

Generación automática de mallas tridimensionales para la simulación numérica de procesos medioambientales



Rafael A. Montenegro Armas

Catedrático de Matemática Aplicada y Director de la División de Discretización y Aplicaciones del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería de la Universidad de Las Palmas de Gran Canaria

Resumen

El objetivo de este trabajo es presentar los principales aspectos de un código capaz de generar automáticamente mallas de tetraedros a partir de una distribución adecuada de nodos en el dominio de estudio, así como plantear una versión de un algoritmo de refinamiento local basado en la subdivisión en 8-subtetraedros. En concreto, el dominio está limitado en su parte inferior por la superficie del terreno considerada como $z = f(x, y)$, tal que $\forall (x, y) \in [a, b] \times [c, d] \subset \mathbb{R}^2 \Rightarrow \exists ! z \in \mathbb{R}$, y en su parte superior por un plano horizontal. Las paredes laterales están formadas por cuatro planos verticales, paralelos dos a dos. Las ideas básicas para la construcción de la malla combinan, por un lado, la utilización de un algoritmo de refinamiento y desrefinamiento para dominios bidimensionales [6], por otro lado, un algoritmo de generación de mallas de tetraedros basado en la triangulación de Delaunay [5], y además, una función para definir el espaciado vertical de los nodos distribuidos en el dominio. Este código ha sido utilizado anteriormente para generar mallas donde la superficie $z = f(x, y)$ correspondía a una digitalización topográfica del sur de la isla de la Palma, [19] y [20]. Asimismo, en el generador de mallas entran en juego un procedimiento para optimizar la malla resultante y diferentes estrategias para construir la nube de puntos. Una vez que se ha generado la malla atendiendo a la geometría del dominio, es necesario adaptarla a la solución numérica del problema. Por ello, dedicamos una segunda parte de este trabajo al desarrollo e implementación de un algoritmo de refinamiento local de mallas encajadas de tetraedros, cuyo principal interés es su aplicabilidad, rapidez y sencillez. El algoritmo propuesto se basa en la división expuesta en [16] cuyo análisis de calidad se estudia en [15].

1. Introducción

El trabajo se ha dividido en dos partes. La primera se desarrolla en la sección 2 y se dedica al análisis del generador automático de mallas de tetraedros de un dominio tridimensional definido sobre un terreno irregular. En la sección 3 presentamos el algoritmo de refinamiento local de mallas de tetraedros. Estas dos grandes líneas se engloban entre los aspectos básicos necesarios para la simulación numérica del problema de dispersión de contaminantes en la atmósfera, que pretendemos afrontar.

2. Generación de mallas sobre orografía irregular

Inicialmente, nos planteamos crear una malla de tetraedros que respete la topografía de la región de estudio con una precisión determinada. Para ello, se dispone únicamente de la

información digitalizada del terreno. Por otra parte, deseamos que la malla esté adaptada con una mayor densidad de nodos donde sea necesario para definir las características geométricas de nuestro dominio a partir de una interpolación lineal a trozos. 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 [27] y [21], propagación de fuego [18], contaminación atmosférica, 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, que únicamente se adapta a las características geométricas del dominio, se podrán aplicar posteriormente algoritmos de refinamiento y desrefinamiento de tetraedros para mejorar la solución numérica del problema [16] y [15]. Estos algoritmos tendrán un especial interés en los problemas evolutivos.

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 parte superior del dominio. En concreto, se propone plantear una malla rectangular 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 definida en la entrada de datos, 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, [6] y [23], 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 resultante.

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

La distribución de puntos en el dominio entrará como dato en el 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 inferior 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 a realizar una transformación, 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.

La calidad de la malla final, obtenida mediante la transformación inversa al dominio real, depende de la distribución de los puntos definida en ambos dominios, ya que respetaremos la topología de la malla obtenida en la triangulación del paralelepípedo auxiliar. Proponemos una estrategia 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 sus características fundamentales. Con esta estrategia se generan capas 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 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, y en la correspondiente a la frontera superior del dominio. En concreto, se 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 parte superior del dominio, en función de los tamaños de los elementos existentes sobre ambas superficies.

Una vez que se ha construido la triangulación de Delaunay de la nube de puntos en el paralelepípedo, procedemos a situar los puntos en sus posiciones reales manteniendo la topología de la malla. Hay que tener en cuenta que este proceso de compresión de la malla puede dar lugar a cruces de tetraedros que habrá que deshacer posteriormente. Asimismo, será aconsejable aplicar una etapa de suavizado para mejorar la calidad de los elementos de la malla resultante.

