Una vez definida la nube de puntos será necesario generar una malla tridimensional que conecte esos puntos de forma adecuada y que sea conforme con la frontera del dominio, esto es, que respete todas las fronteras establecidas. Aunque la triangulación de Delaunay es apropiada para crear mallas de elementos finitos con un alto grado de regularidad para una determinada nube de puntos, no lo es tanto en lo referente al problema de conformidad con la frontera, puesto que ésta crea una malla de la envolvente convexa de la nube de puntos. Por esta razón, es posible que no se pueda recuperar la frontera del dominio a partir de las caras y aristas generadas por la triangulación. Hemos comentado que para evitar esto, existen básicamente dos tipos diferentes de técnicas. Unas, *Conforming Delaunay Triangulation* [51], están basadas en la colocación de los puntos siguiendo ciertos criterios de espaciado, de manera que la triangulación resultante sea conforme con la frontera. Las otras, *Constrained Delaunay Triangulation* [25], están basadas en la modificación de la triangulación en las zonas próximas a la frontera no respetada, mediante intercambio de aristas y caras (*swapping*) de manera que ésta sea recuperada. En nuestro caso, la primera alternativa no resulta adecuada, debido a que deseamos que la malla resultante contenga ciertos puntos definidos de antemano. Además, debido a la complejidad de la superficie del terreno, esta estrategia conllevaría un alto coste computacional. En principio, la segunda alternativa sería válida, si bien requiere unos algoritmos bastante complejos para recuperar la frontera del dominio. Para solventar los problemas de conformidad con la frontera proponemos el siguiente procedimiento.

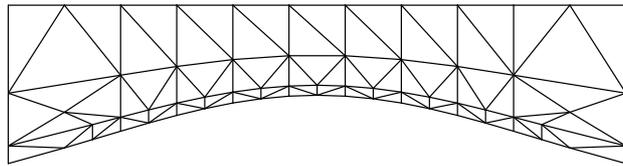
Comenzamos recolocando la distribución de puntos del dominio en un paralelepípedo auxiliar (ver figura 6), tal que todos los puntos de la superficie del terreno estén situados en las coordenadas x, y originales, pero a una altura igual a la cota



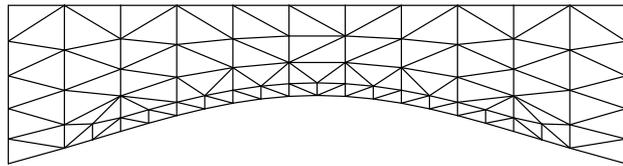
(a) Estrategia 1



(b) Estrategia 2



(c) Estrategia 3



(d) Estrategia 4

Figura 5: *Distribución de puntos obtenida sobre la frontera $ADD'A'$ para el ejemplo de la figura 4 mediante las diferentes estrategias.*

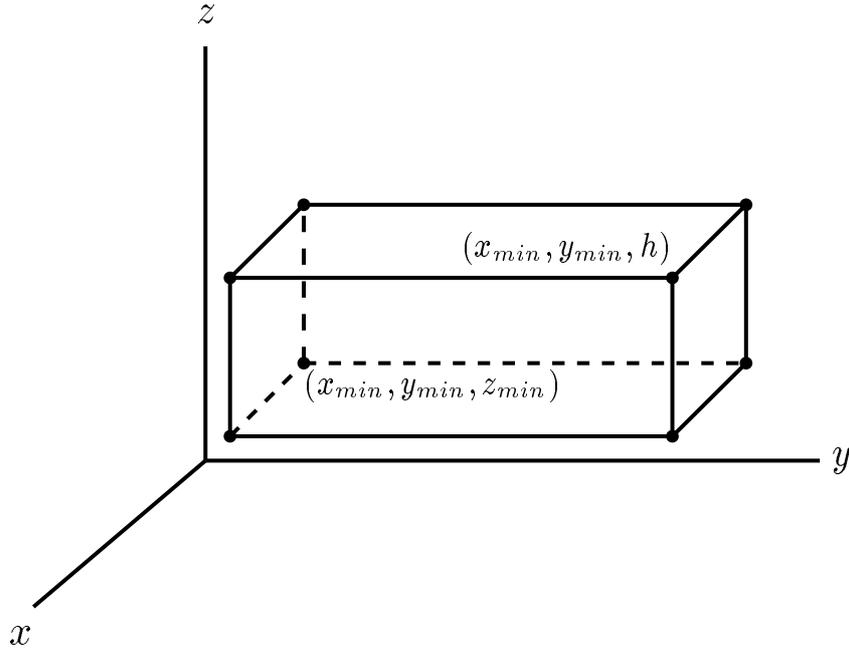


Figura 6: Definición del paralelepípedo auxiliar.

mínima del terreno, z_{min} . En el plano superior del paralelepípedo situamos los nodos del nivel τ_1 , de la secuencia de mallas que define la superficie del terreno, a una altura igual a h . En general, los restantes puntos también mantienen sus coordenadas x , y , pero sus cotas se obtienen sustituyendo su correspondiente z_0 por z_{min} en la ecuación (14). La triangulación de esta nube de puntos se realiza utilizando una variante del algoritmo incremental de Watson [18] que resuelve de manera efectiva los problemas derivados de los errores de redondeo que se producen cuando se trabaja con números en coma flotante. Una vez construida la triangulación en el paralelepípedo, obtenemos la malla final restableciendo las cotas originales de los puntos. Este último proceso puede entenderse como una compresión de toda la malla definida en el paralelepípedo, tal que su plano inferior se transforma en la superficie del terreno, con lo que se asegura la conformidad.

En ocasiones puede ocurrir que al restablecer las posiciones de los puntos a sus cotas reales se produzcan elementos de muy mala calidad o, incluso, *invertidos*, es decir, elementos para los que su volumen V_e , evaluado como el determinante jacobiano $|J_e|$ asociado a la transformación del tetraedro e al de referencia, pasa a ser negativo. Así, si las densidades de puntos van disminuyendo de forma progresiva con la altura y si los puntos se sitúan sobre capas *suficientemente* separadas en el paralelepípedo, entonces la probabilidad de existencia de elementos de baja calidad o de *cruces* se reduce. En cualquier caso, necesitamos algún procedimiento capaz de deshacer los cruces que pudieran aparecer y suavizar la malla resultante. Esta cuestión será abordada en la sección 1.8. Por otra parte, se debe tener en cuenta que la posibilidad de obtener una malla de elevada calidad mediante algoritmos de suavizado basados en movimientos de los nodos alrededor de sus posiciones iniciales depende, además del procedimiento concreto utilizado, de la ‘calidad topológica’ de la malla. Se entiende que ésta es alta cuando la valencia de cada nodo, esto es el número de nodos conectados a él, se aproxima a la que tendría una malla regular formada por tetraedros equiláteros.

La triangulación de Delaunay es capaz de crear una malla de elevada calidad, óptima en 2-D, para una nube de puntos dada. Así pues, un adecuado criterio para la distribución de éstos tendrá como consecuencia la obtención de una malla inicial de elevada calidad. La malla de nuestro dominio conserva la calidad topológica de la triangulación obtenida en el paralelepípedo y, por tanto, un suavizado apropiado conduciría a mallas de gran calidad.

1.8. Optimización de la malla. Suavizado y desenredo simultáneo

Uno de los aspectos fundamentales de la técnica de elementos finitos es la utilización de mallas de calidad. Algunos generadores automáticos de mallas tridimensionales construyen en una primera aproximación mallas válidas, pero de poca calidad. Además, en ciertos casos especiales, pueden aparecer elementos invertidos, dando lugar a lo que llamamos mallas *enredadas*. Por tanto, es necesario desarrollar procedimientos capaces de suavizar y desenredar mallas ya existentes.

El suavizado es una de las técnicas más comunes para mejorar la calidad de una malla válida, esto es, una malla que no tiene elementos invertidos. En esencia, esta técnica consiste en desplazar cada nodo de la malla hasta una nueva posición que optimiza una cierta función objetivo. Esta función está construida a partir de una medida de calidad de la submalla local $N(v)$, formada por los tetraedros conectados al nodo libre v . Dado que es un proceso de optimización local, no se puede garantizar que la malla final sea globalmente óptima. Sin embargo, tras repetir este proceso un determinado número de veces en todos los nodos de la malla se alcanzan resultados satisfactorios.

Normalmente este tipo de función objetivo es adecuado para mejorar la calidad de mallas válidas. Sin embargo, no funcionan bien cuando hay tetraedros invertidos debido a que presentan singularidades cuando algún tetraedro de $N(v)$ cambia el signo de su determinante jacobiano. Para resolver este problema se puede proceder tal y como se propone en [22; 23; 21]. En una primera etapa se desenredan los elementos invertidos usando un algoritmo que maximice sus determinantes jacobianos negativos [23]; en una segunda etapa se suaviza la malla resultante usando otra función objetivo basada en alguna medida de calidad de los tetraedros de $N(v)$ [21]. Más adelante se presentan dos de estas funciones objetivo. Tras el proceso de desenredo la malla tiene una calidad muy pobre porque la técnica no busca la creación de elementos de buena calidad. Como queda dicho en [22], para optimizar la función objetivo asociada al proceso de suavizado no se puede aplicar un algoritmo basado en el gradiente porque ésta no está definida en todo \mathbb{R}^3 . Es necesario entonces utilizar otras técnicas para evitar este problema. Algunos autores [65; 66] han propuesto el desplazamiento de las barreras (singularidades) mediante la introducción de parámetros globales calculados en función del área, con el propósito de obtener mallas convexas estructuradas en dominios bidimensionales.

Nosotros proponemos una alternativa a estas técnicas, de forma que el desenredo y el suavizado de la malla se realicen de manera simultánea. De esta forma se obtienen mallas con mejor calidad tras el desenredo y además se necesitan menos iteraciones de suavizado para alcanzar una cierta calidad. Para ello se usa una modificación de la función objetivo de forma que esté definida sobre \mathbb{R}^3 , pero preservando las posiciones en que se encuentran los mínimos, de forma que cuando existe una

región factible (subconjunto de \mathbb{R}^3 donde puede situarse v siendo $N(v)$ una submalla válida) los mínimos de la función original y la modificada están muy cercanos. Cuando ésta región no existe, el mínimo de la función objetivo modificada es tal que tiende a desenredar $N(v)$. Esto último ocurre, por ejemplo, cuando la frontera fija de $N(v)$ está enredada. De esta forma se pueden usar métodos de optimización estándar para localizar el mínimo de la función objetivo modificada (véanse por ejemplo [13], [26] y [5]).

En nuestros trabajos se han aplicado las modificaciones propuestas para dos funciones objetivo distintas derivadas de las *medidas de calidad algebraicas* estudiadas en [32], aunque sería posible aplicarlas también a otras funciones objetivo que presentaran singularidades, tales como las estudiadas en [31].

1.9. Funciones objetivo

Las funciones objetivo se pueden construir partiendo de alguna medida de calidad de los tetraedros [15]. Sin embargo, aquellas que se obtienen mediante operaciones algebraicas son especialmente adecuadas para nuestro propósito ya que el coste computacional requerido para su evaluación puede ser muy bajo.

Sea T un tetraedro en el espacio físico cuyos vértices están dados por $\mathbf{x}_k = (x_k, y_k, z_k)^T \in \mathbb{R}^3$, $k = 0, 1, 2, 3$ y sea T_R el tetraedro de referencia con vértices $\mathbf{u}_0 = (0, 0, 0)^T$, $\mathbf{u}_1 = (1, 0, 0)^T$, $\mathbf{u}_2 = (0, 1, 0)^T$ y $\mathbf{u}_3 = (0, 0, 1)^T$. Si se toma \mathbf{x}_0 como vector de traslación, la aplicación afín que transforma T_R en T es $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, donde A es la matriz jacobiana de la aplicación afín referida al nodo \mathbf{x}_0 , y cuya expresión es

$$A = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix} \quad (15)$$

Sea ahora T_I un tetraedro equilátero con todas sus aristas de longitud uno y vértices en $\mathbf{v}_0 = (0, 0, 0)^T$, $\mathbf{v}_1 = (1, 0, 0)^T$, $\mathbf{v}_2 = (1/2, \sqrt{3}/2, 0)^T$ y $\mathbf{v}_3 = (1/2, \sqrt{3}/6, \sqrt{2}/\sqrt{3})^T$. Sea $\mathbf{v} = W\mathbf{u}$ la aplicación afín que convierte T_R en T_I y siendo su matriz jacobiana

$$W = \begin{pmatrix} 1 & 1/2 & 1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/6 \\ 0 & 0 & \sqrt{2}/\sqrt{3} \end{pmatrix} \quad (16)$$

La aplicación afín que transforma T_I en T viene dada por $\mathbf{x} = AW^{-1}\mathbf{v} + \mathbf{x}_0$, y su matriz jacobiana es $S = AW^{-1}$. La matriz S es independiente del nodo que se toma como referencia y se dice que es invariante respecto a los nodos [32]. Para construir medidas de calidad algebraicas de T puede usarse alguna norma matricial o la traza de S . Por ejemplo, la norma de Frobenius de S , definida por $|S| = \sqrt{\text{tr}(S^T S)}$, es especialmente adecuada porque es fácilmente computable. Así, se demuestra en [32] que tanto $q_\eta(S) = \frac{3\sigma^{2/3}}{|S|^2}$ como $q_\kappa(S) = \frac{3}{|S||S-1|}$ son medidas de calidad algebraicas de T , donde $\sigma = \det(S)$. El valor máximo de estas medidas es la unidad y corresponde al tetraedro equilátero, mientras que cualquier tetraedro plano o degenerado tiene medida nula.

A partir de estas medidas de calidad se pueden obtener funciones objetivo. Así, sea $\mathbf{x} = (x, y, z)^T$ la posición del nodo libre v , y S_m la matriz jacobiana correspondiente al m -ésimo tetraedro de $N(v)$. Se define la función objetivo de \mathbf{x} , asociada

al m -ésimo tetraedro como

$$\eta_m = \frac{|S_m|^2}{3\sigma_m^{\frac{2}{3}}} \quad (17)$$

o

$$\kappa_m = \frac{|S_m| |S_m^{-1}|}{3} = \frac{|S_m| |\Sigma_m|}{3\sigma_m} \quad (18)$$

donde $\Sigma_m = \sigma_m S_m^{-1}$ es la matriz de los adjuntos de S_m . Entonces, las funciones objetivo correspondientes de $N(v)$ pueden construirse usando la p -norma de $(\eta_1, \eta_2, \dots, \eta_M)$ o $(\kappa_1, \kappa_2, \dots, \kappa_M)$ como

$$|K_\eta|_p(\mathbf{x}) = \left[\sum_{m=1}^M \eta_m^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (19)$$

o

$$|K_\kappa|_p(\mathbf{x}) = \left[\sum_{m=1}^M \kappa_m^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (20)$$

donde M es el número de tetraedros de $N(v)$. La función objetivo $|K_\eta|_1$ es deducida y utilizada en [3] para suavizar y adaptar mallas 2-D. La misma función se utiliza en [14] para el suavizado de mallas 2-D y 3-D como resultado de un método *force-directed*. Las funciones $|K_\kappa|_p$ están propuestas y ampliamente analizadas en [22; 21] y finalmente ambos tipos de funciones, entre otras, están estudiados y comparados en [31]. Todas las funciones citadas únicamente pueden usarse para suavizar mallas válidas, es decir, aquellas en las que se cumple que $\sigma_m > 0, \forall m = 1, \dots, M$.

