

x

TJ 217

M45

REPUBLICA DE VENEZUELA
UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
POSTGRADO EN INGENIERIA ESTRUCTURAL

**GENERADOR DE MALLAS H-ADAPTATIVO APLICADO A
PROBLEMAS TRIDIMENSIONALES USANDO INDICES DE
DENSIDAD DE ENERGIA DE DEFORMACION**

Por

Carlos Gregorio Meléndez Dugarte

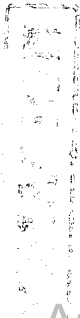
WWW.BDIGITAL.ULA.VE

Mérida, Octubre de 1999

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

U.L.A. Consejo de Estudios de Postgrado



INDICE GENERAL

	Pag.
Resumen	v
Abstrac	vi
Planteamiento del Problema	vii
Justificación e Importancia	ix
Objetivos	x
Introducción	1
Capítulo 1 Formulación con Elementos Finitos a problemas Tridimensionales.	
1.1 Introducción.....	4
1.2 Utilidad del Método de los Elementos Finitos.....	5
1.3 Concepto de Elemento Finito.....	5
1.4 Forma Geométrica de los Elementos.....	6
1.5 Discretización de una Estructura.....	8
1.6 Solución del Problema de Elementos Finitos.....	9
1.7 Pasos de un Análisis con Elementos Finitos.....	9
1.7.1 Pre-Procesamiento.....	10
1.7.2 Análisis.....	10
1.7.3 Post-Procesamiento.....	10
1.8 Análisis de los Resultados.....	11
1.9 Conceptos básicos de elementos finitos en tres dimensiones.....	11
1.9.1 Trabajo realizado por las cargas externas.....	12
1.9.2 Energía de deformación almacenada en el cuerpo cargado.....	12
1.9.3 Relación de esfuerzo-deformación para problemas tridimensionales.....	13
1.9.4 Campos de desplazamiento asumido.....	14
1.9.5 Elemento hexahédrico básico para análisis tridimensional.....	16
1.10 Principio de los Trabajos Virtuales.....	16
1.11 El Método de los Desplazamientos como Minización de la Energía Potencial Total.....	18
Capítulo 2 Método de los Elementos Finitos Adaptativos	
2.1 Introducción.....	24
2.2 Estrategias de refinamiento Adaptativo.....	26
2.2.1 Refinamiento tipo R.....	26
2.2.2 Refinamiento tipo H.....	27
2.2.3 Refinamiento tipo P.....	27
2.2.4 Refinamiento tipo HP.....	28
2.2.5 Refinamiento tipo D.....	28
2.2.6 Refinamiento tipo HM (H-Method).....	28
2.3 Análisis sobre estrategias adaptativas.....	29
2.4 Fuentes de Error.....	30
2.4.1 Error del Modelado Matemático.....	31
2.4.2 Error Computacional.....	31
2.4.3 Error debido a Singularidades.....	32
2.4.4 Error de la Aproximación del Funcional.....	32

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

2.4.5 Error en la Geometría.....	32
2.4.6 Error de Discretización.....	33
2.4.7 Error de Redondeo.....	33
2.4.7.1 Error Básico.....	33
2.4.7.2 Número de condición del problema numérico.....	34
2.4.8 Error por selección incorrecta del elemento.....	34
2.4.9 Error debido a Inestabilidad.....	35
2.4.10 Error en el número de elementos utilizados al discretizar.....	36
Capítulo 3 Proceso de Refinamiento	
3.1 Introducción.....	38
3.2 Refinamiento por Indices de Densidad de Energía de Deformación.....	38
3.3 Distribución Uniforme del error.....	40
3.4 Nuevos Criterios de refinamiento y Estimación de Error.....	43
3.4.1 Criterio de refinamiento h-adaptativo modificado.....	43
3.4.2 Estimador de Error Direccional.....	45
3.4.3 Estimador en términos de Momento Flector y Fuerza Cortante.....	46
3.4.4 Técnicas de Reconstrucción de Parches Superconvergentes.....	48
Capítulo 4 Generadores de Mallas Tridimensionales	
4.1 Clasificación de los Generadores de Malla.....	51
4.1.1 Técnicas de Macro Elementos.....	52
4.1.2 Técnicas de Mallado Libre.....	52
4.2 Generador H-Adaptativo.....	52
4.2.1 Características de GEMATRI.....	55
4.2.2 Proceso de generación adaptativa.....	63
4.2.3 Imposición de restricciones.....	63
Capítulo 5 Resultados Numéricos	
5.1 Introducción.....	67
5.2 Junta de unión viga-columna.....	69
5.3 Viga en voladizo con empotramiento defectuoso.....	73
Conclusiones.....	76
Referencias Bibliográficas.....	78
Anexos.....	81

RESUMEN

Es esta investigación se presenta un proceso de refinamiento adaptativo de mallas tridimensionales. Este h-refinamiento usa técnica de macro elementos para la generación de mallas, de igual manera utiliza los índices de energía de deformación SEDs en orden de localizar y evaluar la presencia de singularidades, con la finalidad de lograr un mallado fino en los sitios donde sea necesario, a un costo computacional bajo. Además se presentan algunos ejemplos numéricos elásticos tridimensionales para mostrar la efectividad del procedimiento desarrollado.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - ^vCompartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

ABSTRACT

An process for adaptive 3D mesh refinement is presented. This h-refinement uses a macro-element technique for generating the mesh and relies upon the strain energy density indices SEDs in order to locate and evaluate the presence of singularities, allowing the process to produce a finer mesh in those areas where it is needed with a low computational cost. Several numerical examples from three dimensional elasticity are presented to show the effectiveness of the procedure developed.

WWW.BDIGITAL.ULA.VE

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el análisis de importantes estructuras de la mecánica aplicada se desarrollan con la utilización del método de los elementos finitos, lo que implica que la escogencia adecuada de la malla y especialmente el tamaño de los elementos finitos en cada parte de ella, determina la calidad de la solución numérica y el trabajo computacional involucrado en el proceso. Obviamente, el desarrollo de técnicas de mallado para modelos tridimensionales se ha producido más lentamente que para modelos planos por dos razones fundamentales:

- En primer lugar, la dificultad de almacenar y disponer de todos los datos necesarios para la correcta definición de un modelo tridimensional de forma cómoda. En este sentido cabe destacar que los modernos sistemas de un modelo sólido han resuelto en gran parte esta dificultad al permitir sus bases de datos disponer de la información necesaria para definir correctamente los modelos tridimensionales de manera eficaz.
- En segundo lugar, la propia dificultad de desarrollar un método de mallado válido para discretizar un sólido de forma general o con el menor número posible de restricciones.

Para la creación del mallado de modelos tridimensionales hay que desarrollar el trabajo tanto en forma manual como intuitiva debido a la carencia de un sistema automatizado. Esto trae como consecuencia la dificultad de discretizar el mallado y reducir el número de restricciones; lo cual conforma el problema a plantear en este trabajo.

Se hace necesario entonces la creación de un software Generador de Mallas Tridimensionales que permita discretizar al máximo dicho mallado y reducir al mínimo el número de restricciones antes mencionadas.

El tipo de investigación emprendida en el presente trabajo, desde el punto de vista de propósito o razón de la misma es del tipo “Investigación Aplicada”, ya que se satisface necesidades en términos específicos y concretos. Además permite generar conocimientos de aplicación inmediata, así como también aportar soluciones

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

a la problemática de generación de mallas automáticas adaptativas a problemas tridimensionales sólidos, tema este de mucha actualidad dentro de Centros de Investigación Nacional e Internacional.

Desde otro punto de vista como lo es el nivel de conocimiento a obtener con la investigación , la misma debe ser catalogada como una “Investigación Explicativa”, ya que se indaga y estudia las variables y/o características que tienen que ver con el uso de índices de densidad de energía de deformación como indicadores de refinamiento y de esta manera poder evaluar y localizar las zonas de singularidades, presentes en la malla de los elementos finitos de una estructura tridimensional a analizar.

A la luz de este razonamiento, nuestro problema a investigar será formulado de la siguiente manera: ¿Es relevante establecer un proceso de refinamiento automático adaptativo de mallas aplicado a problemas tridimensionales sólidos, donde se logre discretizar al máximo el mallado ?.

WWW.BDIGITAL.ULA.VE

JUSTIFICACION E IMPORTANCIA

Una respuesta pertinente a esta interrogante impulsa a realizar una investigación sobre la creación de un proceso de refinamiento automático adaptativo de mallas aplicado a problemas tridimensionales sólidos, utilizando criterios de refinamiento que permitan discretizar al máximo el mallado.

En esta investigación se elaborará un software, que no sólo va facilitar la creación y el refinamiento automático del mallado de una estructura tridimensional (3D), sino que además va permitir abarcar los siguientes objetivos:

- La creación de mallas de elementos que permitan lograr modelos de geometría compleja, lo que traería como consecuencia favorable el poder modelar con mayor precisión la estructura y por ende un mejor análisis de la misma.
- La discretización al máximo de la malla, disminuyendo a la mínima expresión los errores que esta discretización conlleva.
- El control de la densidad del mallado en diferentes zonas de la estructura, permitiendo de esta manera, mejorar el diseño de la misma.
- Lograr la menor intervención del usuario, lo que se traduciría, en esfuerzo mínimo y costo computacional bajo.
- Lograr patrones de exactitud para mejorar la calidad de la solución numérica por medio de resultados más precisos y confiables.

OBJETIVOS

Objetivo General:

Establecer un software que permita el proceso de refinamiento automático h-adaptativo aplicado a problemas tridimensionales sólidos, con el propósito de discretizar al máximo el mallado.

Objetivos Específicos:

- Diseñar un generador de malla adaptativo fundamentado en la técnica de macro elementos que permita evaluar y localizar las singularidades automáticamente haciendo uso como indicadores de refinamiento los índices de densidad de energía de deformación (SED), bajo el lenguaje de programación Fortran.
- Crear un proceso de refinamiento adaptativo basado en la versión H-refinamiento, en lenguaje de Programación Fortran.
- Elaborar un algoritmo para incorporar al generador de mallas adaptativo la capacidad de transformación de coordenadas, de forma tal que, la geometría de algunos o de todos los superelementos del dominio pueda ser especificada en sistemas de coordenadas diferentes al global.
- Elaborar un algoritmo que permita localizar los nodos de la malla de cualquier estructura tridimensional que presenten incompatibilidad y poder asociar a ellos los nodos necesarios para la imposición de restricciones.

INTRODUCCION

El método de los elementos finitos se ha convertido en una herramienta poderosa para resolver los complejos problemas que aparecen en las ciencias aplicadas. La existencia de una gran cantidad de programas basados en el método de los elementos finitos y otros métodos aproximados, abren posibilidades casi ilimitadas para el progreso de la ingeniería. Problemas insolubles anteriormente pueden ser tratados hoy en día mediante estas técnicas y al menos en principio se puede conseguir un mejoramiento en el diseño.

Hasta hace muy poco tiempo el diseño y forma de la malla, de cualquier problema de elementos finitos, eran dejados a la intuición y experiencia de los analistas estructurales. Por esto, actualmente investigadores de este campo le están dando mayor importancia a la obtención de una forma automática de mallas de muy buena calidad dedicando grandes esfuerzos al estudio del mallado adaptativo, de modo que se pueda corregir una malla a partir de la cuantificación de alguna magnitud representativa de la calidad de la misma, siguiendo algún esquema de refinamiento.

La necesidad de elaborar procesos adaptativos con la finalidad de automatizar la elaboración de mallas en el análisis de estructuras complejas de ingeniería ha despertado mucho interés en la comunidad científica especializada. Son muchos los trabajos que a lo largo del tiempo se han realizado sobre problemas en 2D, mientras que para el caso de problemas tridimensionales (3D) las investigaciones se han desarrollado más lentamente y esto se justifica por la complejidad de la geometría de dichos problemas.

Los profesionales de la Ingeniería, especialistas en el análisis estructural, requieren resolver una gran cantidad de problemas complejos. Estos analistas están conscientes de que la etapa más costosa del método consiste en la elaboración de una malla adecuada a las exigencias de la estructura para representar lo más fielmente posible el modelo a analizar. Tomando en consideración esta premisa el trabajo que se presenta es un aporte importante en la solución técnica de los problemas confrontados en el quehacer diario de estos especialistas.

En este trabajo se presenta un proceso de refinamiento automático H-adaptativo de mallas aplicado a problemas de sólidos tridimensionales, por lo que el objetivo central fue el diseño de un generador de malla adaptativo que lleva por nombre *GEMATRI*, en función del tipo de problemas que permite resolver, es decir, *GE*nerador de *MA*llas *TRI*dimensionales.

La elaboración de esta investigación, implicó en primer lugar, la revisión bibliográfica de trabajos relacionados con el tema (Febres ,1984, 1988), (Buenaño y Febres ,1996), entre otros, con la finalidad de obtener información acerca de cuanto se ha logrado hasta la fecha sobre los elementos finitos adaptativos, así como también, para reforzamiento y profundidad de los conocimientos que posee el autor. En segundo lugar se efectuó una constante consulta a distintas fuentes de información escrita actualizada sobre el tema en la biblioteca de Intevp - Caracas, dentro de las cuáles se pueden destacar, *Computers & Structures*, *International Journal for Numerical Methods in Engineering*, *Revista Internacional de Métodos Numéricos para el cálculo y diseño en Ingeniería*, entre otras.

Para la realización de este trabajo de investigación, además de analizar críticamente dichas fuentes de información escrita sobre el tema, se realizaron algoritmos para la generación de mallas automáticas, los cuáles fueron finalmente validados con el análisis de algunos problemas existentes en la literatura.

Este software fue elaborado en lenguaje Fortran y funciona bajo ambiente Unix, y puede considerarse un instrumento de trabajo muy útil para los profesionales de la Ingeniería que incursionen en el análisis de estructuras complejas aplicando el método de los elementos finitos.

El proceso de refinamiento presentado tiene como soportes fundamentales la técnica de Macro-elementos, Los Indices de Densidad de Energía de Deformación (SED) y un proceso de refinamiento adaptativo basado en la versión H-refinamiento.

El contenido de este trabajo se encuentra dividido en cinco capítulos. En el primero de estos se describe el método de los elementos finitos en forma general y se presenta la formulación de un elemento finito tridimensional.

El capítulo dos intenta presentar en forma muy breve las diferentes estrategias adaptativas estudiadas en la actualidad y un análisis comparativo de ellas. El capítulo

tres se presenta las fuentes de error en el análisis por elementos finitos y las medidas adoptadas para su medición. Debido a la importancia que presentan los estimadores de error en los procesos adaptativos, al guiar los procesos de refinamiento y validar los resultados obtenidos, se presenta una clasificación de los estimadores encontrados en la literatura.

El capítulo cuatro está enfocado a los criterios de refinamiento haciendo énfasis en el uso de Los Indices de Energía de Deformación (SED) y su utilización por otros autores, así como también se hace referencia a los tipos de generadores de mallas automáticos y su importancia en los procesos adaptativos. Se hace una clasificación de estos basado en la técnica empleada para su generación. Se presenta el generador de mallas adaptativas GEMATRI el cual está basado en la técnica de superelementos y tiene la capacidad de crear mallas en tres dimensiones, así como también se especifica el tratamiento que se aplica al problema de imposición de restricciones.

El capítulo cinco contiene resultados de varios ejemplos numéricos obtenidos del procedimiento adaptativo presentado en los capítulos anteriores. Por último se presentan las conclusiones del presente trabajo, así como algunas perspectivas para futuras investigaciones.

CAPITULO 1

FORMULACION EN ELEMENTOS FINITOS DE PROBLEMAS TRIDIMENSIONALES

1.1 Introducción

A lo largo de las últimas décadas y en el ámbito de la Ingeniería Civil, se puede constatar la creciente complejidad de los problemas que son abordados de manera cotidiana. Este fenómeno nace fundamentalmente de la necesidad de aproximar cada vez más entre sí la realidad y la idealización que de ella realizamos. Dicha complejidad exige desde el punto de los métodos numéricos por un lado, contar con herramientas de cálculo cada vez más potentes, y por otro lado, disponer de programas y estrategias de análisis capaces de afrontar de manera eficiente situaciones cada vez más cercanas a la realidad. [Bretones, Rodríguez y Huerta, 1996]

El método de los elementos finitos es una técnica numérica para resolver los problemas complejos de la mecánica aplicada. Como punto de partida este método discretiza la región de definición del problema por medio de una partición geométrica denominada malla. A cada elemento de esta malla se le asocia un espacio funcional, usualmente un polinomio, y un conjunto de condiciones conocidas en la literatura como " grados de libertad". A esta trilogía se le denomina elemento finito.

El problema inicial es planteado ahora en el nuevo espacio de dimensión finita, convirtiéndose de esta forma en un problema discreto que puede ser resuelto por métodos aproximados. El problema final que se formula en cualquier análisis del elemento finito, así como en el análisis de cualquier estructura, tiene la forma de un sistema de ecuaciones simultáneas. Pudiéramos decir con toda propiedad, que problemas insolubles anteriormente, pueden ser tratados hoy en día mediante estas técnicas y al menos en principio se puede conseguir un mejoramiento significativo del diseño.

El método de los elementos finitos se ha convertido en una herramienta poderosa para resolver los complejos problemas que aparecen en las ciencias

aplicadas. Por esta razón, la gran cantidad de programas que se encuentran disponibles basados en elementos finitos y otros métodos aproximados, abren posibilidades casi ilimitadas para el progreso de la ingeniería. Como se quiera, muchos peligros han surgido, ya que estas facilidades son puestas al alcance de una comunidad muy amplia, que frecuentemente no posee los conocimientos necesarios ni la adecuada educación, obteniéndose resultados desastrosos cuando los análisis se realizan con el enfoque de “Caja Negra”. [Buenaño y Febres, 1996].

El diseño de modelos tridimensionales es un proceso en el que se consume mucho tiempo, tanto en la preparación de los datos como en el análisis, debido a que el número de grados de libertad es muy alto y las formas geométricas de las estructuras son usualmente complejas. En este capítulo se verá una introducción del método de los elementos finitos en forma general y los conceptos básicos para la formulación de un elemento finito tridimensional, que son la base fundamental de esta investigación..

1.2 Utilidad del Método de los Elementos Finitos

No es necesario hacer especial hincapié de la importancia y extensión que hoy en día tiene el método numérico de los elementos finitos para resolver en forma aproximada gran cantidad de problemas complejos de la ingeniería. [Hormaza et al., 1995]. Este método se usa principalmente para resolver problemas para los cuales no existe una solución exacta y requiere de un gran trabajo computacional cuando se aplica a problemas reales. Las tres áreas principales en Ingeniería donde se viene aplicando con éxito el método de los elementos finitos son: Mecánica de los Sólidos, Transferencia de Calor y Mecánica de Fluidos.

1.3 Concepto de Elemento Finito

Un elemento finito es una sub-región de un continuo discretizado. El método de los elementos finitos permite convertir un problema de un número infinito de grados de libertad a uno con un número finito, con la intención de simplificar el proceso de

solución. Aún cuando su aplicación inicial estaba orientada a la Mecánica de los Sólidos, su uso ha sido extendido a otros campos de bases matemáticas similares [Weaver, Jr. Y Johnston, 1984]

Cualquier problema físico puede ser dividido en cierto número de áreas o volúmenes que son llamados elementos finitos. Para resolver problemas de análisis de comportamiento lineal (estático y dinámico) de las estructuras continuas sometidas a acciones diversas con infinitos *grados de libertad* es necesario estudiar un modelo matemático de estructura dividida (*discretizada*) en partes pequeñas denominadas elementos finitos. Estos elementos finitos están conectados entre sí solamente a través de los llamados *puntos nodales* o nodos de borde de los elementos finitos reproduciendo la estructura real lo más fielmente posible y pasando de una estructura de infinitos grados de libertad a una de un número finito. [Fornons, 1982]. Al ensamblaje de estos elementos finitos o pequeñas áreas o volúmenes se le denomina malla.

El método de los elementos finitos, como técnica numérica que permite obtener la solución aproximada a un problema de ingeniería, convierte el manejo de ecuaciones diferenciales en un conjunto de ecuaciones algebraicas lineales. Su gran aceptación se debe principalmente a la facilidad con que estas ecuaciones pueden reunirse y resolverse por computadora.

1.4 Forma Geométrica de los Elementos Finitos

En este apartado se efectuará una breve descripción de los elementos más empleados en el análisis de comportamiento de distintos tipos de estructuras. Los elementos finitos están conformados por nodos, los cuáles están colocados en las esquinas de dichos elementos y también en los puntos medios de las caras o aristas que lo conforman y en algunos casos dentro de ellos, específicamente en su centro de gravedad. Los nodos son los puntos donde se conectan los elementos finitos entre sí. Los elementos finitos pueden ser unidimensionales (Lineal para modelar barras o vigas), bidimensionales (Triangular y Cuadrilátero para modelar membranas o placas) y tridimensionales (Tetraédro y Hexaedro para modelar sólidos, placas gruesas o

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

conchas). Debido a esta condición se puede modelar con elementos finitos líneas, superficies o volúmenes.

1.4.1 Elemento barra

El elemento barra tiene una gran importancia debido a que su empleo en las estructuras constituidas por barras en conexión con partes estructurales es muy grande.

1.4.2 Elementos triangulares

El elemento triangular es el elemento finito por excelencia. Es un "comodín", ya que con él se puede aproximar cualquier forma estructural continua, plana o espacial, en forma de lámina y bordes curvos, siempre que disminuya adecuadamente el tamaño de los mismos y en consecuencia se aumente su número en la discretización de la estructura.

1.4.3 Elementos Rectangulares

El elemento rectángulo es un elemento útil, por su facilidad computacional, cuando sus lados son paralelos a los ejes coordenados globales y la estructura no tiene bordes curvos. Este elemento tiene gran importancia en el estudio de flexión de placas delgadas [Buenaño y Febres, 1995], y al igual que en el caso del elemento triángulo, es muy importante para el estudio de elementos hexaédricos con caras curvas.

1.4.4 Elemento Cuadrilátero

Este elemento puede estudiarse como la suma de dos elementos triángulo.

1.4.5 Elementos Tetraédricos

Estos elementos son empleados principalmente para el estudio de problemas de elasticidad espacial y en el estudio de placas gruesas sometidas a cargas transversales.

1.4.6 Elementos Curvos

Estos refinados elementos de bordes curvos o distorsionados son especialmente útiles para ser adaptados a estructuras continuas con perímetros no rectos.

1.4.7 Elementos Asimétricos

Estos elementos tienen gran interés práctico para el estudio de las estructuras resistentes con ejes de simetría, como pueden ser: depósitos, estructuras de laminas delgadas, etc., por la posibilidad de simplificación de formulación que presentan.

1.5 Discretización de una Estructura

La discretización o formación de una malla de la estructura que se pretende estudiar con elementos finitos, es una de las etapas más importantes y difíciles de precisar del método de los elementos finitos, ya que está condicionada por dos factores importantes como son: el grado de exactitud de la solución a obtener y el costo computacional involucrado.

La mayor exactitud al mínimo costo computacional parece que es la opción más adecuada; pero para tal fin, no existen reglas fijas que permitan decidir al ingeniero respecto del tamaño y forma de los elementos a emplear en su problema. Esto es, emplear elementos pequeños y de pocos nodos, o bien, elementos grandes y de muchos nodos. [Fornons, 1982].

En la discretización se deben emplear elementos finitos de la misma forma aún cuando en algunos problemas estructurales se pueden tomar de forma diferente. Cuando usamos el método de los elementos finitos para el modelado numérico de un problema físico, la selección de un elemento apropiado es un factor muy importante que afecta la seriedad y eficiencia del modelo numérico. Una selección incorrecta del elemento puede, en muchos casos, resultar en soluciones divergentes del problema. [Lee y Lo, 1997] El sentido común del ingeniero debe jugar un papel muy importante en cuanto a la exigencia con que deben ser aplicados los criterios de convergencia del modelo físico-matemático a su problema práctico.

Es bien sabido que una malla de elementos finitos debe converger a resultados exactos a medida que la malla inicial es discretizada o refinada, es decir, que la solución aproximada de los desplazamientos tiendan a la solución exacta.

Para lograr esta convergencia se deben cumplir las siguientes condiciones:

- a) Los elementos deben hacerse lo suficientemente pequeños para que cualquier punto del dominio de la estructura estudiada esté siempre dentro de un elemento.
- b) Cualquier malla de discretización debe estar contenida en las mallas que se construyan más refinadas.
- c) *Las funciones de interpolación o de forma* deben continuar siendo las mismas; esto es, deben continuar teniendo la misma forma dentro del proceso de refinamiento de la malla.

1.6 Solución del Problema de Elementos Finitos

El análisis del ensamblaje de los elementos finitos nos provee de los desplazamientos, dependiendo cual sea el caso, de cada uno de los nodos para ciertas cargas aplicadas al modelo. Los desplazamientos de los otros puntos dentro del elemento se interpolan a partir de los valores de los desplazamientos en los nodos que correspondan al elemento.

Cada nodo tiene cierto número de posibles desplazamientos llamados *grados de libertad*. En el espacio o tridimensionalmente cada nodo tiene seis (6) grados de libertad, tres (3) desplazamientos y tres (3) rotaciones. Conocidos los desplazamientos se pueden obtener también las reacciones, las deformaciones y los esfuerzos. Este análisis requiere de gran cantidad de esfuerzo computacional si se desea estudiar un modelo realista de un problema de ingeniería.

