Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVIII, págs. 713-721 (artículo completo)
H.G. Castro, J.L. Mroginski, R.R. Paz, M.A. Storti (Eds.)
Resistencia, 1-5 Noviembre 2021

GAMMAS: UN PROGRAMA DE ELEMENTOS FINITOS PARA RESOLVER EDP EN 2D COMO PRÁCTICA BÁSICA DEL MÉTODO

GAMMAS: A FINITE ELEMENT PROGRAM FOR SOLVING 2D PDE AS A BASIC PRACTICE OF THE METHOD

Fernando G. Flores[a,b] **y Alejandro T. Brewer**[b]

[a]*Instituto de Estudios Avanzados en Ingeniería y Tecnología(IDIT) UNC-CONICET*

[b]*Departamento de Estructuras, FCEFYN, Universidad Nacional de Córdoba, Av. Velez Sarsfield 1611 , 5016 Cordoba, Argentina. fernando.flores@unc.edu.ar, http://www.inv.idit.efn.uncor.edu*

Palabras clave: Elementos Finitos, Enseñanza MEF, Práctica básica MEF,.

Resumen:

En este trabajo se describen algunos aspectos de un programa de elementos finitos para la solución de ecuaciones a derivadas parciales en dos dimensiones, destinado a ser usado como práctica básica en materias de grado y posgrado. El código sólo incluye elementos cuadráticos y condiciones de contorno sencillas. Resuelve problemas de campo (ecuación de Helmholtz) para distintos casos, elasticidad plana y flexión de placas de Reissner. El código se complementa con una interfaz gráfica (programa GiD) para el ingreso de datos que permite generar la malla y analizar los resultados. Se discuten los aspectos de implementación, cuáles debieran enfatizarse para la compresión global de la técnica y cuáles son irrelevantes para el usuario de un programa.

Keywords: Keywords: Finite Elements, FEM teaching, FEM basic practice.

Abstract:

This work describes some aspects of a finite element program for the solution of partial differential equations in two dimensions, intended to be used as basic practice at graduate and undergraduate levels. The code includes only quadratic elements and simple boundary conditions. The code solves field problems (Helmholtz equation) for different cases, planar elasticity and bending of Reissner plates. The code is complemented with an graphical interface (program GiD) for data entry, which allows to generate the mesh and to analyze the results. Implementation aspects are discussed, which should be emphasized for the overall understanding of the technique and which ones are irrelevant to the use of a program.

1. INTRODUCCIÓN

Tradicionalmente se ha considerado que un aspecto crucial para la comprensión del Método de Elementos Finitos (MEF) es la programación del mismo, es decir, los aspectos de implementación: la lectura de datos y la generación automática de parte de ellos, los arreglos internos, el almacenado de las variables, la numeración de las incógnitas, la determinación del ancho de banda, la solución del sistema de ecuaciones, la consideración de distintos casos de carga, la salida de resultados, etc. En esa línea casi todos los libros sobre el MEF incluyen códigos o partes de código fuente mostrando los detalles. Muchas veces en un lenguaje FORTRAN desactualizado en el que inicialmente se programó el método, sin apelar a técnicas modernas de programación. Esto conspira contra el interés del lector que no conozca el lenguaje FORTRAN o esté acostumbrado a formas más elaboradas de programación y documentación. La aparición de nuevos lenguajes de programación y sistemas de cómputo numérico ha permitido que otros autores hayan preferido utilizarlos para enseñar e implementar el MEF.

Por otro lado, los estudiantes de grado y especialmente los de posgrado suelen tener conocimientos básicos de lenguajes de programación como Python o sistemas de cómputo numérico como Matlab©, GNU Octave©, Scilab©, etc. Esto conlleva que las posibilidades para introducir los aspectos principales y particulares de implementación sean amplias, a todo lo cual, debe agregarse la propia experiencia y preferencia del docente con estos lenguajes y sistemas.

Por otro lado debe distinguirse entre un programa de elementos finitos (PEF) para un propósito específico, donde tal vez se programe un único elemento con un único propósito y una única técnica de resolución, de un PEF de propósito general, es decir, que pueda resolver un conjunto importante de problemas dentro de, por ejemplo, la mecánica de sólidos y estructuras, que incluya una variedad de elementos finitos con características muy distintas y que a su vez tenga la flexibilidad suficiente, para implementar nuevos elementos, técnicas alternativas de solución o nuevos problemas.