Estas funciones objetivo son suaves en aquellos puntos donde $N(v)$ es una submalla válida, pero se vuelven discontinuas cuando el volumen de cualquier tetraedro de $N(v)$ se hace cero. Esto se debe a que $|K_\eta|_p$ y $|K_\kappa|_p$ tienden a infinito cuando σ_m tiende a cero, ya que sus numeradores están acotados. Se puede demostrar que $|S_m|$ y $|\Sigma_m|$ alcanzan sus mínimos, con valores estrictamente positivos, cuando v está situado en el centro geométrico de la cara fija del m -ésimo tetraedro. Las posiciones de v tales que $N(v)$ sea válida, es decir, la región factible, se corresponden con el interior del poliedro P definido como

$$P = \bigcap_{m=1}^M H_m \quad (21)$$

donde H_m son los semiespacios definidos por $\sigma_m(\mathbf{x}) \geq 0$ (la región sombreada de la figura 7a). Este conjunto puede estar vacío, por ejemplo, cuando la frontera fija de $N(v)$ está enredada (figura 7c). En esa situación, las funciones $|K_\eta|_p$ y $|K_\kappa|_p$ dejan de útiles como funciones objetivo. Por otra parte, si existe región factible, esto es, $\text{int } P \neq \emptyset$, las funciones objetivo tienden a infinito a medida que v tiende a la frontera de P . Debido a esas singularidades, se forma una barrera que impide que se alcance el mínimo usando algoritmos basados en el gradiente cuando se parte de un nodo libre que se encuentra fuera de la región factible (véase figura 7b). En resumen, no podemos optimizar una malla enredada $N(v)$ trabajando con estas funciones objetivo, ni con algoritmos de tipo gradiente, aunque exista región factible.

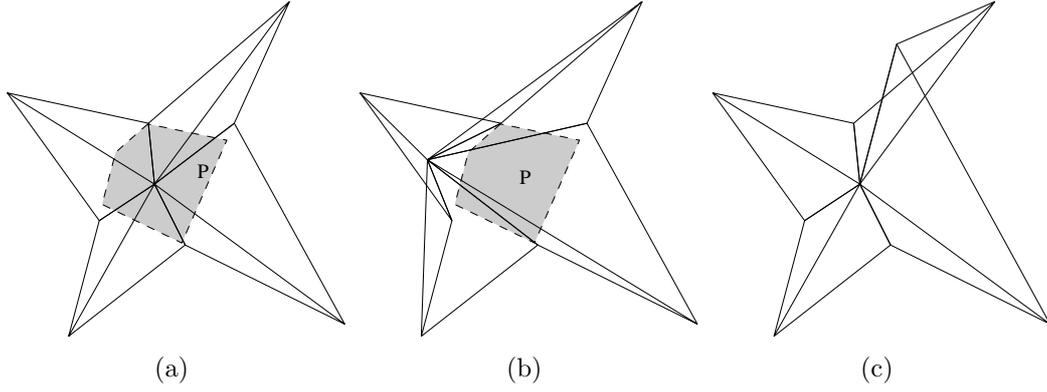


Figura 7: (a) Malla válida con el nodo libre en el interior de la región factible. (b) Malla enredada con el nodo libre fuera de la región factible. (c) Malla enredada con la frontera fija enredada.

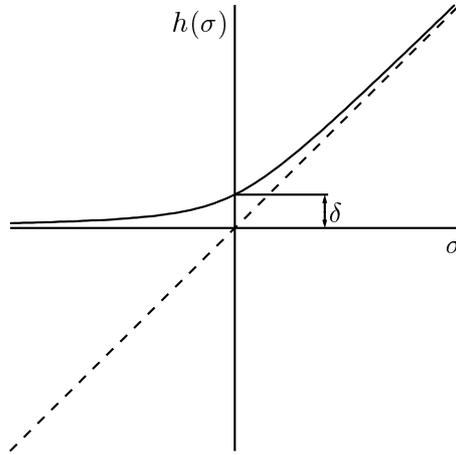


Figura 8: Representación de la función $h(\sigma)$.

1.10. Funciones objetivo modificadas

A continuación proponemos una modificación de las funciones objetivo (19) y (20), de manera que se elimine la barrera asociada a las singularidades y se consiga que la nueva función sea suave en \mathbb{R}^3 . Un requisito esencial es que los mínimos de la funciones objetivo originales y las modificadas sean casi idénticos cuando $\text{int } P \neq \emptyset$. Nuestra modificación consiste en sustituir σ en (19) y (20) por la función creciente y positiva

$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \quad (22)$$

donde $\delta = h(0)$. En la figura 8 está representada la función $h(\sigma)$.

Así, las nuevas funciones objetivo propuestas aquí están dadas por

$$|K_\eta^*|_p(\mathbf{x}) = \left[\sum_{m=1}^M (\eta_m^*)^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (23)$$

y

$$|K_\kappa^*|_p(\mathbf{x}) = \left[\sum_{m=1}^M (\kappa_m^*)^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (24)$$

donde

$$\eta_m^* = \frac{|S_m|^2}{3h(\sigma_m)^{\frac{2}{3}}} \quad (25)$$

y

$$\kappa_m^* = \frac{|S_m| |\Sigma_m|}{3h(\sigma_m)} \quad (26)$$

son las funciones objetivo modificadas para el tetraedro m -ésimo.

El comportamiento de $h(\sigma)$ en función del parámetro δ es tal que, $\lim_{\delta \rightarrow 0} h(\sigma) = \sigma$, $\forall \sigma \geq 0$ y $\lim_{\delta \rightarrow 0} h(\sigma) = 0$, $\forall \sigma \leq 0$. Así, si $\text{int } P \neq \emptyset$, entonces $\forall \mathbf{x} \in \text{int } P$ tenemos $\sigma_m(\mathbf{x}) > 0$, para $m = 1, 2, \dots, M$ y, a medida que vayamos eligiendo valores de δ más pequeños, $h(\sigma_m)$ se va pareciendo más a σ_m , de manera que las funciones originales y sus correspondientes versiones modificadas son muy próximas en la región factible. Así, en dicha región, $|K_\eta^*|_p$ y $|K_\kappa^*|_p$ convergen puntualmente a $|K_\eta|_p$ y $|K_\kappa|_p$ cuando $\delta \rightarrow 0$. Además, considerando que $\forall \sigma > 0$, $\lim_{\delta \rightarrow 0} h'(\sigma) = 1$ y $\lim_{\delta \rightarrow 0} h^{(n)}(\sigma) = 0$, para $n \geq 2$, es fácil comprobar que las derivadas de las funciones objetivo verifican la misma propiedad de convergencia. Como resultado de estas consideraciones, podemos concluir que las posiciones de v que minimizan las funciones objetivo originales y las modificadas son casi idénticas cuando el valor de δ es *pequeño*. En realidad, δ se selecciona en función del punto v bajo consideración, haciéndolo tan pequeño como sea posible pero de manera que la evaluación del mínimo de la función objetivo modificada no presente ningún problema computacional.

En particular, sea γ el *épsilon de la máquina* ($0 < \gamma \ll 1$), entonces para evitar divisiones por cero al calcular las funciones objetivo modificadas (25) y (26) hay que imponer que $h(\sigma) \geq \gamma$ en todos los tetraedros de $N(v)$. Como $h(\sigma)$ es una función creciente el peor caso corresponde a $\sigma = \sigma_{min}$, donde $\sigma_{min} = \min_{m=1, \dots, M}(\sigma_m)$. Se puede demostrar que para los siguientes valores de δ ,

$$\delta \geq \delta_{min} = \begin{cases} \sqrt{\gamma(\gamma - \sigma_{min})} & \text{si } \sigma_{min} < \gamma \\ 0 & \text{si } \sigma_{min} \geq \gamma \end{cases} \quad (27)$$

se verifica la condición $h(\sigma) \geq \gamma$. En la práctica se ha multiplicado γ por un factor de seguridad. Hay que destacar que cuando el nodo libre está *claramente* dentro de la región factible ($\sigma_{min} \geq \gamma$), nuestras funciones objetivo modificadas coinciden con las originales.

Supongamos ahora que, $\text{int } P = \emptyset$, entonces las funciones objetivo originales, $|K_\eta|_p$ y $|K_\kappa|_p$, no son adecuadas para nuestro propósito ya que no están correctamente definidas. Sin embargo, las funciones objetivo modificadas están correctamente definidas y tienden a resolver el enredo. Podemos razonar esto desde un punto de vista cualitativo considerando que los términos dominantes en $|K_\eta^*|_p$ o $|K_\kappa^*|_p$ son aquellos que están asociados a tetraedros con valores de σ más negativos y, por ello, la minimización de estos términos implica el incremento de estos valores. Hay

que destacar que $h(\sigma)$ es una función creciente y que tanto $|K_\eta^*|_p$ como $|K_\kappa^*|_p$ tienden a ∞ cuando el volumen de cualquier tetraedro de $N(v)$ tiende a $-\infty$, ya que $\lim_{\sigma \rightarrow -\infty} h(\sigma) = 0$.

En resumen, mediante las funciones objetivo modificadas, podemos desenredar la malla a la vez que mejoramos su calidad. Obviamente, la modificación propuesta aquí puede aplicarse fácilmente a otras funciones objetivo del mismo tipo.

1.11. Optimización de las funciones objetivo modificadas

Los algoritmos convencionales de optimización, como el descenso mínimo o el algoritmo de Newton, necesitan evaluar el gradiente y, en algunos casos, el hessiano de la función objetivo. Por eso en esta sección se van a obtener las expresiones de las derivadas de η^* y κ^* con respecto a parámetros arbitrarios α y β que representan cualquiera de las coordenadas x , y y z del nodo libre.

Considérese el producto interno de dos matrices, R y S , como

$$(R, S) = \text{tr}(R^T S) \quad (28)$$

de forma que, la norma de Frobenius de S es $|S| = \sqrt{(S, S)}$. Si llamamos ∂_α al operador derivada parcial respecto de α , tal que $\partial_\alpha R = [\partial_\alpha r_{ij}]$ para una matriz $n \times n$ $R = [r_{ij}]$, $1 \leq i, j \leq n$, se puede demostrar que la derivada de η^* es

$$\partial_\alpha \eta^* = 2\eta^* \left[\frac{(\partial_\alpha S, S)}{|S|^2} - \frac{\partial_\alpha \sigma}{3\sqrt{\sigma^2 + 4\delta^2}} \right] \quad (29)$$

Para κ^* se tiene

$$\partial_\alpha \kappa^* = \kappa^* \left[\frac{(\partial_\alpha S, S)}{|S|^2} + \frac{(\partial_\alpha \Sigma, \Sigma)}{|\Sigma|^2} - \frac{\partial_\alpha \sigma}{\sqrt{\sigma^2 + 4\delta^2}} \right] \quad (30)$$

Para obtener las segundas derivadas hay que considerar que S , Σ y σ son funciones lineales de x , y , z , y así $\partial_{\alpha\beta} S$, $\partial_{\alpha\beta} \Sigma$ y $\partial_{\alpha\beta} \sigma$ son cero. Con estas consideraciones tenemos

$$\begin{aligned} \partial_{\alpha\beta} \eta^* &= \frac{\partial_\alpha \eta^* \partial_\beta \eta^*}{\eta^*} + \\ &+ 2\eta^* \left[\frac{(\partial_\alpha S, \partial_\beta S)}{|S|^2} - \frac{2(\partial_\alpha S, S)(\partial_\beta S, S)}{|S|^4} + \frac{\sigma \partial_\alpha \sigma \partial_\beta \sigma}{3(\sigma^2 + 4\delta^2)^{\frac{3}{2}}} \right] \end{aligned} \quad (31)$$

y para κ^* se tiene

$$\begin{aligned} \partial_{\alpha\beta} \kappa^* &= \frac{\partial_\alpha \kappa^* \partial_\beta \kappa^*}{\kappa^*} + \kappa^* \left[\frac{(\partial_\alpha S, \partial_\beta S)}{|S|^2} + \frac{(\partial_\alpha \Sigma, \partial_\beta \Sigma)}{|\Sigma|^2} - \right. \\ &\left. - \frac{2(\partial_\alpha S, S)(\partial_\beta S, S)}{|S|^4} - \frac{2(\partial_\alpha \Sigma, \Sigma)(\partial_\beta \Sigma, \Sigma)}{|\Sigma|^4} + \frac{\sigma \partial_\alpha \sigma \partial_\beta \sigma}{(\sigma^2 + 4\delta^2)^{\frac{3}{2}}} \right] \end{aligned} \quad (32)$$

Las ecuaciones (31) y (32) pueden simplificarse dado que $(\partial_\alpha S, \partial_\alpha S) = \frac{3}{2}$, $(\partial_\alpha S, \partial_\beta S) = 0$, si $\alpha \neq \beta$.

Si δ satisface la ecuación (27), las derivadas de las funciones objetivo modificadas, $|K_\eta^*|_p$ y $|K_\kappa^*|_p$, pueden evaluarse fácilmente utilizando las anteriores expresiones sin que presenten problemas computacionales, puesto que $\gamma \leq h(\sigma) \leq \sqrt{\sigma^2 + 4\delta^2}$.

1.12. Enlace entre códigos

La implementación de las técnicas de mallado descritas anteriormente se ha realizado aprovechando códigos que ya estaban disponibles. En concreto, utilizamos un generador de mallas 2-D, el programa de refinamiento/desrefinamiento en 2-D (NEPTUNO) y un generador de malla 3-D basado en la triangulación de Delaunay.

Los anteriores programas fueron creados con el lenguaje de programación FORTRAN 77 y no fueron pensados para interactuar entre ellos. A pesar de que han sido modificados a lo largo del tiempo, bien para incorporar ciertas mejoras o bien para adaptarlos a las sucesivas plataformas informáticas donde se han ido ejecutando, siguen teniendo algunas características que los hacen muy poco amigables con el usuario:

- Los ficheros de entradas son complejos y hay que atenerse a un formato tan estricto como incómodo para el usuario. Una de las mayores dificultades era la adaptación manual de la salida de un programa para que sirviera como entrada para el siguiente, en la secuencia de generación de la malla 3-D.
- Las salidas que proporcionan carecen de flexibilidad; como ejemplo, baste decir que no se puede elegir el nombre ni el directorio del fichero de salida de ninguno de los programas.
- Debido a que FORTRAN 77 no tiene capacidad de memoria dinámica, en ocasiones es necesario recompilarlos para adaptarlos a la capacidad de la máquina donde se ejecutan y a los requerimientos del problema concreto.