1.7 Pasos de un Análisis con Elementos Finitos

Un análisis usando elementos finitos requiere de la disponibilidad de un programa de computación especializado en estos análisis. Existen numerosos programas de elementos finitos en el mercado actual : ABAQUS, NASTRAN, FEA, AQUA, ADINA, ANSYS, mTABS, etc. que pueden resolver un gran rango de problemas en ingeniería. El uso de cualquiera de ellos para el análisis de un problema real sigue los siguientes pasos:

- a) Pre-procesamiento.
- b) Análisis.
- c) Post-procesamiento.
- d) Análisis de los resultados.

1.7.1 Pre-procesamiento:

El pre-procesamiento consiste en la creación del modelo y la definición de la manera como será cargada y apoyada la estructura a modelar. El modelo suele hacerse con un paquete gráfico que a menudo viene incluido con el programa o importado de un programa de CAD. El software que se propone en esta investigación puede funcionar como un simple preprocesador, ya que permite, una vez conocida la topología del modelo a analizar, crear un archivo de entrada que puede ser leído por el programa comercial ABAQUS. El éxito del análisis es determinado en gran parte por la destreza del usuario para definir que simplificaciones deber ser introducidas en el modelo para que represente el objeto real, así como también la escogencia de elementos a ser usados y el diseño de la malla y por último la imposición apropiada de restricciones o apoyos.

El resultado final de este paso es un archivo de datos en el que se le indica al programa que hacer y con que trabajar.

1.7.2 Análisis:

El programa de elementos finitos primeramente lee los datos del archivo de entrada para luego realizar ciertos chequeos de esa información a fin de detectar errores. Si no existen errores se realiza el análisis y se produce un archivo de resultados. Este paso es completamente transparente para el usuario, por lo que el proceso de análisis está contenido en una especie de "caja negra" donde el usuario no tiene acceso.

1.7.3 Post-procesamiento:

El post-procesador toma la información de los archivos de resultados y los presenta en forma gráfica o tabulada. Regularmente se usan colores para localizar zonas de valores de esfuerzo o temperatura semejantes. El modelo puede ser rotado para ser examinado desde diferentes puntos de vista.

1.7.4 Análisis de los resultados:

Es el paso final y uno de los más importantes del análisis por el método de los elementos finitos. Aún cuando los programas comerciales son excelentes en cuanto a la representación de los resultados en forma gráfica y facilitan la interpretación de los mismos haciendo uso de colores para diferenciar las zonas de grandes gradientes, sólo queda de parte del analista estructural interpretar correctamente dicha información para la toma de decisiones.

La existencia de un gran número de programas comerciales de elementos finitos, con los cuales puede contar cualquier analista estructural, le dan a estos profesionales la oportunidad de obtener información muy detallada sobre el comportamiento de los modelos que diseña. Sin embargo, la existencia de esta posibilidad incrementa, en vez de reducir, la necesidad de un juicio firme en ingeniería sobre la parte de uso del programa. La información resultante de los paquetes computacionales nunca puede sustituir el entendimiento y al sentido común.

1.8 Conceptos básicos de Elementos Finitos en tres dimensiones

Los elementos finitos para el análisis tridimensional se obtienen principalmente utilizando el principio de la Mínima Energía Potencial de una estructura cargada, definido como la diferencia entre la energía de deformación almacenada en la estructura (U) y el trabajo realizado por las cargas externas (V), es decir:

$$\Pi = U - V \quad (1.1)$$

Varios autores han demostrado que la Energía Potencial total en problemas de elasticidad, no sólo es estacionaria, sino también mínima. De esta manera, el método de los elementos finitos busca dicho mínimo con la condición de que satisfaga una determinada configuración de desplazamientos. Mientras mayor sea el número de grados de libertad más se acercará la solución a la exacta, la cual asegura el equilibrio completo siempre y cuando los desplazamiento tiendan, en el límite, hacia los desplazamientos verdaderos. Es importante observar que, si bien el equilibrio verdadero requiere la minimización completa de la energía potencial total, Π , una

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

solución aproximada mediante elementos finitos proporciona siempre una energía aproximada Π mayor que la correcta. Por consiguiente, siempre se obtiene una cota superior en el valor de la energía potencial total. [Zienkiewicz y Taylor, 1994].

1.9.1 Trabajo realizado por las Cargas Externas:

El trabajo (V) realizado por las tres cargas en cada nodo viene dado por la expresión:

$$V = P_{x1}u_1 + P_{y1}v_1 + P_{z1}w_1 + P_{x2}u_2 + P_{y2}v_2 + P_{z2}w_2 + \dots + P_{xn}u_n + P_{yn}v_n + P_{zn}w_n \quad (1.2)$$

que generalmente se escribe en notación matricial como :

$$V = \{\mathbf{u}\}^T \{\mathbf{f}\} \quad (1.3)$$

donde:

$$\{\mathbf{u}\}^T = [u_1 \quad v_1 \quad w_1 \quad u_2 \quad v_2 \quad w_2 \quad \dots \quad u_n \quad v_n \quad w_n] \quad (1.4)$$

$$\{\mathbf{f}\}^T = [P_{x1} \quad P_{y1} \quad P_{z1} \quad P_{x2} \quad P_{y2} \quad P_{z2} \quad \dots \quad P_{xn} \quad P_{yn} \quad P_{zn}] \quad (1.5)$$

Como puede observarse, el elemento finito tridimensional rectangular básico tendrá 8 nodos y por tanto 24 grados de libertad.

1.9.2 Energía de Deformación almacenada en el cuerpo cargado elásticamente:

La energía de deformación almacenada por el cuerpo tridimensional cargado será:

$$U = \int_{Vol} \frac{\{\sigma\}^T \{\epsilon\}}{2} dVol \quad (1.6)$$

donde $\{\sigma\}$ es la matriz de esfuerzos y $\{\epsilon\}$ la matriz de deformación que vienen dadas por:

$$\{\sigma\}^T = [\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{zx}] \quad (1.7)$$

$$\{\epsilon\} = \{\epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{zx}\}^T \quad (1.8)$$

y están relacionadas entre si por:

$$\{\sigma\} = [K] \{\epsilon\} \quad (1.9)$$

1.9.3 Relación de Esfuerzo-Deformación para Problemas Tridimensionales:

En el caso más general, podemos expresar las relaciones esfuerzo-deformación en notación matricial de la siguiente manera: [Lee y Lo, 1997]

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \quad (1.10)$$

donde E es el Módulo de Young y ν el Coeficiente de Poisson.

De manera mas concisa se puede expresar la relación esfuerzo-deformación como:

$$\{\sigma\} = [D]\{\epsilon\} \quad (1.11)$$

donde $[D]$ representa la matriz de rigidez elástica que depende del comportamiento del material y relaciona los componentes de esfuerzo con los componentes de deformación. [Lee y Lo, 1997]

La energía de deformación almacenada en el cuerpo cargado será entonces:[Zienkiewicz y Taylor,1994]

$$U = \int_{Vol} \frac{\{\epsilon\}^T [D] \{\epsilon\}}{2} dVol \quad (1.12)$$

En un análisis tridimensional completo, el vector deformación en un punto tiene seis componentes; si definimos las componentes del desplazamiento de un punto de un material arbitrario en las direcciones x , y , y z respectivamente se tendría:

$$\mathbf{u} = \mathbf{u}(x,y,z); \quad \mathbf{v} = \mathbf{v}(x,y,z); \quad \mathbf{w} = \mathbf{w}(x,y,z) \quad (1.13)$$

Si las deformaciones y las rotaciones son pequeñas, los gradientes de las deformaciones y los desplazamientos en coordenadas cartesianas son relacionados por las siguientes ecuaciones:[Cook,1995]

$$\begin{aligned}
 \varepsilon_x &= \frac{\partial u}{\partial x} & \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\
 \varepsilon_y &= \frac{\partial v}{\partial y} & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\
 \varepsilon_z &= \frac{\partial w}{\partial z} & \gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}
 \end{aligned}
 \tag{1.14}$$

1.9.4 Campo de Desplazamiento asumido:

Las relaciones esfuerzo-deformación demostradas en la sección 1.9.3 se pueden expresar para el nodo i como:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} \{u\}_i
 \tag{1.15}$$

$$\{\varepsilon\}_{6 \times 1} = [B]_{6 \times 3n} \{u\}_{3n \times 1}
 \tag{1.16}$$

donde $[B]$ es la matriz que relaciona las deformaciones con los desplazamientos

La energía de deformación almacenada por el cuerpo cargado puede ser expresada en notación matricial de acuerdo a las ecuaciones (1.6) y (1.12) como:

$$U = \int_{Vol} \frac{\{\sigma\} \{\varepsilon\}}{2} dVol = \int_{Vol} \frac{\{\varepsilon\}^T [D] \{\varepsilon\}}{2} dVol
 \tag{1.17}$$

y de igual manera, la energía de deformación al escribirla en función de los desplazamientos $\{u\}$ y en base a la ecuación (1.16) vendría dada por:

$$U = \frac{1}{2} \int_{Vol} (\{u\}^T [B]^T [D][B] \{u\}) dVol \quad (1.18)$$

$$U = \frac{1}{2} \{u\}^T \left\{ \int_{Vol} ([B]^T [D][B]) dVol \right\} \{u\} = \frac{1}{2} \{u\}^T [K] \{u\} \quad (1.19)$$

donde la integral es sobre x, y, z .

En la presente investigación la expresión de la energía de deformación es de gran importancia ya que es la base fundamental del indicador de refinamiento utilizado, como también para la obtención de la matriz de rigidez del elemento. En capítulo posteriores será explicado este concepto.

El campo de desplazamiento u, v, w para cualquier posición x, y, z en un elemento de n nodos puede ser definido en función de:

- valores nodales u_i, v_i y w_i
- funciones de interpolación N_i que son funciones de x, y, z

$$\begin{aligned} u &= N_1 u_1 + N_2 u_2 + \dots + N_n u_n & u &= [N] \{u\} \\ v &= N_1 v_1 + N_2 v_2 + \dots + N_n v_n & v &= [N] \{v\} \end{aligned} \quad (1.20)$$

$$w = N_1 w_1 + N_2 w_2 + \dots + N_n w_n \quad w = [N] \{w\}$$

Las funciones u, v y w pueden ser utilizadas para obtener las deformaciones:

$$\begin{aligned} \epsilon_x &= \frac{\partial u}{\partial x} = \sum_{i=1}^n \frac{\partial N_i}{\partial x} u_i & \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \sum_{i=1}^n \left(\frac{\partial N_i}{\partial y} u_i + \frac{\partial N_i}{\partial x} v_i \right) \\ \epsilon_y &= \frac{\partial v}{\partial y} = \sum_{i=1}^n \frac{\partial N_i}{\partial y} v_i & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = \sum_{i=1}^n \left(\frac{\partial N_i}{\partial z} v_i + \frac{\partial N_i}{\partial y} w_i \right) \\ \epsilon_z &= \frac{\partial w}{\partial z} = \sum_{i=1}^n \frac{\partial N_i}{\partial z} w_i & \gamma_{xy} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \sum_{i=1}^n \left(\frac{\partial N_i}{\partial x} w_i + \frac{\partial N_i}{\partial z} u_i \right) \end{aligned} \quad (1.21)$$

Como se puede observar solo las funciones N_i dependen de la posición x, y, z .

1.9.5 Elemento Hexaédrico Básico para Análisis Tridimensional:

El elemento finito 3D más simple que se muestra en la figura 1.1 tiene nodos en sus ocho esquinas y grados de libertad u , v y w en cada nodo, por tanto posee veinticuatro (24) grados de libertad.

Las Funciones deben ser lineales en cada arista del elemento ya que sólo tiene dos nodos por arista y son de la forma :

$$\begin{aligned} N_1 &= \frac{1}{8} \left[\left(1 - \frac{x}{a}\right) \left(1 - \frac{y}{b}\right) \left(1 - \frac{z}{c}\right) \right] \\ N_2 &= \frac{1}{8} \left[\left(1 + \frac{x}{a}\right) \left(1 - \frac{y}{b}\right) \left(1 - \frac{z}{c}\right) \right] \\ &\vdots \\ N_8 &= \frac{1}{8} \left[\left(1 - \frac{x}{a}\right) \left(1 + \frac{y}{b}\right) \left(1 + \frac{z}{c}\right) \right] \end{aligned} \quad (1.22)$$

La matriz de rigideces puede ahora ser evaluada

$$[K] = \int \int \int_{x \ y \ z} ([B]^T [D] [B]) \, dz \, dy \, dx \quad (1.23)$$

Donde $[B]$ es la matriz que relaciona las deformaciones con los desplazamientos y $[D]$ representa la matriz de rigidez elástica que depende del comportamiento del material.

1.10 Principio de los Trabajos Virtuales

El principio de los trabajos virtuales permite el desarrollo de métodos numéricos de formulación equivalente a las ecuaciones de equilibrio, permitiendo el análisis de estructuras reales que no pueden ser calculadas analíticamente. El teorema de los trabajos virtuales viene dado por:

$$W_{\text{int}} - W_{\text{ext}} = 0 \quad (1.24)$$

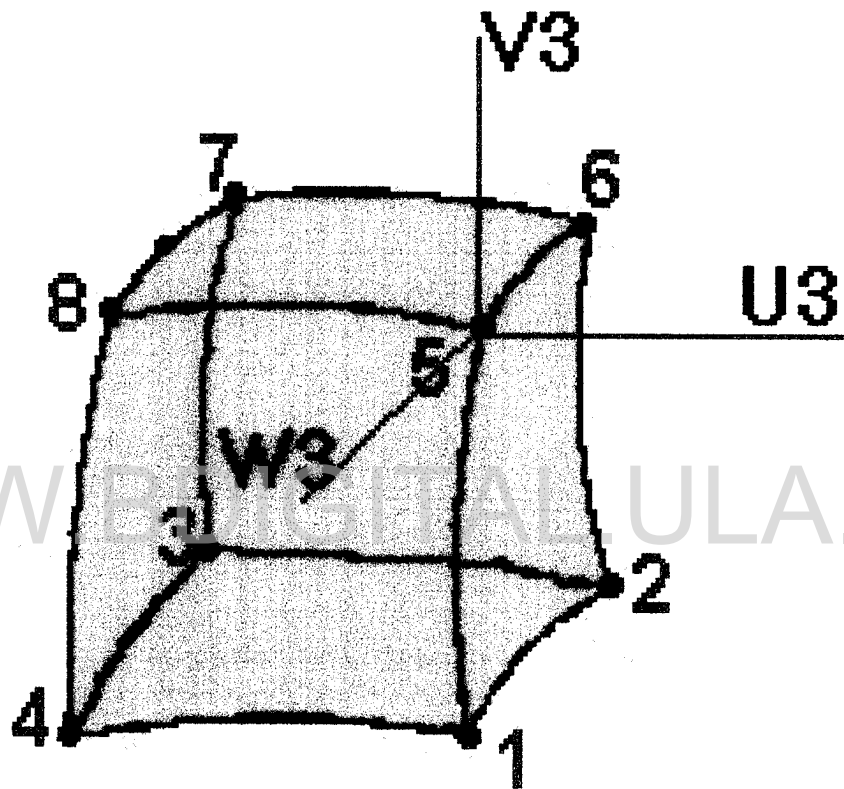


Figura 1.1 Elemento Hexaédrico básico de 8 nodos

donde el trabajo virtual interno es:

$$W_{\text{int}} = \iiint_{\text{vol}} \{\varepsilon^*\}^T \{\sigma\} \, d\text{vol} \quad (1.25)$$

y el trabajo virtual externo es:

$$W_{\text{ext}} = \iiint_{\text{VOL}} \{\mathbf{u}^*\}^T \{\mathbf{f}^d\} \, d\text{vol} - \iint_{\partial\sigma^v} \{\mathbf{u}^*\}^T \{\mathbf{F}^d\} \, ds \quad (1.26)$$

Cuando el desplazamiento considerado es continuo y compatible con las condiciones de borde (admisibles), y los esfuerzos son calculados a partir de las deformaciones usando las relaciones apropiadas, la ecuación (1.24) contiene los requerimientos constitutivos y de compatibilidad necesarios para el análisis de los problemas de mecánica estructural.

1.11 El Método de los Desplazamientos como Minimización de la Energía Potencial Total

Zienkiewicz (1994), - refieren Buenaño y Febres (1995) en su investigación - plantea que al hacer uso de los teoremas energéticos y considerando las cantidades virtuales como variaciones (o diferenciales) de las cantidades reales; el principio de los trabajos virtuales puede replantearse, haciendo uso del teorema de la energía potencial total como:

$$\Pi = U + W \quad (1.27)$$

y ser sustituido por el principio de la energía potencial estacionaria:

$$\delta(\Pi) = \delta(U + W) = 0 \quad (1.28)$$

donde U es la energía de deformación interna y W energía potencial de las fuerzas externas. La ecuación anterior significa que para asegurar el equilibrio, la energía potencial total (Π) debe ser estacionaria para las variaciones de los desplazamientos admisible. En problemas de elasticidad, la energía potencial total no solo es estacionaria, sino también mínima. De esta manera, el método de los elementos finitos busca dicho mínimo con la condición de que se satisfaga una determinada configuración de desplazamiento.

Mientras mayor sea el número de grados de libertad, la solución se acercará más rápidamente a la solución exacta, la cual asegura el equilibrio total siempre y cuando los desplazamientos tiendan en el límite a los desplazamientos verdaderos.

Se puede resaltar que, una solución aproximada mediante elementos finitos proporciona siempre una energía potencial total aproximada mayor que la correcta, a sabiendas de que el equilibrio verdadero requiere la minimización completa de Π . Por lo tanto, siempre se obtiene una cota superior en el valor de la energía potencial total.

En este concepto se fundamenta el proceso de refinamiento que se propone en esta investigación. Es decir, a medida que incrementamos el número de elementos (refinamiento tipo h) los valores de las energías potenciales y de deformación tienden a sus valores exactos. Esto debido a que, la energía potencial total correspondiente a los desplazamientos calculados u es un límite superior a la energía potencial exacta del problema y la energía de deformación es un límite inferior a la energía de deformación total exacta.

En el análisis de sistemas continuos y frecuentemente en análisis elásticos puede ser utilizando el método de Rayleigh-Ritz, que consiste en la minimización de la energía potencial total con respecto a una expresión previamente formulada de la energía potencial y de la configuración de desplazamientos en función de un conjunto finito de parámetros, para obtener un serie de ecuaciones simultáneas. Estos desplazamientos se especifican mediante expresiones válidas en toda la región, obteniéndose por consiguiente ecuaciones simultaneas sin forma de banda, ya que la matriz está llena de coeficientes.

El método de los elementos finitos es semejante al método de Rayleigh-Ritz, y sólo se diferencian en la manera de imponer los desplazamientos. Como se indicó, el

método de Rayleigh-Ritz los especifica mediante expresiones válidas en toda la región, mientras que el método de los elementos finitos la especificación de la imposición de dichos desplazamientos se hace parceladamente. Esto trae como consecuencia que para el método de los elementos finitos la matriz que se obtiene posee una gran cantidad de términos nulos, por lo tanto una matriz en banda.

De este modo nace el método de los elementos finitos, el cual toma como punto de partida la discretización de la región del problema por medio de una partición geométrica denominada malla, a cada elemento j de esta malla se asocia un espacio funcional Π_j y se establece que están conectados entre si por un numero discreto de nodos, donde los desplazamientos \mathbf{d}_i de estos son las incógnitas fundamentales del problema.

Definimos entonces un campo de desplazamiento admisible, medido en un sistema local, de tal forma que el desplazamiento dentro de cualquier elemento sea interpolado de los \mathbf{d}_i de dicho elemento,

$$\{\mathbf{u}\}_j = [\mathbf{N}]_j \{\mathbf{d}\}_j \quad (1.29)$$

donde $[\mathbf{N}]_j$ es la matriz de funciones de forma o interpolación del elemento, $\{\mathbf{d}\}_j$ son los grados de libertad del elemento. La energía potencial total de la estructura será igual a la contribución de los m elementos finitos,

$$\Pi = \sum_{j=1}^m \Pi_j = \sum_{j=1}^m U_j + \sum_{j=1}^m W_j \quad \text{para } j = 1, 2, \dots, m. \quad (1.30)$$

Logrando de esta manera reducir el problema del continuo, donde la solución es una función \mathbf{u} que hace estacionario a Π con respecto a variaciones pequeñas de $\delta\mathbf{u}$, al caso discreto de elementos finitos donde para cualquier variación pequeña $\delta\mathbf{d}_i$ se obtiene un sistema de ecuaciones de donde podemos determinar los parámetros \mathbf{d}_i

que minimizan el funcional Π y que al mismo tiempo representan la configuración de equilibrio.

$$\delta \Pi = \frac{\partial \Pi}{\partial \mathbf{d}_i} \delta \mathbf{d}_i \quad \Rightarrow \quad \frac{\partial \Pi}{\partial \{\mathbf{u}\}} \equiv \frac{\partial \Pi}{\partial \mathbf{d}_i} = \mathbf{0} \quad \text{para } i = 1, 2, \dots, n \quad (1.31)$$

donde n es el número de grados de libertad de la estructura y $\{\mathbf{u}\}$ es el vector de desplazamientos nodales globales. Para calcular la matriz de rigidez local $\{\mathbf{k}\}_j$ y el vector de fuerzas nodales $\{\mathbf{r}\}_j$ de un elemento cualquiera y al mismo tiempo referirnos al proceso de ensamblaje de las matrices locales de los elementos, en la matriz global de la estructura, la cual es automáticamente realizada como en el método de rigidez directa,

$$\Pi = \sum_1^m \frac{1}{2} \{\mathbf{u}\}_j^T [\mathbf{k}]_j \{\mathbf{u}\}_j + \sum_1^m \{\mathbf{u}\}_j^T \{\mathbf{r}\}_j \quad (1.32)$$

$$\Pi = \frac{1}{2} \{\mathbf{u}\}^T [\mathbf{K}] \{\mathbf{u}\} - \{\mathbf{u}\}^T \{\mathbf{R}\} \quad (1.33)$$

donde $\{\mathbf{K}\}$ y $\{\mathbf{R}\}$ son la matriz de rigidez y el vector de fuerzas nodales globales respectivamente. Valiéndonos de esta analogía podemos generar a partir de (1.32), en una manera casi mecánica la bien conocida ecuación de equilibrio del sistema, al minimizar el potencial

La Energía Potencial Total sería:

$$\Pi = \mathbf{U} - \mathbf{V} = \frac{1}{2} \{\mathbf{u}\}^T \left\{ \int_{\text{Vol}} ([\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}]) d\text{Vol} \right\} \{\mathbf{u}\} - \{\mathbf{u}\}^T \{\mathbf{r}\} \quad (1.34)$$

$$\frac{\partial \Pi}{\partial \{\mathbf{u}\}} = \left[\int \int \int ([\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}]) dz dy dx \right] \{\mathbf{u}\} - \{\mathbf{r}\} = \{\mathbf{0}\} \quad (1.35)$$

El término entre paréntesis es conocido como Matriz de Rigideces, es decir:

$$[\mathbf{K}] \{\mathbf{u}\} = \{\mathbf{f}\} \quad (1.36)$$

$$\frac{\partial \Pi}{\partial \{\mathbf{u}\}} = [\mathbf{K}] \{\mathbf{u}\} - \{\mathbf{R}\} = \mathbf{0} \quad (1.37)$$

Las deformaciones y los esfuerzos en cada elemento vendrían dados al sustituir el campo continuo $\{\mathbf{u}\}_j$ por su forma discreta

$$\{\boldsymbol{\varepsilon}\}_j = [\mathbf{B}]_j \{\mathbf{u}\}_j \quad (1.38)$$

$$\{\boldsymbol{\sigma}\}_j = [\mathbf{E}] \{\boldsymbol{\varepsilon}\}_j \Rightarrow \{\boldsymbol{\sigma}\}_j = [\mathbf{E}] [\mathbf{B}]_j \{\mathbf{u}\}_j \quad (1.39)$$

donde $[\mathbf{S}]$ es la matriz operadora para la deformación y $[\mathbf{B}]_j$ es una matriz que relaciona deformación-desplazamiento y viene dada por la apropiada diferenciación y combinación de la matriz $[\mathbf{N}]_j$,

$$[\mathbf{B}]_j = [\mathbf{S}] [[\mathbf{N}]_j] \quad (1.40)$$

A partir del procedimiento anterior se puede desarrollar una metodología casi automática para la creación de elementos finitos. Una vez conocida para la teoría dada las expresiones de energía de deformación interna, a partir de la cual puede ser obtenida la matriz de rigidez de un elemento $\{\mathbf{k}\}_j$,

$$U_j = \frac{1}{2} \iiint_{\text{vol}} \{\varepsilon^*\}^T \{\sigma\} \, d\text{vol} = \frac{1}{2} \iiint_{\text{vol}} \{\mathbf{u}\}_j^T [\mathbf{B}]_j^T [\mathbf{E}] [\mathbf{B}]_j \{\mathbf{u}\}_j \, d\text{vol} \quad (1.41)$$

$$U_j = \frac{1}{2} \{\mathbf{u}\}_j^T [\mathbf{k}]_j \{\mathbf{u}\}_j = \frac{1}{2} \{\mathbf{u}\}_j^T \left(\iiint_{\text{vol}} [\mathbf{B}]_j^T [\mathbf{E}] [\mathbf{B}]_j \, d\text{vol} \right) \{\mathbf{u}\}_j \quad (1.42)$$

y conocida la energía potencial de las fuerzas externas, puede obtener el vector fuerzas nodales $\{\mathbf{r}\}_j$,

$$W_j = \iiint_{\text{vol}} \{\mathbf{u}\}_j^T \{\mathbf{f}^d\} \, d\text{vol} - \iint_{\partial_{\sigma}V} \{\mathbf{u}\}_j^T \{\mathbf{F}^d\} \, ds \quad (1.43)$$

$$W_j = \{\mathbf{u}\}_j^T \{\mathbf{r}\}_j = \{\mathbf{u}\}_j^T \left(\iiint_{\text{vol}} [\mathbf{N}]_j^T \{\mathbf{f}^d\} \, d\text{vol} - \iint_{\partial_{\sigma}V} [\mathbf{N}]_j^T \{\mathbf{F}^d\} \, ds \right) \quad (1.44)$$

WWW.BDIGITAL.ULA.VE

CAPITULO 2

METODO DE LOS ELEMENTOS FINITOS ADAPTATIVOS

2.1 Introducción

El método de los Elementos Finitos Adaptativos (M.E.F.A.) parte de una malla somera diseñada sobre la base de criterios puramente geométricos, para posteriormente y con el fin de mejorar la calidad de la solución obtenida, se van realizando modificaciones de dicha malla inicial, hasta obtener soluciones de una exactitud predeterminada. Es bien sabido por quienes trabajan con el método de los elementos finitos, que no es posible diseñar una malla perfecta en un primer paso, especialmente si se trata de modelos complejos, lo que implica una experiencia del usuario.[S. Lakshmanam y otros, 1997]

El método de los elementos finitos adaptable puede ser definido como un proceso automático para el análisis por elementos finitos que permiten al usuario alcanzar predeterminados patrones de exactitud a partir de sucesivas discretizaciones de un mallado inicial. La escogencia adecuada de la malla, especialmente el tamaño y tipo de los elementos finitos en cada parte de ella, determina la calidad de la solución numérica y el trabajo computacional involucrado en el proceso.[A. Umar, et al 1996].

