

Cálculo de volúmenes en modelos remallados de elementos finitos

Andrés Lapuebla-Ferri, Fernando Giménez, A. J. Jiménez, Juan A. Monsoriu
UNIVERSIDAD POLITÉCNICA DE VALENCIA
anlafer0@mes.upv.es, fgimenez@mat.upv.es, ajimene@mes.upv.es, jmonsori@fis.upv.es

Amaya Pérez Del Palomar
INSTITUTO DE INVESTIGACIÓN EN INGENIERÍA DE ARAGÓN (I3A)
amaya@unizar.es

Abstract

El Método de los Elementos Finitos asistido por ordenador es una herramienta muy extendida en ingeniería. En ciertas aplicaciones, se precisa modificar la malla del modelo a lo largo del proceso de simulación, lo que se conoce como remallado. En estos casos, además, suele ser necesario realizar mediciones en el modelo de elementos finitos, como por ejemplo calcular el volumen de ciertas regiones del mismo. Los paquetes comerciales disponibles en la actualidad permiten satisfacer algunas de estas necesidades, aunque adolecen de ciertas limitaciones. En este trabajo se presenta una aplicación desarrollada en el código propio de MATLAB[®], que permite el cálculo aproximado del volumen de una región del espacio comprendida entre una malla de elementos finitos cuadriláteros lineales y otra malla topológicamente equivalente a la anterior, esta última obtenida a través de un proceso de remallado.

Computer-assisted Finite Element Method is a widespread tool in the field of engineering. In certain applications, it is necessary to modify the mesh of the model during the analysis, which is known as 'remeshing'. In these situations, it is usually needed to perform some measurements, p.e., to calculate the volume of a region. Commercial packages can cover these needed, but they reveal to be limited in some situations. In this work, an application developed in MATLAB[®] is presented to approximately calculate the volume of a spatial region comprised between two meshes. The first is a quadrilateral finite element mesh, and the second mesh is topologically equivalent to the first and obtained through a remeshing process.

Keywords: Modelo de elementos finitos, remallado.

1 Introducción

El Método de los Elementos Finitos (en adelante, MEF) es quizá el método numérico de simulación de uso más frecuente en ingeniería. Se trata de un método general para la resolución aproximada de ecuaciones diferenciales, que se basa en los principios variacionales o energéticos. En este método, la geometría objeto de estudio se discretiza en un número, por lo general elevado, de elementos finitos, que junto a los nodos que los delimitan constituyen los dos tipos de entidades que caracterizan al MEF [1]. La geometría particular de los elementos (barras, cuadriláteros, hexaedros, tetraedros, etc.) permite el uso de unas funciones de interpolación determinadas, con las que se obtiene un sistema de ecuaciones matemáticas en el cual se consideran asimismo las condiciones de contorno del modelo, las características del medio y las acciones actuantes. El sistema se resuelve mediante ordenadores a través de técnicas de cálculo numérico específicas para simular el problema que se aborda.

De este modo, cualquier problema que pueda formularse como un sistema de ecuaciones diferenciales –como es la mayoría de los problemas de ingeniería– puede resolverse mediante el MEF. Así, áreas como la aeronáutica, la biomecánica o el análisis estructural se han beneficiado de este método numérico desde su aparición en la segunda mitad del siglo XX. La justificación del MEF estriba en innegables ventajas como su potencia, su gran versatilidad y su capacidad para modelar geometrías muy complejas. Además, el MEF es el único medio disponible para simular la forma en la que se distribuyen ciertas variables internas en el medio que se estudia, como es el caso de las tensiones y deformaciones en un material, ya que no pueden obtenerse de forma directa mediante los métodos clásicos de ensayo de laboratorio.

En la actualidad existen numerosos programas comerciales de elementos finitos capaces de cubrir un gran abanico de situaciones de análisis. En su mayor parte, tales paquetes disponen de una interfaz de usuario con múltiples opciones con las que el usuario puede, además de simular el modelo, realizar muchas mediciones y modificaciones en el mismo. Sin embargo, al tratarse en su mayor parte de códigos cerrados al usuario, desvelan una serie de carencias como la que ha motivado la realización del presente trabajo y se describe a continuación.