Cada aspecto de un programa de propósito general: flexibilidad en la entrada de datos, combinación de elementos, técnicas de solución, condiciones de contorno naturales, condiciones de contorno esenciales, restricciones multipunto, solución en etapas múltiples con posibles modificaciones en el modelo, salida de resultados, comunicación con programas de pre y postproceso, etc., que hacen a la potencialidad de un código, implican una enorme variedad de aspectos y detalles que hacen imposible intentar explicar y asimilar durante un curso semestral.

El uso de un PEF comercial, ya sea en una versión demo (limitada) o su versión completa, implica un aprendizaje y entrenamiento en las posibilidades del código y en las facilidades que ofrece para la construcción de un modelo. Cuanto más complejo es el código, mayor tiempo requiere el entrenamiento; y si el objetivo del curso es realizar una práctica básica, el tiempo dedicado a esa actividad debe ser el adecuado.

Por todo ello, en el desarrollo de un curso de elementos finitos resulta necesario elegir cuidadosamente:

- los aspectos de implementación a considerar y el detalle con el cual abordarlos,
- el lenguaje de programación o el sistema de computo numérico, en el caso que se haga énfasis en aspectos de implementación y
- el PEF a utilizar en la práctica básica.

En la experiencia de los autores, muy pocos estudiantes desarrollarán un código, por lo cual, resulta más importante hacer énfasis en los aspectos conceptuales del método y en realizar

una práctica básica que incluya el uso de una interfaz gráfica sencilla, que permita generar e interpretar resultados de modelos simples.

En la Sección 2 se presentan las generalidades del programa GAMMAS propuesto para ser utilizado en la práctica. En la Sección 3 se describen los aspectos de implementación que, en opinión de los autores, resultan de mayor relevancia y merecen incluirse en la temática del curso. En la sección 4 se describen los aspectos más importantes de la práctica básica y en la sección 5 se agrupan algunas conclusiones.

2. EL PROGRAMA GAMMAS

La primera versión del programa (1996) fue escrita en el estándar del lenguaje FORTRAN 90. Inicialmente, se concibió como un programa de código abierto, sencillo, con programación estructurada, que fuera "incompleto", en el sentido de que permitiera ser modificado y completado por los asistentes al curso. En ese momento, el código se acompañaba con el compilador Essential Lahey© Fortran (gratuito entonces). Este último elemento, era indispensable para el trabajo en el entorno Windows©. Pero sólo un par de alumnos de posgrado intentaron modificar el programa original. Por lo cual no se insistió con ello y el programa fue completado en sus aspectos básicos. Los resultados se podían visualizar con una antigua versión de Tecplot(Tecplot (2005)). Actualmente, el programa se comparte ya compilado pero el fichero fuente está disponible para quien lo requiera.

2.1. Características del programa

El objetivo es la solución de ecuaciones diferenciales a derivadas parciales de segundo orden lineales en dos dimensiones. Se han implementado dos elementos cuadráticos isoparamétricos: el triángulo de seis nudos (con integración numérica en 3 puntos, uno a la mitad de cada lado) y el cuadrilátero de nueve nudos (con integración numérica con 4 o 9 puntos).

Las ecuaciones diferenciales que se han implementado son (ver Flores y Brewer (2020)) :

- La ecuación de Helmholtz, con una variable escalar, cuya solución permite describir:
  - la distribución de tensiones de corte en la sección transversal de una viga debida a un esfuerzo de corte constante,
  - la distribución de tensiones de corte en la sección transversal de una viga debida a un momento torsor constante sin restricción de alabeo (Saint Venant), usando como incógnita el alabeo de la sección o la función de tensión (Prandtl),
  - el escurrimiento en un medio poroso (Darcy),
  - el flujo de calor, el flujo potencial con función potencial o líneas de corriente, etc.
- Las que describen problemas de elasticidad plana, con una variable vectorial: los desplazamientos en las dos direcciones
  - tensión plana,
  - deformación plana, y
  - sólido axilsimétrico.
- La correspondiente a flexión de placas, incluyendo deformaciones transversales de corte (Reissner-Mindlin), con tres incógnitas nodales, el desplazamiento transversal y las dos

componentes del giro de la normal. En este caso se utilizó un esquema de deformaciones impuestas para aliviar el bloqueo por corte transversal.

Los términos de fuente interna se evalúan automáticamente en función de la ecuación diferencial a resolver. Estos términos son estándar en la mayoría de los programas, salvo en el caso de torsión usando la función de alabeo y en el caso de esfuerzo de corte que incluyen términos que provienen de la integral por partes. Sobre el contorno, donde se conoce el flujo, se pueden incluir flujos distribuidos de variación cuadrática a lo largo de un lado de un elemento y también flujos puntuales en nudos elegidos. En el caso de problemas de elasticidad, es posible incluir cambios térmicos en cada nudo de la malla como estado de solicitación.