En este sentido una mejora sustancial consiste en obtener una malla en la que los elementos posean una buena relación de aspecto, y por tanto una adecuada regularidad. Los métodos de Generación Laplaciana permiten mejorar la regularidad de la malla, si bien con ellos resulta difícil controlar una posible variación de la densidad del mallado en distintas zonas del modelo. [Hormaza y otros, 1995].

El desarrollo de procesos adaptativos ha recibido, en la última década, especial atención tanto por parte de los investigadores, así como de las firmas comerciales que desarrollan software. Un gran número de investigadores en este campo dedican esfuerzos considerables para lograr el desarrollo de estrategias adaptativas destinadas a lograr soluciones numéricas a los problemas de ingeniería de

una manera eficiente, confiable y económica, a un costo computacional bajo y que además requiera de la menor intervención posible del usuario.

El objetivo del diseño de cualquier proceso adaptativo es obtener automáticamente la malla óptima, en el sentido de que el error pueda ser minimizado para un número de grados de libertad y de esta manera alcanzar económicamente la relación máxima de convergencia de las aproximaciones de elementos finitos.[Choi y Lee, 1996].

De acuerdo a Lee y Lo,(1992), - refieren Buenaños y Febres,(1995) en su investigación - un proceso adaptativo consta de los siguientes módulos principales los cuales se encuentran relacionados unos con otros:

- Un módulo que realiza el análisis de elementos finitos.
- Un módulo indicador de refinamiento y estimación del error.
- Un módulo para la generación de mallas adaptativas de elementos.

Por otra parte para Selman, Hinton y Bicanic (1997), el procedimiento de refinamiento adaptativo de mallas requiere de tres módulos:

- Un módulo para la indicación del refinamiento local
- Una técnica para convertir la información de refinamiento en una densidad de malla deseada y
- Un generador de malla automático que facilite la producción de mallas con una densidad establecida.

La importancia del procedimiento de refinamiento de mallas adaptativo en aplicaciones industriales tiene ventajas para el incremento de investigaciones sobre generadores de malla completos, los cuales requieren sólo de la especificación de las condiciones de apoyo y distribución del tamaño de la malla sobre el dominio bajo consideración. Los próximos procedimientos de refinamiento de mallas adaptativos dependerán de la gran atención que se le preste a la completa eficiencia entre el análisis adaptativo y el generador automático de la malla.

En este capítulo se describirán brevemente las diferentes estrategia de refinamiento adaptativo que se encuentran en la literatura y se efectuará un análisis comparativo entre ellas. Así mismo se describirán una de las posibles fuentes de error

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

en la escogencia de la malla original y del tipo de elemento en un diseño cualquiera, al igual que de los efectos de bloqueo "Locking" que se producen debido a la distorsión del elemento seleccionado.

2.4 Estrategias de Refinamiento Adaptativo

Recientemente han sido desarrolladas técnicas de refinamiento adaptativo automatizando del proceso de refinamiento de la malla de elementos finitos donde sea más necesario y en problemas transitorios, permitiendo que el refinamiento de la misma ocurra en regiones diferentes.[D. Morgan, et al, 1995].

En la literatura existen una serie de estrategias utilizadas para generar refinamiento de mallas de elementos finitos, las cuáles serán referidas muy brevemente.

2.2.1 Refinamiento tipo R

Esta estrategia implica la redistribución de los nodos y consiste en encontrar la localización optima de los mismos, manteniendo fijo el número de elementos y grados de libertad de la estructura bajo consideración. En esta aproximación las coordenadas nodales, \mathbf{C} , son incluidas como incógnitas en el apropiado funcional de energía potencial.

$$\Pi(\mathbf{u}, \mathbf{c}) \Rightarrow \begin{cases} \frac{\partial \Pi}{\partial \mathbf{u}} = 0 \\ \frac{\partial \Pi}{\partial \mathbf{c}} = 0 \end{cases} \quad (2.1)$$

Varios autores encontraron inconveniente este proceso, que intenta obtener la malla optima para el análisis, ya que resulta en ecuaciones no-lineales, inclusive para problemas lineales, por lo cual las técnicas matemáticas resultan en un esfuerzo computacional alto[Buenaño y Febres, 1995]. Para otros investigadores, esta estrategia tiene la ventaja de no incrementar el tiempo de computación puesto que se preserva el mismo número de puntos nodales a lo largo del proceso.[Lakshmanam y otros, 1997].

2.2.2 Refinamiento tipo H:

Esta estrategia consiste en el refinamiento local de la malla, incrementando de esta forma el número de elementos de la misma. Con esta estrategia se reduce el tamaño h del elemento manteniendo el grado del polinomio de la función de interpolación constante durante los sucesivos refinamientos de la malla, por lo que puede realizarse usando un sólo tipo de elemento. De todas las estrategias esta es la vía más natural para el ingeniero.

Zienkiewicz. (1992), plantea que el refinamiento local de la malla puede ser llevado a cabo al subdividir los elementos que así lo requieran o al aumentar el número de elementos mediante un remallado total del dominio con incremento de la densidad en zonas determinadas.[Buenaño y Febres,1995].

La estrategia de refinamiento tipo h es preferida en el caso de singularidades en el dominio y tiene como desventaja que la restricción de la malla limita las características locales del refinamiento. Siempre que un elemento restringido es subdividido, los elementos vecinos tendrán que ser bisectados primeros. Esto causa una secuencia de subdivisiones a los elementos vecinos, además, que requiere frecuentes cambios de los nodos restringidos en nodos activos.[S. Lakshmanam y otros, 1997].

2.2.3 Refinamiento tipo P:

Esta estrategia mantiene el número de elementos fijos, incrementando el grado del polinomio p , o lo que es lo mismo, ajusta la malla de elementos finitos e incrementa el orden de interpolación de los elementos. La convergencia en este tipo de refinamiento es exponencial si la solución exacta es suavizada.[S. Lakshmanam y otros, 1997]. La malla es refinada sumando términos de orden superior, usualmente polinomios, a las funciones de forma básicas del elemento finito. El uso de elementos jerárquicos ha demostrado ser el mejor enfoque en el P-refinamiento, y su campo de desplazamiento viene dado por:

$$\{\mathbf{u}\}_j = [\mathbf{N}]_j \{\mathbf{d}\}_j + [\mathbf{N}]_k \{\mathbf{a}\}_k \quad (2.2)$$

donde $\{a\}_k$ representa los desplazamientos jerárquicos y $[N]_k$ la función de forma jerárquica, la cual satisface la propiedad de ser nula en los nodos. De esta manera los grados de libertad jerárquicos pueden ser agregados al elemento sin aumentar el número de nodos o cambiar las funciones de forma, o los elementos de la matriz de rigidez, o el vector de fuerzas existentes. [Buenaño y Febres, 1996].

2.2.4 Refinamiento tipo H-P:

Esta estrategia selecciona simultáneamente la malla y el grado del polinomio haciendo uso apropiado del refinamiento h- y p-. Generalmente el mejoramiento en la economía computacional y en la convergencia puede ser obtenido usando un h-refinamiento hasta alcanzar una exactitud dada, seguido de un incremento de p sobre todo el dominio. Esta versión h-p converge exponencialmente a la solución exacta aún en la presencia de singularidades y tiene tanto los beneficios de las estrategia h y p como las limitaciones de estos. [Lakshmanam y otros, 1997].

2.2.5 Refinamiento tipo D:

Esta estrategia consiste en la adaptabilidad dimensional d, acoplada con refinamientos h- y p-. Es necesario generar saltos dimensionales con la idea de garantizar una solución global adecuada. Stein y Ohnimus (1992), indican que este refinamiento es usado por ejemplo, cuando las singularidades tienen orígenes y soluciones 3d y no pueden ser representadas por modelos 2d. [Buenaño y Febres, 1995].

2.2.6 Refinamiento tipo (H-Method):

Esta estrategia es una modificación a la técnica de refinamiento tipo h, debido a que al igual que ésta se reduce el tamaño h del elemento manteniendo el grado del polinomio de la función de interpolación constante durante los sucesivos refinamientos de la malla, pero con la condicionante de que se toma en cuenta las componentes direccionales de discretización del error. Por ejemplo las contribuciones del error del sistema de coordenadas locales en las direcciones ξ, η, ζ para un elemento, serían con cantidades diferentes. [Oh y Lim, (1997)]

2.3 Análisis sobre Estrategias Adaptativas

[Buenaño y Febres,1996], refieren que de todas las estrategias existentes las que mayor atención recibe son la h-, p- y hp-, mientras que las mallas móviles debido a las dificultades que presentan demandan futuros estudios para su implementación práctica. Por otro lado, la estrategia de refinamiento dimensional, -d, es de reciente estudio y se encuentra en pleno desarrollo.

En cuanto a la versión de refinamiento p-, se puede decir que tiene mejor convergencia que la versión de refinamiento h-. De todas maneras, la versión p- es considerada de mayor dificultad para implementar que la h-. En la versión p- es necesaria una gran librería de elementos, pues cuando todas las funciones de forma polinómicas se agoten, no se puede llevar a cabo ningún mejoramiento en la exactitud de los resultados. Mientras que el H-refinamiento esta solo limitado por la estabilidad numérica y la capacidad del hardware para manejar grandes tamaños de información.

La versión p tiene sus propios puntos débiles en el enriquecimiento del orden espectral para cada elemento. El orden de apoyo del nodo común es el máximo fuera de los dos ordenes espectrales p1 y p2 de los elementos vecinos y esto resulta en una malla no óptima para la aproximación deseada de la solución. [Lakshmanam y otros, 1997].

El ancho de banda de la matriz de rigidez en el P-refinamiento tiende a incrementarse drásticamente en cada iteración, lo que implica el aumento del número de operaciones, ya que este es proporcional al número de grados de libertad por el cuadrado del ancho de banda. También el tamaño de la matriz del elemento se incrementa con cada iteración por lo cual el tiempo para su formulación es proporcional al cuadrado del tamaño de la matriz del elemento. En la estrategia h las matrices de rigidez de los elementos que no hallan sido divididos en el nuevo refinamiento no necesitan ser calculadas de nuevo.[Buenaño y Febres,1996].

Ya se dijo que una de las desventajas del criterio de refinamiento tipo h es que si un elemento restringido es subdividido, los elementos vecinos tendrán que ser bisectados primero. Este tipo de inconveniente de obtener gradaciones de mallas

abruptas, varios autores [Choi y Lee, 1996] lo evitan, con el establecimiento de algoritmos adaptativos cinemáticos, como es la implementación de una regla de nivel simple "*single-level rule*". Con la aplicación de este tipo de regla se plantea que un elemento simple no puede ser refinado una vez más sin el refinamiento acoplado de los elementos vecinos.

Al usar una estrategia combinada como la hp- se aprovecha las ventajas de los dos procedimientos, ya que la versión p- requiere de una malla inicial buena, la cual es obtenida con un refinamiento previo h-, para después con una librería limitada de elementos acelerar la convergencia y mejorar la exactitud.

Obviamente es importante que el error de esta aproximación sea conocido en orden de aplicar eficientemente el método de los elementos finitos a los problemas de la ingeniería. La estimación del error en la solución de elementos es actualmente un tema abierto, al cual los investigadores dedican un considerable esfuerzo.

En esta investigación el tipo de estrategia aplicada es la del refinamiento tipo h debido a su independencia de la librería de los elementos disponibles, lo que permite la implementación de procesos adaptativos en códigos con pocos elementos en dicha librería.

2.4 Fuentes de Error

En el capítulo 1 de esta investigación se definió el método de los elementos finitos como una técnica numérica para resolver problemas complejos que aparecen en las ciencias aplicadas, obteniéndose soluciones aproximadas a problemas físicos definidos matemáticamente. Además se indicó que el método de los elementos finitos se basa en la discretización del continuo en un número finito de elementos, de forma tal que con el uso de diferentes tipos de elementos y diferentes subdivisiones, se obtienen diferentes grados de aproximación.

Se debe recordar por lo tanto, - acota J. García (1994), que las soluciones numéricas son esencialmente aproximaciones plausibles y consecuentemente deben ir acompañadas de estimadores de error. Además en ingeniería de Proyectos hay que tratar habitualmente con sucesiones de soluciones y en la mayoría de los casos es necesario utilizar métodos numéricos, basados en algoritmos iterativos, que en su caso convergen hacia soluciones exactas.

En este sentido puede decirse que la teoría (analítica) y la experimentación (física), si bien han permitido desde hace siglos definir modelos matemáticos aceptables en muchas aplicaciones (estructuras, Fluidodinámica, etc.) no han sido suficientes todavía para obtener soluciones numéricas satisfactorias. En cambio la computación (numérica) ha permitido hacerlo dentro de las limitaciones de los modelos, las computadoras y también los métodos utilizados en la resolución de problemas diversos.

En este contexto conviene advertir que estos modelos sólo pueden validarse con la experimentación y por el contrario, las computadoras y los métodos introducen unos errores inevitables que pueden y deben verificarse en la propia computación; estos son los debido a :

- Redondeo (aritmético).
- Truncamiento (algebraico)
- Discretización (Funcional).

Buenaño y Febres (1995), plantean que el error de estas aproximaciones debe ser conocido, a fin de aplicar eficientemente el método de los elementos finitos a los problemas de ingeniería y refieren una serie de fuentes de error como lo son:

2.4.1 Error del Modelado Matemático:

El error del modelado matemático puede ser definido como la diferencia entre las cantidades medidas del sistema físico y las cantidades obtenidas del modelo matemático. El análisis, numérico o analítico, es llevado a cabo en el modelo matemático, el cual esta lleno de suposiciones hechas en orden de simplificarlo, tales como: la definición de la geometría del dominio de la solución, las incógnitas envueltas en la solución, leyes de comportamiento, compatibilidad, modelado de las condiciones de apoyo y limites de la magnitud de la respuesta entre otros. Este error es principalmente inherente a la ecuación diferencial que gobierna el modelo y no esta relacionado al error producido por la aproximación en elementos finitos.

2.4.2 Error Computacional:

Se incurre en este error a causa de las limitaciones de una computadora en representar, almacenar y manipular números de precisión infinita. Podemos definir el error de 'redondeo' el cual es causado por el uso de un numero finito de 'bit' o

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

dígitos para representar números reales. Teniendo en cuenta que el último dígito retenido es siempre *truncado o aproximado*. Se incurre también en errores de *manipulación* introducidos por un algoritmo, dependiendo de su precisión numérica y del tipo de operaciones que este realiza. En cualquier momento del cálculo se puede obtener un error *heredado*, que sería la suma de los error para ese momento.

2.4.3 Error debido a Singularidades:

Este error es debido a razones geométricos (tales como esquinas entrantes), cambios súbitos en las condiciones de apoyo, cambios repentinos en las cargas (fuerzas concentradas) y cambios bruscos en la propiedades de los elementos. La aproximación de elementos finitos es susceptible a este tipo de error, debido a que en estas zonas la malla debe estar adecuadamente refinada para representar la singularidad.

2.4.4 Error de la aproximación del Funcional:

La solución aproximada de elementos finitos requiere que el campo de desplazamiento sea rigurosamente continuo en un orden menor de la derivada de mayor orden de la expresión del funcional, C^{m-1} , mientras que para satisfacer la ecuación diferencial es necesaria la continuidad de las derivadas hasta un orden $2m-1$. Esto es debido a que las condiciones de borde esenciales involucran derivadas de orden cero a $m-1$, y las no esenciales derivadas de orden m hasta $2m-1$. Esta diferencia de los requerimientos de continuidad entre la aproximación de elementos finitos y la ecuación diferencial gobernante resulta en un error de la solución, la cual es exhibida por la discontinuidad en las condiciones no esenciales. Como se menciono antes, la aproximación del campo de desplazamiento tiende hacia la solución exacta mientras el tamaño de los elementos decrece. El campo de desplazamiento debe ser escogido, también, de tal forma que represente los modos de deformación físicamente posibles.

2.4.5 Error en la Geometría:

Este error proviene de las discrepancias entre el dominio del problema físico y el representado por el sistema discreto de elementos finitos. Este sucede generalmente cuando las fronteras del dominio están formados por curvas o superficies mas complejas que la de los elementos. Este error puede ser combatido por elementos con

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

fronteras curvas o mediante la refinación de los elementos en la frontera. La solución de elementos finitos puede contener serios errores si condiciones de compatibilidad entre la frontera de los elementos no son satisfechas, por lo tanto gaps y overlaps no son admisibles.

2.4.6 Error de Discretización:

El error de Discretización se refiere a la diferencia entre la solución exacta del modelo y la solución numérica obtenida para una determinada malla de elementos finitos,

$$\mathbf{e}_u = \mathbf{u} - \hat{\mathbf{u}} \quad (2.3)$$

es claro que el presente error nunca podrá ser calculado a menos que la solución exacta sea conocida. De todas maneras, cuando \mathbf{u} es desconocido, es posible desarrollar estimadores de error. Es de notar que los errores provenientes de la geometría, funcional y singularidad disminuyen si el error de discretización decrece. Por estas razones el error que interesa conocer y controlar en las soluciones aproximadas es este, tanto a nivel local como global, para atacar de esta manera las áreas donde una mejora de la malla sea necesaria.

2.4.7 Error de Redondeo:

J. García (1994) plantea toda una formulación para la estimación de los errores de redondeo, de truncamiento y discretización. Dentro de estos el error de redondeo, que fue despreciado en esta investigación para efectos de simplicidad, lo clasifica en dos tipos : error básico y el número de condición del problema numérico.

2.4.7.1 Error Básico: Es un error relativo de la computadora digital que puede definirse o estimarse en:

$$\varepsilon = \beta^{-t} \quad (2.4)$$

donde

β : base digital (generalmente 2)

t : precisión mantisa (sin signo)

que dependen de la computadora digital (hardware) y del programa Fortran (software):

t_1 : simple precisión (real)

t_2 : doble precisión (igual o mayor que $2t_1$)

2.4.7.2 Número de condición del problema numérico: que puede definirse o estimarse en :

$$K = e_R / \varepsilon \quad (2.5)$$

donde e_R es el error de redondeo del resultado

La estimación de K puede hacerse de muchas manera, pero a una de ellas es la definición siguiente:

$$e_R = \frac{x_2 - x_1}{x_1} \quad (2.6)$$

donde:

x_1 : resultado de simple precisión

x_2 : resultado de doble precisión

2.4.8 Error por selección incorrecta del Elemento:

C. K. Lee y S. H Lo (1997), advierten sobre la necesidad de selección de un elemento apropiado para el modelado numérico de un problema físico, cuando se utiliza el método de los elementos finitos, debido a que la selección de dicho elemento afecta la seriedad y eficiencia del modelo numérico. Indican, que la selección incorrecta del elemento puede, en muchos casos, resultar en soluciones divergentes del problema. La eficiencia del elemento puede ser valorada por la precisión de la solución y el costo computacional involucrado.

Usualmente, - acotan Lee y Loo - elementos de alto orden son más precisos pero computacionalmente más caros que los elementos de bajo orden. Esto es debido a que elementos de alto orden usualmente requieren norma de integración de alto orden para la formulación de la matriz de rigidez del elemento y tienen más grados de libertad por elemento.

La distorsión del elemento es otro factor importante que afecta la precisión de los resultados como en muchos casos, el dominio del problema ha ser añadido es irregular y la distorsión del elemento es inevitable en la discretización de dicho dominio. En el análisis tridimensional , elementos tipo bloque, son más sensibles a la distorsión de forma que los elementos tetraédricos. Es importante puntualizar que

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

para el análisis de elementos finitos, la efectiva distorsión de la forma de un elemento llegará a ser tan importante, como son frecuentemente necesarias las mallas altamente gradadas para la óptima solución de los problemas.

2.4.9 Error debido a Inestabilidad:

Robert Cook (1995), refiere que un modelo por elementos finitos es usualmente inexacto, y este error lo hace muy rígido. La sobre rigidez se hace crítica por el uso de más puntos de Gauss para integrar las matrices de rigidez del elemento, debido a que puntos adicionales captan más términos de alto orden en \mathbf{k} . Estos términos resisten algunos modos de deformación, lo que no ocurre con términos de bajo orden, debido a esto se rigidiza un elemento. En consecuencia, una gran precisión en la integración de \mathbf{k} generalmente produce baja precisión en la solución de elementos finitos, en adición a más requerimientos computacionales.

En otro orden de ideas, el uso de muy pocos puntos de Gauss produce una situación crítica conocida por varios nombres : inestabilidad, modo singular puro, mecanismo, modo cinemático, modo de cero energía, o modo Hourglass. El termino de modo de cero energía se refiere al vector de desplazamiento nodal $\{\mathbf{u}\}$ que no es un movimiento de cuerpo rígido pero sin embargo produce energía de deformación cero.

Para explicar el término modo de cero energía y mostrar como tal modo puede presentarse sustituimos la relación (1.16)

$$\{\boldsymbol{\varepsilon}\} = \{\mathbf{B}\}\{\mathbf{u}\} \quad (2.7)$$

en la expresión de energía de deformación para un elemento vista en el capítulo 1 (1.17) y (1.18)., es decir:

$$U = \frac{1}{2} \{\mathbf{u}\}^T [\mathbf{K}] \{\mathbf{u}\} = \frac{1}{2} \{\mathbf{u}\}^T \left\{ \int_{Vol} ([\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}]) dVol \right\} \{\mathbf{u}\} = \frac{1}{2} \int_{Vol} \{\boldsymbol{\varepsilon}\}^T [\mathbf{K}] \{\boldsymbol{\varepsilon}\} dVol \quad (2.8)$$

donde $[\mathbf{k}]$ está formada por integración numérica y contiene solo la información que puede ser detectada de los puntos de muestreo de la regla de cuadratura.

Si sucediera que los términos de la expresión (2.4) son cero en todos los puntos de muestreo para cierto modo $\{\mathbf{u}\}$, entonces la energía de deformación pudiera desaparecer para $\{\mathbf{u}\}$, y de acuerdo a la ecuación de rigidez $[\mathbf{k}]$ pudiera ser una matriz de rigidez cero en el sentido que la energía de deformación es cero para

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$\{u\}$. Se espera que la energía de deformación sea igual a cero si $\{u\}$ es un movimiento de cuerpo rígido. Pero si la energía de deformación es igual a cero cuando $\{u\}$ no es un movimiento de cuerpo rígido, entonces se presenta una inestabilidad.

La inestabilidad ocurre (no por pandeo) en uno o más modos de deformación produciendo la visualización de deformaciones cero en todos los puntos de Gauss. Hay que considerar los puntos de Gauss como un detector de esfuerzos. Si no se perciben deformaciones en los puntos de Gauss bajo un cierto modo de deformación, resultaría que $[k]$ pudiera no tener resistencia para esos modos de deformación.

Un elemento que presenta un mecanismo es llamado de rango deficiente. Esto es, el rango de $[k]$ es menor que el número de elementos de grados de libertad menos el número de modos de cuerpo rígido.

2.4.10 Error en el Número de Elementos utilizados al Discretizar:

Umar y otros (1996), plantean por otro lado, que la precisión en los resultados del análisis usando el método de los elementos finitos (FEM) requiere que la estructura deba ser analizada y discretizada usando una malla fina en adición a la correcta representación de la estructura. Sin embargo, la discretización de malla finas requiere una gran memoria computacional tan igual como de tiempo de ejecución.

En su investigación, Umar y otros, hacen un análisis de la conveniencia de dividir el modelo a analizar en un número de elementos adecuado, para lo cual divide el rango de número de elementos en tres zonas.

Zona de representación incorrecta (IR): Cuando el número de elementos es pequeño ($n < n_i$) entonces los resultados de los elementos finitos no son de fiar. Donde n_i es el valor umbral del número de elementos.