2.1. Experimentos numéricos. Como aplicación real del generador de mallas se ha considerado una región rectangular del sur de la Isla de La Palma de 22.8×15.6 km, en la que las cotas extremas varían de 0 a 2279 m de altitud. La parte superior del dominio se ha establecido a una altitud $h = 6$ km. Para definir la topografía se dispuso de una digitalización de la zona en la que las alturas estaban definidas sobre una cuadrícula con un paso espacial de 200 m según los ejes x e y .

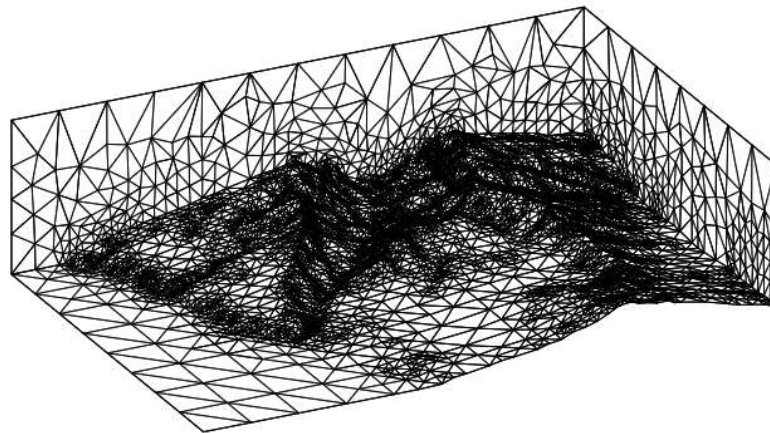


Figura 1. Malla resultante después de diez pasos del proceso de optimización para una región de la Isla de La Palma.

A partir de una malla uniforme de la región rectangular con un tamaño de elementos aproximadamente de $2 \times 2 \text{ km}$, se realizaron seis refinamientos globales utilizando el algoritmo 4-T de Rivara [25]. Una vez que se interpolaron los datos digitalizados sobre esta malla refinada, se empleó el algoritmo de desrefinamiento desarrollado en [6] y [23] con un parámetro de desrefinamiento $\varepsilon = 25 \text{ m}$. Esto asegura que la malla adaptada aproxima la superficie del terreno con un error menor que este valor. La distribución de nodos de la malla base regular utilizada antes de los refinamientos globales es la que se consideró sobre la frontera superior del dominio. En la figura 1 se representa la malla obtenida después de diez pasos del proceso de optimización.

También mostramos otras aplicaciones para una serie de superficies *test*. Las mallas que se exponen a continuación han sido generadas por nuestro código antes de la aplicación del proceso de optimización. Todas las superficies han sido definidas en una región rectangular de $10 \times 5 \text{ km}$. La parte superior del dominio se ha establecido a una altitud $h = 5 \text{ km}$. La primera aplicación corresponde a una superficie gaussiana relativamente suave y se representa en la figura 2. Hemos fijado un parámetro de desrefinamiento $\varepsilon = 20 \text{ m}$. Esta estrategia introduce 6 capas, 1155 nodos y no produce ningún tetraedro invertido.

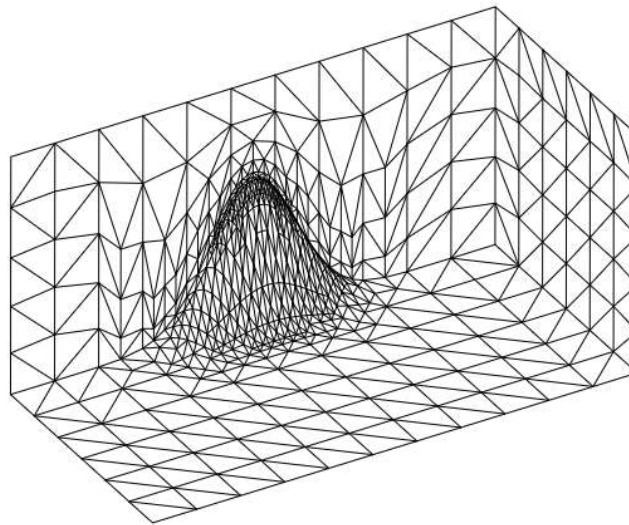


Figura 2. Malla resultante, antes de aplicar el proceso de optimización, para una superficie gaussiana.