En ciertas situaciones de simulación numérica, especialmente en el campo de la biomecánica, no es difícil encontrarse con situaciones en las que el modelo de elementos finitos deba ser modificado a lo largo del proceso, lo que se conoce como remallado. Esto, además, suele llevar aparejada la necesidad de realizar mediciones en el modelo, como el cálculo de áreas o de volúmenes, lo que no siempre es posible a través de paquetes comerciales. El presente trabajo consiste en el desarrollo de un código para el cálculo de volúmenes en modelos remallados de elementos finitos. Dicho código se ha implementado en el paquete de cálculo MATLAB[®] dada su gran versatilidad [2].

2 Planteamiento del problema.

El problema que se plantea en el presente trabajo es el cálculo del volumen del espacio comprendido entre dos mallas de elementos finitos cuadriláteros lineales (figura 1), habiéndose obtenido una de ellas a través de un proceso de remallado. Tales mallas son topológicamente equivalentes, puesto que ambas tienen el mismo número de entidades (elementos finitos y nodos) que se encuentran distribuidas de forma idéntica en el espacio. Así pues, la diferencia entre ambas mallas radica en su aspecto y en las posiciones nodales. Se supone que cada elemento de una malla tiene su homólogo en la malla topológicamente equivalente. Una primera solución al

problema consiste en calcular dicho volumen a través del programa comercial MSC.PATRAN 2006r1 [3] mediante el siguiente procedimiento:

- 1) Generación de elementos hexaédricos lineales, de modo que cada uno de ellos conecta un elemento cuadrilátero con su homólogo a través de dos caras opuestas entre sí (figura 2).
- 2) Cálculo del volumen comprendido entre las mallas como la suma de los volúmenes de los hexaedros generados en el paso anterior.

Este procedimiento, aunque válido, presenta dos inconvenientes destacados. En primer lugar, la generación de elementos hexaédricos por un programa comercial puede dar lugar de forma inesperada a la degeneración de las geometrías de algunos de ellos, lo que impide el cálculo del volumen deseado. La razón estriba en que, si bien los elementos finitos hexaédricos tienen unas características particulares que los hacen más favorables que los tetraedros en algunas aplicaciones [4], ya que estos últimos son muy rígidos y pueden bloquearse durante el análisis del modelo, la forma de los primeros se altera de forma excesiva si las superficies topológicamente equivalentes tienen un aspecto muy dispar, lo que se pone de manifiesto en la figura 1. En segundo lugar, este procedimiento puede ser en ocasiones excesivamente lento y tedioso. Con el fin de soslayar estos problemas, se plantea la formulación y el código implementado en MATLAB[®] que se describen en los apartados siguientes.

3 Formulación

Se considera que el volumen total V_T del espacio comprendido entre las dos mallas es la suma de los volúmenes $V_{ii'}$ comprendidos entre cada par de elementos cuadriláteros homólogos $i - i'$:

$$V_T = \sum_{i=1}^n V_{ii'}, \quad (23.1)$$

siendo n el número de elementos en una de las dos mallas.

En lo sucesivo, se denomina *hexaedro virtual* al volumen hueco comprendido entre dos elementos cuadriláteros homólogos, de modo que los vértices de dicho hexaedro virtual son los nodos de tales elementos. No obstante, se tiene como inconveniente que el hexaedro virtual no es, en general, un prisma en el sentido estricto de la palabra, ya que dos de sus caras o aristas opuestas cualesquiera no son necesariamente paralelas, lo que impide el cálculo directo del volumen $V_{ii'}$. Así pues, se supone que cada hexaedro virtual se compone de dos prismas de bases triangulares denominados α y β (figura 3a) de modo que:

$$V_{ii'} = V_{ii'}^{\alpha} + V_{ii'}^{\beta}. \quad (23.2)$$

Estos nuevos *prismas* tampoco lo son tales por las mismas razones que las aducidas para los hexaedros virtuales, con lo que no es posible el cálculo directo de sus respectivos volúmenes. Sin embargo, en estos nuevos prismas los vértices de cada base triangular están contenidos en un mismo plano, lo que posibilita el tratamiento numérico de su volumen, aunque las bases no sean paralelas. Este último aspecto puede corregirse fácilmente descomponiendo cada prisma triangular en dos partes: un prisma triangular de caras opuestas paralelas (figura 3b) y una pirámide de 5 vértices formada, a su vez, por dos tetraedros $T1$ y $T2$ (figura 3c). Luego el volumen de cada prisma triangular (con $k = \alpha, \beta$) se calcula como sigue:

$$V_{ii'}^k = (V_{ii'}^k)' + (V_{ii'}^k)^{T1} + (V_{ii'}^k)^{T2}. \quad (23.3)$$

siendo $(V_{ii'}^k)'$ el volumen de la porción del prisma triangular de caras paralelas, $(V_{ii'}^k)^{T1}$ y $(V_{ii'}^k)^{T2}$, los volúmenes de los tetraedros en los que se descompone la pirámide de 5 vértices.

Se considera a continuación la *Regla de Simpson* [5], un método numérico para obtener de forma aproximada el valor de la integral de una función $f(x)$ en un intervalo $[a, b]$:

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right). \quad (23.4)$$

Si se particulariza la expresión anterior para el cálculo del volumen de un prisma triangular de caras opuestas paralelas separadas una distancia d , se obtiene:

$$(V_{ii'}^k)' = \int_0^d A(x)dx \approx \frac{d}{6} \left(A(0) + 4A\left(\frac{d}{2}\right) + A(d) \right), \quad (23.5)$$

siendo $A(x)$ el área de una sección transversal del prisma, que es paralela a sus caras opuestas, que se evalúa a una distancia $x \in [0, d]$ de una de las bases inferiores (figura 3b). No obstante, no siempre puede asegurarse de que el intervalo en el que se evalúa la integral en la Regla de Simpson sea lo suficientemente pequeño, con lo que el error de aproximación cometido puede ser muy grande. Para soslayar este inconveniente, se recurre a la Regla de Simpson compuesta, en virtud de la cual el intervalo de separación entre bases $[0, d]$ se divide en n subintervalos de igual tamaño $[x_{j-1}, x_j]$, de tal modo que la Regla de Simpson se aplica a cada uno de ellos. Procediendo de este modo, la formulación queda como sigue:

$$x_j = jh, \quad h = \frac{d}{n}, \quad j = 0, 1, \dots, n,$$

$$(V_{ii'}^k)' = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} A(x)dx \approx \sum_{j=1}^n \frac{x_j - x_{j-1}}{6} \left(A(x_{j-1}) + 4A\left(\frac{x_{j-1} + x_j}{2}\right) + A(x_j) \right). \quad (23.6)$$

De este modo, sumando las integrales de cada subintervalo se obtiene la aproximación buscada de $(V_{ii'}^k)'$ en el intervalo $[0, d]$.

El número n de subintervalos debe ser, a priori, lo suficientemente grande para garantizar unos buenos resultados, sin que ello implique un tiempo de cálculo excesivo. En la actualidad, la potencia de los ordenadores y el software disponible permiten que el valor de n sea elevado sin comprometer el tiempo computacional. En el contexto del presente trabajo, la formulación anterior se ha implementado en el paquete matemático multifunción MATLAB[©] 7.6.0 [2] a través de un código específico cuya estructura se describe en el apartado siguiente.

4 Implementación en MATLAB[©]

El código implementado en MATLAB[©] requiere de un archivo de entrada en formato texto en el que deben especificarse los siguientes datos:

- Las posiciones en el espacio de cada nodo N_i de las dos mallas: [Etiqueta X_i, Y_i, Z_i].
- La *conectividad* de cada uno de los elementos cuadriláteros Q_i de ambas mallas, que se define como una ordenación de los nodos que lo componen: [Etiqueta $N_{1i}, N_{2i}, N_{3i}, N_{4i}$].
- La correspondencia entre cada elemento i y su homólogo i' en la otra malla.

A su vez, el código se ha dividido en las siguientes funciones, con el fin de lograr una programación más estructurada y modular, además de poder reutilizarlas en otras posibles aplicaciones (figura 4):

- **PuntosPrisma.** Forma cada hexaedro virtual leyendo el archivo de entrada y considerando la conectividad y las coordenadas nodales de cada elemento cuadrilátero y su homólogo.
- **TrataPrisma.** Divide cada hexaedro virtual en dos prismas de bases triangulares.
- **ParalelPrisma.** Toma cada prisma de bases triangulares y lo divide en dos partes: un prisma de bases triangulares paralelas y una pirámide de 5 vértices.
- **VSimpson.** Calcula el volumen del prisma de bases triangulares paralelas mediante la Regla de Simpson.
- **VPiramide.** Calcula el volumen de la pirámide de 5 vértices como la suma de los volúmenes de los dos tetraedros en los que puede descomponerse.