Zona de representación adecuada (PR): El incremento en el número de elementos más allá de n_i resulta en una convergencia gradual, siempre que la discretización sea hecha tomando en consideración la variación esperada de los parámetros dentro del cuerpo de la estructura, es decir que ($n_i < n < n_j$). Donde n_j es el número de elementos después del cual comienza la acumulación de error.

Zona donde termina la acumulación del error (AR): El incremento en el número de elementos más allá de n_j resulta en la acumulación del redondeo del error

consecutivamente de los resultados de los elementos finitos llegando a desviarse de los resultados correctos en lugar de obtener convergencia. Esto es obvio ya que el incremento en el número de elementos aumenta los requerimientos de memoria computacional y del tiempo de ejecución .

Si el número mínimo de elementos n requerido para la aceptación del error satisface tanto a la maquina como al usuario (por ejemplo $n < n_t$ y $n > n_m$) entonces los resultados de los elementos finitos son de fiar. Donde n_t es el número máximo de elementos dependientes del tiempo de ejecución y n_m es el número máximo de elementos dependientes de la capacidad de la memoria de la computadora.

Sin embargo, cuando el número de elementos n requeridos para la aceptación del error no satisface a la maquina o al usuario (por ejemplo $n > n_t$ o $n > n_m$) entonces los resultados de los elementos finitos llegan a ser irreales. Por otra parte, el número mínimo de elementos requeridos para la aceptación del error es desconocido. Por lo tanto, es usualmente aconsejable para llevar a cabo el análisis para dos o tres configuraciones de mallas teniendo número diferentes de elementos en cada caso, pero teniendo similares modelos de discretización.

WWW.BDIGITAL.ULA.VE

CAPITULO 3

PROCESO DE REFINAMIENTO

3.1 Introducción

El proceso de refinamiento adaptativo utilizado en el presente trabajo está fundamentado en la versión h, donde se mantiene constante el orden del polinomio, incrementándose el número de los elementos por medio del refinamiento de la malla. Este refinamiento es realizado al subdividir localmente los elementos o por la generación de una malla nueva, en función de ciertos criterios de refinamiento.

3.2 Refinamiento por Indices de Densidad de Energía de Deformación

En este proceso de refinamiento las regiones con grandes valores de energía de deformación son refinadas mediante la subdivisión local de la malla. Dicho refinamiento es llevado a cabo en cada iteración al subdividir ,usualmente bisectando, los elementos en los cuales el indicador de refinamiento excede los límites permisibles.

H. Febres, (1984), utiliza con éxito por primera vez los índices de densidad de energía de deformación (SED) como un indicador de refinamiento en problemas de elasticidad plana y después es Buenaño(1995) quien los aplica en problemas de deflexión en placas. Los índices de densidad de energía de deformación permiten localizar y evaluar los elementos con altas concentraciones de esfuerzos y singularidades. Así mismo en otra investigación H. Febres, (1988), define el índice SED de un elemento i , con la intención de poder encontrar un indicador de la magnitud de la energía de deformación presente en cada elemento de la siguiente manera:

$$SED_i = \frac{U_i}{V_i} \quad (3.1)$$

donde U_i y V_i son la energía de deformación y el volumen del elemento i , respectivamente. Similarmente define el índice de concentración de energía global, (GSED), como:

$$\text{GSED} = \frac{\text{Energía Total de Deformación}}{\text{Volumen Total}} \quad (3.2)$$

Con el objeto de establecer comparaciones entre elementos podemos calcular el indicador ϕ , definido para cada elemento de la malla como:

$$\phi_i = \frac{\text{SED}_i}{\text{GSED}} \quad (3.3)$$

con esta definición los nuevos elementos serán adicionados en aquellas regiones donde se cumpla que:

$$\phi_i \geq \beta \quad (3.4)$$

donde β es un indicador de la magnitud del refinamiento deseado, el cual puede tomar diferentes valores a lo largo de las iteraciones y regiones para un problema dado. Un valor de β igual a 1 indicaría que aquellos elementos con SED_i mayores que el promedio (GSED) serán refinados. Si el valor de β se hace cero todos los elementos serán refinados.

Un punto interesante de resaltar es el criterio de refinamiento en la dirección del gradiente propuesta por C. González.(1993), que es aplicable cuando el problema esta representado por la minimización de un funcional, y el cual esta basado en la variación de un funcional al variar el espacio de aproximación representado por la malla de elementos finitos. La evaluación de la variación del funcional viene dada por funciones u_i y u_{i+1} pertenecientes a sub-espacios diferentes. De este modo, la magnitud de la variación es indicativa de la calidad de la aproximación u_i y mostrara si conviene refinar o no en esa dirección. Este criterio es tratado en conjunto con el índices de densidad de energía de deformación, ya que estos no requieren de estimadores de error, como la mayoría de los criterios adaptativos empleados por los investigadores.

3.3 Distribución uniforme del error

Se describe aquí una metodología de refinamiento suponiendo que se dispone de un estimador de error confiable. Este esquema de refinamiento es desarrollado a partir de las siguientes propiedades y comportamientos identificados para las mallas óptimas, a saber:

- La contribución al error de discretización de todos los elementos en una malla óptima son del mismo orden, lo que significa que el error es equidistribuido entre todos los elementos de la malla.
- El error de discretización de una malla óptima se mantiene estable para perturbaciones de la misma, lo que significa que no hay diferencias apreciables entre el error de discretización calculado para una malla óptima y mallas cercanas a la óptima.

El método de refinamiento, basado en una distribución uniforme del error, consiste en evaluar la estimación del error a nivel de cada uno de los elementos y refinar según su magnitud hasta que el error relativo global cometido en la malla sea menor que la tolerancia prefijada.

Zienkiewicz. (1987), resalta que: “*el esquema de refinamiento toma como punto de partida una malla inicial, a la cual se le calcula el error relativo de la malla, η , conocida la norma energética del estimación del error*

$$\eta = \frac{\|\mathbf{e}\|}{\|\mathbf{u}\|} \leq \eta_0 \quad (3.5)$$

donde $\|\mathbf{e}\|$ es la norma energética del estimador del error y $\|\mathbf{u}\|$ la norma de desplazamiento, valores estos conocidos.” Por lo que se calcula el error relativo actual de la malla y se compara con el patrón de exactitud predeterminado η_0 .

Efectuando las operaciones de sustitución necesarias se obtiene que la ecuación del error relativo vendría dada por:

$$\eta = \frac{\|\mathbf{e}_u\|}{\left(\|\hat{\mathbf{u}}\|^2 + \|\mathbf{e}_u\|^2\right)^{\frac{1}{2}}} \quad (3.6)$$

de este modo se sabe si el proceso se detiene o continúa.

Seguidamente se calcula el error a nivel de cada elemento, tomando en cuenta que el requerimiento de equidistribución del error puede ser alcanzado imponiendo un límite local, $\|e_o\|$, para la norma del error para cada elemento dada por:

$$\|e_o\| = \eta_o \left(\frac{\|u\|^2}{m} \right)^{\frac{1}{2}} \cong \eta_o \left(\frac{\|u\|^2 + \|e_u\|^2}{m} \right)^{\frac{1}{2}} = \overline{\|e_o\|} \quad (3.7)$$

Es de esperar que el error en todos los elementos cumpla con:

$$\|e_i\|^2 \leq \overline{\|e_o\|}^2 \quad (3.8)$$

de donde se obtiene como criterio de refinamiento, la relación entre el error de la norma energética del elemento i y el error permisible

$$\varphi_i = \frac{\|e_i\|}{\overline{\|e_o\|}} \quad (3.9)$$

Para elementos con valores de φ_i mayores de uno, la malla será refinada.

Cuando se cuenta con generadores de malla que tienen como parámetro el tamaño del elemento, podemos suministrarle como dato el nuevo tamaño, dado como:

$$h_{\text{nuevo}} = \frac{h_i}{\varphi^{-\alpha}} \quad (3.10)$$

donde α viene dada por la relación de convergencia del error en el área cubierta por el elemento. El parámetro toma el valor de p si no hay singularidades o si el elemento está lejos de estas y valores iguales a λ para elementos cercanos a la singularidad. Si se obliga que α sea igual a p para todos los elementos de la malla entonces ocurrirían refinamientos inadecuados en las regiones cercanas a las singularidades y consecuentemente, se necesitarían de mas pasos de refinamientos para alcanzar la misma precisión. Por otra parte, si α es igualada a λ para todos los elementos, ocurriría un sobrefinamiento de la malla. En orden de obtener la exactitud deseada en una mínima cantidad de esfuerzo computacional, los investigadores Li y Wiberg.1994. y Lee y Lo. 1992. han tomado las ventajas de los índices de densidad de energía de deformación, para balancear el valor de α a lo largo de la malla.

En orden de establecer comparación entre los elementos y juzgar la presencia de singularidades, es que se ha introducido un valor de tolerancia ϑ . Asumiremos que para un elemento i , donde ϕ_i es menor que ϑ , el elemento estará localizado lejos de una singularidad, y por lo tanto, haríamos a $\alpha = p$. Para los valores de α de aquellos elementos donde ϕ_i sea mayor o igual que ϑ , primero definiremos a

$$SED_m = \max (SED_i) \quad (3.11)$$

como el índice de máxima concentración de densidad de energía de deformación y por ende:

$$\phi_m = \frac{SED_m}{SEDG} \quad (3.12)$$

Valiéndonos de estos valores, se tiene que para elementos donde $\phi_m \geq \vartheta$, podríamos calcular:

$$\alpha_i = \frac{1}{2} \left[(1 - \psi_i) \lambda + (1 + \psi_i) p \right] \quad (3.13)$$

donde ψ_i es un parámetro que depende del valor ϕ_i del elemento y debe ser igual a -1 cuando $\phi_i = \phi_m$ y igual a 1 cuando $\phi_i = \vartheta$. Lee y Lo. (1992), en orden de determinar ψ_i , asumen que “cuando se aproxima al punto de singularidad el valor de ϕ_i es directamente proporcional a la cantidad $\frac{1}{r}$, donde r , es la distancia desde la singularidad. De esta manera ψ_i viene dada por:

$$\psi_i = \frac{1}{\left(\frac{1}{\vartheta} - \frac{1}{\phi_m} \right)} \left(\frac{2}{\phi_i} - \frac{1}{\vartheta} - \frac{1}{\phi_m} \right) \quad (3.14)$$

Si existiera más de una singularidad presente en el dominio del problema, se deberán realizar refinamientos mayores en las regiones cercanas a la singularidad más fuerte ya que los valores de ϕ_i serán mayores que para los elementos cercanos a las singularidades mas débiles.

3.4 Nuevos Criterios de Refinamiento y Estimación de Error

Existen una serie de criterios que se han implementado recientemente y se consiguen en la literatura, que versan sobre nuevas técnicas de refinamiento y estimación de error. Consideramos importante resaltar brevemente dichos trabajos ya que ellos marcan el rumbo a seguir en futuras investigación, que vendrían a complementar la investigación que se presenta.

3.4.1 Criterio de refinamiento h-adaptativo modificado

La mayoría de los métodos adaptativos subdivide un elemento basado sobre la consideración de que el error de discretización es constante en el elemento. Sin embargo, el error de discretización no es constante en el elemento, por lo que deben ser consideradas las componentes direccionales en el sistema coordenado. El no tomar en cuenta esta consideraciones implica que los nuevos grados de libertad generados son ineficientes y si por el contrario se incluyen en el procedimiento de subdivisión se mejoraría notablemente el método *h-refinamiento*. Oh y Lim (1997) proponen una modificación al criterio de refinamiento tipo *h* al plantear factores de refinamiento direccionales.

Muchos autores han propuesto variados algoritmos para subdividir un elemento, como es el caso de Krishnamoorthy u Umesh (1993), quienes con la intención de obtener una malla con un gradación suavizada, proponen un método donde introducen un parámetro de refinamiento, RL 'refinement level' y utilizan la regla 'one-level-rule'. Oh y Lim (1997) presentan en su variante de la estrategia de refinamiento tipo *h*, una función de penalización para el control de los nodos de transición con la imposición de restricciones de acuerdo a la ecuación:

$$\frac{1}{2}(u_i + u_j) - u_k = 0 \quad (3.15)$$

donde u_i y u_j son los desplazamiento de los nodos extremos y u_k es el desplazamiento a restringir del nodo de transición.

El nivel de refinamiento RL, es un valor entero que representa el grado de subdivisión y puede ser expresado como:

$$\begin{aligned} \mathbf{RL}_i &= \frac{1 \log \xi_i}{p \log 2}, & \text{para } \xi_i > 1 \\ \mathbf{RL}_i &= 0 & \text{para } \xi_i < 1 \end{aligned} \quad (3.16)$$

donde $p=1$ en el caso de elementos lineales y la subdivisión del i es realizada de acuerdo al parámetro de nivel de refinamiento \mathbf{RL}_i . Debe quedar claro que cuando $\xi_i > 1$ el elemento i necesita ser refinado.

El termino ξ_i sería calculado por la expresión:

$$\xi_i = \frac{\|\mathbf{e}_\sigma\|_i}{\mathbf{e}_m} \quad (3.17)$$

donde \mathbf{e}_m es el error máximo permisible y $\|\mathbf{e}_\sigma\|_i$ es la norma del error de los esfuerzos planteada por Zienkiewicz y Zhu y ambos valores vendría dada por las ecuación siguiente:

$$\mathbf{e}_m = \bar{\eta} \left[\frac{\|\mathbf{u}\|^2 \times \|\mathbf{e}_\sigma\|}{n} \right]^{\frac{1}{2}} ; \quad \|\mathbf{e}_\sigma\|^2 = \int_{\Omega} (\mathbf{e}_\sigma)^T \mathbf{D}^{-1} (\mathbf{e}_\sigma) d\Omega \quad (3.18)$$

donde n es el número total de elementos pertenecientes al dominio, $\bar{\eta}$ que un valor máximo permisible, como por ejemplo 5% en muchas aplicaciones de ingeniería y \mathbf{D} es la matriz de elasticidad.

El valor promedio que representa el vector de esfuerzos exactos, según Zienkiewicz y Zhu, vendría dado por:

$$\sigma \cong \sigma^* = \mathbf{N} \hat{\sigma}_{ij}^{av} \quad ; \quad \hat{\sigma}_{ij}^{av} = \frac{1}{m} \sum_{e=1}^m \mathbf{D} \mathbf{B}_j^e \Delta^e \quad (3.19)$$

Y por último el estimador de error vendría dada por la ecuación:

$$\mathbf{e}_\sigma = \sigma - \hat{\sigma} \quad (3.20)$$

donde σ es el vector de esfuerzos exactos y $\hat{\sigma}$ es el vector de esfuerzos aproximados sobre un elemento.

3.4.2 Estimador de Error Direccional

La norma del error del error de Zienkiewicz y Zhu puede ser expresada con sus componentes direccionales para problemas tridimensionales como:

$$\|\mathbf{e}_\sigma\|^2 = \|\mathbf{e}_\sigma\|_x^2 + \|\mathbf{e}_\sigma\|_y^2 + \|\mathbf{e}_\sigma\|_z^2 \quad (3.21)$$

donde

$$\|\mathbf{e}_\sigma\|_x^2 = \int_{\Omega} (\sigma_x^* - \sigma_x)^2 d\Omega + \frac{1}{3} \int_{\Omega} (\sigma_{xy}^* - \sigma_{xy})^2 + (\sigma_{zx}^* - \sigma_{zx})^2 d\Omega \quad (3.22)$$

$$\|\mathbf{e}_\sigma\|_y^2 = \int_{\Omega} (\sigma_y^* - \sigma_y)^2 d\Omega + \frac{1}{3} \int_{\Omega} (\sigma_{xy}^* - \sigma_{xy})^2 + (\sigma_{yx}^* - \sigma_{yx})^2 d\Omega \quad (3.23)$$

$$\|\mathbf{e}_\sigma\|_z^2 = \int_{\Omega} (\sigma_z^* - \sigma_z)^2 d\Omega + \frac{1}{3} \int_{\Omega} (\sigma_{zx}^* - \sigma_{zx})^2 + (\sigma_{yz}^* - \sigma_{yz})^2 d\Omega \quad (3.24)$$

donde (x,y,z) son las coordenadas del sistema de coordenadas globales, aunque las subdivisiones hay que realizarlas en el sistema coordenado local. Por lo que la norma del error direccional en el sistema de coordenadas locales son expresadas como sigue:

$$\begin{Bmatrix} \|\mathbf{e}_\sigma\|_s \\ \|\mathbf{e}_\sigma\|_t \\ \|\mathbf{e}_\sigma\|_r \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} \|\mathbf{e}_\sigma\|_x \\ \|\mathbf{e}_\sigma\|_y \\ \|\mathbf{e}_\sigma\|_z \end{Bmatrix} \quad (3.25)$$

$$T_{ij} = \frac{J_{ij}^{-1}}{\sqrt{\sum_{k=1}^3 J_{kj}^{-1} \times J_{kj}^{-1}}} \quad (3.26)$$

donde los subíndices (s,t,r) son las coordenadas locales y (x,y,z) son las coordenadas globales del sistema coordinado de un elemento. T es la matriz inversa Jacobiana normalizada; J_{kj} es una componente de la matriz jacobiana. Para estas normas del error direccional se pueden definir el índice de refinamiento direccional ξ_q de la siguiente manera:

$$\|e_\sigma\|_{\min} = \min(\|e_\sigma\|_s, \|e_\sigma\|_t, \|e_\sigma\|_r) \quad (3.27)$$

$$\xi_q = \frac{\|e_\sigma\|_q}{\|e_\sigma\|_{\min}}, \quad q = (r, s, t). \quad (3.28)$$

Se pueden definir los niveles de refinamiento direccionales como:

$$n_q = \frac{\xi_q}{2} \times n, \quad q = (r, s, t). \quad (3.29)$$

3.4.3 Estimador de error expresado en términos de momentos flexionantes y fuerzas cortantes

Selman et al, (1997), hacen uso de una norma de energía del error expresado en términos de momentos flexionantes y fuerzas cortantes de la forma:

$$\|e\|^2 = \int_{\Omega} [M - \hat{M}]^T D_b^{-1} [M - \hat{M}] d\Omega + \int_{\Omega} [S - \hat{S}]^T D_s^{-1} [S - \hat{S}] d\Omega \quad (3.30)$$

donde

$$M = [M_x, M_y, M_{xy}]^T \quad y \quad \hat{M} = [\hat{M}_x, \hat{M}_y, \hat{M}_{xy}]^T \quad (3.31)$$

y son los momentos flexionantes del análisis por elemento finito y los momentos flexionantes exactos y

$$\mathbf{S} = [\mathbf{S}_x, \mathbf{S}_y]^T, \quad \hat{\mathbf{S}} = [\hat{\mathbf{S}}_x, \hat{\mathbf{S}}_y]^T \quad (3.32)$$

son los valores de las fuerzas cortantes del análisis por elemento finito y los valores de fuerza cortante exacta, respectivamente.

Como no se tiene acceso a la solución exacta para los momentos flexionantes $\hat{\mathbf{M}}$ y la fuerza cortante $\hat{\mathbf{S}}$, se sustituye la mejor estimación de estos valores como son las obtenidas por la técnica de suavización de los mínimos cuadrados, promediando los resultados nodales del análisis de elementos finitos o haciendo uso de la técnica reciente de la reconstrucción del parche superconvergente; denotando estos valores como \mathbf{M}^* y \mathbf{S}^* , de forma que:

$$\hat{\mathbf{M}} \approx \mathbf{M}^* \quad \text{y} \quad \hat{\mathbf{S}} \approx \mathbf{S}^* \quad (3.33)$$

donde los valores de los momentos flexionantes suavizados y las fuerzas cortantes suavizadas van a depender del tipo de elemento usado.

La energía de deformación de la solución exacta es estimada en :

$$\|\mathbf{w}^*\|^2 \approx \int_{\Omega} [\mathbf{M}^*]^T \mathbf{D}_b^{-1} \mathbf{M}^* d\Omega + \int_{\Omega} [\mathbf{S}^*]^T \mathbf{D}_s^{-1} \mathbf{S}^* d\Omega \quad (3.34)$$

Tanto el valor de $\|\mathbf{e}\|$ y $\|\mathbf{w}^*\|$ pueden ser evaluados a nivel de cada elemento como

$$\|\mathbf{e}\|^2 = \sum_{i=1}^n \|\mathbf{e}_i\|^2 \quad \text{y} \quad \|\mathbf{w}^*\|^2 = \sum_{i=1}^n \|\mathbf{w}_i^*\|^2 \quad (3.35)$$

donde n es el número de elementos dentro del dominio. Cuando durante el proceso de refinamiento el error relativo global llegue a alcanzar un valor permisible.

3.4.4 Técnica de reconstrucción de parches superconvergentes para problemas tridimensionales

Es bien sabido, que un criterio de refinamiento adaptativo requiere para su implementación el uso de un estimador de error confiable. En años recientes, muchos métodos adaptativos han sido propuestos usando el estimador de error presentado por Zienkiewicz y Zhu (1987), logrando incrementar la calidad de solución de los elementos finitos.

Los esfuerzos en los nodos y en los apoyos del elemento, que son obtenidos por extrapolación de los esfuerzos en los puntos de integración del análisis de los elementos finitos no poseen continuidad inter-elemental y son de baja precisión en los nodos y en los apoyos. Una gran variedad de técnicas de post-procesamiento han sido desarrolladas para la obtención de un mejor campo de esfuerzos. Dentro de estas se encuentra *la técnica de suavización global por mínimos cuadrados*, que aunque da aproximaciones razonablemente exactas y puede ser fácilmente generalizada a problemas tridimensionales, no es suficiente para el cálculo de los esfuerzos localizados cerca de los apoyos, donde no existen elementos vecinos y también dependen fuertemente de la regularidad de la malla.

Otra de las técnicas existentes para la suavización de campos de esfuerzos es *la técnica de reconstrucción de parche superconvergente propuesta* por Zienkiewicz y Zhu (1992), la cuál fue propuesta en sus inicios para problemas de una y dos dimensiones y extendida a 3D por Choi y Lee. (1996).

El procedimiento de reconstrucción de parche superconvergente hace uso del hecho de que alrededor de un nodo, los elementos que colindan con él poseen puntos de superconvergencia a partir de los cuáles puede obtenerse una expansión polinomial continua suavizada producto del ensamblaje del conjunto de estos puntos, lo que permite conocer el valor nodal de $\bar{\sigma}^*$, con orden de exactitud superior. Estos parches son construidos por la unión de hexaedros tridimensionales que comparten un nodo esquina común.

El campo de esfuerzo continuo sobre el parche, el cuál contiene elementos de transición propuesto por Choi y Lee (1996), puede también se asumido por los ocho

términos de Legendre y los parámetros desconocidos correspondientes, como se expresa en la ecuación siguiente:

$$\sigma_p^* = \mathbf{P}(\mathbf{x}, \mathbf{y}, \mathbf{z})\mathbf{a} \quad (3.36)$$

$$\sigma_p^* \equiv \sigma_p^*(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{a}_1 + \mathbf{a}_2\mathbf{x} + \mathbf{a}_3\mathbf{y} + \mathbf{a}_4\mathbf{z} + \mathbf{a}_5\mathbf{xy} + \mathbf{a}_6\mathbf{xz} + \mathbf{a}_7\mathbf{yz} + \mathbf{a}_8\mathbf{xyz} \quad (3.37)$$

En la solución por elementos finito, los puntos de integración de Gauss son generalmente los puntos de muestreo con mejor valor de esfuerzo y aún cuando los esfuerzos del elemento finito σ_h en los puntos de integración no son superconvergentes, los esfuerzos en los puntos de integración son mucho mejores que los localizados en otro punto cualquiera. Los parámetros desconocidos \mathbf{a} del polinomio de la ecuación (3.37) pudieran por lo tanto ser apropiadamente determinados empleando un mínimo cuadrado conveniente de σ_p^* a σ_h en los puntos de integración de Gauss en el parche construido. El problema entonces se inicia al conseguir primero $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ al minimizar el funcional:

$$\begin{aligned} \mathbf{F}(\mathbf{a}) &= \sum_{i=1}^n (\sigma_h(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) - \sigma_p^*(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i))^2 \\ \mathbf{F}(\mathbf{a}) &= \sum_{i=1}^n (\sigma_h(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) - \mathbf{P}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\mathbf{a})^2 \end{aligned} \quad (3.38)$$

donde (x_i, y_i, z_i) son las coordenadas globales de los puntos de muestreo y n es el número de puntos de muestreo en el parche dado. Por medio de la minimización de $\mathbf{F}(\mathbf{a})$, un conjunto de ecuaciones lineales en \mathbf{a}_i pueden ser definidas como se muestra en la formulación (3.39), de forma tal que los coeficientes \mathbf{a}_i para cada parche pudieran fácilmente obtenerse de técnicas de soluciones convencionales.

$$\sum_{i=1}^n \mathbf{P}^T(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\mathbf{P}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\mathbf{a} = \sum_{i=1}^n \mathbf{P}^T(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\sigma_h(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) \quad (3.39)$$

El lado derecho de la ecuación anterior es construido una vez para cada parche y entonces los parámetros desconocidos \mathbf{a} pudieran ser calculados de los seis componentes de esfuerzos para los problemas tridimensionales. Una vez que los parámetros \mathbf{a}_i desconocidos son determinados, los esfuerzos nodales construidos

pueden ser calculados simplemente por inserción de coordenadas globales por nodos dentro del parche dado en la expresión para σ_p^* en ecuación (3.37).