En lo que se refiere a condiciones de contorno esenciales, sobre cualquier nudo del contorno y en cualquier otro nudo interno, se puede imponer el valor de la variable (variables escalares) o el valor de las componentes cartesianas de la variable (en el caso de variables vectoriales). Se hace notar que, debido a que es un código con prestaciones básicas, no es posible incluir restricciones multipunto o indicar el valor de las componentes de una variable vectorial en un sistema distinto del global.

La entrada de datos del programa (*solver*) se realiza a partir de un archivo ASCII con una sintaxis bastante estricta que debe responder a lo descripto en la entrada de datos del programa; de otra forma conducirá a un error y la detención del programa. Para facilitar la entrada de datos se ha escrito una interfaz con el programa GiD-15.0. (2021) que permite ingresar los datos del modelo, generar la malla y escribir el archivo de datos ASCII sin cometer errores de sintaxis. La descripción de la interfaz se realiza más adelante.

La solución del sistema de ecuaciones simétrico se realiza en forma directa usando un esquema de almacenamiento que sólo incluye los elementos por debajo del *skyline* con la rutina `COLSOL` (Bathe y Wilson (1976)). Como facilidades adicionales, el programa incluye un renumerador de nudos cuyo objetivo es minimizar el número de elementos que quedan dentro del *skyline* de la matriz de coeficientes. Esto es innecesario en el caso de usar la interfaz con GiD que numera la malla en forma optimizada.

Los resultados se presentan en archivos ASCII: uno de ellos hace un eco de los datos leídos y los pasos que se van dando para la resolución y otros están destinados a ser leídos por post-procesadores gráficos. Los dos post-procesadores gráficos que se han considerado son: GiD, que de esta forma permite realizar la carga de datos y la lectura de resultados desde una misma interfaz gráfica, y Tecplot que requiere valores nodales para las variables, por lo cual es necesario realizar un suavizado nodal de las variables derivadas (flujos).

2.2. La interfaz de entrada de datos con GiD

La práctica básica con el MEF requiere una interfaz gráfica amigable para la generación de los datos en forma automática. Esta interfaz debe permitir:

- Introducir los datos generales del problema, indicando opciones y definiendo materiales
- Definir y modificar la geometría en forma sencilla, en este caso restringido a dominios bidimensionales
- Introducir las condiciones de contorno con facilidad
- Generar mallas con distintas características: estructuradas, no estructuradas, de tamaño variable con distintos criterios, que permita concentrar los elementos en zonas de interés

- Preferentemente ejecutar el *solver* desde dentro de la interfaz
- Preferentemente analizar los resultados desde la misma interfaz gráfica, para ver mapas de colores, deformadas en el caso de problemas de elasticidad, ver perfiles y/o variación de las variables a lo largo de cortes.

Estos requisitos hacen de GiD una buena elección teniendo en cuenta que, si bien es un programa comercial, tiene una versión académica que permite hacer mallas limitadas en la cantidad de nudos y además, una versión de prueba completa durante 30 días, que es suficiente para hacer la práctica básica. Por otro lado, la programación de GiD para generar un denominado *problem type* es sencilla para los principales aspectos de un programa acotado como GAMMAS. A partir del año 2005 se cuenta con la posibilidad de usar GAMMAS desde la interfaz de GiD, lo que permite la generación amigable del modelo numérico.

3. ASPECTOS DE IMPLEMENTACIÓN A CONSIDERAR

Un PEF de propósito general implica aspectos de implementación y recursos muy variados que son difíciles de presentar en un curso corto, por lo cual resulta necesario discernir los aspectos que deben abordarse y con qué nivel de detalle. Esto conduce a la necesidad de fijar un criterio para la elección de los temas; lo que a la vez depende de que el curso esté orientado a futuros desarrolladores del método o a usuarios que puedan hacer un uso inteligente de los recursos.

3.1. Entrada de datos y bases de datos

Casi todos los PEF leen los datos del modelo desde un archivo ASCII**,** o al menos tienen una opción para hacerlo. Originalmente la sintaxis de dichos archivos era muy poco flexible, con campos de longitud fija. Actualmente se hace uso de "palabras claves" y formato libre para darle la mayor flexibilidad posible. Sin embargo**,** esto no es relevante si el usuario no va a acceder y o modificar los archivos de datos. Es decir, si el usuario sólo va a trabajar desde la interfaz gráfica. Aún así, es muy bueno disponer de un archivo de datos amigable y de fácil lectura para aquellos que decidan modificar detalles o parámetros directamente en el archivo de datos.