El bucle que se indica en la figura 4 hace referencia a que el proceso indicado tiene lugar para cada uno de los hexaedros virtuales cuyo volumen se calcula.

5 Resultados y conclusiones

El código desarrollado en MATLAB[®] se aplicó al ejemplo numérico ilustrado en la figura 1. De forma paralela, el volumen comprendido entre las dos mallas se calculó mediante el software comercial MSC.PATRAN 2006R1. La diferencia entre los volúmenes obtenidos resultó ser menor del 1%. Se logró asimismo una gran economía en tiempo invertido con el empleo del código desarrollado.

Es interesante observar que para este problema en particular la convergencia del algoritmo implementado resultó ser independiente del número n de subintervalos. De hecho, el resultado fue el mismo variando n entre 1 y 1000. De este modo, resulta innecesario implementar la *Regla de Simpson* en su versión compuesta para la situación particular considerada, pudiendo haberse escogido la formulación general. No obstante, el código queda listo para otros casos en los que sí pueda ser necesaria esta distinción.

De este modo, se verifican los buenos resultados del código implementado, así como el uso que se le ha dado a la formulación considerada a través del mismo y a las asunciones tenidas en cuenta a lo largo del proceso. El código resultó ser robusto, ya que los resultados obtenidos no se modificaron al alterar la conectividad de los nodos de cada elemento cuadrilátero.

Agradecimientos. Los autores agradecen al Ministerio de Ciencia e Innovación (DPI2008-02953) y a la Universidad Politécnica de Valencia (PAID-06-08) su apoyo financiero. J.A. Monsoriu forma parte del Grupo de Innovación Docente e-MACAFI de la Universidad Politécnica de Valencia.

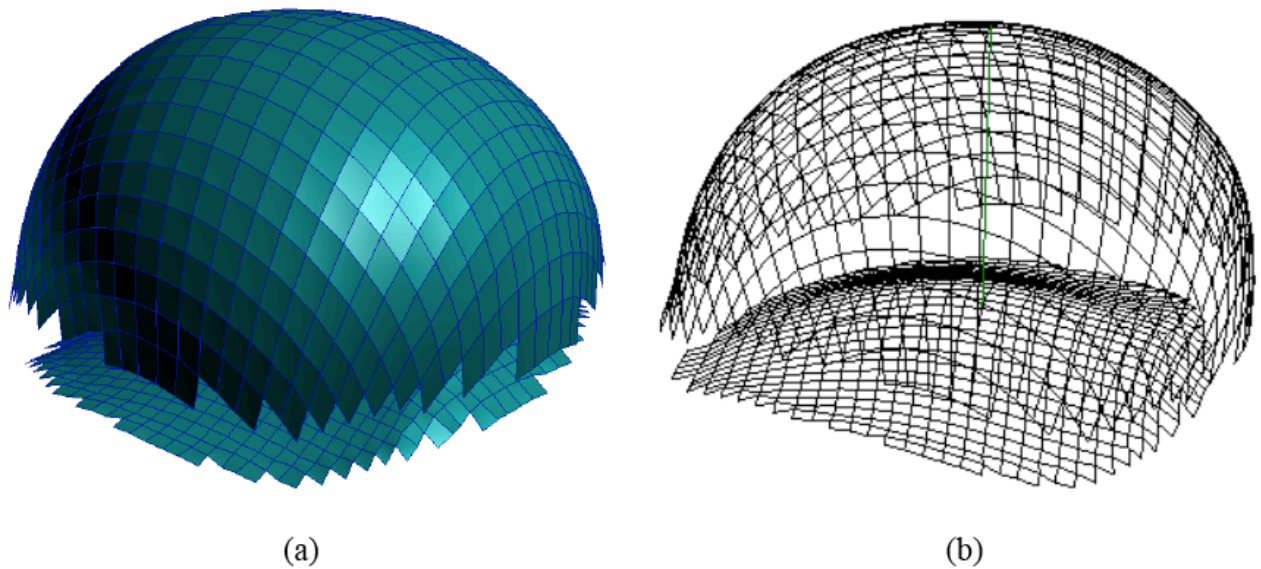


Figura 1: Mallas de elementos cuadriláteros lineales topológicamente equivalentes, cuyo volumen comprendido entre ambas se pretende calcular. (a) Vista renderizada. (b) Vista alámbrica.