$\bar{\eta}$ especificado por el usuario, la solución es aceptada y puede expresarse como:

$$\|\mathbf{e}\| \leq \bar{\eta} \|\mathbf{w}^*\| \quad (3.40)$$

WWW.BDIGITAL.ULA.VE

CAPITULO 4

GENERADORES DE MALLAS TRIDIMENSIONALES

4.1 CLASIFICACION DE LOS GENERADORES DE MALLA

Los generadores de malla, de acuerdo a la técnica empleada para su generación, pueden ser clasificados en dos categorías, como los son técnicas de Macro Elementos y técnicas de Mallado Libre , las cuales se definen a continuación:

4.1.1 Técnicas de Macro Elementos:

En esta técnica el usuario tiene que dividir la estructura en regiones básicas simples, generalmente descritas por elementos isoparamétricos. En este tipo de esquema generador de malla, los elementos son directamente generados tanto individualmente como por parcelas. Varias técnicas de ‘mapping’ o mapeo han sido empleadas para la generación de mallas uniformes. Krishnamoorthy y Rajeshirke (1993). indican que *“la calidad de estos elementos es usualmente alta por lo que no son necesarios algoritmos de suavización de la malla.”* El gran problema al usar funciones de ‘mapping’ o mapeo es que el generador de malla sólo puede manipular dominios simplemente convexos.

Buenaño y Febres, 1996, en su investigación refieren que si el dominio del problema necesariamente debe ser cóncavo o con aberturas internas, este debe ser primero partido en regiones simples convexas, manualmente o por algoritmos especiales, antes de que tenga inicio la generación de la malla. De todas manera, en este tipo de generador de malla, los efectos de gradación pueden ser alcanzados solamente en una, dos o tres direcciones particulares (para 1D, 2D y 3D) y su aplicación esta limitada a la generación de mallas uniformes o gradadas con relaciones de cambio del tamaño del elemento en valores adecuados

Generalmente este tipo de técnica produce mallas estructuradas regulares, al usarse en procesos adaptativos, ya que las nuevas regiones refinadas contienen como base las mallas de la iteración anterior. Schweingruber y Rank.(1992), le asigna el nombre a esta metodología como técnica de Superelementos o Esquema Directo.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.1.2 Técnicas de Mallado Libre:

Buenaño y Febres, 1996, explican ampliamente esta técnica cuando indican que un enfoque completamente diferente de este tipo de esquema es capaz de generar automáticamente mallas no estructuradas en dominios de polígonos arbitrarios convexos. Zhu y coautores, (1991), Lee y coautores (1992), y C. K. Lee (1997), han publicado diferentes algoritmos los cuales usan funciones de densidad de malla para generar mallas fuertemente gradadas, muy necesarias para el cálculo de elementos finitos adaptativos. Este tipo de generador define inicialmente una malla muy somera sobre la región del dominio del problema denominada “malla de parámetros”. En esta malla se definen funciones de interpolación y se asocia un grado de libertad a cada nodo, dado por el tamaño de la malla para ese punto, de la misma forma como lo plantea el método de los elementos finitos. Basándose en esta malla inicial se genera la malla definitiva de elementos finitos que discretiza la región del problema.

Este esquema resulta ideal para los procesos de refinamiento donde el tamaño del nuevo elemento es calculado y basta con tomar la malla analizada como malla de parámetros y asociar a cada nodo el nuevo tamaño, para obtener la nueva malla refinada. Esta metodología es conocida también como técnica de remallado, ya que la malla es hecha de nuevo, sin mantener vestigios de la anterior.

La principal desventaja de estos códigos es que ellos resultan en mallas triangulares cuyos elementos son muy rígidos. Debido a esta razón, los usuarios de elementos finitos han preferido el uso de elementos rectangulares, ya que ellos poseen dependencia en la solución con la orientación en una dirección dada..

Comparando este esquema de generación de cuadriláteros, con el método directo, este es más efectivo en alcanzar los efectos de gradación rápida en los dominios irregulares y complicados. Como sea, uno de los inconvenientes del esquema de generación indirecta es que la calidad de los elementos cuadriláteros producidos no es usualmente tan buena como los generados por la generación directa.

4.2 GENERADOR H-ADAPTATIVO

La generación de mallas adaptativas debería ser totalmente automática de forma tal que una vez que la información necesaria es introducida en el generador de

malla, ninguna intervención manual posterior debe ser permitida. Además, estos generadores deben estar capacitados para generar elementos de buena forma con la menor distorsión posible.

Un elemento tiende a ser rígido y a perder exactitud si su relación de forma ‘aspect ratio’ se incrementa, como también si la diferencia entre sus ángulos internos se vuelve marcada o si el espaciamiento de los nodos de los lados, si existen, no es uniforme.

Estos requerimientos de la calidad geométrica del elemento se vuelven más importantes cuando se emplea en los análisis adaptativos mallas de cuadriláteros, pues es bien sabido, que su comportamiento es altamente sensitivo a la distorsión de su forma.

El generador de malla h-adaptativo tridimensional GEMATRI presentado en esta investigación, está basado en la técnica de macro elementos y es una versión modificada del generador de mallas tridimensional Model-3, desarrollado por Martins P. Y Barata M.(1992). Este generador tiene la capacidad de desarrollar mallas h-adaptativas en tres dimensiones, haciendo uso para el refinamiento de la misma, de los Índices de Energía de Deformación utilizados con éxito por Buenaño y Febres (1996) en su investigación cuando presentan la versión Genera-3. Estos generadores utilizan superelementos isoparamétricos con los cuales la estructura es particionada en sub-bloques (superelementos) convexos hasta que es totalmente definida. Para la generación de sólidos tridimensionales se utilizan elementos isoparamétricos hexahédricos de 20 nodos.

Las coordenadas de un punto cualquiera en el superelemento son referidas a las coordenadas naturales de la siguiente manera:

$$\mathbf{x} = \sum_{i=1}^{20} N_i \mathbf{x}_i \quad ; \quad \mathbf{y} = \sum_{i=1}^{20} N_i \mathbf{y}_i \quad ; \quad \mathbf{z} = \sum_{i=1}^{20} N_i \mathbf{z}_i \quad (4.15)$$

donde $N_i (\xi, \eta, \zeta)$ son las funciones de forma asociadas a cada nodo definido en términos de las coordenadas curvilíneas cuyo valor varia de 1 a -1 en los lados

opuestos del elemento considerado. Las funciones de forma, N_i , en el caso tridimensional del elemento de 20 nodos pueden ser escritas como:

en los nodos esquineros:

$$N_i(\xi, \eta, \zeta) = \frac{1}{8} (1 + \xi \xi_i) (1 + \eta \eta_i) (1 + \zeta \zeta_i) (\xi \xi_i + \eta \eta_i + \zeta \zeta_i - 2) \quad i=1, \dots, 8 \quad (4.16)$$

en los nodos centrales:

$$N_i(\xi, \eta, \zeta) = \frac{1}{4} (1 - \xi^2) (1 + \eta \eta_i) (1 + \zeta \zeta_i) \quad i=9, 11, 17, 19 \quad (4.17)$$

$$N_i(\xi, \eta, \zeta) = \frac{1}{4} (1 - \eta^2) (1 + \xi \xi_i) (1 + \zeta \zeta_i) \quad i=10, 12, 18, 20 \quad (4.18)$$

$$N_i(\xi, \eta, \zeta) = \frac{1}{4} (1 - \zeta^2) (1 + \xi \xi_i) (1 + \eta \eta_i) \quad i=13, 14, 15, 16 \quad (4.19)$$

Si las coordenadas de los puntos nodales son conocidas entonces las coordenadas cartesianas de un punto en particular $p(\xi, \eta, \zeta)$ pueden ser fácilmente calculadas a partir de las ecuaciones anteriores.

Una vez definido el dominio en sub-regiones los superelementos son divididos en elementos más pequeños de acuerdo al número de subdivisiones indicados por el usuario o en función de los criterios internos del programa, así como también el gradado a realizarse en cada eje local de coordenadas. Tomando como ejemplo la división en el eje local ξ , las coordenadas naturales son incrementadas de acuerdo a:

$$\xi_i = \xi_{i+1} + \frac{2(W_\xi)_i}{W_\xi^T} \quad (4.20)$$

donde

$$W_\xi^T = \sum (W_\xi)_i \quad (4.21)$$

siendo $(W_\xi)_i$ el peso relativo de gradado o la distancia entre los nuevos nodos. Para el primer nodo $\xi_i = -1$ y $(W_\xi)_i = 0$, una vez calculadas las coordenadas curvilíneas se obtienen por medio de las ecuaciones (4.16) a (4.19), las coordenadas naturales.

4.2.1 Características de GEMATRI

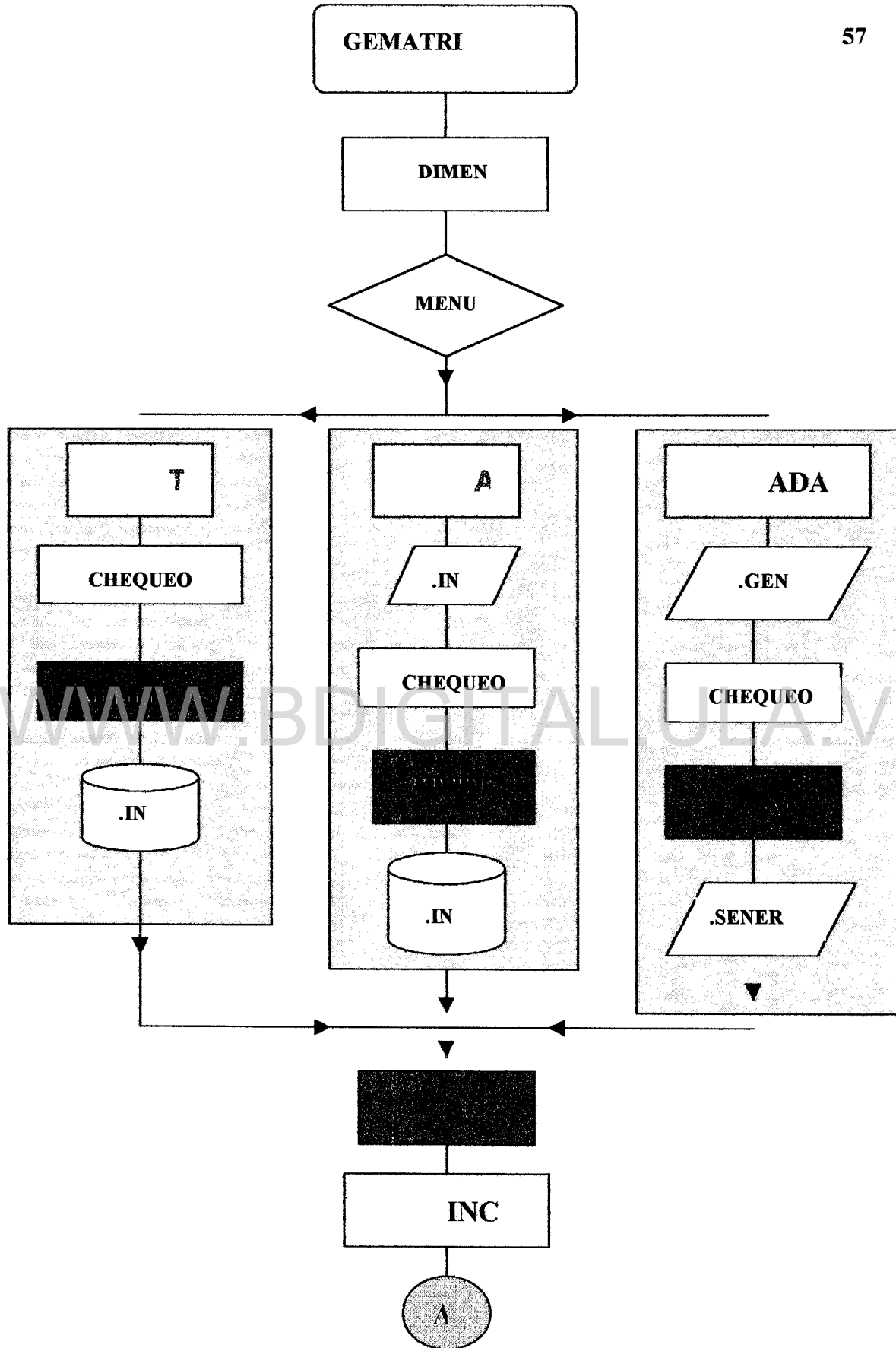
El Generador de Mallas Tridimensionales GEMATRI tiene la ventaja particular de permitir el refinamiento de los elementos que conforman la malla de la estructura que se quiere analizar, haciendo uso de los valores de Energía de Deformación, de acuerdo al criterio anteriormente expuesto. Así mismo posee la capacidad de transformación de coordenadas de forma tal que la geometría de algunos o de todos los superelementos del dominio puede ser especificada en sistemas de coordenadas diferentes al global. Estos sistemas locales de referencia pueden ser definidos en coordenadas naturales, cilíndricas o esféricas de acuerdo al caso, para luego ser transformados a coordenadas naturales para su manipulación interna.

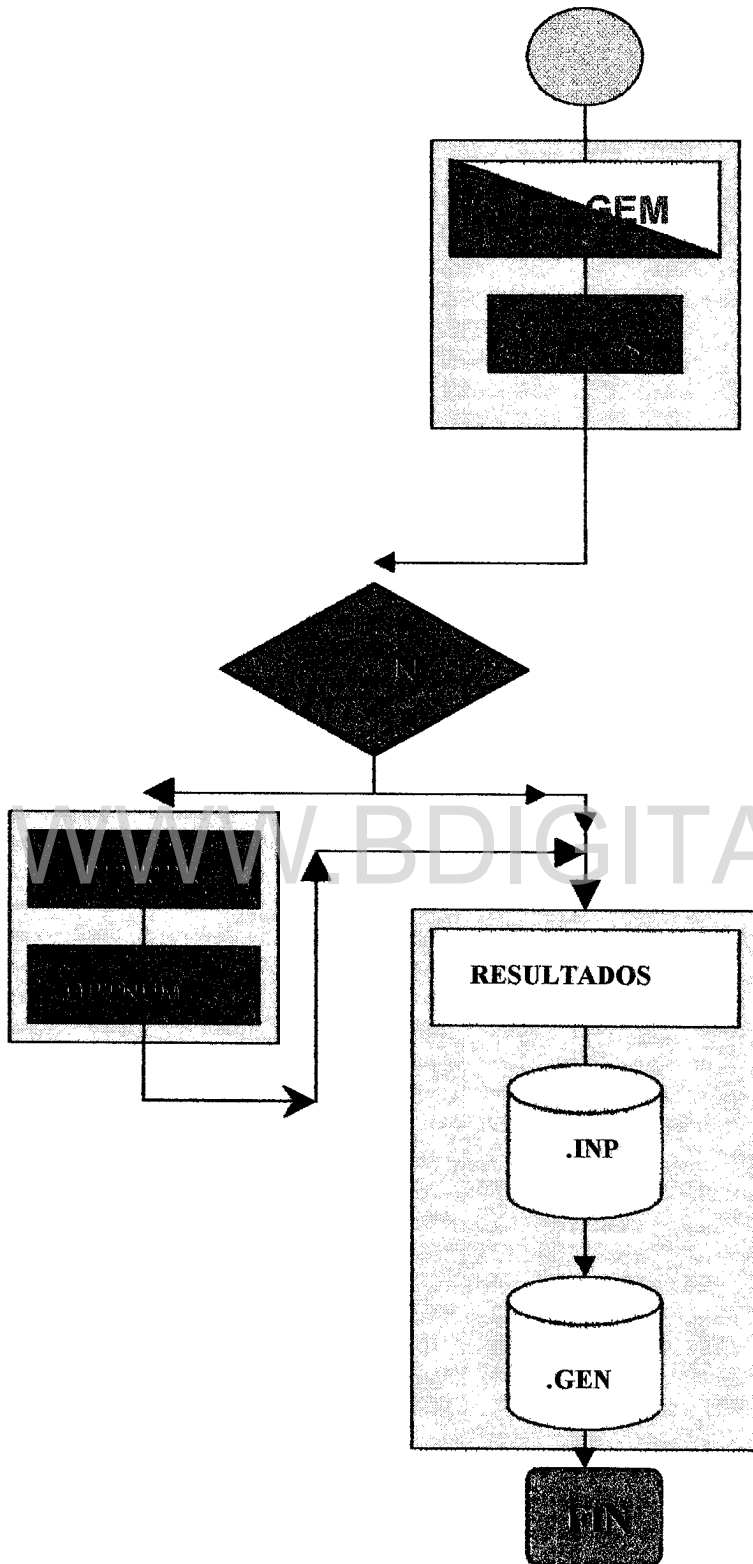
La dificultad inevitable en los generadores de mallas, como lo es, la presencia de nodos repetidos presente en las técnicas de macro elementos, es atacada mediante el algoritmo propuesto por M. E. R. Wu (1991). El programa optimiza la numeración de los nodos para reducir en lo mínimo el ancho de banda de la matriz de rigidez y evitar el incremento desmedido de esta durante los refinamientos posteriores, reduciendo el tiempo en el subsecuente análisis de elementos finitos.

El Generador de Mallas Tridimensionales GEMATRI actúa en forma interactiva con el Programa Comercial Abaqus bajo ambiente Unix; aún cuando existe una versión de dicho programa comercial para PC con el cual no debería presentar ningún problema de incompatibilidad.

Como se indicó anteriormente, el Generador de Mallas Tridimensionales H-Adaptativo GEMATRI se apoya en el generador de mallas Model-3 presentado por P. Martins y M. Barata (1992), por lo que varias de las subrutinas de éste fueron utilizadas en su versión original. Como una manera de indicar cuáles son las subrutinas creadas en esta investigación se presenta un diagrama de flujo general donde se indica en color verde las subrutinas originales de la versión y en amarillo las creadas por el autor de esta investigación. Los algoritmos generalizados de dichas Subrutinas son presentados en este capítulo, con la finalidad de indicar con claridad, el desarrollo de los mismos. Además en los anexos se presenta el listado del programa en lenguaje Fortran.

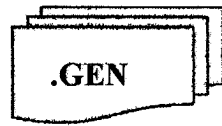
- Adaptive: Esta Subrutina permite la lectura de los valores de la Energía de Deformación que genera el programa abaqus para iniciar el proceso de refinamiento h-adaptativo de acuerdo al fundamento explicado anteriormente.
- Archi: Esta Subrutina lee los datos de entrada desde un archivo almacenándolos en un archivo .in
- Genmsh: Esta Subrutina genera una malla de elementos hexahédricos tridimensionales dentro de cada superelemento. Así mismo determina los nodos que son incompatibles, lineal y bilinealmente, para la aplicación de restricciones.
- Incomp: Esta Subrutina determina las coordenadas del nodo de los posible nodos incompatible, asociando a ellos las restricciones de los nodos vecinos.
- Optnum: Esta subrutina realiza la renumeración nodal en orden de obtener una diferencia nodal mínima.
- Resultados: Esta subrutina genera los datos de salida en formato .inp para el análisis posterior por el programa software y en formato .gen para el análisis de GEMATRI.
- Mside: Esta subrutina interpola los nodos centrales sobre los lados localizados en el lado recto del superelemento hexahédrico de 20 nodos.
- Renmbr: Esta subrutina calcula cuantos nodos están directamente conectados a un nodo dado, y almacena los números de los nodos.
- Reordr: Esta subrutina renumera los superelementos de acuerdo a la proximidad relativa entre ellos
- Tecla: Esta subrutina permite que el usuario introduzca los datos de entrada por teclado y los almacena en un archivo .in
- Ubilsd: Esta subrutina localiza los nodos lineales incompatibles tanto en elementos no refinados como en refinados.
- Shp20: Esta subrutina evalúa las funciones de forma para el elemento tridimensional hexahédrico de 20 nodos.





ADAPTIVO

LECTURA DE ARCHIVO .GEN (8 Nodos)



No. de Bloques
No. de Puntos
Coordenadas
Conectividad
No. Divisiones

- ❖ Lectura del No. De Bloques
- ❖ Transforma las coordenadas cilíndricas o esféricas en Cartesianas, si fuera necesario.
- ❖ Calcula las coordenadas del resto de los 20 nodos del elemento, con uso de la subrutina Mdside.
- ❖ Relaciona la conectividad de los bloques
- ❖ Calcula el volumen de cada elemento $\text{Vol}(i)$

- ❖ Calcula el volumen total del modelo
$$V_{\text{total}} = \sum_{i=1}^n \text{Vol}(i)$$

- ❖ Lectura de los Valores de Energía de Deformación por cada elemento.

$$\text{Sed}(i) = \sum_{i=1}^8 \text{Sener}(i).$$

- ❖ Calculo de Energía de Deformación Total
$$U_{\text{total}} = \sum_{i=1}^n (\text{Sed}(i) * \text{Vol}(i))$$

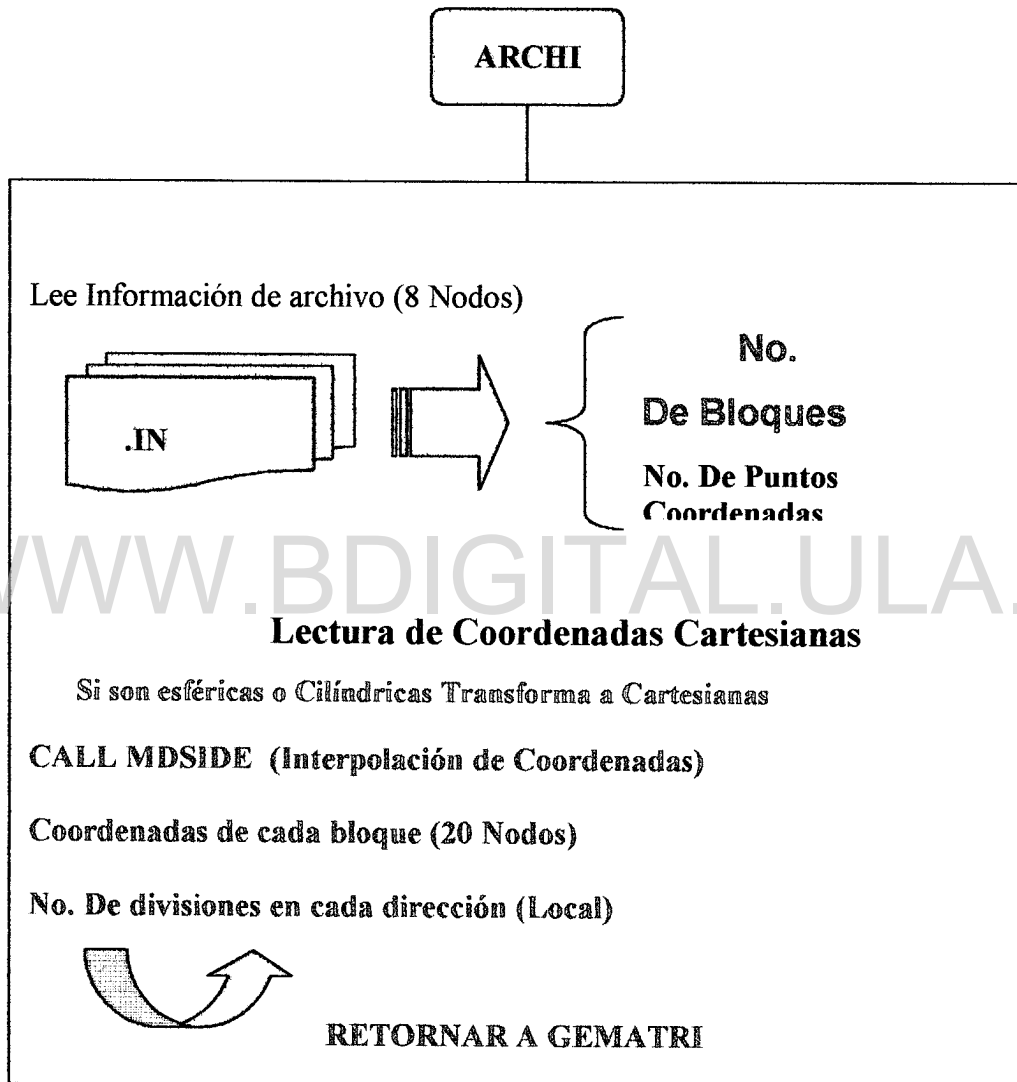
- ❖ Calcular el Índice de Concentración de Energía Global
$$G_{\text{sed}} = \frac{U_{\text{total}}}{V_{\text{total}}}$$

- ❖ Asumir el indicador de magnitud de refinamiento β

- ❖ Refinar todo elemento que cumpla con
$$\text{Sed}(i) \geq \beta * G_{\text{sed}}$$

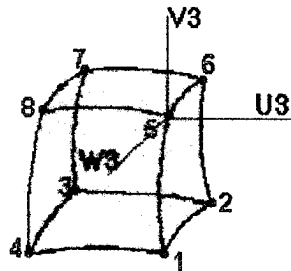


Retornar a GEMATRI



TECLA

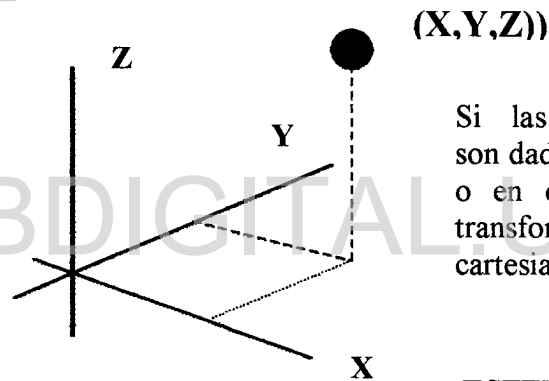
TOPOLOGIA DE CADA BLOQUE



NUMERACIÓN 8 NODOS
4 INFERIORES
4 SUPERIORES
SENTIDO ANTIHORARIO

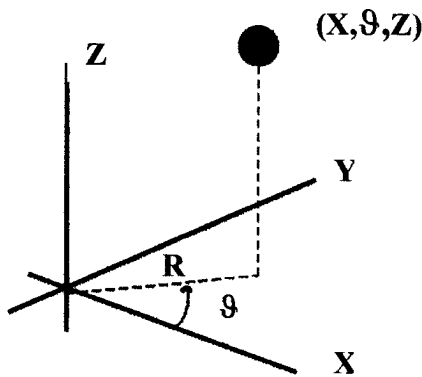
COORDENADAS DE LOS 8 NODOS (X,Y,Z)

CARTESIANAS

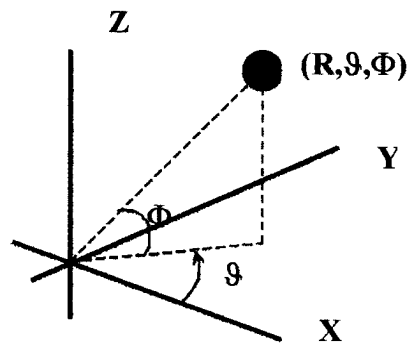


Si las Coordenadas son dadas en esféricas o en cilíndricas, las transforma en cartesianas

CILINDRICAS



ESFERICAS



- ❖ CALL MDSIDE (Interpolación de Puntos Medios)
- ❖ No. De Divisiones de cada Bloque en la dirección local. X,Y,Z
- ❖ Conectividad de cada elemento (20 nodos)

Almacena información en archivos .in



RETORNAR A GEMATRI



No. De Bloques
 No. De Puntos
 Coordenadas
 No. Divisiones
 Conectividad

Licencia Creative Commons

RESULTADOS

- ❖ Genera Archivo de Lectura ABACUS (.INP)
- ❖ Genera Archivo de Lectura GEMATRI (.GEN)

* NODE
 * NSET,NSET=Apoyo (No Nodo,X,Y,Z)
 * ELEMENT,TYPE=C3D8,ELSET= TODOS
 No. Elemento, Conectividad (8 Nodos)
 * SOLID SECTION,ELSET= TODOS,MATERIAL=Mat1
 * MATERIAL,NAME=Mat1
 * ELASTIC
 * MPC
 * LINEAR
 1,2,3
 * BILINEAR
 1,2,3,4,5
 * DEFINICION CARGA
 * CONDICIONES DE APOYO
 * SALIDA POSTPROCESADOR
 * SENER

ARCHIVO ABACUS (INP)

- ◆ No. Bloques, No. Puntos, No. Sist. Coord. ,Parametro.
- ◆ No. Bloques , Conectividad (8 Nodos)
- ◆ Identificación del sistema coordinado, Nodo, X,Y,Z
- ◆ Bloque, Divisiones x,y,z

ARCHI
 VOS
 GENMA



RETORNAR A GEMATRI

4.2.2 Proceso de Generación Adaptativa.

El programa GEMATRI durante el proceso de generación adaptativa toma la malla anterior como dato de entrada, generando los nuevos nodos y elementos donde los valores del criterio de refinamiento sobrepase el límite introducido como dato por el usuario. El programa también localiza los nodos que presentan incompatibilidad y asocia a ellos los nodos necesarios para la imposición de restricciones. En la nueva malla las propiedades del material, carga y condiciones de apoyo adecuadas son asociadas a los nuevos nodos y elementos.