Planteamos ahora una superficie con forma de *volcán* que fue definida mediante la composición de dos superficies gaussianas; la cota máxima en el dominio es $h = 1 \text{ km}$, mientras que la cota mínima del terreno es $z_{\min} = -1 \text{ km}$. En la figura 3 se representa un detalle de la malla obtenida con un parámetro de desrefinamiento $\varepsilon = 5 \text{ m}$. La malla resultante posee 3973 nodos sin existencia de cruces de tetraedros.

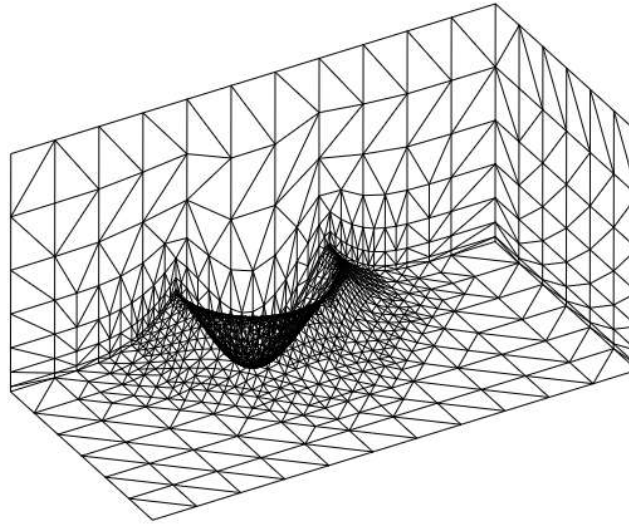


Figura 3. Malla resultante, antes de aplicar el proceso de optimización, para una superficie con forma de volcán.

2.2. Conclusiones. Hemos establecido y analizado los aspectos principales para generar una malla tridimensional de tetraedros que se adapta a una superficie definida sobre una región rectangular con una mínima intervención del usuario. En concreto se ha planteado una generación de puntos, bien distribuidos en el dominio de estudio, capaz de captar la información de la superficie y que posee una densidad menor a medida que aumenta la altura con respecto al terreno. Los puntos se generan aplicando técnicas de refinamiento/desrefinamiento en 2-D, una función de espaciado vertical y diferentes estrategias. Seguidamente, con la ayuda de un paralelepípedo auxiliar, se ha planteado un procedimiento basado en la triangulación de Delaunay para construir automáticamente la malla, asegurando la conformidad con la superficie del terreno. No obstante, la distribución de puntos obtenida también podría tener interés para generar la malla tridimensional con otras técnicas clásicas, tales como avance frontal [11] y *normal offsetting* [12]. Finalmente, el procedimiento propuesto para optimizar la malla generada ha permitido resolver al mismo tiempo los problemas de cruces de tetraedros y de calidad de la malla.

3. Refinamiento local de mallas de tetraedros

En la actualidad, la mayor parte de los programas que utilizan el método de elementos finitos se apoya 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 generación de mallas adaptables podemos considerar dos aspectos diferentes: la discretización del dominio atendiendo a su geometría o a la solución numérica. 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 [17] o en

algoritmos basados en la triangulación de Delaunay [10] y [5]. 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. 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 local 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 y 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.

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 [2]; 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], [26], [24], y los que utilizan la subdivisión en 8-subtetraedros [3], [15], [16]. En concreto, el algoritmo desarrollado en [24] 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 [3], [15], [16], que a su vez generalizan a 3-D la partición en cuatro subtriángulos propuesta en [2], 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 en C++ una versión del algoritmo que utiliza la subdivisión en 8-subtetraedros.

3.1. Aplicaciones. Como ejemplo de aplicación del algoritmo de refinamiento, se presenta en la figura 4(a) una triangulación inicial τ_1 formada por 5072 tetraedros y 1140 nodos, que fue generada a partir del código introducido en [5]. Para proceder a su refinamiento se ha utilizado un indicador de error atendiendo a las distancias desde los centros de gravedad de los tetraedros a un vértice del dominio. En las figuras 4(b) y 4(c) se representan las mallas refinadas después de 1 y 2 etapas de refinamiento, compuestas por 5386 tetraedros y 1201 nodos, y 6270 tetraedros y 1433 nodos, respectivamente. Destacamos la rapidez del proceso de refinamiento, así como la calidad de la malla resultante.