Todo esto justifica la creación de un programa que, enlazando todos los anteriores, presente al usuario una interfaz coherente y cómoda. Los detalles engorrosos, los ficheros de entrada ininteligibles y la rigidez de las salidas quedan ocultos tras esta interfaz que se ocupa de gestionarlos. En la figura 9 se puede ver el proceso que se sigue para la construcción de la malla 3D. Primero hay que leer los parámetros que definen la malla, después se ejecuta el mallador 2D, a continuación el programa NEPTUNO de refinamiento/desrefinamiento 2D y por último el mallador 3D. El programa de enlace de códigos, llamado MALLA, es el encargado de ejecutar la secuencia descrita, para lo que debe realizar automáticamente las adaptaciones necesarias con el objeto de que la salida de un programa sirva como entrada al siguiente en la secuencia.

El programa está escrito en PERL, un lenguaje de programación interpretado inicialmente diseñado para tareas de administración de sistemas, pero que debido a su versatilidad, portabilidad y facilidad de uso se ha extendido a muchas otras áreas de aplicación. Tiene dos modos de funcionamiento. El primero de ellos es interactivo: el usuario va contestando una serie de preguntas que le va formulando el programa. En el otro modo, el usuario invoca al programa pasándole como parámetro el nombre de un fichero de texto que ha de contener los datos que el programa requiere. La sintaxis del fichero de entrada es muy sencilla, lo que permite al usuario crearlo y/o modificarlo fácilmente. Está compuesto por una serie de líneas con el formato que se observa en la figura 10. Las líneas que comienzan por el carácter “#” son comentarios, que pueden insertarse en cualquier sitio y permiten que los ficheros de entrada puedan ser autoexplicativos. El resto de las líneas sirve para definir los distintos parámetros que necesita el programa. El formato que siguen es el de una palabra clave que define un parámetro, el signo “=” y el valor que se desea asignar

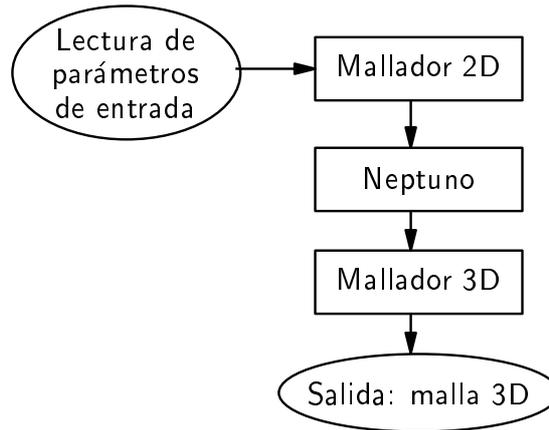


Figura 9: *Diagrama de la secuencia de construcción de la malla 3D.*

```

# Esto es un comentario que aclara el significado
# de PALABRA_CLAVE

PALABRA_CLAVE_1 = valor_1

PALABRA_CLAVE_2 = valor_2
  
```

Figura 10: *Formato del fichero de entrada al programa MALLA.*

al parámetro. El modo interactivo tiene la característica de que el usuario, tras contestar las preguntas, puede optar por guardar un fichero de parámetros con las respuestas que ha dado, de forma que si posteriormente desea volver a ejecutar el programa pueda hacerlo directamente, sin entrar en modo interactivo.

A continuación se describe brevemente el significado de las palabras clave del mallador:

h: Coordenada z del plano horizontal que delimita la frontera superior del dominio.

D: Distancia entre los dos últimos puntos introducidos en cada vertical en la función de espaciado vertical (ver sección 1.5).

temg: Tamaño de triángulo de la malla grosera inicial (τ_1 de la sección 1.2), que además se usará para discretizar la frontera superior del dominio.

n1: Número de refinamientos globales sobre la malla grosera inicial.

estrategia: Estrategia utilizada para generar la nube de puntos (véase la sección 1.6):

1. Grado de espaciado y número de capas fijos.
2. Grado de espaciado variable y número de capas fijo.
3. Grado de espaciado fijo y número de capas variable.
4. Grado de espaciado y número de capas variables.

- alfa:** Grado de la función de espaciado vertical: parámetro α de la ecuación (5).
- capas:** Número de capas. Sólo es necesario indicarlo en la estrategia 1, ya que en las restantes se calcula automáticamente.
- epsilon:** Parámetro ε de desrefinamiento de la malla 2D del terreno. Determina la precisión con que la discretización se aproxima a la topografía (véase la sección 1.2).
- epsilonmax:** Parámetro ε máximo de desrefinamiento. En cada capa se obtiene el ε correspondiente interpolando entre el ε inicial y el máximo. Un valor extremadamente pequeño del ε máximo (por ejemplo, -1^{-10}) en las estrategias 1 y 2, implica que sólo se aplica el algoritmo de desrefinamiento a la malla correspondiente a la superficie del terreno. En las estrategias 3 y 4 no interviene este parámetro.
- ftopografia:** Fichero que contiene la topografía digitalizada.
- cxii:** Coordenada X del vértice inferior izquierdo de la base del dominio.
- cyii:** Coordenada Y del vértice inferior izquierdo de la base del dominio.
- cxsd:** Coordenada X del vértice superior derecho de la base del dominio.
- cysd:** Coordenada Y del vértice superior derecho de la base del dominio.

Atendiendo a los datos necesarios, destacamos que la generación de la malla tridimensional, ajustada a la orografía irregular del terreno, se obtiene utilizando muy pocos parámetros de entrada. El proceso es completamente automático y la sintaxis del fichero de entrada es muy sencilla. Esto, unido al hecho de que el tiempo de ejecución del programa es, para los problemas más complejos que se han resuelto, de alrededor de menos de un minuto y medio, permite que el usuario pueda ejecutar el programa tantas veces como desee, probando distintas estrategias y parámetros, hasta generar la malla que más se ajuste a sus necesidades. Estas características hacen del mallador una herramienta potente, versátil y fácil de utilizar.

2. Modelización de campos de viento

En este capítulo se desarrolla un modelo de masa consistente para ajuste de campos de viento en tres dimensiones a partir de medidas experimentales. El modelo sirve tanto de herramienta generadora de mapas de viento de una zona determinada, como de preproceso para otros modelos que traten fenómenos que tiene lugar en la atmósfera, tales como dispersión de contaminantes o propagación de incendios.

El modelo hace uso del método de elementos finitos (MEF) sobre un mallado tridimensional adaptado a la orografía del dominio. Se emplean técnicas iterativas de resolución de sistemas de ecuaciones, así como refinamiento adaptativo de la malla para mejorar a la solución numérica.

2.1. Modelo de masa consistente

Los modelos de masa consistente son modelos de diagnóstico para definir campos de velocidades de viento a partir de un número determinado de medidas experimentales [33]. En general, estos modelos están gobernados por las leyes físicas definidas para un fluido incompresible, por el diseño de los perfiles de viento de forma empírica y por medidas experimentales realizadas en la zona de estudio.

La ecuación de continuidad de masa del aire que se mueve en un dominio tridimensional Ω con un campo de velocidades \vec{u} y la condición de impenetrabilidad sobre Γ_b (terreno y frontera superior del dominio) son, respectivamente

$$\vec{\nabla} \cdot \vec{u} = 0 \text{ en } \Omega \quad (33)$$

$$\vec{n} \cdot \vec{u} = 0 \text{ en } \Gamma_b \quad (34)$$

asumiendo que la densidad del aire es constante en todo el dominio.

A partir de estas condiciones, los modelos de masa consistente plantean un problema de mínimos cuadrados en el dominio Ω con las velocidades a ajustar $\vec{u}(\tilde{u}, \tilde{v}, \tilde{w})$ a partir de las observadas $\vec{v}_0(u_0, v_0, w_0)$,

$$E(\tilde{u}, \tilde{v}, \tilde{w}) = \int_{\Omega} [\alpha_1^2 ((\tilde{u} - u_0)^2 + (\tilde{v} - v_0)^2) + \alpha_2^2 (\tilde{w} - w_0)^2] d\Omega \quad (35)$$

siendo α_1 y α_2 los módulos de precisión de Gauss, considerados idénticos para las direcciones horizontales. Los modelos de masa consistente son muy sensibles a los valores elegidos para α_1 y α_2 . Por consiguiente debe prestarse especial atención a este problema. Si dividimos el funcional a minimizar por un valor constante, por ejemplo α_2^2 , éste no resulta distinto excepto a efectos de su minimización. Se introduce entonces el parámetro de estabilidad α ,

$$\alpha = \frac{\alpha_1}{\alpha_2} \quad (36)$$

Obsérvese que los coeficientes α_1 y α_2 son los pesos de los ajustes horizontal y vertical de las componentes de la velocidad. Así para $\alpha \gg 1$ 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 ella. Mientras que para $\alpha \ll 1$, el ajuste del flujo ocurre primeramente en el plano horizontal, por tanto el aire pasará alrededor de las barreras del terreno más que sobre ella. En particular,

$\alpha \rightarrow \infty$ significa ajuste vertical puro, mientras $\alpha \rightarrow 0$ significa ajuste horizontal puro [11].

El campo buscado $\vec{v}(u, v, w)$ será la solución del problema:

“Encontrar $\vec{v} \in K$ tal que,

$$E(\vec{v}) = \min_{\vec{u} \in K} E(\vec{u}), \quad K = \left\{ \vec{u}; \vec{\nabla} \cdot \vec{u} = 0, \vec{n} \cdot \vec{u}|_{\Gamma_b} = 0 \right\} ” \quad (37)$$

Este problema es equivalente a encontrar el punto silla (\vec{v}, ϕ) del Lagrangiano [73],

$$L(\vec{u}, \lambda) = E(\vec{u}) + \int_{\Omega} \lambda \vec{\nabla} \cdot \vec{u} d\Omega \quad (38)$$

La técnica de los multiplicadores de Lagrange permite obtener el punto silla de la expresión (38), $L(\vec{v}, \lambda) \leq L(\vec{v}, \phi) \leq L(\vec{u}, \phi)$, tal que el campo solución \vec{v} se obtiene a partir de las ecuaciones de Euler-Lagrange,

$$\vec{v} = \vec{v}_0 + T \vec{\nabla} \phi \quad (39)$$

siendo ϕ el multiplicador de Lagrange y $T = (T_h, T_h, T_v)$ el tensor diagonal de transmisión

$$T_h = \frac{1}{2\alpha_1^2}, \quad T_v = \frac{1}{2\alpha_2^2} \quad \text{y} \quad \frac{T_v}{T_h} = \alpha^2 \quad (40)$$

Si α_1 y α_2 se consideran constantes en todo el dominio, la formulación variacional conduce a una ecuación elíptica definida en ϕ . En efecto, sustituyendo la ecuación (39) en (33) resulta

$$-\vec{\nabla} \cdot (T \vec{\nabla} \phi) = \vec{\nabla} \cdot \vec{v}_0 \quad (41)$$

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

$$\phi = 0 \quad \text{en} \quad \Gamma_a \quad (42)$$

y una condición de Neumann en las impermeables (terreno y frontera superior)

$$\vec{n} \cdot T \vec{\nabla} \phi = -\vec{n} \cdot \vec{v}_0 \quad \text{en} \quad \Gamma_b \quad (43)$$

Obsérvese que en la frontera superior, al ser el campo inicial \vec{v}_0 horizontal, la condición (43) se transforma en

$$\vec{n} \cdot T \vec{\nabla} \phi = 0 \quad (44)$$

Al considerar T_h y T_v constantes, la ecuación (41) se convierte en

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \alpha^2 \frac{\partial^2 \phi}{\partial z^2} = -\frac{1}{T_h} \left(\frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y} + \frac{\partial w_0}{\partial z} \right) \quad (45)$$

2.2. 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. El campo inicial \vec{v}_0 se construye en tres etapas. 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. Con esta información se realiza una *extrapolación vertical* para definir el campo de velocidades en todo el dominio. Finalmente, la componente vertical del campo de velocidades es corregida en el entorno de posibles fuentes de emisión de contaminantes (chimeneas) con el fin de simular el movimiento de salida de los gases.

2.2.1. Análisis de las medidas de las estaciones

Los datos de viento se toman de estaciones de medida 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.

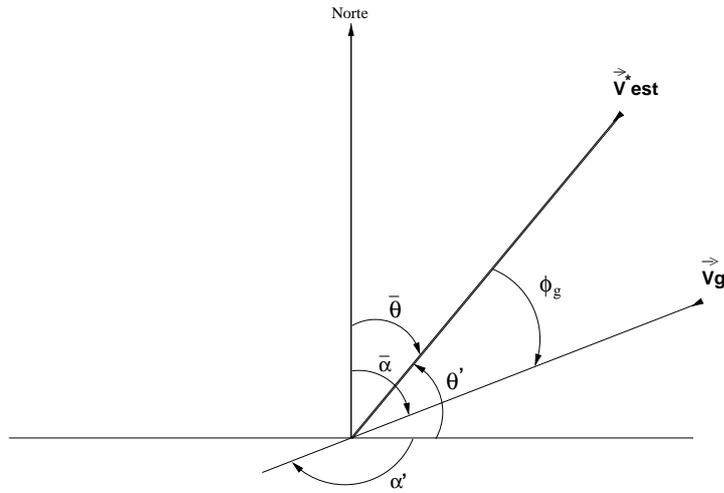


Figura 11: *Ángulo de desviación del viento sobre la superficie respecto al viento geostrofico.*

El cambio se realiza de la siguiente manera (ver figura 11),

$$\bar{\alpha} = \bar{\theta} + \phi_g$$

o de una manera más simplificada

$$\alpha' = 180 + \theta' + \phi_g$$

donde, $\bar{\theta}$ es el ángulo tomado por las estaciones de medida en grados norte en sentido horario y θ' es su complementario. Asumimos que el viento sobre la superficie gira un ángulo ϕ_g con respecto a la dirección del viento geostrofico \vec{V}_g , $\bar{\alpha}$ es el ángulo respecto a la vertical y α' es el ángulo que se desea obtener en sentido antihorario sobre la horizontal.

2.2.2. Interpolación horizontal

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 [73]. Sin embargo, otros autores usan simplemente la altitud de los puntos de medida [52]. Aquí se propone

una fórmula que tiene en cuenta ambas consideraciones,

$$\vec{v}_0(z_e) = \varepsilon \frac{\sum_{n=1}^N \frac{\vec{v}_n}{d_n^2}}{\sum_{n=1}^N \frac{1}{d_n^2}} + (1 - \varepsilon) \frac{\sum_{n=1}^N \frac{\vec{v}_n}{|\Delta h_n|}}{\sum_{n=1}^N \frac{1}{|\Delta h_n|}} \quad (46)$$

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 ε es un parámetro de peso que toma valores entre 0 y 1. Cuando $\varepsilon \rightarrow 1$ 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 $\varepsilon \rightarrow 0$ 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 ε suele ser lo más apropiado.

2.2.3. Extrapolación vertical

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 $f = 2\Theta \text{sen } \phi_l$ se denomina parámetro de Coriolis, siendo $\Theta = 7.292 \times 10^{-5} \text{ s}^{-1}$ 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 geostrofcos.

a. Estratificación atmosférica

En este modelo consideraremos 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 12.