Los archivos de salida del programa han sido construidos de tal manera que sirven para ser usados en el código comercial ABAQUS , donde los mismos son a la vez los archivos de entrada de dicho paquete comercial. De igual manera, para la representación gráfica de los resultados se hace uso del post-procesador que contiene dicho programa comercial.

4.2.3 Imposición de Restricciones

Durante el proceso adaptativo y como consecuencia de usar elementos rectangulares, el nodo irregular localizado en la mitad de un elemento no refinado es generado como consecuencia del refinamiento local del elemento vecino Fig 4.1. Este nodo no es físicamente definible y por lo tanto es necesario imponer restricciones al desplazamiento de dicho nodo o usar otras técnicas para preservar la compatibilidad entre elementos refinados y no refinados.

En la literatura se encuentran varias técnicas para asegurar la compatibilidad de las mallas cuadriláteras refinadas, entre otras podemos mencionar:

1. El uso de triángulos o de elementos cuadriláteros distorsionados, los cuales evitan la aparición del nodo irregular
2. El empleo de elementos de transición los cuales tienen la capacidad de originar los nodos requeridos en el centro de los lados irregulares, con el propósito de conseguir la conformidad tanto de mallas gruesas como de

mallas finas o como en elementos de bajo orden y/o elementos de orden superior .

El uso de la primera de estas técnicas crea el inconvenientes de sensibilidad del cambio de forma o rigidizacion de los elementos discutida anteriormente. La última necesita el desarrollo y prueba de estos elementos, los cuales no se encuentran en la mayoría de los códigos comerciales existentes.

Cuando la presencia del nodo irregular no es combatida mediante las técnicas mencionadas anteriormente, se impone entonces restringir el desplazamiento de este para conseguir la compatibilidad. L. Quiros y coautores (1992), clasifican los problemas de unión no conforme de mallas de acuerdo al tipo de no conformidad en tres categorías :

1. Elementos de igual tipo con nodos no coincidentes
2. Elementos de tipo diferente con nodos coincidentes .
3. Elementos de tipo diferentes con nodos no coincidentes .

Las restricciones las podemos clasificar según el objeto que persiguen en:

1. Restricciones de punto simple (single-point constraint), cuando impone a un grado de libertad un valor dado.
2. Restricciones de puntos múltiples (multi-point constraint), cuando se impone una relación entre dos o mas grados de libertad.

Para combatir es tipo de incompatibilidad se han desarrollado diferentes metodología, entre las cuales podemos mencionar:

1. Multiplicadores de Lagrange.
2. Funciones de penalización.
3. Métodos de transformación.

El método de los Multiplicadores de Lagrange, aunque efectivo, fue descartado como esquema de restricción, ya que produce una matriz de rigidez expandida de naturaleza definida no positiva, debido a que los elementos de la diagonal de la matriz de rigidez correspondientes a los multiplicadores son cero. Por lo cual se hace necesario el uso de un algoritmo de resolución basado en una estrategia de pivotes alternantes. Este método es más atractivo cuando son requeridas pocas ecuaciones de restricción.

MPC Linear y Bilinear

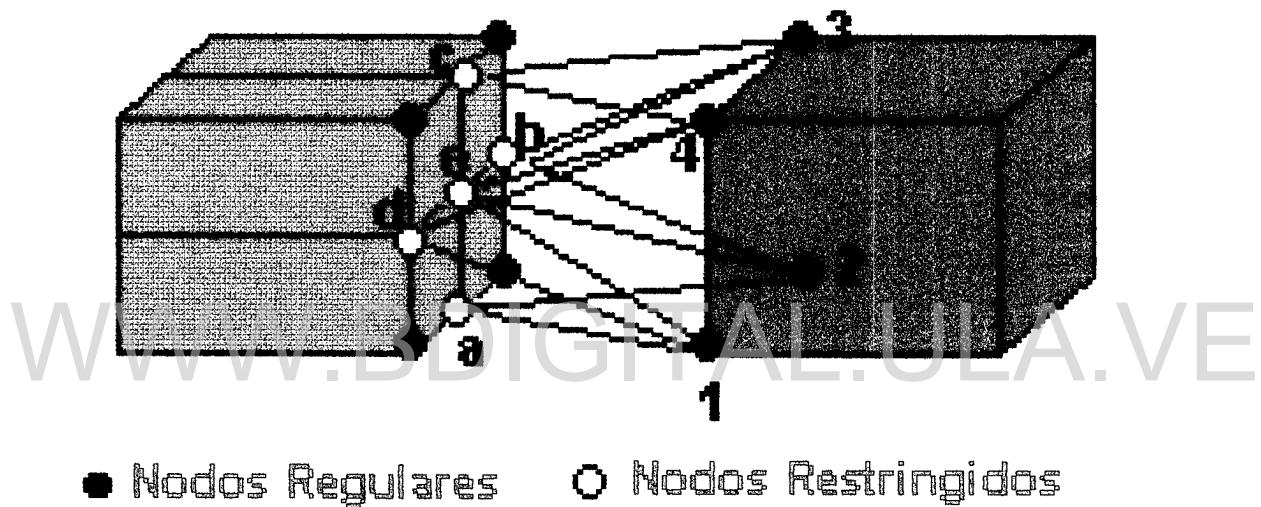


Fig. 4.1 Dependencia de los nodos nuevos

El método de la función de penalización presenta la ventaja de no producir incógnitas adicionales y la matriz de rigidez resultante sigue siendo positivamente definida. La desventaja principal es que hay que determinar el factor de penalización apropiado a cada elemento. Si el factor es muy pequeño la violación a las restricciones es inaceptable y si es muy alto se producen problemas numéricos.

Dentro de los métodos de transformación, podemos citar el propuesto por el prof. R. D. Cook (1989), que hace uso de matrices de transformación y el método directo propuesto por J. L. Batoz y G. Dhatt (1990). En estos métodos la matriz de rigidez permanece positiva y la transformación requiere de una cantidad significativa de memoria y tiempo. Las restricciones aplicadas en este trabajo entran de esta categoría.

Debido a que las restricciones aplicadas en este trabajo entran en las categorías anteriormente descritas, se amplía el algoritmo general para la restricciones a casos de 3D desarrollado por Buenaño y Febres, 1996, que permite la imposición de restricciones del tipo de puntos múltiples. Este algoritmo consiste en promediar los desplazamientos de los nodos vecinos para preservar la compatibilidad entre elementos.

$$\bar{\mathbf{u}}_c = \frac{1}{2}(\bar{\mathbf{u}}_a + \bar{\mathbf{u}}_b) \quad (4.22)$$

donde $\bar{\mathbf{u}}_c$ es el nodo irregular en el centro del lado no conforme del elemento no refinado.

CAPITULO 5

RESULTADOS NUMERICOS

5.1 Introducción

Para lograr evaluar la aplicabilidad y efectividad del generador de mallas tridimensional adaptativo GEMATRI que se propone, se presentan algunos ejemplos numéricos encontrados en la literatura [Choi y Lee, 1996, Lee y Lo, 1997]. En la generación de mallas tridimensionales adaptativas se utilizan elementos isoparamétricos hexaédrico de 20 nodos como se muestra en la figura 5.1, mientras que para el caso del análisis por elementos finitos con el paquete comercial ABAQUS se hace uso del elemento de 8 nodos S3D8R, de acuerdo a la figura 5.2.

El programa GEMATRI fue desarrollado en el lenguaje Fortran 77 con la finalidad de aprovechar la ventaja que ofrece el hecho de que el paquete comercial ABAQUS, con el cual intereectúa, se encuentra en ambiente Unix.

Los resultados numéricos son obtenidos del análisis de dos problemas típicos estudiados por Choi y Lee, (1996) y Lee y Lo, (1997) y que fueron mostrados en forma preliminar por Meléndez, Buenaño y Febres. (1997). El primer problema trata del estudio de una unión tridimensional viga-columna con discontinuidad geométrica y el segundo se refiere a una viga en voladizo que soporta una carga uniforme con condición de apoyo irregular.

Para el análisis de los resultados de estos ejemplos se presenta por una parte la secuencia de mallas adaptativas generadas por el programa GEMATRI y por la otra una tabla de resultados donde se compara la tendencia de valores representativos entre las mallas adaptadas y mallas uniformemente refinadas.

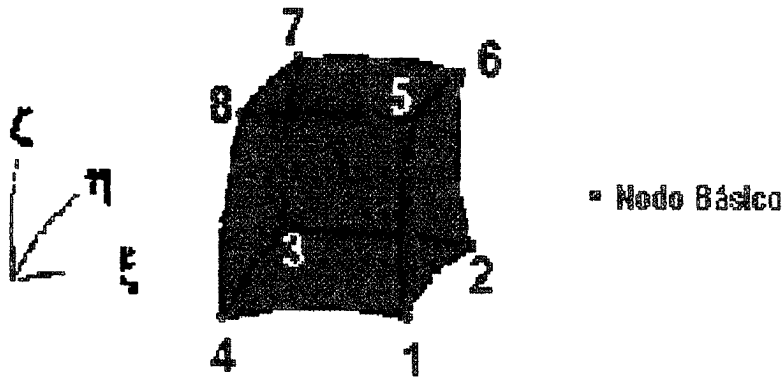


Figura 5.1 Elemento Hexaédrico de 8 nodos S3D8R
de la librería de ABAQUS

WWW.BDIGITAL.ULA.VE

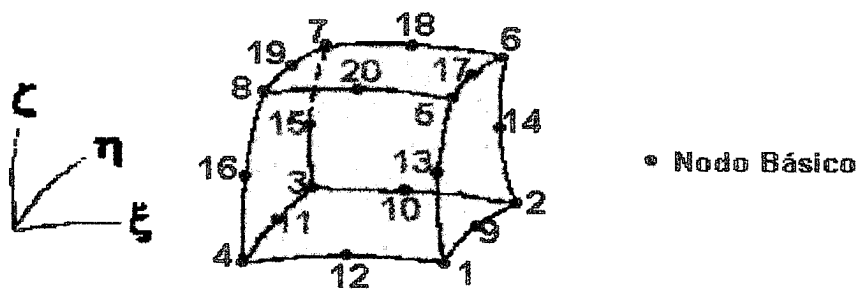


Figura 5.2 Elemento Hexaédrico de 20 nodos

5.2 JUNTA DE UNIÓN VIGA-COLUMNA

Se presenta un primer ejemplo de una unión tridimensional viga-columna con discontinuidad geométrica sometida a la carga, condiciones de apoyo, propiedades y el punto control del desplazamiento “A” como se muestra en la figura 5.3.

La información obtenida, una vez efectuado los análisis de cada una de las mallas adaptativas, se presenta en forma de tabla con el objeto de visualizar las variaciones propias de dichas mallas. A tal efecto, se presenta la tabla 5.1 que permite la comparación de resultados de la Junta Unión-Viga Columna y muestra la malla original 5.4(a), segunda 5.4(b) y tercera 5.4(c), así como también la malla control 5.4(d), la uniformemente refinada.

Del análisis comparativo de los resultados de dicha tabla 5.1 se desprende que existe una rápida convergencia del parámetro desplazamiento del punto “A” de la tercera malla con respecto a la malla uniformemente refinada. Esta convergencia al valor real del desplazamiento del punto “A” se logra en un tiempo de utilización del CPU muy reducido.

En la actualidad, debido a los costos que representa el tiempo de utilización de paquetes computacionales para el análisis de estructuras complejas, se hace necesario la búsqueda de metodología que permitan lograr obtener resultados satisfactorios con un esfuerzo computacional menor (Costo Computacional). El hecho de utilizar la combinación de los Índices de Energía de Deformación con procesos de h-refinamiento, permiten en este ejemplo de la Junta Unión Viga-Columna, reducir el tiempo de computación en un 70 %. Esta reducción es debida a que en el análisis de la tercera malla sólo es requerido un 30% de los grados de libertad y un 27 % del número de elementos que se requieren para un análisis completo de la malla uniformemente discretizada.

De esta manera queda demostrado que el método presentado permite la solución numérica eficiente, confiable y económica, a un costo computacional bajo.

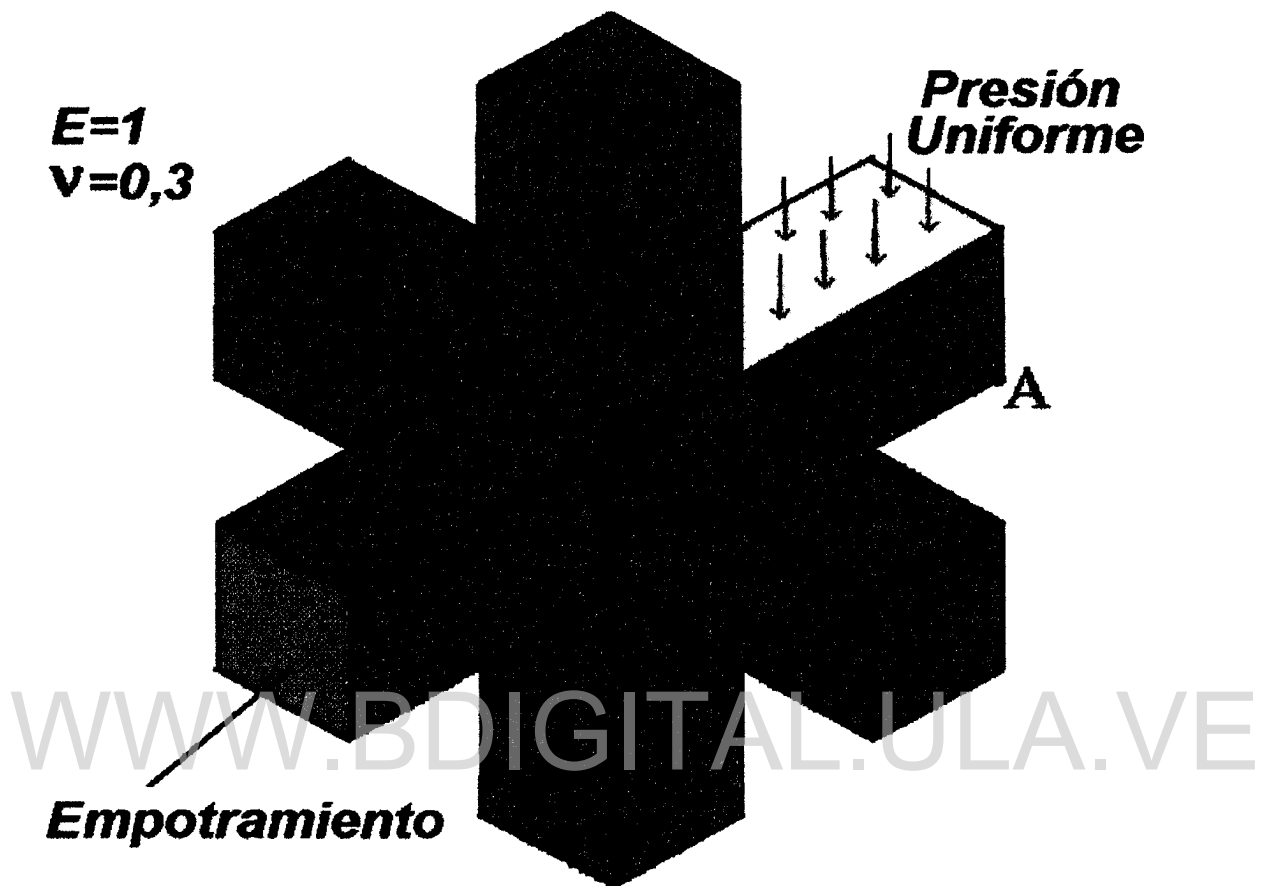
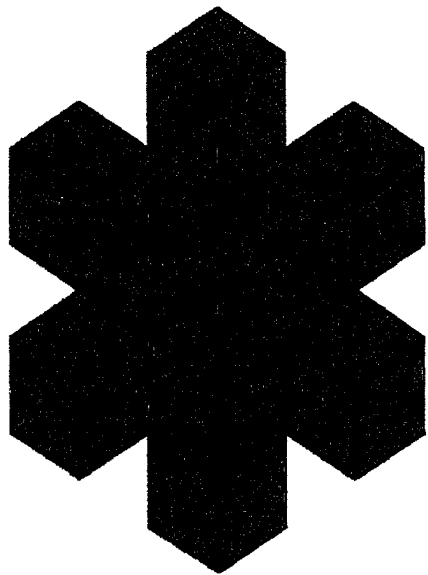
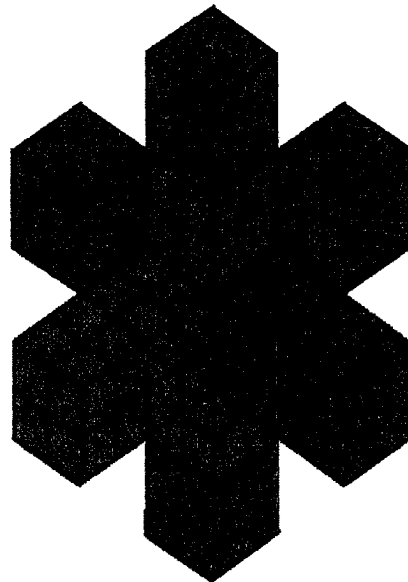


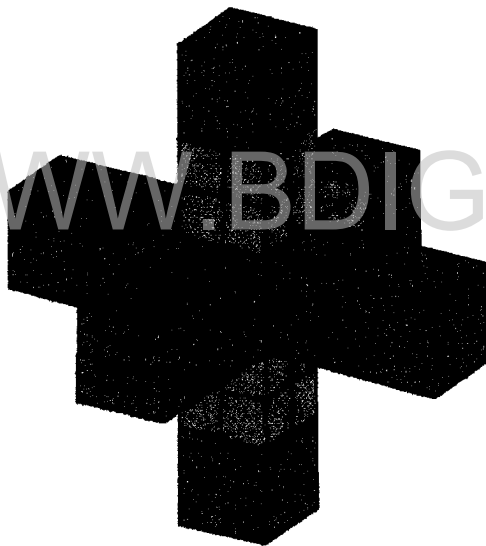
Fig. 5.3 Junta Unión viga-columna



a) Malla original



b) Segunda malla



c) Tercera malla



d) Malla uniforme

Fig. 5.4 Secuencia de mallas adaptadas de junta de unión Viga-columna

Mallas Relacionadas	Malla Original	Segunda Malla	Tercera Malla	Uniformemente Refinada
Figura N°	5.4(a)	5.4(b)	5.4(c)	5.4(d)
No de elementos	13	55	167	832
No. nodos	56	145	325	1325
Grados de libertad	156	420	927	3900
Tiempo (sec)	0,21	0,45	0,95	3,6
Desplazamiento	-1543	-1286	-1299	-1320
Energía de Deformación Total	$4,56 \times 10^3$	$3,81 \times 10^3$	$3,85 \times 10^3$	$3,91 \times 10^3$

TABLA 5.1
Comparación de resultados de la junta unión viga-columna

5.3 VIGA EN VOLADIZO CON EMPOTRAMIENTO DEFECTUOSO.

Se presenta un segundo ejemplo de una viga en voladizo que soporta una carga uniforme con condiciones de apoyo poco convencional, ya que el empotramiento es irregular. La geometría del modelo, las condiciones de apoyo, las propiedades, la carga aplicada y el punto control del desplazamiento “A”, se muestran en la Figura 5.5. Se presenta la tabla 5.2 que permite la comparación de resultados de la Viga en Voladizo y muestra la malla original 5.6(a), segunda 5.6(b) y tercera 5.6(c), así como también la malla control 5.6(d), la uniformemente refinada.

De nuevo en este ejemplo se demuestra del análisis de la tabla 5.2, la rápida convergencia del parámetro desplazamiento del punto “A” de la tercera malla con respecto a la malla uniformemente refinada, lo que representa un tiempo de utilización del CPU muy reducido.

La utilización de los Índices de Energía de Deformación en el proceso h-refinamiento, permiten en este ejemplo, reducir el tiempo de computación en un 94 %, con un 95% menos de los grados de libertad y solo un 4 % de elementos que se requieren para un análisis completo de la malla uniformemente discretizada.