Para un curso introductorio parece suficiente con utilizar la interfaz para escribir el archivo de datos (sin errores sintácticos) y mostrar un par de ejemplos de ellos. Sin embargo, y debido a que no todas las posibilidades que ofrece el código son accesibles a través de la interfaz, resulta conveniente contar con la posibilidad de modificar directamente el archivo de datos. Por ejemplo, GAMMAS permite considerar cambios térmicos cuyos valores deben ingresarse para cada nudo; y esta opción no ha sido programada en la interfaz, por lo cual, si quiere considerarse dicha solicitación debe modificarse el archivo de datos en forma directa.

Las bases de datos de los programas han evolucionado sustancialmente; la gestión de memoria se realiza en forma dinámica (no disponible en el estándar de FORTRAN 77); se ha incorporado el uso de tipos, listas, punteros, módulos, etc, lo que permite una gran flexibilidad y una amplia gama de posibilidades para el manejo interno de los datos. En el caso de querer abordar estos aspectos, un prerequisito necesario es el conocimiento en detalle de los alcances de un lenguaje de programación de alto nivel. En el caso de un curso introductorio, la base de datos interna debiera tratarse a partir de los requerimientos del código, pero sin entrar en los detalles de como puede programarse el mismo.

Un análisis del diagrama de flujo del programa permite entender el orden en que se leen los datos y lo que hace con ellos. Esto también permite las distintas etapas y los datos necesarios

en cada una de ellas.

3.2. Restricciones multipunto

Para la imposición de las condiciones de contorno esenciales es habitual y necesario indicar cómo establecer valores de las variables (nulos o no) en los nudos asociados al contorno en cuestión. Un tema mucho menos tratado son las restricciones multipunto. En un programa orientado al análisis de sólidos y estructuras o a la simulación de procesos industriales, es imprescindible contar con la posibilidad de imponer dicho tipo de restricciones, lo cual permite contemplar

- Apoyos que no coinciden con las direcciones del sistema global de coordenadas
- Nudos maestros
- Modelado de dominios con simetría cíclica
- Unión de elementos finitos con distintos grados de interpolación en el contorno o con diferentes grados de libertad
- Articulaciones en pórticos
- Rigidizadores de láminas
- Movimiento de cuerpos rígidos y herramientas
- Mecanismos, pasadores, mordazas
- etc.

Por lo cual, las restricciones multipunto tienen una importancia conceptual importante para el desarrollo de modelos numéricos. Resulta entonces necesario una descripción de las posibilidades que representan y las distintas técnicas que suelen usarse para su implementación, mencionando sus ventajas y desventajas. Nuevamente, se prefiere mantener un punto de vista conceptual y no entrar en la descripción de los detalles necesarios para la programación de las distintas técnicas. La ejemplificación de casos de restricciones multipunto permite tener una visión más amplia de las posibilidades que brindan los códigos para el desarrollo de modelos numéricos. Al contar con estos recursos, que simplifican las etapas de modelado, el analista puede enfocar su atención en otros aspectos y detalles relevantes del modelo.

3.3. Técnicas de solución del sistema de ecuaciones

Originalmente, debido a las limitaciones en almacenamiento tanto de memoria RAM como de espacio en disco, se hacía mucho énfasis en la forma de resolución del sistema de ecuaciones. Las características de la matriz de coeficientes: simetría, en banda o rala, no sólo condicionaban los problemas que podían efectivamente resolverse, sino también, la lógica interna del programa y la gestión de la memoria RAM. Actualmente este aspecto no aparece como un tema en el que deba profundizarse durante un curso básico del MEF. Sólo resulta necesario describir las posibilidades que pueden presentarse (directas, iterativas, seriales, paralelizadas, etc) en función de los recursos que brinda el hardware y el software de que se disponga, pero sin entrar en los detalles de los algoritmos.