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 [62],

$$z_{pbl} = \frac{\gamma |\bar{v}^*|}{f} \quad (47)$$

siendo γ una constante comprendida entre 0.15 y 0.45 que depende de la estabilidad de la atmósfera y \bar{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

$$h_m = \gamma' \sqrt{\frac{|\bar{v}^*| L}{f}} \quad (48)$$

donde usualmente se toma $\gamma' = 0.4$ [74] y L es la longitud de Monin-Obukov, que se calcula a través de la fórmula de Liu [57],

$$\frac{1}{L} = az_0^b \quad (49)$$

con a y b , definidas por la clase de estabilidad de Pasquill (ver tabla 2).

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 [74],

$$z_{sl} = \frac{h_m}{10} \quad (50)$$

b. 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:

- **Estables:** 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.
- **Inestables:** 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.

- **Neutras:** 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 (1) definida por Pasquill.

Clase de estabilidad de Pasquill					
Velocidad del viento en la superficie (m/s)	Insolación			Noche	
	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 1: 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.

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

Normalmente los anemómetros ofrecen medidas de la intensidad de turbulencia que pueden ayudar a completar la información sobre la clase de estabilidad atmosférica en la zona de medida. La intensidad de turbulencia i se define como la

raíz cuadrada de la suma de varianzas σ_u^2 , σ_v^2 , σ_w^2 , de las tres componentes de la velocidad u_0 , v_0 , w_0 , respectivamente, dividida por la velocidad media del viento que se ha medido,

$$i = \frac{\sqrt{\sigma_u^2 + \sigma_v^2 + \sigma_w^2}}{|\vec{v}_0|} \quad (51)$$

En la práctica disponemos sólo de medidas de variaciones en intensidad de la velocidad y no en la dirección. En estos casos la ecuación (51) se reduce a:

$$i = \frac{\sigma_{v_0}}{|\vec{v}_0|} \quad (52)$$

donde σ_{v_0} representa la desviación típica de las intensidades de viento medidas.

Una atmósfera inestable implica un elevado nivel de turbulencia, con un rango de intensidades de turbulencia entre 0.2 y 0.4 aproximadamente. En cambio, una atmósfera estable, con una turbulencia pequeña o casi cero, se caracteriza por intensidades en el rango 0.05 – 0.01.

La siguiente tabla compatibiliza las relaciones anteriores de la intensidad de turbulencia y la estabilidad atmosférica, con la tabla 3 definida por Pasquill.

Clase de estabilidad de Pasquill							
Velocidad del viento en la superficie (m/s)	Insolación				Noche		
	$i > 0.35$	$0.35 \geq i > 0.25$	$0.25 \geq i > 0.15$	$0.15 \geq i$	$i > 0.075$	$0.075 \geq i > 0.03$	$0.03 \geq i$
$ v_0 < 2$	A	B	B	B	F	F	F
$2 \leq v_0 < 3$	A	B	C	C	E	E	F
$3 \leq v_0 < 5$	B	B	C	C	D	E	E
$ v_0 \geq 5$	C	C	C	D	D	D	D

Tabla 3: Clases de estabilidad de Pasquill según la velocidad del viento en la superficie y la intensidad de turbulencia.

c. Perfil vertical de velocidades de viento

Como se muestra en la figura 12, se considera un perfil logarítmico-lineal [33] en la capa límite planetaria, que tiene en cuenta la interpolación horizontal [45], 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,

$$\vec{v}_0(z) = \frac{\vec{v}^*}{k} \left(\ln \frac{z}{z_0} - \Phi_m \right) \quad z_0 < z \leq z_{sl} \quad (53)$$

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*),

$$\vec{v}^* = \frac{k \vec{v}_0(z_e)}{\ln \frac{z_e}{z_0} - \Phi_m(z_e)} \quad (54)$$

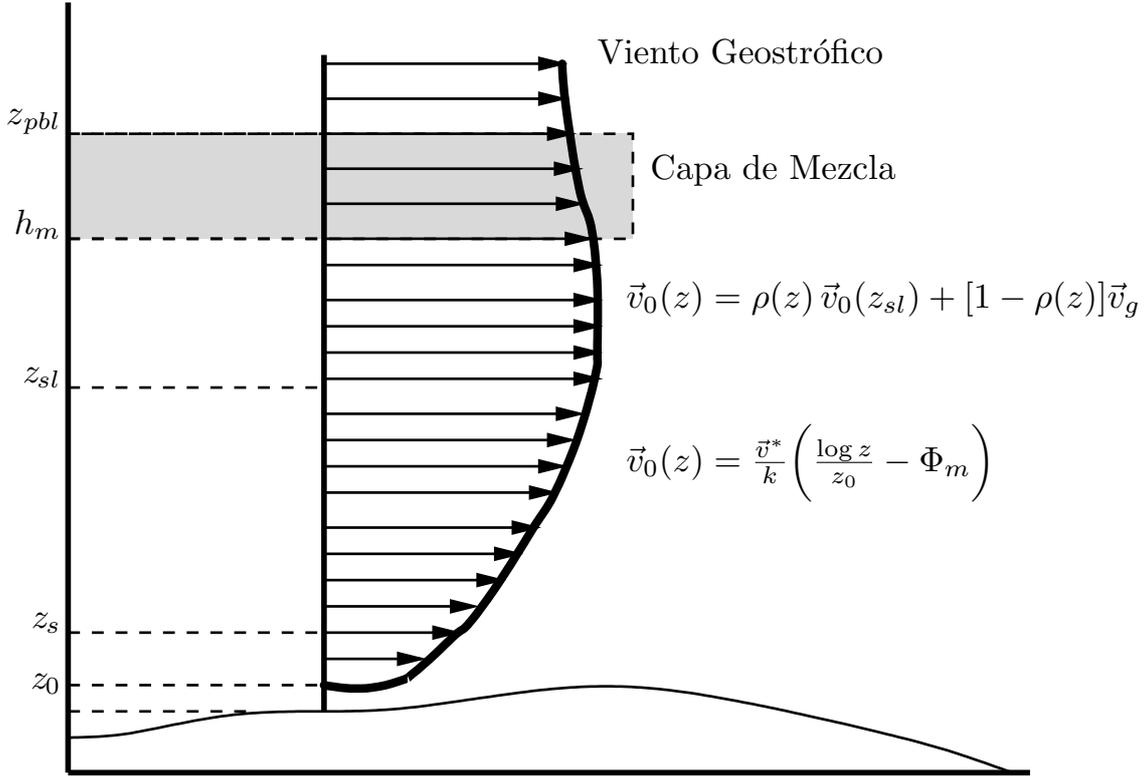


Figura 12: Perfil vertical de viento definido sobre cada capa de la estratificación atmosférica.

Lógicamente, si disponemos de n medidas en la misma vertical de una estación de medida, la expresión anterior daría lugar a n velocidades de fricción diferentes,

$$\vec{v}_i^* = \frac{k \vec{v}_{0_i}(z_{e_i})}{\ln \frac{z_{e_i}}{z_0} - \Phi_m(z_{e_i})} \quad i = 1, \dots, n \quad (55)$$

Para obtener un valor óptimo de \vec{v}^* , realizamos un ajuste de mínimos cuadrados con las velocidades medidas a distintas alturas, considerando fija la velocidad de fricción. Sea

$$A_i = \frac{1}{k} \left(\ln \frac{z_{e_i}}{z_0} - \Phi_m(z_{e_i}) \right) \quad i = 1, \dots, n \quad (56)$$

tal que se tiene

$$\vec{v}_{0_i}(z_{e_i}) = \vec{v}^* A_i \quad i = 1, \dots, n \quad (57)$$

Entonces la función a minimizar resulta,

$$F_{obj} = \sum_{i=1}^n (\vec{v}_{0_i}(z_{e_i}) - \vec{v}_{s_i}(z_{e_i}))^2 = \sum_{i=1}^n (\vec{v}^* A_i - \vec{v}_{s_i}(z_{e_i}))^2 \quad (58)$$

El mínimo de esta función se obtiene para una velocidad de fricción,

$$\vec{v}^* = \frac{\sum_{i=1}^n A_i \vec{v}_{s_i}(z_{e_i})}{\sum_{i=1}^n A_i^2} \quad (59)$$

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 [38]; ver figura 13. 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 [74],

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

donde

$$\theta_m = \left(1 - 16\frac{z}{L} \right)^{1/4} \quad (60)$$

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 law*) [57].

$$|\vec{V}_g| = \frac{|\vec{v}^*|}{k} \sqrt{\left(\ln \frac{|\vec{v}^*|}{f z_0} - A \right)^2 + B^2} \quad (61)$$

Los valores de los coeficientes A y B tienden a ser ~ 1.8 y ~ 4.5 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

$$\phi_g = \arctan2 \left(\ln \frac{|\vec{v}^*|}{f z_0} - A, B \right) \quad (62)$$

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

$$\vec{v}_0(z) = \rho(z) \vec{v}_0(z_{sl}) + [1 - \rho(z)] \vec{v}_g \quad \text{con} \quad z_{sl} < z \leq z_{pbl} \quad (63)$$

donde $\rho(z)$ es

$$\rho(z) = 1 - \left(\frac{z - z_{sl}}{z_{pbl} - z_{sl}} \right)^2 \left(3 - 2 \frac{z - z_{sl}}{z_{pbl} - z_{sl}} \right) \quad (64)$$

Finalmente, este modelo considera

$$\vec{v}_0(z) = \vec{v}_g \quad \text{si} \quad z > z_{pbl} \quad (65)$$

$$\vec{v}_0(z) = 0 \quad \text{si} \quad z \leq z_0 \quad (66)$$

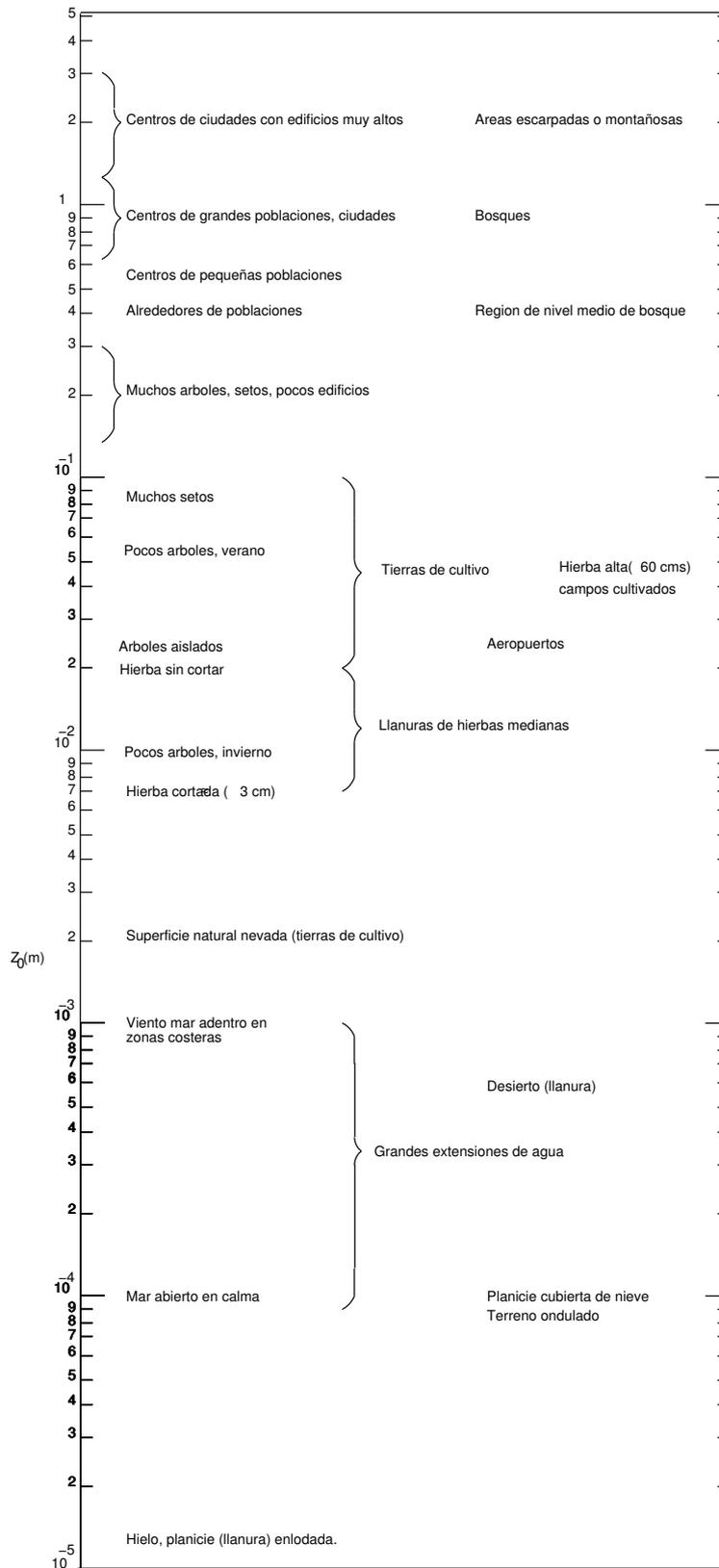


Figura 13: Longitud de rugosidad: valores aproximados de z_0 para distintos tipos de terreno definidos por McRae (1982)

2.3. Discretización mediante elementos finitos

Para la discretización mediante elementos finitos de la formulación clásica del problema dada en (41), (42) y (43) se ha utilizado una malla de tetraedros, generada mediante las técnicas ya descritas, e interpolación lineal. Nótese que en la formulación variacional del problema, las integrales de contorno en la parte de la frontera con condición Neumann se cancelan utilizando la ecuación (43) y las de tipo Dirichlet se eliminan anulando la correspondiente función test.

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 $\hat{\Omega}_e$,

$$\begin{aligned} \{\mathbf{A}^e\}_{ij} = & \int_{\hat{\Omega}_e} \left\{ \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial x} \right) \left(\frac{\partial \hat{\psi}_j}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_j}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_j}{\partial \varphi} \frac{\partial \varphi}{\partial x} \right) + \right. \\ & \left. + \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial y} \right) \left(\frac{\partial \hat{\psi}_j}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_j}{\partial \eta} \frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_j}{\partial \varphi} \frac{\partial \varphi}{\partial y} \right) + \right. \\ & \left. + \frac{T_v}{T_h} \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial z} \right) \left(\frac{\partial \hat{\psi}_j}{\partial \xi} \frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_j}{\partial \eta} \frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_j}{\partial \varphi} \frac{\partial \varphi}{\partial z} \right) \right\} \cdot |\mathbf{J}| \, d\xi \, d\eta \, d\varphi \end{aligned} \quad (67)$$

y de vectores elementales de 4×1 ,

$$\begin{aligned} \{\mathbf{b}^e\}_i = & \int_{\hat{\Omega}_e} -\frac{1}{T_h} \left\{ u_0 \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial x} \right) + \right. \\ & \left. + v_0 \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial y} \right) + \right. \\ & \left. + w_0 \left(\frac{\partial \hat{\psi}_i}{\partial \xi} \frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \eta} \frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \varphi} \frac{\partial \varphi}{\partial z} \right) \right\} \cdot |\mathbf{J}| \, d\xi \, d\eta \, d\varphi \end{aligned} \quad (68)$$

2.4. Resolución del sistema de ecuaciones

La aplicación del método de elementos finitos en este tipo de problemas implica la resolución de grandes sistemas de ecuaciones, que se caracterizan porque la matriz de coeficientes es escasa (*sparse*), ya que muchos de los términos de la matriz de los coeficientes son cero. Ambas características, tamaño y elevado número de términos nulos, justifican la elección de métodos iterativos de resolución de sistemas de ecuaciones. Se ha optado por representar las matrices en un formato compacto denominado morse, lo que permite un gran ahorro de memoria en comparación con la representación estándar. En este formato se utiliza un vector para almacenar los valores no nulos de la matriz, un vector de enteros con tantos elementos como valores no nulos tiene la matriz donde se almacenan las columnas a las que pertenece cada valor, un vector de enteros con tantos elementos como la dimensión de la matriz más uno y un entero que contiene la dimensión original de la matriz. Así, el tamaño de memoria necesario para la representación de una matriz escasa es del orden del número de elementos no nulos que contiene, mientras que la representación estándar sería de orden n^2 , siendo n la dimensión de la matriz.