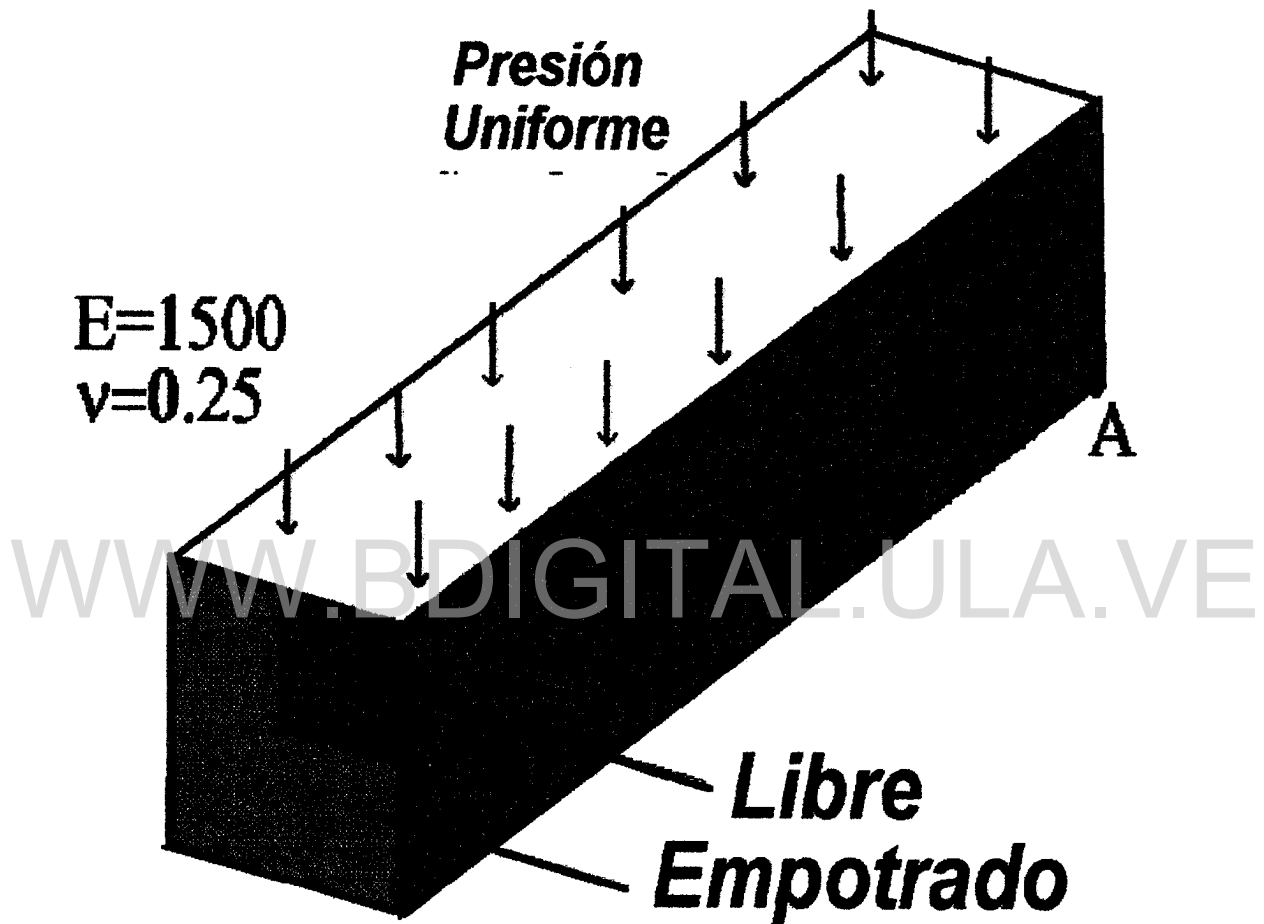
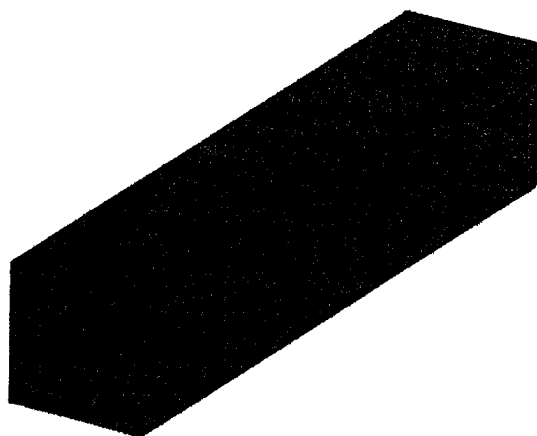
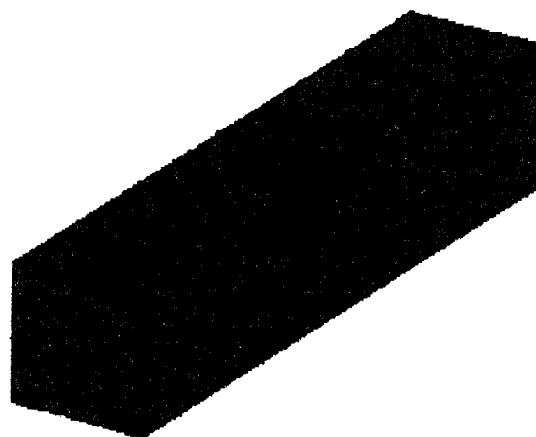


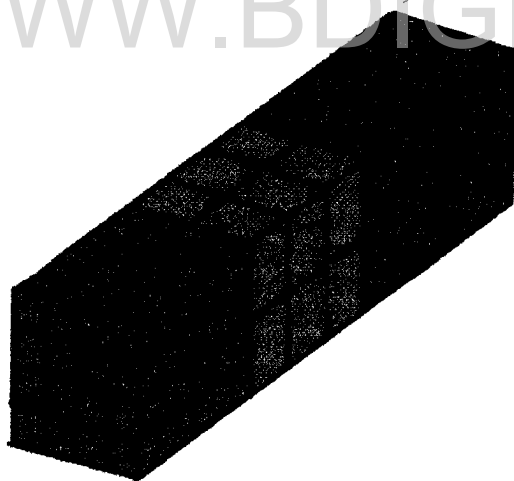
Fig. 5.5 Viga en voladizo con empotramiento defectuoso



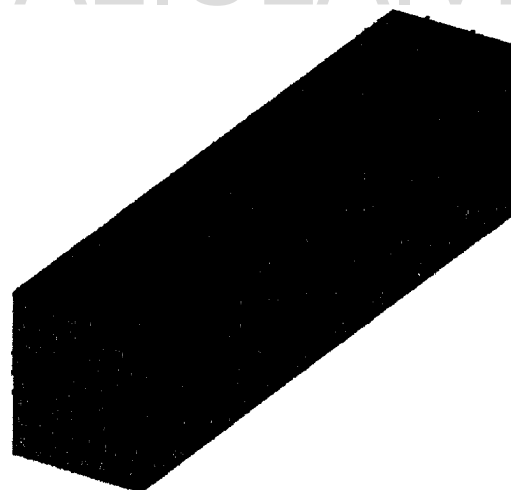
a) Malla original



b) Segunda malla



c) Tercera malla



d) Malla uniforme

Fig. 5.6 Secuencia de mallas adaptadas de viga en voladizo con empotramiento defectuoso

Mallas Relacionadas	Malla Original	Segunda Malla	Tercera Malla	Uniformemente refinada
Figura N°	5.6 (a)	5.6 (b)	5.6 (c)	5.6 (d)
No de elementos	5	26	110	2560
No. nodos	24	71	227	3321
Grados de libertad	63	189	567	9795
Tiempo (sec)	0,16	0,32	0,68	11,3
Desplazamiento	-2,397	-1,377	-1,43	-1,304
Energía de Deformación Total	10,6	5,72	5,96	6,04

TABLA 5.2
Comparación de resultados de la viga en voladizo con empotramiento defectuoso

CONCLUSIONES

En el análisis de sólidos tridimensionales es usual encontrarse con la necesidad de incrementar la precisión en los cálculos sin incrementar grados de libertad innecesarios. Por lo tanto, es necesario densificar la malla de tal forma que sea lo suficientemente fina en regiones críticas, permaneciendo razonablemente gruesa en el resto del dominio. En la presente investigación se demostró que el Índice de Densidad de Energía de Deformación (SED) utilizado con éxito en problemas 2D, [Buenaño y Febres (1996)], es un indicador efectivo de las localizaciones donde se requiere refinar una malla de Elementos Finitos. Así mismo, el Generador de Mallas adaptativas GEMATRI, ha demostrado ser una herramienta de gran utilidad en procesos de modelado computacional, ya que permite resolver problemas de elasticidad 3D, reduciendo notablemente el esfuerzo y capacitación del usuario, así como también el costo computacional. Los dos ejemplo estudiados mostraron la rápida convergencia de los refinamientos sucesivos hacia la mejor solución, reduciéndose considerablemente el tiempo de cálculo.

Los resultados de los ejemplos estudiados mostraron que el método es efectivo en la indicación de las regiones donde es necesaria una malla más fina, en orden de obtener una mejor solución. Se encontró que el procedimiento es capaz de identificar las regiones de grandes gradientes de esfuerzos y localizar las singularidades, produciendo mallas de transición gradual alrededor de estas.

Este procedimiento adaptativo se diferencia de la gran mayoría de los procesos ya que no utiliza estimador de error como criterio de refinamiento; no obstante, la necesidad de contar con un estimador de error para indicar cuando parar el proceso y aún más importante para validar el resultado obtenido es una necesidad.

Para futuras investigaciones es necesario la inclusión de otro tipo de elemento que permita una mejor discretización en el análisis de modelos con superficie curva, así como también la combinación del uso de los Índices de

Energía de deformación con Técnica de Extrapolación demostrada en otras investigaciones.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

REFERENCIAS BIBLIOGRAFICAS

1. **ABAQUS, User's Manual** (1996)., Versión 5.5, Hibbitt, Karlsson & Sorence, Inc.,
2. Bretone M., Rodríguez A. Y Huerta A., (1995), **La Programación Orientada al Objeto Aplicada al Cálculo por El Método de los Elementos Finitos**, Revista Internacional de Métodos Numéricos para el Cálculo y Diseño en Ingeniería. Vol 11, No. 3, pp. 423-449.
3. Buenaño A. Y Febres H. E. ,(1995), **Generador de malla H-adaptativo Aplicado a problemas de Flexión de Placas usando Indices de Densidad de Energía de Deformación**, Boletín Técnico INME, Vol 33, No. 1, pp. 53-63.
4. Buenaño A. Y Febres H. E., (1996), **Proceso Adaptativo Basado en Indices de Energía de Deformación aplicado al Problema de Flexión de Placas**, Tesis de Maestría en Ingeniería Estructural. Universidad de Los Andes. Mérida.
5. Choi C. and Lee N. (1996), **A 3-D Adaptive Mesh Refinement Using Variable-Node Solid Transition Elements**, International Journal for Numerical Methods in Engineering, Vol. 39. Pp. 1585-1606.
6. Cook R. D., Malkus D. S. y Plesha M. E. (1989)., **Concepts and Application of Finite Element Analysis**, 3rd edition, Editorial Jhon Wiley & Sons, Inc
7. Cook R. D., (1995)., **Finite Element Modeling for Stress Analysis**, Editorial Jhon Wiley & Sons, Inc. U.S.A
8. Febres-Cedillo H. E. y Bhatti M. A. (1988), **A Simple Stain Energy Bases Finite Element Mesh Refinemet Scheme**, Computers & Structures, Vol. 28. No. 4, pp 523-533
9. Febres-Cedillo H. E., (1984)., **An Energy-based Adaptive Finite Element Analysis of Two-dimensional Elasticity Problems**, Thesis for Degree of Science in Civil and Environmental Engineering, University of Iowa.
10. Fornons J., **El Método de los Elementos Finitos en Ingeniería de Estructuras**, Universidad Plitécnica de barcelona, España.
11. García J. M, (1994), **Verificación Computacional de Soluciones Numéricas**, Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, Vol 10, No. 4, pp. 341-364.

12. González Castor ,(1993), **El Método de Los Elementos Finitos Adaptable**, Tesis Doctoral en Ciencias de la Computación, Universidad Central de Venezuela, Caracas.
13. Hormaza M., Amezua E., Aviles R. Y Hernández A., (1995), **Un Método de Regularización de Mallas de Elementos Finitos en Modelos Tridimensionales Sólidos**, Revista Internacional de Métodos Numéricos para el Cálculo y Diseño en Ingeniería. Vol 11, No. 3, pp. 401-421.
14. Krishnamoorthy and Rajeshirke Umesh (1993)., **Adaptive Mesh Refinement for Two-dimensional Finite Element Stress Analysis**, Computers & Structures, Vol 48, No. 1. pp.121-133.
15. Lakshmanan S, Soni K. Y Balasubramaniam K., (1997), **R-Adaptation in Finite Element Modelling of Elastic Solids** , Computers & Structures, Vol 63, No. 2, pp. 249-257.
16. Lee C. K y Lo S. H.,(1992), **An Automatic Adaptive Refinement Finite Element Produce for 2D Elastostatic Analysis**, International Journal for Numerical Methods in Engineering, Vol. 35, pp 1967-1989.
17. Lee C. K y Lo S. H., (1997), **Automatic adaptprtive 3-D Finite Element Refinemet using different - order tetrahedral elemets**, International For Numerical Methods in Engineering, vol 40, pp 2195-2226
18. Li X. Y Wiberg N., (1994), **A posteriori error estimate by element patch post-procesing, adaptive in energy and L2 norms**, Computers & Structures, Vol. 53, No. 4, pp. 907-919.
19. Meléndez C., Buenaño A. y Febres H., (1997), **Refinamiento Automático H-Adaptativo de mallas Tridimensionales**, XVIII CILAMCE, Congreso Ibero Latino-Americano de Métodos Computacionales para Ingeniería, Vol III, pp 1345-1352
20. Morgan D. J., Tyler J. M. Y Dorroh J. R., (1995), **A New 3D Finite Element for Adaptive H-Refinement in 1-irregular Meshes**, International For Numerical Methods in Engineering, Vol 38, pp 3989-4008.
21. Oh H. S. Y Lim J. K., (1997), **Modified H-Meth with Directional Error Estimate for Finite Element Stress Analysis**, Computers & Structures, Vol 65, No. 2, pp. 191-204.
22. Selman A., Hinton E. Y Bicanic N., (1997), **Adaptive mesh Refinement for Localised Phenomena**, Computers & Structures, Vol 63, No. 3, pp. 475-495.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

23. Stein E. y Ohnimus S. (1992), **Concept and Realisation of Integrated Adaptive Finite Element Methods in Solid and Structural Mechanics**, Numerical Methods in Engineering, Elsevier, .
24. Umar A., Abbas H., Qadder A. Y Sehgal D. K., (1996), **Prediction of error in Finite Element results**, Computers & Structures, Vol 60, No. 3, pp. 471-480.
25. Weaver W. Y Johnston P., **Finite Elements for Structural Analysis**, Editorial Prentice Hall, Inc. , Englewood Clifes, New Jersey.
26. Wu M. E. R., (1991), **Techniques to avoid duplicate nodes and relax restrictions on super elements numbering in a mesh generator**, Computers & Structures, Vol 15, pp 419-428.
27. Zienkiewicz O. C. Y Taylor R. L., (1994), **El Método de los Elementos Finitos**, 4ta Edición, Editorial McGraw-Hill – Interamericana de España, S.A, España.
28. Zienkiewicz C. y Zhu J. Z., (1992), **Mesh Regeneration and Automatic Adaptive Analysis**, Numerical Methods in Engineering, Elsevier, Vol 34, No 2, pp. 245-278

WWW.BDIGITAL.ULA.VE

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

PROGRAM C
COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION CORDB(1200,3),MATNB(60),LNODB(60,20),ORIGB(10,3),
1 LNODR(60,20),NBUSD(60),NBNGH(60),NDIVX(60),NDIVY(60),
2 NDIVZ(60),WEITX(60,30),WEITY(60,30),WEITZ(60,30),
3 LREPN(300),LASOC(300),LFINN(300),LFASC(300),COORD
4 (4800,3),MATNO(600),LNODS(600,8),NDUMY(60,4),DUMYX
5 (60,30),DUMYY(60,30),DUMYZ(60,30),NNCNEC(4800)
DIMENSION ELAST(60),CNABLA(60),POBIL(1200,3),NNCOR(1200,3),
1 PNOBIL(1200,3),CBILI(1200,3),CODAS(1200,8,3),
2 CODASII(1200,3),CODASIS(1200,3),CODASDI(1200,3),
3 CODASDS(1200,3),CLIN(1200,3),LAS01(1200,3),LAS02(1200,3)
DIMENSION CARAI(60,400,3),PLIL(1200,3),CHEQ(3),CALII(1200,3),
1 CALID(1200,3),COLMBD(1200,3),COLMBR(1200,3),
2 COLMBU(1200,3),COLMBL(1200,3),COBIL(1200,3),
3 COBIR(1200,3),COBRD(1200,3),COBRU(1200,3),
4 COBUL(1200,3),COBUR(1200,3),COBLU(1200,3),COBLD(1200,3)
5 ,SALTO(4,4),SED(60),XBLOC(60),XBLOC1(60),VOL(1200)
INTEGER PONTRS(4800),ORIGN,POLA,CAMBIO,IPOL(4800),NODOIS(1200),
1 NODOII(1200),NODODI(1200),NODODS(1200),IPOLI(4800),
2 NODII(1200),NODIS(1200),NODO(1200)
INTEGER COORD,SELEC,NM,NL,ICON,INCON,LINE(1200),LINE1(1200)
1 ,LINE2(1200),LIRCI(3),LIRCD(3),IG,PILAS,ICONTROL1(1200)
2 ,ICONTROL2(1200),ICONTROL3(1200),ICONTROL4(1200),
3 BASE11,BASE12,BASE21,BASE22,BASE31,BASE32,BASA11,
4 BASA12,BASA22,BASA21
DOUBLE PRECISION VALOR
REAL TOTX,TOTY,TOTZ,SEDG,BETA,UTOTAL,VTOTAL
-----
C PROGRAMA PRINCIPAL
C -----
C DATA NNODE,LNODE,NDIME/8,20,3/
C
CALL DIMEN (MBLOC,MPONT,MDIVI,MREP,NM,MPOLA,MELEM,MPOIN)
WRITE(*,100)
WRITE(*,120)'INTRODUCIR DATOS DESDE EL TECLADO.....(1)'
WRITE(*,130)'CARGAR ARCHIVO DE DATOS EXISTENTE .....(2)'
WRITE(*,130)'CARGAR ARCHIVO ADAPTATIVO .....(3)'
WRITE(*,140)'INTRODUZCA SU SELECCION = '
READ(*,*)SELEC
IF (SELEC.EQ.1) THEN
CALL TECLA (NBLOC,MPONT,CORDB,MATNB,LNODB,ORIGB,
1 NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,MBLOC,
2 MPONT,MDIVI,MPOLA,NMATNB,ELAST,CNABLA,
3 POLA,CAMBIO,VALOR,COORD,NABLA,ELASTIC,PILAS)
ELSE IF (SELEC.EQ.2) THEN
CALL ARCHI (NBLOC,MPONT,CORDB,MATNB,LNODB,ORIGB,
1 NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,MBLOC,
2 MPONT,MDIVI,MPOLA,NMATNB,ELAST,CNABLA,
3 POLA,CAMBIO,VALOR,COORD,SELEC,NABLA,ELASTIC,PILAS)
ENDIF
IF (SELEC.EQ.3) THEN
CALL ADAPTIVE (NBLOC,MPONT,MOPLA,MBLOC,MDIVI,SED,CORDB,LNODB,
1 NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,SEDG,BETA,
2 ORIGB,MATNB,PILAS,BASE11,BASE12,BASE21,
3 BASE22,BASE31,BASE32,BASA11,BASA12,BASA22,BASA21,
4 VOL,UTOTAL,VTOTAL)
ENDIF
CALL INCOMP (NBLOC,NDIVX,CORDB,POBIL,LNODB,NCOR,TOTX,
1 TOTY,TOTZ,MPONT,NM,MBLOC,ICON,INCON,PNOBIL,
2 NNCOR,CBILI,CODAS,CODASIS,CODASII,CODASDS,
3 CODASDI,COLMBD,COLMBR,COLMBU,COLMBL,COBIL,COBIR,
4 COBRD,COBRU,COBUL,COBUR,COBLU,COBLD,PLIL,CALII,CALID,
5 CHEQ,SALTO,IG,ICONTROL1,ICONTROL2,ICONTROL3,
6 ICONTROL4)
IF (PILAS.NE.1) THEN
CALL REORDR (NBLOC,MATNB,LNODB,LNODR,NBUSD,NBNGH,
1 NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,
2 NDUMY,DUMYX,DUMYY,DUMYZ,
3 MBLOC,MDIVI)
ENDIF
CALL GENMSH (NBLOC,NELEM,NPOIN,CORDB,MATNB,LNODB,NDIVX,NDIVY
1 ,NDIVZ,WEITX,WEITY,WEITZ,LREPN,LASOC,LFINN,LFASC,
2 COORD,MATNO,LNODS,MBLOC,MPONT,MDIVI,MREP,NM,MELEM,
3 MPOIN,NL,ICON,NRENM,ORIGN,IPOL,NNCNEC,CBILI,CODASIS,
4 ,CODASII,CODASDS,CODASDI,NODOIS,NODOII,NODODI,NODODS,

```

```

5          IPOLI,NODII,NODIS,PLII,CALII,CALID,IG)
CALL UBILSD (NBLOC,NPOIN,NNCNEC,COORD,DARAI,DARAS,DARAB,MPOIN,
1          NRENM,ORIGN,CLIN,LAS01,LAS02,LINE,LINE1,LINE2,LIRCI
2          ,CARAI,CARAS,CARAB,LIRCD,MPONT,IDRB,IDCS)
IF (NRENM.EQ.1)THEN
CALL RENMBR (NELEM,NPOIN,JNODS,MELEM,MPOIN,IDIFF,ORIGN,NNCNEC,
1          PONTRS)
ENDIF
CALL OUTPUT (NELEM,NPOIN,TITLE,
1          COORD,MATNO,LNODS,
2          MELEM,MPOIN,
3          ORIGN,NRENM,NNCNEC,PONTRS,
4          NMATNB,ELAST,CNABLA,IPOL,
5          NODOIT,NODOIS,NODODI,NODODS,IPOLI,NODII,NODIS,ICON,
6          IG,NODO)
100  FORMAT(5X,70(' '),/,5X,'*',68X,'*',
1/,5X,'*',26X,'PROGRAMA GEMATRI',26X,'*',/,5X,'*',68X,'*',
2/,5X,'*',17X,'GENERADOR DE MALLA TRIDIMENSIONAL',18X,'*',/,
35X,'*',68X,'*',/,5X,'*',7X,'USANDO SUPERELEMENTOS E INDICES DE DEN
4SIDAD DE ENERGIA',7X,'*',/,5X,'*',68X,'*',/,5X,'*',19X,'AUTOR: ING.
5 CARLOS MELENDEZ D.',20X,'*',/,5X,'*',68X,'*',/,5X,'*',23X,'FECHA:
6SEPTIEMBRE 1997',24X,'*',/,5X,'*',68X,'*',/,5X,70(' '))
120  FORMAT(/,15X,A)
130  FORMAT(15X,A)
140  FORMAT(/,15X,A,$,A)
END

```

```

SUBROUTINE ADAPTIVE (NBLOC,MPONT,MPOLA,MBLOC,MDIVI,SED
1          ,CORDB,LNODE,NDIVX,NDIVY,NDIVZ,WEITX,WEITY,
2          WEITZ,GSED,BETA,ORIGB,MATNB,PILAS,
3          BASE11,BASE12,BASE21,BASE22,BASE31,
4          BASE32,BASA11,BASA12,BASA22,BASA21,VOL,
5          UTOTAL,VTOTAL)
DIMENSION SED(MBLOC),CORDB(MPONT,NDIME),LNODB(MBLOC,LNODE),
1          NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ(MBLOC),WEITX(MBLOC,
2          MDIVI),WEITY(MBLOC,MDIVI),WEITZ(MBLOC,MDIVI),ORIGB
3          (MPOLA,NDIME),MATNB(MBLOC),VOL(MBLOC)
COMMON/MODEL/NNODE,LNODE,NDIME
REAL GSED,BETA,UTOTAL,VTOTAL
INTEGER JBLOC,PILAS,BASE11,BASE12,BASE21,BASE22,BASE31,BASE32,
1          BASA11,BASA12,BASA22,BASA21
CHARACTER*74 FILE
CHARACTER*74 FILE3
*****
C          LECTURA DEL ARCHIVO EXISTENTE .GEN
C          *****
PILAS=1
WRITE(*,200)'DIRECTORIO Y NOMBRE DEL ARCHIVO(C:.GEN)='
READ(*,'(A)')FILE
OPEN (UNIT=1,FILE=FILE,ACCESS='SEQUENTIAL',STATUS='OLD')
READ (1,*)NBLOC,NPONT,NPOLA,NRENM
C
C          CHEQUEO DE LOS DATOS DE ENTRADA DEL MODELO
C
CALL CHECK (NBLOC,NPONT,NPOLA,MBLOC,MPONT,MPOLA,ISTOP)
C
C          PARAR SI SE CONSIGUE UN ERROR
C
IF (ISTOP.GT.0) STOP
C
C          LECTURA E IMPRESION DE LA CONECTIVIDAD DE LOS SUPERELEMENTOS Y
C          NUMERO DE MATERIALES
DO 61 IBLOC=1,NBLOC
JBLOC=IBLOC
READ (1,*) JBLOC,MATNB(JBLOC),(LNODB(JBLOC,INODE),INODE=1,NNODE)
61  CONTINUE
C          LECTURA E IMPRESION DEL ORIGEN DE LOS SISTEMAS AUXILIARES DE
C          COORDENADAS
C
DO 69 IPOLA=1,MPOLA
DO 69 IDIME=1,NDIME
ORIGB(IPOLA,IDIME)=0.0
69  CONTINUE
IF (NPOLA.GT.0) THEN
WRITE(*,1012)

```