Asociado con la técnica de solución, aparece la numeración interna de los nudos y la numeración de las ecuaciones; esto permite un uso eficiente de los recursos y requiere la definición de arreglos internos a los fines del ensamble del sistema de ecuaciones. En general, en la implementación no habrá una matriz de rigidez cuadrada donde ensamblar la matriz de rigidez elemental, por lo cual, este aspecto es dependiente no sólo de la forma de resolver el sistema de ecuaciones, sino también, de las condiciones de contorno impuestas y en particular de la existencia de restricciones multipunto. Por ello, alcanza con una definición conceptual del ensamble del sistema de ecuaciones sin entrar en otros detalles. Por otro lado, casi todos los alumnos han pasado por un curso de análisis estructural donde el proceso de ensamble básico ha sido explicado al tratar la aplicación del método de rigidez.

3.4. Salida de resultados y suavizado de variables

Obtenidas las incógnitas nodales, el paso siguiente es determinar qué resultados son importantes de mostrar y cómo. Por ejemplo, en el caso de torsión de Saint Venant, resulta relevante evaluar el valor de la rigidez torsional de la sección, o en el caso del esfuerzo de corte, la posición del centro de corte. Las incógnitas que calcula el MEF son las variables principales en los nudos de los elementos. Por otro lado, los flujos se evalúan en los puntos de integración debido a que se los asume como puntos de mayor precisión o de superconvergencia. Para las formulaciones estándar las variables principales resultan continuas y los flujos no. La visualización mediante mapas de colores de las distintas variables se realiza con los programas de posproceso gráfico. Hay algunos de ellos (Tecplot por ejemplo) que sólo admiten como datos los valores nodales, por lo cual, las variables de flujo deben evaluarse en los nudos mediante un proceso llamado "suavizado de variables". Mediante este tratamiento, se obtiene un valor único de la variable de interés para cada nudo, de tal forma que todos los mapas de colores resultan continuos. Otros programas (GiD por ejemplo) permiten incluir variables nodales y en los puntos de integración. Además, GID permite mostrar mapas de colores de las variables en los puntos de integración en forma discontinua entre elementos o presentar mapas continuos haciendo su propio suavizado.

Por un lado, los mapas de colores discontinuos (a partir de los puntos de integración) permiten observar y evaluar la calidad de la malla utilizada, y con ello, decidir si es necesario mejorar la discretización. Por otro lado, las diferencias entre los mapas sin suavizar y suavizados permiten una evaluación del error del modelo. Resulta entonces importante explicar cómo se realiza un suavizado, qué opciones sencillas existen y cómo podría usarse para evaluar el error y eventualmente cómo usar estos resultados para mejorar la malla. Además, es importante mostrar los casos donde las variables son efectivamente discontinuas y por lo tanto, en estas situaciones, un mapa de flujos suavizados es erróneo.

En relación con los nudos de la malla con condiciones de contorno esenciales, se pueden evaluar los flujos (reacciones) que mantienen el balance (equilibrio) en dichos puntos. Estos flujos sólo pueden evaluarse numéricamente como valores nodales equivalentes y habitualmente no se distribuyen sobre el contorno. En general, los códigos no muestran flujos puntuales en la interfaz; pero a veces si pueden verse en un archivo ASCII. Es importante lograr una adecuada interpretación de este tipo de resultados, de su relación con los flujos normales al contorno y del balance global del sistema.

3.5. Uso de sistemas de computo

Si bien en el presente trabajo se describen los enfoques de la enseñanza práctica de un curso de MEF enfatizando la aplicación de un programa de las características de Gammas, resulta necesario no descuidar el proceso educativo que conduce a la incorporación gradual de los distintos aspectos involucrados en su implementación (lectura de datos, cálculo de elementos, integración numérica, ensamble, condiciones de borde, etc.) mencionados previamente. Cada uno (o varios a la vez) de estos tópicos se aborda mediante trabajos prácticos que el estudiante resuelve en forma individual y la herramienta computacional que se utiliza es Matlab©/GNU Octave©. Un primer ejemplo se vincula a la solución de una estructura simple, un reticulado plano de cinco barras tomados del libro de Bhatti (2005), que permite, en muy pocas líneas, presentar aspectos como la lectura de datos (coordenadas de los nudos que conforman los elementos, conectividades, material), cálculo y ensamble de elementos (mediante la matriz de conectividades), imposición de las condiciones de borde naturales y esenciales, solución del sistema de ecuaciones y resultados. Si bien resulta inevitable, dada la diversidad de formación previa de los estudiantes, tener que introducir conceptos del lenguaje de programación, lo acotado del "script" y del modelo analizado, permiten acometer la tarea sin mayores inconvenientes. Modificaciones posteriores de este ejemplo permiten introducir las nociones de convergencia *"h"* (aumento del número de elementos) y convergencia *"p"* (aumento del orden de los polinomios de interpolación en el elemento). Los elementos de viga de Bernouilli y Timoshenko se utilizan para mostrar lo que se entiende por bloqueo de la solución y algunas estrategias para evitarlo. Las facilidades gráficas de este tipo de programas permiten visualizar las funciones de interpolación tanto en elementos unidimensionales como en elementos planos. Los distintos órdenes de aproximación de la integración numérica y la comparación de los resultados obtenidos, se presentan resolviendo áreas, centros de gravedad y momentos de inercia de figuras planas construidas con elementos planos triangulares o cuadriláteros. Esta apretada síntesis muestra las estrategias utilizadas para incorporar los distintos conceptos y aspectos computacionales que luego se ordenan y concatenan para conformar un código que permita plasmar las características enunciadas en la sección 2 de este trabajo.