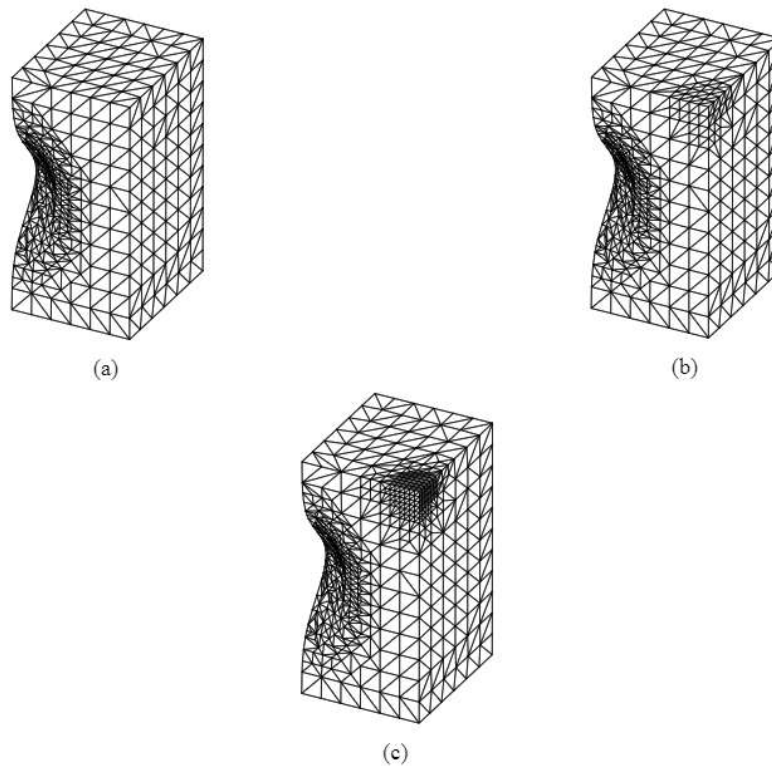


Figura 4. Aplicación del algoritmo de refinamiento: (a) malla inicial; (b) y (c) mallas resultantes tras 1 y 2 etapas de refinamiento, respectivamente.

3.2. Conclusiones. En este apartado se estudia un algoritmo de refinamiento local de mallas de tetraedros basado en la subdivisión en 8-subtetraedros propuesta en [3], [15] y [16]. Se ha aplicado de forma eficiente en mallas tridimensionales generadas mediante una versión del método de triangulación de Delaunay [5]. En trabajos futuros se propone desarrollar el correspondiente algoritmo de desrefinamiento y aplicarlo especialmente en problemas evolutivos de dispersión de contaminantes en la atmósfera.

Reconocimientos

El trabajo que aquí se expone es fruto de una investigación conjunta con G. Montero, J.M. Escobar, E. Rodríguez y J.M. González-Yuste, parcialmente subvencionada por el Ministerio de Ciencia y Tecnología y fondos FEDER a través del proyecto REN2001-0925-C03-02/CLI.