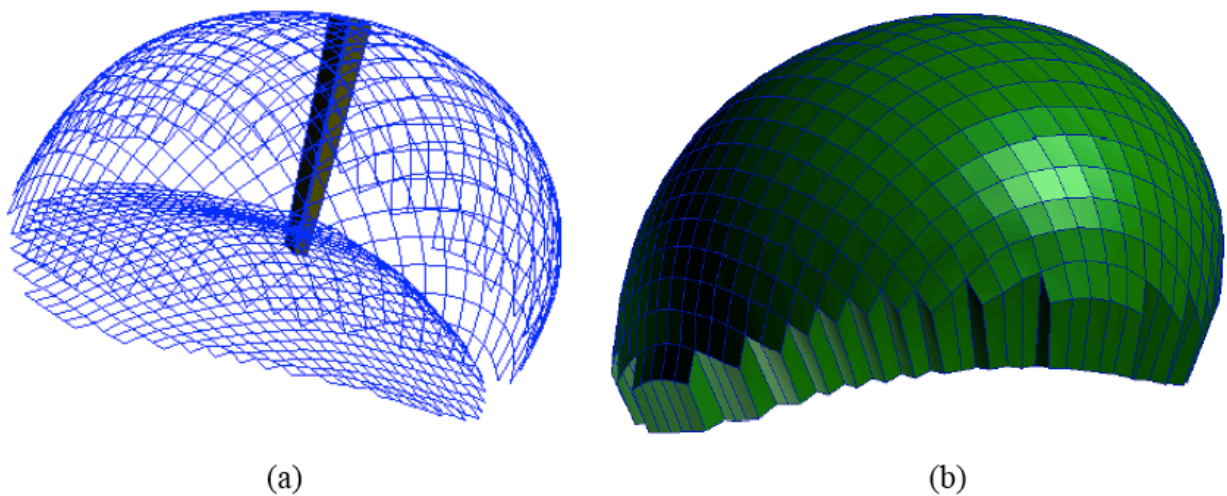


Figura 2: Generación de elementos hexaédricos en el volumen comprendido entre las dos mallas topológicamente equivalentes, empleando un programa comercial. (a) Esquema de disposición de un hexaedro que conecta un elemento cuadrilátero con su topológicamente equivalente. (b) Generación completa de hexaedros entre ambas mallas.