4. LA PRÁCTICA BÁSICA

Todo programa tiene una descripción detallada de su entrada de datos y toda interfaz gráfica una extensa documentación respecto a sus posibilidades. Pero no se pretende, en un primer contacto, la lectura de la documentación, sino, que la práctica básica tiene como objetivo que el estudiante tenga una primera experiencia en la confección de un modelo numérico. Esto requiere de la existencia de tutoriales introductorios y, en el caso de programas de propósito general, de tutoriales avanzados para el abordaje de diferentes problemas físicos. Los tutoriales introductorios son fundamentales para guiar al estudiante en los distintos pasos, en el caso que nos interesa aquí:

1. Instalación del programa, obtención de una licencia temporal
2. Ejemplos guiados para la generación de un modelo
3. Datos básicos del modelo, tipo de problema, materiales
4. Generación de la geometría, bidimensional en este caso**,** que es más sencilla que una tridimensional

5. Imposición de las condiciones de contorno, naturales y esenciales
6. Generación de mallas: elección del tipo de elemento, mallas estructuradas y no estructuradas, graduación de la misma.
7. Ejecución del programa
8. Análisis de los resultados, deformadas (elasticidad), mapas de variables, cortes, gráficos, etc.

Los aspectos 4, 6 y 8 en los cuales la interfaz gráfica ofrece muchas variantes, son los que pueden desalentar al estudiante, y son estos aspectos en los que se debe enfatizar la guía, para permitirle hacer un modelo básico, generar confianza y alentar el aprendizaje de nuevas posibilidades.

Para GAMMAS se han escrito dos tutoriales**:**

- Uno para elasticidad 2D, orientado a estudiantes de grado que toman un curso de elasticidad que incluye un capítulo dedicado a los fundamentos del MEF. Su lectura les permite resolver también problemas de placas.
- Determinación de la distribución de tensiones de corte en una sección debidas a esfuerzo de corte y debidas a torsión de Saint Venant. Orientado a estudiantes que han tomado un curso de resistencia de materiales, y que no conocen los fundamentos del MEF. En este caso se hace particular énfasis en la elección de las condiciones de contorno, condiciones de simetría y antisimetría, necesarias para la construcción correcta del modelo numérico.

Ambos tutoriales pueden ser utilizados por estudiantes del curso de posgrado.

5. CONCLUSIONES

En este trabajo se han presentado los aspectos principales de un programa de elementos finitos para la práctica básica del mismo. Se ha hecho énfasis en los aspectos que a juicio de los autores son los que debe darse mayor importancia. Esto implica que en el curso se combinen:

- El uso de sistemas de cómputo, que con pocos elementos de programación permitan ver en forma sencilla el llevado a la práctica del diagrama de flujo de un programa.
- El uso de un software comercial para generar el modelo numérico (definición del dominio, condiciones de contorno y mallado) y también para su uso como visualizador de resultados
- Un programa académico limitado que permite resolver una variedad de problemas de interés dentro del área de la elasticidad y problemas de campo.

REFERENCIAS

Bathe K. y Wilson E. *Numerical Methods in finite element analysis*. Prentice Hall, Englewood Cliffs, 1976.

Bhatti M. *Fundamental Finite Element Analysis and Applications*. John Wiley & sons, New Jersey, 2005.

Flores F. y Brewer A. *El método de elementos finitos. Curso Introductorio*. Departamento de Estructuras, FCEFyN, UNC, 2020.

GiD-15.0. *The personal pre and post processor*. International Center for Numerical Methods in Engineering, UPC, Barcelona, 2021.

Tecplot. *Tecplot 10 Release 6*. Tecplot, Inc. Bellevue, Washington, EEUU, 2005.