Bibliografía

- [1] D.N. Arnold, A. Mukherjee y L. Pouly, Locally adapted tetrahedral meshes using bisection, *SIAM J. Sci. Comput.*, **22**, 2, 431-448 (2000).
- [2] R.E. Bank, A.H. Sherman y A. Weiser, Refinement algorithms and data structures for regular local mesh refinement, in *Scientific Computing IMACS*, Amsterdam, North-Holland, (1983), 3-17.
- [3] F. Bornemann, B. Erdmann y R. Kornhuber, Adaptive multilevel methods in three space dimensions", *Int. J. Numer. Meth. Eng.*, **36**, 3187-3203 (1993).
- [4] H.N. Djidjev, *Force-directed methods for smoothing unstructured triangular and tetrahedral meshes*, Tech. Report, Dep. of Computer Science, Univ. of Warwick, Coventry, UK, (2000).
[Disponible en <http://www.andrew.cmu.edu/user/sowen/topics/new.html>].
- [5] J.M. Escobar y R. Montenegro, Several aspects of three-dimensional Delaunay triangulation, *Advances in Engineering Software*, **27**, 1/2, 27-39 (1996).
- [6] L. Ferragut, R. Montenegro y A. Plaza, Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems, *Comm. Num. Meth. Eng.*, **10**, 403-412 (1994).
- [7] L.A. Freitag y P.M. Knupp, *Tetrahedral element shape optimization via the jacobian determinant and condition number*, en *Proceedings of the Eighth International Meshing Roundtable*, Sandia National Laboratories (1996), pp. 247-258.
- [8] L.A. Freitag y P. Plassmann, Local optimization-based simplicial mesh untangling and improvement, *Int. J. Numer. Methods Engng.*, **49**, 109-125 (2000).
- [9] L.A. Freitag y P.M. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, *Int. J. Numer. Methods Engng.*, **53**, 1377-1391 (2002).
- [10] P.L. George, F. Hecht y E. Saltel, Automatic mesh generation with specified boundary, *Comp. Meth. Appl. Mech. Eng.*, **92**, 269-288 (1991).
- [11] H. Jin y R.I. Tanner, Generation of unstructured tetrahedral meshes by advancing front technique, *Int. J. Num. Meth. Eng.*, **36**, 1805-1823 (1993).
- [12] B.P. Johnston y J.M. Sullivan, Jr., A normal offsetting technique for automatic mesh generation in three dimensions, *Int. J. Num. Meth. Eng.*, **36**, 1717-1734 (1993).
- [13] P.M. Knupp, Algebraic mesh quality metrics, *SIAM J. Sci. Comput.*, **23**, 193-218 (2001).
- [14] 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. Methods Engng.*, **48**, 1165-1185 (2000).
- [15] A. Liu y B. Joe, Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision, *Mathematics of Computations*, **65**, 215, 1183-1200 (1996).
- [16] R. Löhner y J.D. Baum, Adaptive *h*-refinement on 3D unstructured grids for transient problems, *Int. J. Num. Meth. Fluids*, **14**, 1407-1419 (1992).

- [17] R. Löhner y P. Parikh, Three-dimensional grid generation by advancing front method, *Int. J. Num. Meth. Fluids*, **8**, 1135-1149 (1988).
- [18] R. Montenegro, A. Plaza, L. Ferragut y I. Asensio, Application of a nonlinear evolution model to fire propagation, *Nonlinear Analysis, Th., Meth. & App.*, **30**, 5, 2873-2882 (1997).
- [19] R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez y J.M. González-Yuste, Tetrahedral mesh generation for environmental problems over complex terrains, *Lecture Notes in Computer Science*, **2329**, 335-344 (2002).
- [20] R. Montenegro, G. Montero, J.M. Escobar y E. Rodríguez, Efficient strategies for adaptive 3-D mesh generation over complex orography, *Neural, Parallel & Scientific Computation*, **10**, 57-76 (2002).
- [21] G. Montero, R. Montenegro y J.M. Escobar, A 3-D diagnostic model for wind field adjustment, *J. of Wind Eng. and Ind. Aerodynamics*, **74-76**, 249-261 (1998).
- [22] M. Murphy, D.M. Mount y C.W. Gable, *A point-placement strategy for conforming Delaunay tetrahedralization*, en *Symposium on Discrete Algorithms*, (2000), pp. 67-74.
- [23] A. Plaza, R. Montenegro and L. Ferragut, An improved derefinement algorithm of nested meshes, *Advances in Engineering Software*, **27**, 1/2, 51-57 (1996).
- [24] A. Plaza y G.F. Carey, Local refinement of simplicial grids based on the skeleton, *Appl. Numer. Math.*, **32**, 195-218 (2000).
- [25] M.C. Rivara, A grid generator based on 4-triangles conforming. Mesh-refinement algorithms, *Int. J. Num. Meth. Eng.*, **24**, 1343-1354 (1987).
- [26] M.C. Rivara y C. Levin, A 3-d refinement algorithm suitable for adaptive multigrid techniques, *J. Comm. Appl. Numer. Meth.*, **8**, 281-290 (1992).
- [27] G. Winter, G. Montero, L. Ferragut y R. Montenegro, Adaptive strategies using standard and mixed finite elements for wind field adjustment, *Solar Energy*, **54**, 1, 49-56 (1995).

En Internet

<http://ceani.ulpgc.es/index.html>

CEANI-IUSIANI

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería de la Universidad de Las Palmas de Gran Canaria.