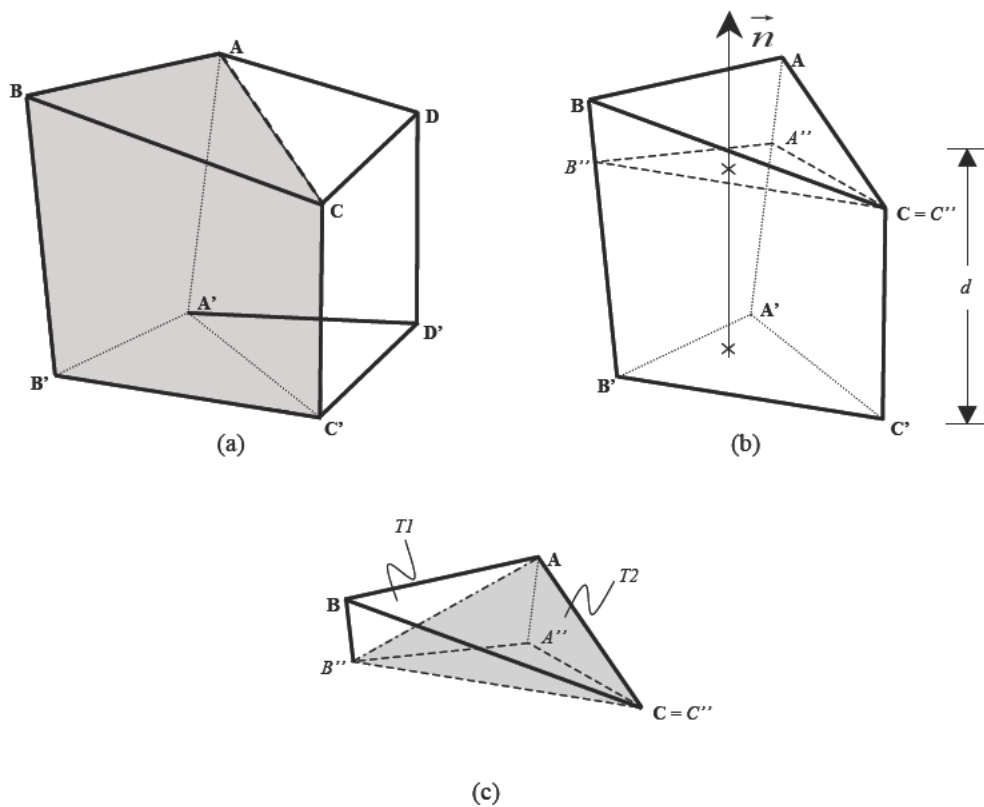


Figura 3: Descomposición de un hexaedro virtual $A'B'C'D' - ABCD$ en varias regiones con el fin de calcular su volumen. (a) Descomposición del elemento en dos prismas triangulares $A'B'C' - ABC$ y $A'C'D' - ACD$ de bases triangulares no paralelas entre sí en el caso general. (b) Descomposición del prisma triangular $A'B'C' - ABC$ en un prisma $A'B'C' - A''B''C''$ de caras triangulares paralelas y una pirámide $A''B''C''AB$. (c) División de la pirámide $A''AB''BC''$ en dos tetraedros T1 y T2.

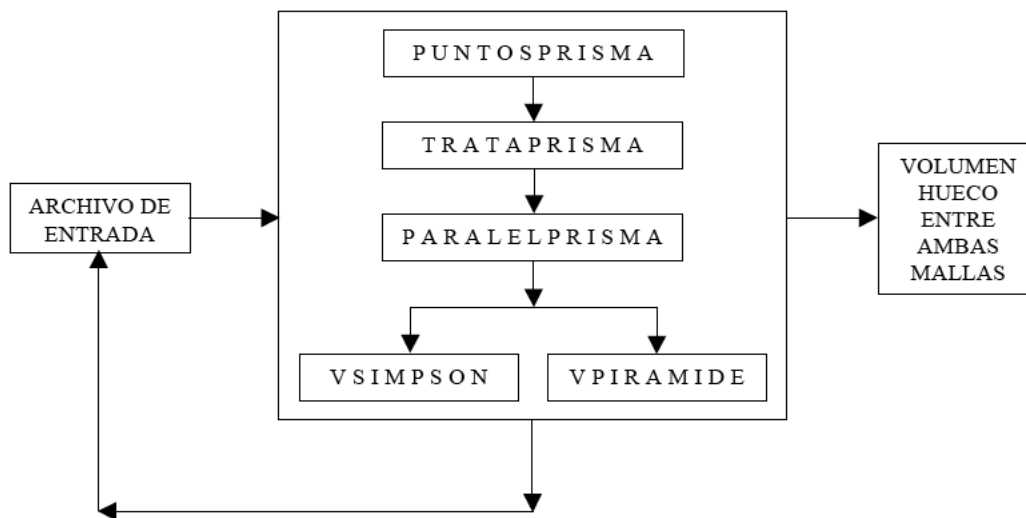


Figura 4: Esquema que ilustra el funcionamiento del código MATLAB[®] para calcular el volumen del espacio comprendido entre las dos mallas de elementos finitos.

Referencias

- [1] Zienkiewicz, O. C. y Taylor, R. L. *The Finite Element Method*. McGraw-Hill, Berkshire, UK, 1994.
- [2] The Mathworks, INC. MATLAB[®] 7.4.0.287 (R2007a) User's Guide. The Mathworks, INC. Natick, MA, USA.
- [3] MSC Software Corporation. MSC. PATRAN 2006 User Documentation. MSC Software Corporation, Santa Ana, CA, USA.
- [4] Hughes, T. J. R., *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs, NJ: Prentice-Hall, Inc. 1987.
- [5] Burden, R. L. y Faires, J. D., *Análisis numérico*. International Thomson, México, 2002.