En nuestro problema tenemos dos tipos de condiciones de contorno. Las de tipo Neumann se introducen en la formulación variacional. En cambio, aquellos nodos con condición de tipo Dirichlet pueden ser eliminados del sistema de ecuaciones. Para ello eliminamos de la matriz del sistema la fila y la columna correspondiente al nodo ya que el valor de la condición Dirichlet es, en este caso, cero.

De esta manera el sistema se convierte en simétrico y, por tanto, puede emplearse el método del gradiente conjugado para resolverlo, con el consiguiente ahorro de tiempo de ejecución, ya que la aplicación del gradiente conjugado resulta más eficiente que otros algoritmos basados en los subespacios de Krylov para sistemas no simétricos. Esto es debido a que utiliza un sólo producto matriz–vector frente a los dos de los otros algoritmos.

Además se han implementado una serie de preconditionadores que permiten mejorar la convergencia de aquellos sistemas que estuvieran mal condicionados.

Una vez calculada la solución del sistema “reducido”, se completa con las aportaciones de los nodos con condición Dirichlet, obteniendo de esta manera la solución completa del sistema de ecuaciones.

Una vez obtenido ϕ como solución del sistema de ecuaciones, se calcula el campo el campo de viento usando la ecuación (39).

2.4.1. Precondicionamiento

La convergencia de los métodos basados en los subespacios de Krylov, y en particular la del gradiente conjugado, mejora con el uso de las técnicas de preconditionamiento. Éstas consisten generalmente en cambiar el sistema original $\mathbf{Ax} = \mathbf{b}$ por otro de idéntica solución, de forma que el número de condicionamiento de la matriz del nuevo sistema sea menor que el de \mathbf{A} , o bien tenga una mejor distribución de autovalores. Generalmente, se considera una matriz de preconditionamiento \mathbf{M}^{-1} , siendo \mathbf{M} una aproximación de \mathbf{A} , esto es,

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$$

tal que, $K(\mathbf{M}^{-1}\mathbf{A}) < K(\mathbf{A})$.

El menor valor corresponde a $\mathbf{M} = \mathbf{A}$, $K(\mathbf{A}^{-1}\mathbf{A}) = 1$, que es el caso ideal y el sistema convergería en una sola iteración, pero el coste computacional del cálculo de \mathbf{A}^{-1} equivaldría a resolver el sistema por un método directo. Se sugiere que \mathbf{M} que sea una matriz lo más próxima a \mathbf{A} sin que su determinación suponga un coste elevado.

Por otro lado, la matriz \mathbf{M} debe ser fácilmente invertible para poder efectuar los productos \mathbf{M}^{-1} por vector que aparecen en los algoritmos preconditionados sin excesivo coste adicional.

Dependiendo de la forma de plantear el producto de \mathbf{M}^{-1} por la matriz del sistema obtendremos distintas formas de preconditionamiento. Éstas son,

$$\begin{aligned} \mathbf{M}^{-1}\mathbf{Ax} &= \mathbf{M}^{-1}\mathbf{b} && \text{(Precondicionamiento por la izquierda)} \\ \mathbf{AM}^{-1}\mathbf{Mx} &= \mathbf{b} && \text{(Precondicionamiento por la derecha)} \\ \mathbf{M}_1^{-1}\mathbf{AM}_2^{-1}\mathbf{M}_2\mathbf{x} &= \mathbf{M}_1^{-1}\mathbf{b} && \text{(Precondicionamiento por ambos lados)} \end{aligned} \quad (69)$$

si \mathbf{M} puede ser factorizada como $\mathbf{M} = \mathbf{M}_1\mathbf{M}_2$.

El campo de posibles preconditionadores es muy amplio. Algunos de los más usados son el de Jacobi, SSOR, $\mathbf{IC}(\mathbf{0})$ y el Diagonal Óptimo.

a. Precondicionador de Jacobi

Surge comparando la fórmula de recurrencia para la solución que resulta de aplicar el método de Richardson, cuya relación de recurrencia viene dada por $\mathbf{x}_{i+1} = \mathbf{x}_i + \mu(\mathbf{b} - \mathbf{A}\mathbf{x}_i)$, con $\mu > 0$, al sistema preconditionado con la fórmula correspondiente que se obtiene aplicando el método de Jacobi al sistema sin preconditionar. De la aplicación del método de Richardson al sistema preconditionado $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$, se obtiene para el cálculo de los sucesivos valores de la solución,

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mu(\mathbf{M}^{-1}\mathbf{b} - \mathbf{M}^{-1}\mathbf{A}\mathbf{x}_i)$$

Multiplicando por la matriz de preconditionamiento \mathbf{M} , queda,

$$\mathbf{M}\mathbf{x}_{i+1} = \mathbf{M}\mathbf{x}_i + \mu(\mathbf{b} - \mathbf{A}\mathbf{x}_i) \quad (70)$$

Por otro lado, descomponiendo la matriz del sistema en $\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$, (\mathbf{D} matriz diagonal formada por los elementos de la diagonal de \mathbf{A} y \mathbf{E} y \mathbf{F} matrices triangulares inferior y superior respectivamente), y utilizando el método de Jacobi para la resolución del sistema, se obtiene,

$$\mathbf{x}_{i+1} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\mathbf{x}_i + \mathbf{D}^{-1}\mathbf{b}$$

que multiplicando por \mathbf{D} y operando resulta,

$$\mathbf{D}\mathbf{x}_{i+1} = \mathbf{D}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i) \quad (71)$$

Comparando las expresiones de recurrencia finales de ambos métodos, se observa que el método de Jacobi aplicado al sistema sin preconditionar, equivale al de Richardson, con $\alpha = 1$, menos robusto y más simple, cuando este se aplica al sistema preconditionado con la matriz diagonal $\mathbf{D} = \text{diag}(\mathbf{A})$.

b. Precondicionador SSOR

Si aplicamos el método SSOR al sistema sin preconditionar, considerando la descomposición de la matriz \mathbf{A} en $\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$, como en el caso anterior, y siendo ω el parámetro de relajación, se obtiene,

$$\begin{aligned} & \frac{1}{\omega(2-\omega)} (\mathbf{D} - \omega\mathbf{E}) \mathbf{D}^{-1} (\mathbf{D} - \omega\mathbf{F}) \mathbf{x}_{i+1} \\ &= \frac{1}{\omega(2-\omega)} (\mathbf{D} - \omega\mathbf{E}) \mathbf{D}^{-1} (\mathbf{D} - \omega\mathbf{F}) \mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i) \end{aligned} \quad (72)$$

con lo que resulta como matriz de preconditionamiento,

$$\mathbf{M} = (\mathbf{I} - \omega\mathbf{E}\mathbf{D}^{-1}) \left(\frac{\mathbf{D} - \omega\mathbf{F}}{\omega(2-\omega)} \right) \quad (73)$$

que en el caso de sistemas simétricos, podemos expresarla como,

$$\mathbf{M} = \left[\frac{(\mathbf{D} - \omega\mathbf{E}) \mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}} \right] \left[\frac{(\mathbf{D} - \omega\mathbf{E}) \mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}} \right]^T \quad (74)$$

c. Precondicionador IC(0)

Resulta de la aproximación de \mathbf{A} por una factorización incompleta de Cholesky, conservando las mismas entradas nulas en las matrices triangulares \mathbf{L} y \mathbf{L}^T ,

$$\mathbf{A} \approx \mathbf{IC}(0) = \mathbf{M} \quad (75)$$

donde $l_{ij} = l_{ji}$ son las entradas de las matrices triangulares incompletas tal que,

$$l_{ij} = 0 \quad \text{si} \quad a_{ij} = 0 \quad (76)$$

$$\{\mathbf{A} - \mathbf{LL}^T\}_{ij} = 0 \quad \text{si} \quad a_{ij} \neq 0 \quad (77)$$

Es decir, que los elementos nulos de la matriz del sistema siguen siendo nulos en las posiciones respectivas de las matrices triangulares para no incrementar el coste computacional.

d. Precondicionador diagonal óptimo

Se trata de un caso particular de inversa aproximada de estructura diagonal. Si se ha precondicionado por la izquierda, el mejor precondicionador diagonal del sistema resulta,

$$\mathbf{M} = \text{diag} \left(\frac{a_{11}}{\|\mathbf{e}_1^T \mathbf{A}\|_2^2}, \frac{a_{22}}{\|\mathbf{e}_2^T \mathbf{A}\|_2^2}, \dots, \frac{a_{nn}}{\|\mathbf{e}_n^T \mathbf{A}\|_2^2} \right) \quad (78)$$

$$\|\mathbf{MA} - \mathbf{I}\|_F^2 = n - \sum_{i=1}^n \frac{a_{ii}}{\|\mathbf{e}_i^T \mathbf{A}\|_2^2} \quad (79)$$

De forma similar se puede definir el precondicionador por la derecha.

e. Precondicionador basado en una inversa aproximada factorizada

Buscamos un precondicionador para el sistema de matriz variable simétrica definida positiva, $(M + \varepsilon N)x = b$, para su posterior aplicación en el algoritmo del Gradiente Conjugado precondicionado. Seguiremos una línea similar a la propuesta por Benzi [6] para el sistema $(M + \varepsilon I)x = b$.

Con el algoritmo SAINV obtenemos una inversa aproximada factorizada de M ,

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

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

$$P_\varepsilon^{-1} = \tilde{Z} \left(\tilde{D} + \varepsilon E \right)^{-1} \tilde{Z}^T, \quad (80)$$

donde E es una matriz genérica simétrica a determinar, fácilmente computable, que haga que $(\tilde{D} + \varepsilon E)$ sea SDP y tal que el producto P_ε^{-1} por vector que figura en el CG no suponga un coste elevado.

A efectos de definirla y supuesto que $P^{-1} = ZD^{-1}Z^T$ sea la inversa exacta, establezcamos la diferencia,

$$P_\varepsilon - A_\varepsilon = Z^{-T} (D + \varepsilon E) Z^{-1} - (M + \varepsilon N) = \varepsilon (Z^{-T} E Z^{-1} - N). \quad (81)$$

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

$$E = \tilde{Z}^T N \tilde{Z} \quad (82)$$

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

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

y

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

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

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

$$E_{h,k} = \tilde{Z}_k^T N_h \tilde{Z}_k \quad (83)$$

En la práctica y a efectos de no incrementar el coste por iteración en el CG, parecen ser viables las parejas $h = 1$ y $k = 2$ o $h = 2$ y $k = 1$, que dan lugar, respectivamente, a las matrices $E_{1,2}$ y $E_{2,1}$, tridiagonales. Más simplificación se obtendría con $h = 1$ y $k = 1$, con $E_{1,1}$ matriz diagonal.

Otra posibilidad, para valores pequeños de ε , siempre a efectos de reducir el coste computacional, es efectuar un desarrollo de $\left(\tilde{D} + \varepsilon E \right)^{-1}$,

$$\begin{aligned} \left(\tilde{D} + \varepsilon E \right)^{-1} &= \left[\tilde{D}^{\frac{1}{2}} \left(I + \varepsilon \tilde{D}^{-\frac{1}{2}} E \tilde{D}^{-\frac{1}{2}} \right) \tilde{D}^{\frac{1}{2}} \right]^{-1} \\ &= \tilde{D}^{-\frac{1}{2}} \left[I - \varepsilon \tilde{D}^{-\frac{1}{2}} E \tilde{D}^{-\frac{1}{2}} + \left(\varepsilon \tilde{D}^{-\frac{1}{2}} E \tilde{D}^{-\frac{1}{2}} \right)^2 + \dots \right] \tilde{D}^{-\frac{1}{2}} \end{aligned} \quad (84)$$

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

$$P_\varepsilon^{-1} = \tilde{Z} \tilde{D}^{-\frac{1}{2}} \left(I - \varepsilon \tilde{D}^{-\frac{1}{2}} E \tilde{D}^{-\frac{1}{2}} \right) \tilde{D}^{-\frac{1}{2}} \tilde{Z}^T \quad (85)$$

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

$$P_\varepsilon^{-1} = \tilde{Z} \tilde{D}^{-1} \tilde{Z}^T \left(I - \varepsilon N \tilde{Z} \tilde{D}^{-1} \tilde{Z}^T \right) \quad (86)$$

$$P_\varepsilon^{-1} = P^{-1} \left(I - \varepsilon N P^{-1} \right) \quad (87)$$

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

$$P_\varepsilon^{-1} = P_{\varepsilon_0}^{-1} \left(I - \Delta\varepsilon N P_{\varepsilon_0}^{-1} \right) \quad (88)$$

2.4.2. Algoritmo de gradiente conjugado precondicionado

Algoritmo 2.1 Gradiente Conjugado precondicionado (PCG).

Aproximación inicial \mathbf{x}_0 . $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;
 Resolver $\mathbf{M}\mathbf{z}_0 = \mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{z}_0$;
Mientras $\|\mathbf{r}_j\| / \|\mathbf{r}_0\| \geq \varepsilon$ ($j = 0, 1, 2, 3, \dots$) **Hacer**
 $\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}{\langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j \rangle}$;
 $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$;
 $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j$;
 Resolver $\mathbf{M}\mathbf{z}_{j+1} = \mathbf{r}_{j+1}$;
 $\beta_j = \frac{\langle \mathbf{r}_{j+1}, \mathbf{z}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}$;
 $\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j$;
Fin Mientras

El método del Gradiente Conjugado está basado en una técnica de proyección ortogonal sobre el subespacio de Krylov $\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ donde \mathbf{r}_0 es el residuo inicial, y fundamentado en el algoritmo D-Lanczos de la siguiente forma. Una aproximación \mathbf{x}_{j+1} puede ser expresada,

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j \quad (89)$$

entonces, los vectores residuos deben satisfacer que,

$$\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j \quad (90)$$

Estos vectores residuos son ortogonales,

$$\langle \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j, \mathbf{r}_j \rangle = 0$$

por tanto,

$$\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}{\langle \mathbf{A}\mathbf{p}_j, \mathbf{r}_j \rangle} \quad (91)$$

Como las siguientes direcciones \mathbf{p}_{j+1} son una combinación lineal de \mathbf{r}_{j+1} y \mathbf{p}_j , después de reescalar los vectores \mathbf{p} apropiadamente, se tiene que

$$\mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j \quad (92)$$

con lo cual,

$$\langle \mathbf{A}\mathbf{p}_j, \mathbf{r}_j \rangle = \langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j - \beta_{j-1} \mathbf{p}_{j-1} \rangle = \langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j \rangle$$

ya que $\mathbf{A}\mathbf{p}_j$ es ortogonal a \mathbf{p}_{j-1} . Entonces, teniendo en cuenta (91) y (92) obtenemos que,

$$\beta_j = -\frac{\langle \mathbf{r}_{j+1}, \mathbf{A}\mathbf{p}_j \rangle}{\langle \mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle}$$

y como de (90), podemos escribir,

$$\mathbf{A}\mathbf{p}_j = -\frac{1}{\alpha_j} (\mathbf{r}_{j+1} - \mathbf{r}_j) \quad (93)$$

entonces,

$$\beta_j = -\frac{1}{\alpha_j} \frac{\langle \mathbf{r}_{j+1}, (\mathbf{r}_{j+1} - \mathbf{r}_j) \rangle}{\langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j \rangle} = \frac{\langle \mathbf{r}_{j+1}, \mathbf{r}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}$$

2.5. Refinamiento adaptativo

En la actualidad, la mayor parte de los programas que utilizan el método de elementos finitos se apoyan en técnicas adaptables basadas en una estimación del error cometido con nuestra solución numérica, o al menos en indicadores de error fiables que nos señalen los elementos que deben ser refinados o desrefinados en la malla.

En la generación de mallas adaptables podemos considerar dos aspectos diferentes: la discretización del dominio atendiendo a su geometría o a la solución numérica, mediante estimadores de error o mediante indicadores de error. Existen muchas formas de abordar estos aspectos. La primera cuestión es: ¿mallas estructuradas o no estructuradas?. En este sentido, está claro que el uso de mallas no estructuradas nos proporciona más flexibilidad a la hora de mallar geometrías complejas utilizando un número óptimo de nodos. En este caso, los métodos más clásicos para la obtención de triangulaciones tridimensionales se basan fundamentalmente en algoritmos de avance frontal [37] o en algoritmos basados en la triangulación de Delaunay [25; 18]. Una vez que se ha discretizado la geometría del dominio, la malla debe adaptarse atendiendo a las singularidades de la solución numérica. Este proceso implica la introducción (refinamiento) o eliminación (desrefinamiento) de nodos de la malla actual. Por esta razón, se debe definir una nueva malla. Los cambios pueden afectar a la malla actual de forma local o global, dependiendo del método de triangulación elegido. Diferentes estrategias de refinamiento han sido desarrolladas para triangulaciones en 2-D, y han sido generalizadas a 3-D. Si se ha optado por un refinamiento que afecte localmente a la malla actual, cabe plantearse otra cuestión: ¿mallas encajadas o no encajadas?. La respuesta en este caso no es tan clara. El uso de mallas encajadas tiene varias ventajas importantes. Podemos conseguir familias de secuencias de mallas encajadas en un mínimo tiempo de CPU. Además, se puede aplicar más fácilmente el método multimalla para resolver el sistema de ecuaciones asociado al problema. Por otra parte, se puede controlar automáticamente la suavidad y la degeneración de la malla, y el mantenimiento de las superficies definidas en el dominio, en función de las características de la malla inicial. Si el dominio posee una geometría compleja, un buen modo de proceder es obtener la malla inicial empleando un generador de mallas no estructuradas y, posteriormente, aplicar una técnica de refinamiento y desrefinamiento de mallas encajadas atendiendo a un indicador de error apropiado al problema. Además, si tratamos de resolver un problema evolutivo, podemos aproximar automáticamente cualquier solución inicial definida en el dominio. Con la técnica de refinamiento o desrefinamiento conseguimos un óptimo soporte de interpolación a trozos capaz de aproximar esta solución con la precisión deseada. En general, podría aplicarse esta técnica para cualquier función definida en el dominio de forma discreta o analítica.

Con estas ideas, anteriormente se desarrollaron técnicas adaptables en 2-D obteniendo buenos resultados en diferentes problemas estacionarios y evolutivos, véase por ejemplo [19; 55; 43; 73]. En estos trabajos se utilizó una versión del algoritmo de refinamiento local 4-T [58]. La elección particular del algoritmo de refinamiento es muy importante, puesto que el algoritmo de desrefinamiento puede entenderse como el inverso del algoritmo de refinamiento. El algoritmo de refinamiento 4-T de Rivara posee buenas propiedades en cuanto a la suavidad y degeneración de la malla. Además de esto, el número de posibilidades que aparecen en la relación entre un elemento padre y sus hijos es menor que con otros algoritmos de refinamiento en 2-D,

tras asegurar la conformidad de la malla. Por ejemplo, sería más complicado desarrollar un algoritmo de desrefinamiento, acoplado con el algoritmo de refinamiento local propuesto en [3]; todos los triángulos que deben ser refinados, atendiendo al indicador de error, se dividen en cuatro subtriángulos mediante la introducción de un nuevo nodo en los centros de sus lados y uniéndolos entre sí.

En 3-D, el problema es diferente. Aunque parezca paradójico, la extensión de un algoritmo adaptable que sea más simple que otro en 2-D, no tiene porqué ser también más simple en 3-D. Así, entre los algoritmos de refinamiento desarrollados en 3-D podemos mencionar los que se basan en la bisección del tetraedro [1; 59; 56], y los que utilizan la subdivisión en 8-subtetraedros [7; 36; 37]. En concreto, el algoritmo desarrollado en [56] se puede entender como la generalización a 3-D del algoritmo 4-T de Rivara, que a su vez está basado en la bisección del triángulo por su lado mayor. El problema que se produce en esta extensión a 3-D es el gran número de casos posibles en los que puede quedar dividido un tetraedro, respetando las diferentes posibilidades de la división 4-T en sus cuatro caras, durante el proceso de conformidad de la malla. Sin embargo, los algoritmos analizados en [7; 36; 37], que a su vez generalizan a 3-D la partición en cuatro subtriángulos propuesta en [3], son más sencillos debido a que el número de particiones posibles de un tetraedro es mucho menor que en el caso de la generalización del algoritmo 4-T. Por otra parte, puesto que la calidad de la malla está asegurada en todos estos casos, hemos optado por implementar una versión del algoritmo que utiliza la subdivisión en 8-subtetraedros.

El algoritmo de refinamiento que proponemos está basado en la subdivisión en 8-subtetraedros desarrollada en [37]. Consideremos una triangulación inicial τ_1 del dominio formada por un conjunto de n_1 tetraedros $t_1^1, t_2^1, \dots, t_{n_1}^1$. Nuestro objetivo es construir una secuencia de m niveles de mallas encajadas $T = \{\tau_1 < \tau_2 < \dots < \tau_m\}$, tal que el nivel τ_{j+1} se obtiene mediante un refinamiento local del nivel anterior τ_j . El indicador de error ϵ_i^j asociado al elemento $t_i^j \in \tau_j$ que se ha utilizado es de tipo gradiente y se define de la forma siguiente,

$$\epsilon_i^j = (d_i)^p \left| \vec{\nabla} \phi_h \right| \quad (94)$$

donde p es un parámetro que se toma generalmente igual a 1 ó 2, y d_i , la longitud de la arista mayor del tetraedro t_i^j . Obsérvese que si $p = 1$, y se considera una interpolación lineal en los elementos de τ_j , entonces ϵ_i^j representa una cota superior de la máxima variación de ϕ_h en el elemento t_i^j . Una vez calculado el indicador de error ϵ_i^j , dicho elemento debe ser refinado si $\epsilon_i^j \geq \theta \epsilon_{\max}^j$, siendo $\theta \in [0, 1]$ el parámetro de refinamiento y ϵ_{\max}^j el máximo valor de los indicadores de error de los elementos de τ_j . Desde un punto de vista constructivo, plantaremos inicialmente la obtención de τ_2 partiendo de la malla base τ_1 , atendiendo a las siguientes consideraciones:

- a) *Subdivisión en 8-subtetraedros.* Decimos que $t_i^1 \in \tau_1$ es un tetraedro de *tipo I* si se verifica que $\epsilon_i^1 \geq \gamma \epsilon_{\max}^1$. Este conjunto de tetraedros serán posteriormente subdivididos en 8 subtetraedros según la figura 14(a); se introducen 6 nuevos nodos en el punto medio de sus aristas y se subdividen cada una de sus cuatro caras en cuatro subtriángulos según la división propuesta en [3]. Así, cuatro subtetraedros quedan determinados a partir de los cuatro vértices de t_i^1 y las nuevas aristas, y los otros cuatro subtetraedros se obtienen al unir los dos vértices opuestos más cercanos del octoedro que resulta en el interior de t_i^1 . Esta sencilla estrategia es la que se propone en [37] o en [7], frente a

otras basadas en transformaciones afines a un tetraedro de referencia, como la analizada en [36], que aseguran la calidad de los tetraedros resultantes. Sin embargo, resultados comparables fueron obtenidos en [7] con ambas estrategias en sus experimentos numéricos. Por otro lado, para otros autores [37], nuestra elección produce el menor número de tetraedros distorsionados en la malla refinada. Evidentemente, siempre se podría determinar la mejor de las tres opciones existentes para la subdivisión del octoedro interior mediante el análisis de la calidad de sus cuatro subtetraedros, pero esto aumentaría el coste computacional del algoritmo.

Una vez definido el tipo de partición de los tetraedros *tipo I*, nos podemos encontrar con tetraedros vecinos que pueden tener 6, 5, ..., 1 ó 0 nuevos nodos introducidos en sus aristas, los cuales debemos tener en cuenta para asegurar la conformidad de la malla. Cada uno de estos casos son analizados a continuación. Cabe destacar que en todo este proceso únicamente estamos marcando las aristas de los tetraedros de τ_1 en las que se ha introducido un nuevo nodo. En base al número de aristas marcadas se clasifica el correspondiente tetraedro, es decir, hasta que no está asegurada la conformidad de τ_2 , simplemente marcando aristas, no se procederá a la definición de esta nueva malla.

- b) *Tetraedros con 6 nuevos nodos.* Aquellos tetraedros que por razones de conformidad tengan marcadas sus 6 aristas pasan automáticamente al conjunto de tetraedros *tipo I*.
- c) *Tetraedros con 5 nuevos nodos.* Aquellos tetraedros que tengan 5 aristas marcadas pasan también al conjunto de tetraedros *tipo I*. Previamente, habrá que marcar la arista en la que no había sido introducido ningún nuevo nodo.
- d) *Tetraedros con 4 nuevos nodos.* En este caso, se marcan las dos aristas restantes y pasa a ser considerado de *tipo I*.

Al proceder de la forma indicada en (b), (c) y (d) debido al refinamiento global considerado en (a) atendiendo al indicador de error, mejoramos la calidad de la malla y simplificamos considerablemente el algoritmo. Puede pensarse que este procedimiento puede aumentar la zona refinada, pero hay que tener en cuenta que sólo se introducirá 1 ó 2 nuevos nodos sobre un total de 6. Obsérvese que esta proporción es igual o inferior a la que surge en el refinamiento en 2-D del algoritmo 4-T de Rivara para un triángulo, en el que la probabilidad de encontrarnos con un nodo introducido en el lado mayor es 1/3. Este fenómeno se acentúa en el algoritmo propuesto como su generalización a 3-D.

- e) *Tetraedros con 3 nuevos nodos.* En este caso, hay que distinguir dos situaciones:
 - e.1. Si las 3 aristas marcadas correspondientes no están sobre la misma cara, entonces se marcan las 3 restantes y el tetraedro se introduce en el conjunto de tetraedros *tipo I*. Aquí podemos hacer la consideración antes mencionada, al comparar este paso con otros algoritmos basados en bisección por el lado mayor.

En los siguientes casos, ya no marcaremos ninguna nueva arista, lo cual implica que no se introducirá ningún nuevo nodo en el tetraedro que se pretende hacer conforme. Se procederá a subdividirlos de la forma que se indica a continuación, creando subtetraedros que llamaremos *transitorios*.

- e.2. Si las 3 aristas marcadas están sobre la misma cara del tetraedro, entonces se crearán 4-subtetraedros transitorios como se muestra en la figura 14(b); se crean nuevas aristas uniendo entre sí los tres nuevos nodos y conectando éstos con el vértice opuesto a la cara que los contiene. Los tetraedros de τ_1 con estas características se englobarán en el conjunto de tetraedros de *tipo II*.
- f) *Tetraedros con 2 nuevos nodos*. En este caso, también distinguiremos dos situaciones:
- f.1. Si las dos aristas marcadas no están sobre la misma cara, entonces se construirán 4-subtetraedros transitorios como se presenta en la figura 14(c), definidos a partir de las aristas que conectan los dos nuevos nodos y a éstos con los vértices opuestos de las dos caras que los contienen. Los tetraedros que se encuentren en esta situación se denominan de *tipo III.a*.
- f.2. Si las dos aristas marcadas están sobre la misma cara, entonces se crearán 3-subtetraedros transitorios según se expone en la figura 14(d); se divide en tres subtriángulos la cara definida por las dos aristas marcadas, conectando el nuevo nodo situado en la arista mayor de éstas dos con el vértice opuesto y con el otro nuevo nodo, tal que estos tres subtriángulos y el vértice opuesto a la cara que los contiene definen los tres nuevos subtetraedros. Se destaca que de las dos posibles elecciones, se toma como referencia la mayor arista marcada para aprovechar en algunos casos las propiedades de la bisección por el lado mayor. Los tetraedros que se encuentren en esta situación se denominan de *tipo III.b*.
- g) *Tetraedros con 1 nuevo nodo*. Se crearán dos subtetraedros transitorios según la figura 14(e); se une el nuevo nodo con los otros dos que no pertenecen a la correspondiente arista marcada. Este conjunto de tetraedros se denomina de *tipo IV*.
- h) *Tetraedros con ningún nuevo nodo*. Estos tetraedros de τ_1 no se dividen y serán heredados a la malla refinada τ_2 . Los denominaremos de *tipo V*; véase la figura 14(f).

Este proceso de clasificación de los tetraedros de τ_1 se realiza simplemente marcando sus aristas. La conformidad de la malla se va asegurando a nivel local por vecindad entre los tetraedros que contienen una nueva arista marcada, mediante un proceso de expansión que comienza en los tetraedros de *tipo I* definidos en el apartado (a). Esto hace que, al finalizar el recorrido de este subconjunto de tetraedros de *tipo I*, la malla resultante sea conforme y refinada localmente. Además, el proceso resulta de un bajo coste computacional, ya que el proceso de expansión local finaliza cuando nos encontramos con tetraedros en los que no se tiene que marcar ninguna nueva arista.

En general, cuando queremos refinar el nivel τ_j en el que ya existen tetraedros *transitorios* se procederá de igual forma que en el paso de τ_1 a τ_2 , con la siguiente variante: desde el momento en que se tenga que marcar una arista de un tetraedro transitorio que no coincida con una arista de su padre, bien porque tenga que ser refinado atendiendo al indicador de error o bien por razones de conformidad, entonces se eliminan (proceso de borrado) todos los tetraedros transitorios “hijos” de su

tetraedro “padre”, marcamos todas las aristas de éste “padre”, se introduce en el conjunto de tetraedros de *tipo I* y se procede a su división. Si por razones de conformidad debiéramos marcar una arista que compartan un tetraedro transitorio y su padre, entonces se eliminan los “hijos” de su tetraedro “padre” y se debe analizar el tipo resultante del tetraedro “padre” para su posterior división.

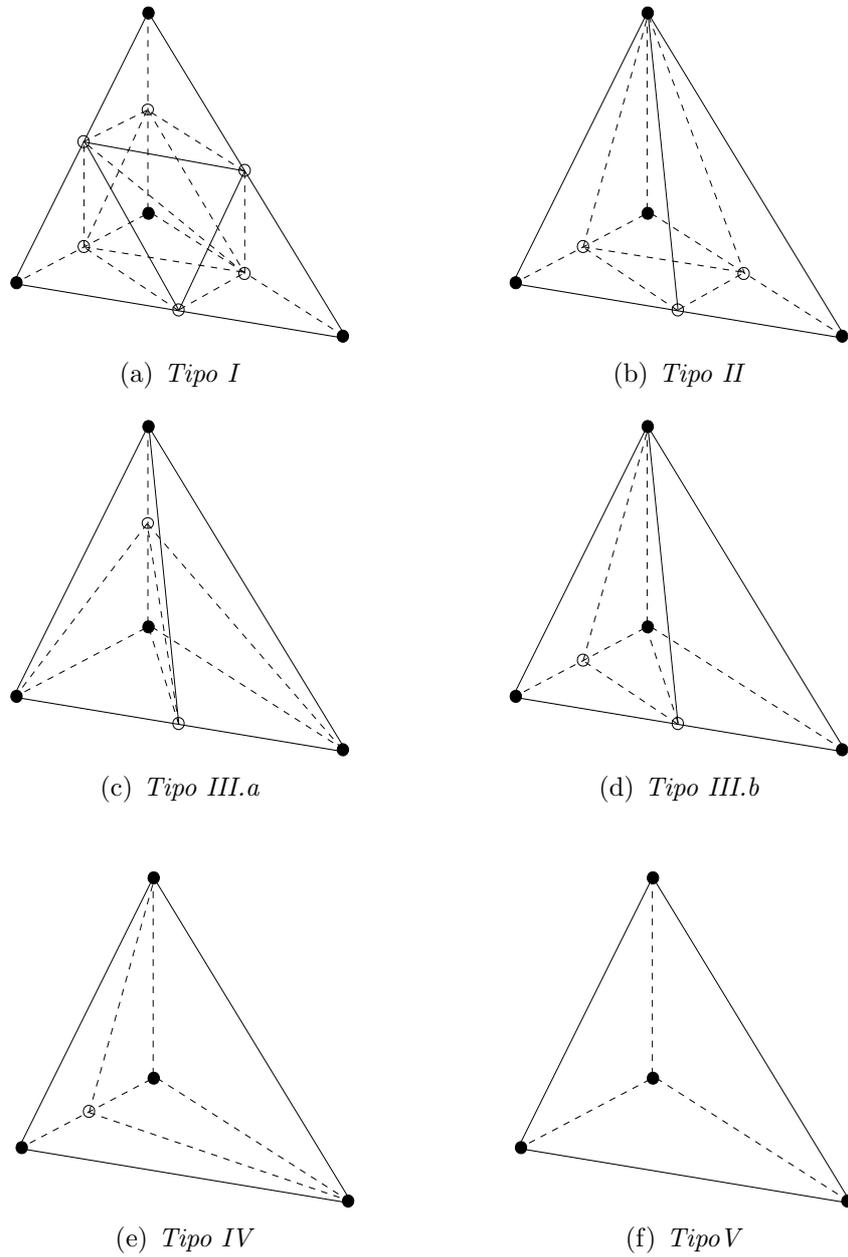


Figura 14: Clasificación de las subdivisiones de un tetraedro en función de los nuevos nodos (círculos huecos).

3. Estimación de parámetros

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. Nosotros hemos planteado 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 [47].

3.1. Definición del problema

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 una mejora sustancial sobre la propuesta en [4] para la estimación de uno solo de los parámetros por un procedimiento de ensayo y error. Nosotros proponemos extenderlo a un total de cuatro de los parámetros que intervienen en el modelo y automatizar su cálculo.

En primer lugar se considera el parámetro de estabilidad

$$\alpha = \frac{\alpha_1}{\alpha_2} = \sqrt{\frac{T_v}{T_h}} \quad (95)$$

que se deriva del funcional (35) y cuyo mínimo no varía si se divide por α_2^2 . Hay que señalar que para $\alpha \gg 1$ predomina el ajuste de viento en la dirección vertical, mientras que para $\alpha \ll 1$ el ajuste tiene lugar predominantemente sobre el plano horizontal. Por lo tanto la elección de α 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 α , 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 [63], [30] y [9], los autores proponen tomar $\alpha = 10^{-2}$, o sea, proporcional a la magnitud de w/u . Otros, como [61] y [50], relacionan α con el número de Froude, mientras que [24], [34] y [67], proponen que el parámetro α varíe en la dirección vertical. Finalmente, [4] 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 α . 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 α que sólo son válidos para cada caso particular y, por tanto, no proporciona valores válidos *a priori* para otras simulaciones. Aquí se estudia una versión del método propuesto en [4], utilizando algoritmos genéticos como herramienta de optimización que permite una selección automática de α .

El segundo parámetro que va a estimarse es el coeficiente de peso ε ($0 \leq \varepsilon \leq 1$) de la ecuación (46), correspondiente a la interpolación horizontal de las medidas de viento observadas. Cuando $\varepsilon \rightarrow 1$ adquiere más importancia la *distancia horizontal* de cada punto a las estaciones de medida, mientras que para $\varepsilon \rightarrow 0$ se da más peso a la *distancia vertical* entre cada punto y las estaciones [45]. En general, para terrenos complejos se utiliza la segunda aproximación [52]. 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 ε .

El siguiente parámetro objeto de estimación es γ , que aparece en la ecuación (47) y está relacionado con la capa límite planetaria en la estratificación atmosférica. Existen diferentes autores que proponen distintos rangos para este parámetro. [53] proponen el intervalo [0.15,0.25]. Sin embargo, en [57] se utiliza directamente el valor $\gamma = 0.3$ en el código de su programa *WINDS*, mientras que para [11] 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, vease (48). 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.

Los parámetros ε , γ y γ' intervienen en la interpolación del viento inicial, mientras que α afecta al cálculo del viento resultante.

3.2. Algoritmos genéticos

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. Hemos basado este trabajo en el modelo de AG desarrollado por [35]. 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, 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 15 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 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

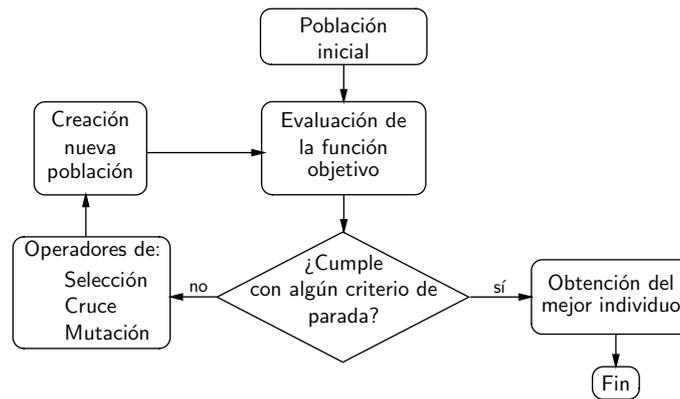


Figura 15: Diagrama del funcionamiento de los AG.

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 lo suficiente complejos como para proporcionar robustos mecanismos de búsqueda de un óptimo.

Normalmente se usan dos técnicas de reemplazamiento de la población. La primera, llamada reemplazo generacional, sustituye la población entera cada vez [29]. La segunda, que se conoce como de estado estacionario (*steady-state*), únicamente reemplaza unos pocos individuos en cada generación [64; 71; 70]. Como criterios de parada se emplean los siguientes: número máximo de iteraciones, población demasiado homogénea y el hecho de que no cambie la mejor solución en un número dado de iteraciones. La población inicial normalmente se genera de forma aleatoria y la única propiedad que ha de verificar es que sus elementos sean lo suficientemente diversos.

La fase de selección crea una población intermedia a partir de la población actual, que estará formada por copias de los individuos que pueden optar a ser progenitores. Los que tengan mayor número de copias en la población intermedia tendrán mayor probabilidad de ser elegidos para la reproducción. En este trabajo se han considerado cuatro esquemas de selección [35]:

Selección Proporcional (P) La probabilidad de que un individuo sea seleccionado para formar parte de la población intermedia es proporcional a la evaluación de la función objetivo para ese individuo.

Selección Universal Estocástica (SU) Los individuos se escogen de la siguiente manera: imaginemos una ruleta donde la casilla que representa a cada individuo de la población es proporcional a la probabilidad de ser escogido como reproductor y ésta a su vez proporcional al resultado de la función objetivo. Esta ruleta tiene tantas flechas equiespaciadas como individuos haya que seleccionar. Realizando un único giro de las flechas de la ruleta se seleccionan los individuos. Este método [2] tiene la ventaja de que es más fácil de implementar que el de selección proporcional y su ejecución es más rápida.

Selección por Torneo Binario (BT) Consiste en elegir aleatoriamente dos individuos de la población y de entre ellos se selecciona el mejor, repitiendo el proceso hasta llenar la población intermedia.

Selección por Torneo Binario Probabilístico (PBT) Consiste en escoger dos individuos al azar y de ellos seleccionar al mejor con una probabilidad p , $0.5 < p < 1$, para formar parte de la población intermedia. Las selecciones por torneo tienen la ventaja de que no es necesario escalar los valores de la función objetivo.

El operador de cruce elige aleatoriamente un par de individuos de los previamente seleccionados para ser progenitores, es decir, de los que forman parte de la población intermedia, y en función de una cierta probabilidad combina o no su material genético para dar lugar a sus hijos. Hay distintos tipos de operadores de cruce, en función del tipo de codificación genética elegida para los individuos. En este trabajo se utiliza codificación real, esto es, los individuos se codifican utilizando números en punto flotante, en contraposición a la codificación con cadenas de bits. Se han usado dos operadores de cruce:

Un Punto (OP) Con codificación con cadenas de bits, selecciona aleatoriamente el lugar donde se parten las cadenas de los progenitores en dos subcadenas; los hijos tendrán sus cromosomas compuestos por estas subcadenas (véase la figura 16(a)). En el caso de la codificación real lo que se hace es crear los hijos de forma que éstos sólo difieran de los padres en que tienen intercambiados los genes correspondientes al lugar elegido para cruzar (véase la figura 16(b)).

Cruce Uniforme (U) En este caso no se usan puntos de corte para las subcadenas, sino que los descendientes se crean decidiendo, para cada alelo de uno de los progenitores, si se intercambia ese alelo con el correspondiente del otro progenitor. Esa decisión se toma usando un “lanzamiento de moneda”, es decir, el intercambio tiene lugar con una probabilidad 0.5 [64] en el caso del cruce uniforme puro, o bien alguna otra, en el caso del cruce uniforme parametrizado [12].

progenitor 1	1 0 0 1 0 1 1
progenitor 2	0 1 0 0 1 1 0
Posición aleatoria	↑
descendiente 1	1 0 0 0 1 1 0
descendiente 2	0 1 0 1 0 1 1

(a) Codificación binaria

progenitor 1	0.07 3.05 8.67 1.34
progenitor 2	1.34 5.00 3.45 1.29
Posición aleatoria	↑
descendiente 1	0.07 5.00 8.67 1.34
descendiente 2	1.34 3.05 3.45 1.29

(b) Codificación real

Figura 16: Operador de cruce de un punto.

El operador de mutación, que en este trabajo, se usa tras el cruce [10], permite generar individuos en el espacio de búsqueda que no se habrían alcanzado de otra manera. Cuando parte de un cromosoma se escoge para ser mutado, los genes correspondientes a esa parte se cambian con probabilidad p . Hemos usado cuatro operadores de mutación. Tres de ellos son de la forma $\nu \leftarrow \nu \pm p \times \nu$, donde ν es el valor original del alelo, y p puede tener un valor constante (C), ser tomado con

probabilidad uniforme del intervalo $(0, \beta)$ con $\beta \leq 1$ (U), o seleccionado de una distribución gaussiana (G). El cuarto operador (R) simplemente reemplaza ν con un valor seleccionado de forma uniformemente aleatoria del rango de inicialización de ese gen.

La función de aptitud u objetivo juega el papel de medio ambiente, evaluando cada individuo de una población. Esto da una medida relativa al resto de la población, de cómo un individuo concreto satisface una métrica específica del problema. Los valores se transforman a una escala de valores positivos crecientes monótonamente. En los experimentos numéricos con nuestro modelo de viento buscamos valores para los parámetros citados anteriormente. Para ello se pretende minimizar el error relativo medio de las velocidades calculadas por el modelo con respecto a las medidas en las estaciones

$$F(\alpha, \varepsilon, \gamma, \gamma') = \frac{1}{N_r} \sum_{n=1}^{N_r} \frac{|\vec{v}_n - \vec{v}(x_n, y_n, z_n)|}{|\vec{v}_n|} \quad (96)$$

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.

3.3. Observaciones

Existen varios aspectos que conviene estudiar: fijada una topografía, ¿sirve la estimación realizada con una configuración de viento para otras configuraciones diferentes?. ¿Influye la topografía en la estimación de parámetros? ¿Es el refinamiento de la malla un factor importante a la hora de estimar los parámetros del modelo?

Los valores de la mejor evaluación dependen de la dirección del viento medida en las estaciones. Los valores obtenidos para α tras el ajuste dependen de la dirección del viento. La desviación típica y el coeficiente de variación son muy grandes en el caso de que la velocidad del viento \vec{v}_n medida en todas las estaciones sea idéntica en módulo y dirección, ε se cancela en la ecuación (46) y no tiene influencia en el modelo. Por tanto, cualquier valor de ε sería factible y la estimación de un valor concreto no tiene sentido. Con esta excepción, tanto γ como ε permanecen prácticamente constantes independientemente de la dirección del viento. La variación del parámetro γ' con la dirección del viento sigue un cierto paralelismo con la de α . A medida que el ajuste de viento es predominantemente horizontal (valores pequeños de α), la altura de la capa de mezcla disminuye (menores valores de γ'). Asimismo, cuando el ajuste comienza a dar más peso a la componente vertical del viento (aumento de α), la altura de la capa de mezcla aumenta. Estos resultados están de acuerdo con la definición dada para la capa de mezcla.

Se deduce de nuestros resultados que los valores de los parámetros dependen de la topografía. Como era de esperar, la estimación de parámetros resulta más complicada en topografías irregulares que en superficies más suaves. Por otro lado, esta dificultad se incrementa para las configuraciones *menos naturales* de viento en las estaciones. Los rangos de variación del parámetro α son distintos para cada orografía. No obstante, los rangos de variación obtenidos para este parámetro son pequeños si se comparan con los propuestos por los diversos autores. El valor de ε 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. γ permaneció prácticamente invariable en varios experimentos, incluso para distintos ángulos de ataque. Para sacar conclusiones sobre γ' habría que realizar diversos experimentos. No obstante, el paralelismo con α apuntado anteriormente se sigue manteniendo para diferentes orografías.

Los parámetros varían con el refinamiento y por lo tanto habría que ajustarlos en cada malla.

Las estimaciones realizadas sobre las 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 nuestros experimentos hemos comprobado 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 nuestro código automático de estimación, que permite mayor flexibilidad en los ajustes. Además, podemos utilizarlo para crear una base de datos correspondiente a un problema concreto, con el consiguiente ahorro de tiempo de ejecución.

Nos hemos planteado la cuestión de cómo afecta, en un problema dado, el hecho de utilizar los parámetros estimados sobre una malla, al usar como soporte otra malla producto del refinamiento o del desrefinamiento de la primera. Para ello, definimos 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:

$$S_T = \frac{|v_T^p - v_T^{p'}|}{|v_T^p|} \quad (97)$$

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 comprueba 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.

La sensibilidad de la gran mayoría de los elementos se mantiene dentro de unos límites razonables. Son relativamente pocos los elementos en los que se produce una gran variación del viento por un cambio en los parámetros.

Existe una cierta continuidad en los parámetros, en el sentido de que el uso de $P(\tau_{i+1})$ en el problema con otra malla τ_i afecta menos a la función objetivo que el uso de parámetros de mallas más “alejadas”, es decir $P(\tau_{i+k})$, $k > 1$. Además se observa que ese efecto sobre la función objetivo es menor cuanto más refinadas están las mallas.

Las gráficas de los isovolúmenes de la sensibilidad indican que existen zonas del dominio que sufren mayor alteración del campo de viento que otras cuando se cambian los parámetros estimados. Se observa que las zonas de mayor sensibilidad se

producen cerca de las laderas de montañas. Además las zonas son muy parecidas en todos los casos. Para reducir la sensibilidad y por tanto mejorar el viento obtenido en el dominio convendría situar estaciones de medida en esas zonas. De hecho, pensamos que 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.

Referencias

- [1] D.N. Arnold. Locally adapted tetrahedral meshes using bisection. *SIAM J. Sci. Comput.*, 22(2):431–448, 2000.
- [2] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In John J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications*, pages 14–21, Cambridge, MA, July 1987. Lawrence Erlbaum Associates.
- [3] R.E. Bank and R.K. Smith. Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.*, 34:979–997, 1997.
- [4] J.C. Barnard, H.L. Wegley, and T.R. Hiester. Improving the performance of mass consistent numerical model using optimization techniques. *J. Climate Appl. Meteorol.*, 26:675–686, 1987.
- [5] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons, Inc., 1993.
- [6] M. Benzi and D. Bertaccini. Approximate inverse preconditioning for shifted linear systems. *BIT Num. Math.*, 43:231–244, 2003.
- [7] F. Bornemann, B. Erdmann, and R. Kornhuber. Adaptive multilevel methods in three space dimensions. *Int. J. Numer. Meth. Engng.*, 36:3187–3390, 1993.
- [8] A. Bowyer. Computing Delaunay tessellations. *Comput. J.*, 24(2):162–166, 1981.
- [9] J. A. Businger and S. P. S. Arya. Heights of the mixed layer in the stably stratified planetary boundary layer. *Adv. Geophys.*, 18A:73–92, 1974.
- [10] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- [11] A.F. de Baas. *Modelling of Atmospheric Flow Fields*, chapter Scaling Parameters and their Estimation, pages 87–102. World Sci., Singapore, 1996.
- [12] K. A. De Jong and W. M. Spears. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematic and Artificial Intelligence*, 5(1):1–26, 1992.
- [13] J.E. Jr. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM’s Classic in Applied Mathematics. SIAM, 1996.
- [14] H.N. Djidjev. Force-directed methods for smoothing unstructured triangular and tetrahedral meshes. Technical report, Dep. of Computer Science, Univ. of Warwick, Coventry, UK, 2000. <http://www.andrew.cmu.edu/user/sowen/topics/new.html>.

- [15] J. Dompierre, P. Labbé, F. Guibault, and R. Camarero. Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization. In *Proceedings of 7th International Meshing Roundtable, Dearborn*, pages 459–478. Sandia National Laboratories, 1998.
- [16] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, and J. M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput. Methods Appl. Mech. Engrg.*, 192:2775–2787, 2003.
- [17] J.M. Escobar. *Generación de Mallas Tridimensionales Mediante la Triangulación de Delaunay*. PhD thesis, Univ. de Las Palmas de Gran Canaria, 1995.
- [18] J.M. Escobar and R. Montenegro. Several aspect of the three-dimensional delaunay triangulation. *Adv. in Eng. Softw.*, 1/2(27):27–39, 1996.
- [19] L. Ferragut, R. Montenegro, and A. Plaza. Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems. *Comm. Numer. Methods Engrg.*, 10:403–412, 1994.
- [20] S. Fortune. Voronoi diagrams and delaunay triangulations. *Computing in Euclidean Geometry*, pages 193–233, 1992.
- [21] L.A. Freitag, M. Jones, and P. Plassmann. A parallel algorithm for mesh smoothing. *SIAM J. Sci. Comput.*, 20:2023–2040, 2000.
- [22] L.A. Freitag and P.M. Knupp. Tetrahedral element shape optimization via the jacobian determinant and condition number. In *Proceedings of the Eighth International Meshing Roundtable*, pages 247–258. Sandia National Laboratories, 1999.
- [23] L.A. Freitag and P.M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Int. J. Numer. Meth. Eng.*, 53:1377–1391, 2002.
- [24] P. Geai. Methode d’interpolation et de reconstitution tridimensionnelle d’un champ de vent: le code d’analyse objective MINERVE. Technical report, Electricité de France, 1985.
- [25] P.L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Comput. Meth. Appl. Mech. Engrg.*, 92:269–288, 1991.
- [26] P.E. Gill, W Murray, and M.H. Wright. *Practical Optimization*. Academic Press, Inc., 1981.
- [27] J.M. González-Yuste. *Un Algoritmo de Refinamiento/Derefinamiento Local para Mallas de Tetraedros*. PhD thesis, Univ. de Las Palmas de Gran Canaria, 2004. En proceso de realización.
- [28] J.M. González-Yuste, R. Montenegro, J.M. Escobar, G. Montero, and E. Rodríguez. Local refinement of 3-D triangulations using object-oriented methods. *Adv. in Eng. Softw.*, 35:693–702, 2004.
- [29] J. Holland. *Adaption in Natural and Artificial Systems*. MIT Press, 1992.

- [30] T. Kitada, A. Kaki, H. Ueda, and L.K. Peters. Estimation of vertical air motion from limited horizontal wind data - a numerical experiment. *Atmos. Environ.*, 17:2181–2192, 1983.
- [31] P.M. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. Part II - a frame work for volume mesh optimization and the condition number of the jacobian matrix. *Int. J. Numer. Meth. Eng.*, 48(1165–1185), 2000.
- [32] P.M. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, pages 193–218, 2001.
- [33] D.P. Lalas and C.F. Ratto, editors. *Modelling of Atmospheric Flow Fields*. World Sci., Singapore, 1996.
- [34] D.P. Lalas, M. Tombrou, and M. Petrakis. Comparison of the performance of some numerical wind energy siting codes in rough terrain. In *European Community Wind Energy Conference*, Herning, Denmark, 1988.
- [35] D. Levine. *A Parallel Genetic Algorithm for the Set Partitioning Problem*. PhD thesis, Illinois Institute of Technology / Argonne National Laboratory, 1994.
- [36] B. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *Mathematics of Computations*, 65(215):1183–1200, 1996.
- [37] R. Löhner and J.D. Baum. Adaptive h -refinement on 3D unstructured grids for transient problems. *Int. J. Num. Meth. Fluids*, 14:1407–1419, 1992.
- [38] G.J. McRae, W.R. Goodin, and J.H. Seinfeld. Development of a second generation mathematical model for urban air pollution. *Atm. Env.*, 16,4:679–696, 1982.
- [39] R. Montenegro, J. M. Escobar, , E. Rodríguez, G. Montero, and J. M. González-Yuste. Improved objective functions for tetrahedral mesh optimization. *Lect. Notes in Comp. Sci.*, 2657:568–578, 2003.
- [40] R. Montenegro, G. Montero, J. M. Escobar, and E. Rodríguez. Efficient strategies for adaptive 3-D mesh generation over complex orography. *Neur., Parar. & Sci. Comp.*, 10:57–76, 2002.
- [41] R. Montenegro, G. Montero, J. M. Escobar, E. Rodríguez, and J. M. González-Yuste. Tetrahedral mesh generation for environmental problems over complex terrain. In P. M. A. Sloot, C. J. Kenneth Tan, J. J. Dongarra, and A. G. Hoekstra, editors, *Lect. Notes in Comp. Sci.*, pages 335–344, 2002.
- [42] R. Montenegro, G. Montero, J. M. Escobar, E. Rodríguez, and J. M. González-Yuste. Tetrahedral mesh generation for environmental problems over complex terrain. *Lect. Notes in Comp. Sci.*, 2329:335–344, 2002.
- [43] R. Montenegro, A. Plaza, L. Ferragut, and I. Asensio. Application of a nonlinear evolution model to fire propagation. *Nonlinear Analysis, Th., Meth.& App.*, 5(30):2873–2882, 1997.

- [44] G. Montero, R. Montenegro, J. M. Escobar, and E. Rodríguez. Generación automática de mallas de tetraedros adaptadas a orografías irregulares. *Rev. Int. Mét. Num. Cál. Dis. Ing.*, 19(2):127–144, 2003.
- [45] G. Montero, R. Montenegro, and J.M. Escobar. A 3-D diagnostic model for wind field adjustment. *J. of Wind Eng. and Ind. Aerodynamics*, 74–76:249–261, 1998.
- [46] G. Montero, R. Montenegro, J.M. Escobar, and E. Rodríguez. *Invited Lectures of ECT'04*, chapter Resolution of Sparse Linear Systems of Equations: the RPK strategy. Saxe-Coburg Publications, Stirling, Scotland, 2004.
- [47] G. Montero, E. Rodríguez, R. Montenegro, J. M. Escobar, and J. M. González-Yuste. Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model. *Adv. Eng. Soft.*, 35:3–10, 2005.
- [48] G. Montero and N. Sanín. 3-D modelling of wind field adjustment using finite differences in a terrain conformal coordinate system. *J. Wind Eng. & Ind. Aer.*, (89), 2001.
- [49] G. Montero, A. Suárez, E. Rodríguez, E. Flórez, and M.D. García. Preconditioning shifted linear systems arising in a wind model. In B.H.V. Topping, editor, *Proceedings of CC'2005*, Stirling, Scotland, 2005. Civil-Comp Press.
- [50] N. Moussiopoulos, Th. Flassak, and G. Knittel. A refined diagnostic wind model. *Environ. Software*, 3:85–94, 1988.
- [51] M. Murphy, D.M. Mount, and C.W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. In *Symposium on Discrete Algorithms*, pages 67–74, 2000.
- [52] I. Palomino and F. Martín. A simple method for spatial interpolation of the wind in complex terrain. *J. Appl. Meteor.*, 34:1678–1693, 1995.
- [53] H.A. Panofsky and J.A. Dutton. *Atmospheric Turbulence*. John Wiley, New York, 1984.
- [54] A. Plaza, R. Montenegro, and L. Ferragut. An adaptive refinement/derefinement algorithm of structured grids for solving time-dependent problems. *Numerical Methods in Engineering*, 1992.
- [55] A. Plaza, R. Montenegro, and L. Ferragut. An improved derefinement algorithm of nested meshes. *Advances in Engineering Software*, 1/2(27):51–57, 1996.
- [56] Angel Plaza and G.F. Carey. Local refinement of simplicial grids based on the skeleton. *Appl. Numer. Math.*, 32:195–218, 2000.
- [57] C.F. Ratto. *Modelling of Atmospheric Flow Fields*, chapter The AIOLOS and WINDS Codes, pages 421–431. World Sci., Singapore, 1996.
- [58] M.C. Rivara. A grid generator based on 4-triangles conforming. Mesh-refinement algorithms. *Int. J. Num. Meth. Eng.*, 24:1343–1354, 1987.

- [59] M.C. Rivara and C. Levin. A 3-d refinement algorithm suitable for adaptive multigrid techniques. *J. Comm. Appl. Numer. Meth.*, 8:281–290, 1992.
- [60] E. Rodríguez, G. Montero, R. Montenegro, J. M. Escobar, and J. M. González-Yuste. Parameter estimation in a three-dimensional wind field model using genetic algorithms. In P. M. A. Sloot, C. J. Kenneth Tan, J. J. Dongarra, and A. G. Hoekstra, editors, *Lect. Notes in Comp. Sci.*, pages 950–959, 2002.
- [61] D.G. Ross, I.N. Smith, P.C. Manins, and D.G. Fox. Diagnostic wind field modelling for complex terrain: Model development and testing. *J. Appl. Meteorol.*, 27:785–796, 1988.
- [62] A.M. Sempreviva. *Modelling of Atmospheric Flow Fields*, chapter Roughness Changes: Response of Neutral Boundary Layers, pages 213–245. World Sci., Singapore, 1996.
- [63] C.A. Sherman. A mass-consistent model for wind fields over complex terrain. *J. Appl. Meteorol.*, 17:312–319, 1978.
- [64] Gilbert Syswerda. Uniform crossover in genetic algorithms. In J. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9. San Mateo, CA: Morgan Kaufmann, 1989.
- [65] J.G. Tinoco-Ruiz and P. Barrera-Sánchez. Area functionals in plane grid generation. In *Proceedings 6th Conference in Numerical Grid Generation in Computational Field Simulations*, pages 293–302, Greenwich, UK, 1998. International Society of Grid Generation.
- [66] J.G. Tinoco-Ruiz and P. Barrera-Sánchez. Area control in generating smooth and convex grids over general plane regions. *J. Comput. Appl. Math.*, 103(1), 1999.
- [67] M. Tombrou and D.P. Lalas. A telescoping procedure for local wind energy potential assessment. In *European Community Wind Energy Conference*. H.S. Stephens & Associates, 1990. Madrid.
- [68] D.F. Watson. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J.*, 24(2):167–172, 1981.
- [69] N.P. Weatherhill and O. Hassan. Efficient three-dimensional grid generation using the Delaunay triangulation. *Comput. Fluid Dyn.*, 2:961–968, 1992.
- [70] D. Whitley. GENITOR: A different genetic algorithm. In *Rocky Mountain Conference on Artificial Intelligence*, 1988.
- [71] D. Whitley. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Third International Conference on Genetic Algorithms*, 1989.
- [72] G. Winter, J. Betancor, and G. Montero. *Ocean Circulation and Pollution Control. A mathematical and Numerical Investigation*, chapter 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.

- [73] G. Winter, G. Montero, L. Ferragut, and R. Montenegro. Adaptive strategies using standard and mixed finite elements for wind field adjustment. *Solar Energy*, 54(1):49–56, 1995.
- [74] P. Zannetti. *Air Pollution Modelling*. Comput. Mech., Boston, 1990.