```

DO 63 IPOLA=1,NPOLA
READ (1,*) JPOLA, (ORIGB(JPOLA, IDIME), IDIME=1, NDIME)
WRITE(*,1013) JPOLA, (ORIGB(JPOLA, IDIME), IDIME=1, NDIME)
63 CONTINUE
ENDIF
C
C
C INICIALIZACION DE TODAS LAS COORDENADAS ANTES DE LEER
C ALGUNA DE ELLAS
C
DO 64 IPONT=1,NPONT
DO 64 IDIME=1,NDIME
CORDB (IPONT, IDIME)=0.0
64 CONTINUE
C
C LECTURA DE LAS COORDENADAS , FINALIZANDO CON EL ULTIMO DE TODOS
C LOS NODOS
C
65 READ (1,*) JPOLA, JPONT, (CORDB(JPONT, IDIME), IDIME=1, NDIME)
C
C EVITA USAR TRANSFORMACION SI SE USA SISTEMA DE COORDENADAS
C CARTESIANAS
IF (JPOLA.GT.0.AND.JPOLA.LE.5) THEN
RADIUS=CORDB(JPONT, 1)
THETA =CORDB(JPONT, 2)*3.141592654/180.
CORDB(JPONT, 1)=RADIUS*COS(THETA)+ORIGB(JPOLA, 1)
CORDB(JPONT, 2)=RADIUS*SIN(THETA)+ORIGB(JPOLA, 2)
CORDB(JPONT, 3)=CORDB(JPONT, 3)+ORIGB(JPOLA, 3)
ELSE IF (JPOLA.GT.5.AND.JPOLA.LE.MPOLA) THEN
RADIUS=CORDB(JPONT, 1)
THETA =CORDB(JPONT, 2)*3.141592654/180.
PHI =CORDB(JPONT, 3)*3.141592654/180.
CORDB(JPONT, 1)=RADIUS*COS(THETA)*SIN(PHI)+ORIGB(JPOLA, 1)
CORDB(JPONT, 2)=RADIUS*SIN(THETA)*SIN(PHI)+ORIGB(JPOLA, 2)
CORDB(JPONT, 3)=RADIUS*COS(PHI)+ORIGB(JPOLA, 3)
ENDIF
IF (JPONT.NE.NPONT) GOTO 65
CLOSE (UNIT=1)
C
C INTERPOLACION DE LAS COORDENADAS DE LOS NODOS DEL CENTRO DEL BORDE
C
C CALL MDSIDE (CORDB, LNODB, MBLOC, MPONT, NBLOC, NPONT)
C *****
C CALCULO DEL VOLUMEN DE CADA ELEMENTO
C *****
DO 1 IBLOC=1,NBLOC
DO 2 INODE=1, 8
NM=LNODB(IBLOC, INODE)
IF (INODE.EQ.1) THEN
BASE11=CORDB(NM, 1)
BASA11=CORDB(NM, 2)
ELSE IF (INODE.EQ.2) THEN
BASE12=CORDB(NM, 1)
BASA12=CORDB(NM, 2)
BASE21=CORDB(NM, 2)
BASA21=CORDB(NM, 1)
ENDIF
IF (INODE.EQ.3) THEN
BASE22=CORDB(NM, 2)
BASA22=CORDB(NM, 1)
BASE31=CORDB(NM, 3)
ELSE IF (INODE.EQ.7) THEN
BASE32=CORDB(NM, 3)
ENDIF
2 CONTINUE
IF (BASE11.EQ.BASE12) GOTO 71
IF (BASE21.EQ.BASE22) GOTO 71
VOL(IBLOC)=ABS((BASE12-BASE11)*(BASE22-BASE21)*(BASE32-BASE31))
GOTO 72
71 VOL(IBLOC)=ABS((BASA12-BASA11)*(BASA22-BASA21)*(BASE32-BASE31))
72 WRITE(*, 3010) 'VOL(', IBLOC, ')=' , VOL(IBLOC)
READ(*,*)
3010 FORMAT(2X, A, I3, A, F12.4)
1 CONTINUE
WRITE(*,*) 'CALCULO EL VOLUMEN'
C *****
C LECTURA DE LOS VALORES DE ENERGIA DE DEFORMACION DE ABAQUS

```

```

C *****
WRITE(*,200)'DIRECTORIO Y NOMBRE DEL ARCHIVO SENER.IN(C:.IN)='
READ(*,'(A)')FILE3
OPEN (UNIT=2,FILE=FILE3,ACCESS='SEQUENTIAL',STATUS='OLD')
DO 3 I=1,NBLOC
DO 4 INODE=1,8
READ (2,*)A,B,SI
COUNT=COUNT+SI
4 CONTINUE
SED(I)=COUNT
WRITE(*,4001)'SED('I,')=',SED(I)
4001 FORMAT(3X,A,I3,A,F12.4)
3 CONTINUE
CLOSE (UNIT=2)
UTOTAL=0.
VTOTAL=0.
DO 5 I=1,NBLOC
UTOTAL=UTOTAL+(SED(I)*VOL(I))
VTOTAL=VTOTAL+VOL(I)
5 CONTINUE
GSED=UTOTAL/VTOTAL
WRITE(*,*)'INTRODUZCA EL VALOR DEL INDICADOR BETA='
READ(*,*)BETA
DO 7 I=1,NBLOC
WRITE(*,4000)'SED('I,')=',SED(I),'BETA=',BETA
4000 FORMAT(3X,A,I3,A,F12.4,A,F12.4)
IF (SED(I).GT.BETA*GSED) THEN
NDIVX(I)=2
NDIVY(I)=2
NDIVZ(I)=2
ELSE
NDIVX(I)=1
NDIVY(I)=1
NDIVZ(I)=1
ENDIF
DO 11 IDIVX=1,NDIVX(I)
WEITX(I,IDIVX)=1.0
11 CONTINUE
DO 12 IDIVY=1,NDIVY(I)
WEITY(I,IDIVY)=1.0
12 CONTINUE
DO 13 IDIVZ=1,NDIVZ(I)
WEITZ(I,IDIVZ)=1.0
13 CONTINUE
7 CONTINUE
5000 FORMAT(3X,A,I3)
RETURN

C
C          FORMATOS DE ENTRADA Y SALIDA
C
100  FORMAT(5X,70('**'),/,5X,'**',68X,'**',
1/,5X,'**',26X,'PROGRAMA GEMATRI',26X,'**',/,5X,'**',68X,'**',
2/,5X,'**',17X,'GENERADOR DE MALLA TRIDIMENSIONAL',18X,'**',/,
35X,'**',68X,'**',/,5X,'**',7X,'USANDO SUPERELEMENTOS E INDICES DE DEN
4SIDAD DE ENERGIA',7X,'**',/,5X,'**',68X,'**',/,5X,'**',19X,'AUTOR: ING.
5 CARLOS MELENDEZ D.',20X,'**',/,5X,'**',68X,'**',/,5X,'**',23X,'FECHA:
6SEPTIEMBRE 1997',24X,'**',/,5X,'**',68X,'**',/,5X,70('**'))
190  FORMAT(25(/),5X,'**',21X,'ENTRADA DE DATOS GENERALES',21X,'**',2(/))
200  FORMAT(5X,A,$,1X)
999  FORMAT(60(/),79('-'),32X,'PROGRAMA GEMATRI',
1//,23X,'GENERADOR DE MALLA TRIDIMENSIONAL',
2//,13X,'USANDO SUPERELEMENTOS E INDICES DE DENSIDAD DE ENERGIA',
3//,79('-'))
1000 FORMAT(//,3X,'**TITULO DEL PROBLEMA',//,3X,A70)
1001 FORMAT(//,3X,'**DEFINICION DE LOS SUPERELEMENTOS PRINCIPALES',
1//,5X,'NBLOC=',I2,5X,'NPOLA=',I2)
1012 FORMAT(//,3X,'** COORDENADA DE LOS SISTEMAS AUXILIARES',
1//,3X,'NUMERO X COORDENADA      Y COORDENADA      Z COORDENADA')
1013 FORMAT(1X,I4,6X,E12.4,5X,E12.4,5X,E12.4)
END

```

```

SUBROUTINE ARCHI (NBLOC,NPONT,CORDB,MATNB,LNOB,ORIGB,
1      NDIWX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,NBLOC,
2      MPONT,MDIVI,MPOLA,ELAST,CNABLA,NMATNB,NUPOLA,

```

```

3          CAMBIO,COORD,VALOR,SELEC,NABLA,ELASTIC,PILAS)
C
COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION CORDB(MPONT,NDIME),MATNB(MBLOC),LNODB(MBLOC,LNODE),
1          ORIGB(MPOLA,NDIME),NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ
2          (MBLOC),WEITX(MBLOC,MDIVI),ELAST(MBLOC),CNABLA(MBLOC),
3          WEITY(MBLOC,MDIVI),WEITZ(MBLOC,MDIVI)
INTEGER PILAS
REAL ELASTIC,NABLA
CHARACTER*70 TITLE
CHARACTER*25 FILE
C
-----
C          ESTA SUBROUTINA CREA LOS DATOS DE ENTRADA
C          Y LOS ALMACENA EN UN ARCHIVO .IN
C
-----
C          PILAS=0
C          *****
C          CARGAR DATOS DESDE ARCHIVO EXISTENTE
C          *****
WRITE(*,200)'DIRECTORIO Y NOMBRE DEL ARCHIVO(C:.IN)='
READ(*,'(A)')FILE
OPEN (UNIT=1,FILE=FILE,ACCESS='SEQUENTIAL',STATUS='OLD')
READ (1,'(A70)') TITLE
READ (1,*)NBLOC,NPONT,NPOLA,NRENM
C
C          CHEQUEO DE LOS DATOS DE ENTRADA DEL MODELO
C
CALL CHECK (NBLOC,NPONT,NPOLA,MBLOC,MPONT,MPOLA,ISTOP)
C
PARAR SI SE CONSIGUE UN ERROR
C
IF (ISTOP.GT.0) STOP
C
C          LECTURA E IMPRESION DE LA CONECTIVIDAD DE LOS SUPERELEMENTOS Y
C          NUMERO DE MATERIALES
DO 61 IBLOC=1,NBLOC
61  READ (1,*) JBLOC,MATNB(JBLOC),(LNODB(JBLOC,INODE),INODE=1,NNODE)
CONTINUE
62  DO 62 IBLOC=1,NBLOC
CONTINUE
C          LECTURA E IMPRESION DEL ORIGEN DE LOS SISTEMAS AUXILIARES DE
C          COORDENADAS
C
DO 69 IPOLA=1,MPOLA
DO 69 IDIME=1,NDIME
ORIGB(IPOLA,IDIME)=0.0
69  CONTINUE
IF (NPOLA.GT.0) THEN
DO 63 IPOLA=1,NPOLA
63  READ (1,*) JPOLA,(ORIGB(JPOLA,IDIME),IDIME=1,NDIME)
CONTINUE
ENDIF
C
C          INICIALIZACION DE TODAS LAS COORDENADAS ANTES DE LEER
C          ALGUNA DE ELLAS
C
DO 64 IPONT=1,NPONT
DO 64 IDIME=1,NDIME
64  CORDB (IPONT,IDIME)=0.0
CONTINUE
C
C          LECTURA DE LAS COORDENADAS , FINALIZANDO CON EL ULTIMO DE TODOS
C          LOS NODOS
C
65  READ (1,*) JPOLA,JPONT,(CORDB(JPONT,IDIME),IDIME=1,NDIME)
C
C          EVITA USAR TRANSFORMACION SI SE USA SISTEMA DE COORDENADAS
C          CARTESIANAS
IF (JPOLA.GT.0.AND.JPOLA.LE.5)THEN
RADIUS=CORDB(JPONT,1)
THETA =CORDB(JPONT,2)*3.141592654/180.
CORDB(JPONT,1)=RADIUS*COS(THETA)+ORIGB(JPOLA,1)
CORDB(JPONT,2)=RADIUS*SIN(THETA)+ORIGB(JPOLA,2)
CORDB(JPONT,3)=CORDB(JPONT,3)+ORIGB(JPOLA,3)
ELSE IF (JPOLA.GT.5.AND.JPOLA.LE.MPOLA) THEN

```

```

RADIUS=CORDB(JPONT,1)
THETA =CORDB(JPONT,2)*3.141592654/180.
PHI   =CORDB(JPONT,3)*3.141592654/180.
CORDB(JPONT,1)=RADIUS*COS(THETA)*SIN(PHI)+ORIGB(JPOLA,1)
CORDB(JPONT,2)=RADIUS*SIN(THETA)*SIN(PHI)+ORIGB(JPOLA,2)
CORDB(JPONT,3)=RADIUS*COS(PHI)+ORIGB(JPOLA,3)
ENDIF
IF (JPONT.NE.NPONT) GOTO 65

C
C   INTERPOLACION DE LAS COORDENADAS DE LOS NODOS DEL CENTRO DEL BORDE
C
CALL MDSIDE (CORDB,LNODE,MBLOC,MPONT,NBLOC,NPONT)
DO 66 IPONT=1,NPONT
66  CONTINUE
C
C   LEER E IMPRIMIR EL NUMERO DE DIVISIONES EN LA DIRECCION LOCAL
C       Y SU PESO RELATIVO
C
DO 67 IBLOC=1,NBLOC
JBLOC=IBLOC
READ (1,*) JBLOC,NDIVX(JBLOC),NDIVY(JBLOC),NDIVZ(JBLOC)
READ (1,*) (WEITX(JBLOC,IDIVX),IDIVX=1,NDIVX(JBLOC))
READ (1,*) (WEITY(JBLOC,IDIVY),IDIVY=1,NDIVY(JBLOC))
READ (1,*) (WEITZ(JBLOC,IDIVZ),IDIVZ=1,NDIVZ(JBLOC))
67  CONTINUE
CLOSE (UNIT=1)
RETURN

C
C           FORMATOS DE ENTRADA Y SALIDA
C
100  FORMAT(5X,70(' '),/,5X,' ',68X,' ',
1/ ,5X,' ',26X,'PROGRAMA GEMATRI',26X,' ',/,5X,' ',68X,' ',
2/ ,5X,' ',17X,'GENERADOR DE MALLA TRIDIMENSIONAL',18X,' ',/,
35X,' ',68X,' ',/,5X,' ',7X,'USANDO SUPERELEMENTOS E INDICES DE DEN
4SIDAD DE ENERGIA',7X,' ',/,5X,' ',68X,' ',/,5X,' ',19X,'AUTOR: ING.
5 CARLOS MELENDEZ D.',20X,' ',/,5X,' ',68X,' ',/,5X,' ',23X,'FECHA:
6SEPTIEMBRE 1997',24X,' ',/,5X,' ',68X,' ',/,5X,70(' '))
200  FORMAT(5X,A,$,1X)
190  FORMAT(25(/),5X,' ',21X,'ENTRADA DE DATOS GENERALES',21X,' ',2(/))
999  FORMAT(60(/),79('-'),32X,'PROGRAMA GEMATRI',
1//,23X,'GENERADOR DE MALLA TRIDIMENSIONAL',
2//,13X,'USANDO SUPERELEMENTOS E INDICES DE DENSIDAD DE ENERGIA',
3//,79('-'))
1000 FORMAT (//,3X,'**TITULO DEL PROBLEMA',//,3X,A70)
1001 FORMAT (//,3X,'**DEFINICION DE LOS SUPERELEMENTOS PRINCIPALES',
1//,5X,'NBLOC=',I2,5X,'NPOLA=',I2)
1002 FORMAT (//,3X,'** DEFINICION DE TOPOLOGIA DE LOS SUPERELEMENTOS')
1004 FORMAT (3X,'** COORDENADAS PUNTOS NODALES SUPERELEMENTO',
1//,3X,'NODO      X COORDENADA      Y COORDENADA      Z COORDENADA')
1005 FORMAT (1X,I4,6X,E12.4,5X,E12.4,5X,E12.4)
1012 FORMAT (//,3X,'** COORDENADA DE LOS SISTEMAS AUXILIARES',
1//,3X,'NUMERO X COORDENADA      Y COORDENADA      Z COORDENADA')
1013 FORMAT (1X,I4,6X,E12.4,5X,E12.4,5X,E12.4)
RETURN
END

C
C   SUBROUTINE DIMEN (MBLOC,MPONT,MDIVI,MREPN,MPOLA,MELEM,MPOIN)
C   COMMON/MODEL/NNODE,LNODE,NDIME
C
C   -----
C           ESTA SUBROUTINA CONTIENE LAS DIMENSIONES MAXIMAS
C           DE LOS ARREGLOS PRINCIPALES
C   -----
C
C   MBLOC=  NUMERO MAXIMO DE BLOQUES O SUPERELEMENTOS
C   MBLOC=60
C
C   MPONT =  NUMERO MAXIMO DE PUNTOS NODALES DE LOS SUPERELEMENTOS
C   MPONT=MBLOC*LNODE
C
C   MDIVI=  NUMERO MAXIMO DE DIVISIONES EN CADA DIRECCION LOCAL Y
C           PARA CADA SUPERELEMENTO O BLOQUE
C   MDIVI=30

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

C MREP= MAXIMO NUMERO DE NODOS REPETIDOS EN LA INTERACCION DE
C SUPERELEMENTOS
MREP=300

C
C MPOLA= MAXIMO NUMERO DE SISTEMA DE COORDENADAS AUXILIARES
C ( 1-5 COORDENADAS CILINDRICAS )
C ( 6-10 COORDENADAS ESFERICAS )
MPOLA=10

C
C MELEM= MAXIMO NUMERO DE ELEMENTOS GENERADOS EN EL MODELO
MELEM=600

C
C MPOIN= MAXIMO NUMERO DE PUNTOS NODALES GENERADOS EN EL MODELO
MPOIN=MELEM*NNODE
RETURN
END

```

```

SUBROUTINE GENMSH (NBLOC,NELEM,NPOIN,CORDB,MATNB,LNODE,NDIVX,
1 NDIVY,NDIVZ,WEITX,WEITY,WEITZ,LREP,N,LASOC,
2 LFINN,LFASC,COORD,MATNO,LNODS,MBLOC,MPONT,
3 MDIVI,MREP,MELEM,MPOIN,NL,ICON,NREN,ORIGN,
4 IPOL,NNCNEC,CBILI,CODASIS,CODASII,CODASDS,
5 CODASDI,NODOIS,NODOII,NODODI,NODODS,IPOLI,
6 NODII,NODIS,PLIL,CALII,CALID,IG)

```

```

COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION CORDB(MPONT,NDIME),MATNB(MBLOC),LNODS(MBLOC,LNODE),
1 NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ(MBLOC),WEITX(MBLOC,
2 MDIVI),WEITY(MBLOC,MDIVI),WEITZ(MBLOC,MDIVI),SHAPE(20)
3 ,LREP(NMREP),LASOC(MREP),LFINN(MREP),LFASC(MREP),
4 COORD(MPOIN,NDIME)
DIMENSION MATNO(MELEM),LNODS(MELEM,NNODE),NNCNEC(MPOIN),CBILI
1 (MPONT,NDIME),CODASIS(MPONT,NDIME),CODASII(MPONT,NDIME)
2 ,CODASDS(MPONT,NDIME),CODASDI(MPONT,NDIME),PLIL(MPONT,
3 NDIME),CALII(MPONT,NDIME),CALID(MPONT,NDIME)
INTEGER IPOL(MPOIN),NODOIS(MPOIN),NODOII(MPOIN),NODODI(MPOIN),
1 NODODS(MPOIN),IPOLI(MPOIN),NODII(MPOIN),NODIS(MPOIN)
INTEGER ORIGN

```

```

C -----
C ESTA SUBROUTINA GENERA UNA MALLA DE ELEMENTOS HEXAHEDRICOS
C TRIDIMENSIONALES DENTRO DE CADA SUPERELEMENTO
C -----

```

```

NPOIN=0
NELEM=0

```

```

C INICIO DE VARIABLES
C

```

```

DO 1 IREP=1,MREP
LREP(IREP)=0
1 CONTINUE
DO 2 IPOIN=1,MPOIN
DO 2 IDIME=1,NDIME
COORD(IPOIN,IDIME)=0.0
2 CONTINUE
DO 3 IELEM=1,MELEM
MATNO(IELEM)=0
DO 3 INODE=1,NNODE
LNODS(IELEM,INODE)=0
3 CONTINUE
C
C

```

```

C PROCESO SOBRE CADA ELEMENTO
C
C

```

```

DO 4 IBLOC=1,NBLOC
C
C

```

```

C ESTABLECER EL PESO RELATIVO PARA CADA DIRECCION LOCAL
C

```

```

TOTAL=0.0
DO 5 IDIVX=1,NDIVX(IBLOC)
IF (WEITX(IBLOC,IDIVX).EQ.0.0) WEITX(IBLOC,IDIVX)=1.0
TOTAL=TOTAL+WEITX(IBLOC,IDIVX)
5 CONTINUE
XNORM=2.0/TOTAL
TOTAL=0.0
DO 6 IDIVY=1,NDIVY(IBLOC)
IF (WEITY(IBLOC,IDIVY).EQ.0.0) WEITY(IBLOC,IDIVY)=1.0
TOTAL=TOTAL+WEITY(IBLOC,IDIVY)

```

```

6      CONTINUE
      YNORM=2.0/TOTAL
      TOTAL=0.0
      DO 7 IDIVZ=1,NDIVZ (IBLOC)
      IF (WEITZ (IBLOC, IDIVZ) .EQ. 0.0) WEITZ (IBLOC, IDIVZ)=1.0
      TOTAL=TOTAL+WEITZ (IBLOC, IDIVZ)
7      CONTINUE
4002   FORMAT (3X, A, F12.4)
      ZNORM=2.0/TOTAL

C
C      NUMERO DE PUNTOS NODALES A SER GENERADOS EN CADA DIRECCION LOCAL
C
      NXNNP=NDIVX (IBLOC)+1
      NYNNP=NDIVY (IBLOC)+1
      NZNNP=NDIVZ (IBLOC)+1

C
C      INICIO DE GENERACION DE MALLA DESDE LAS COORDENADAS LOCALES (-1,-1,-1)
C
      IASEZ=0
      EPSIP=-1.0
      KWETZ=0
      DO 8 IZNNP=1,NZNNP
      IASEZ=IASEZ+1
      IF (IASEZ.EQ.3) IASEZ=2
      IASEY=0
      ETASP=-1.0
      KWETY=0
      DO 9 IYNNP=1,NYNNP
      IASEY=IASEY+1
      IF (IASEY.EQ.3) IASEY=2
      IASEX=0
      EXISP=-1.0
      KWETX=0
      DO 10 IXNNP=1,NXNNP
      IASEX=IASEX+1
      IF (IASEX.EQ.3) IASEX=2

C
C      CALCULO DE LAS COORDENADAS X,Y,Z DE LAS COORDENADAS LOCALES
C      DEL SUPERELEMENTO
C
      CALL SHP20 (EXISP,ETASP,EPSIP,SHAPE)
      NPOIN=NPOIN+1
      DO 11 INODE=1,LNODE
      JTEMP=LNODB (IBLOC, INODE)
      DO 11 IDIME=1,NDIME
      COORD (NPOIN, IDIME)=COORD (NPOIN, IDIME)+
1      SHAPE (INODE) *CORDB (JTEMP, IDIME)
11     CONTINUE

C
C      CONSTRUCCION DE LA DEFINICION TOPOLOGICA DE CADA ELEMENTO
C
      IF (IASEX.NE.2.OR.IASEY.NE.2.OR.IASEZ.NE.2) GOTO 12
      NELEM=NELEM+1
      KPOIN=NPOIN-NXNNP*NYNNP-NXNNP
      JPOIN=NPOIN-NXNNP*NYNNP
      LPOIN=NPOIN-NXNNP
      LNODS (NELEM, 1)=KPOIN-1
      LNODS (NELEM, 2)=KPOIN
      LNODS (NELEM, 3)=JPOIN
      LNODS (NELEM, 4)=JPOIN-1
      LNODS (NELEM, 5)=LPOIN-1
      LNODS (NELEM, 6)=LPOIN
      LNODS (NELEM, 7)=NPOIN
      LNODS (NELEM, 8)=NPOIN-1
      MATNO (NELEM)=MATNB (IBLOC)
12     CONTINUE

C
C      EVALUACION DE LAS COORDENADAS DE LOS PUNTOS NODALES
C
      KWETX=KWETX+1
10     EXISP=EXISP+YNORM*WEITX (IBLOC, KWETX)
      KWETY=KWETY+1
9      ETASP=ETASP+YNORM*WEITY (IBLOC, KWETY)
      KWETZ=KWETZ+1

```

```

8     EPSIP=EPSIP+ZNORM*WEITZ (IBLOC,KWETZ)
4     CONTINUE
C
C     ELIMINACION DE LOS NODOS REPETIDOS
C
      NREP=0
      DO 13 IPOIN=1,NPOIN
      IF (NREP.EQ.0)GOTO 14
      DO 15 IREP=1,NREP
      IF (IPOIN.EQ.LREP(IREP)) GOTO 13
15    CONTINUE
14    CONTINUE
      LPOIN=IPOIN+1
C
C     CHEQUEO DE LAS COORDENADAS DE LOS PUNTOS NODALES
C
      DO 16 JPONT=LPOIN,NPOIN
      DX=ABS(COORD(IPOIN,1)-COORD(JPONT,1))
      DY=ABS(COORD(IPOIN,2)-COORD(JPONT,2))
      DZ=ABS(COORD(IPOIN,3)-COORD(JPONT,3))
      IF (DX.GT.1.0E-3.OR.DY.GT.1.0E-3.OR.DZ.GT.1.0E-3) GOTO 16
      NREP=NREP+1
C
C     ALMACENAMIENTO DE LOS PUNTOS NODALES DE IGUAL COORDENADAS
C
      LREP(NREP)=JPONT
      LASOC(NREP)=IPOIN
16    CONTINUE
13    CONTINUE
      IF (NREP.EQ.0) GOTO 35
C
C     TRABAJO DE ELIMINACION
C
      INDEX=0
      DO 17 IPOIN=1,NPOIN
      DO 18 IREP=1,NREP
      IF (LREP(IREP).EQ.IPOIN) GOTO 19
18    CONTINUE
      GOTO 17
19    INDEX=INDEX+1
      LFINN(INDEX)=LREP(IREP)
      LFASC(INDEX)=LASOC(IREP)
17    CONTINUE
      DO 20 IREP=1,NREP
      LREP(IREP)=LFINN(IREP)
      LASOC(IREP)=LFASC(IREP)
20    CONTINUE
      DO 21 IREP=1,NREP
      DO 21 IELEM=1,NELEM
      DO 21 INODE=1,NNODE
      IF (LNODS(IELEM,INODE).EQ.LREP(IREP))
1     LNODS(IELEM,INODE)=LASOC(IREP)
21    CONTINUE
      DO 22 IPOIN=1,NPOIN
      DO 23 IREP=1,NREP
      IF (IPOIN.EQ.LREP(IREP)) GOTO 22
23    CONTINUE
      IF (IPOIN.LT.LREP(1)) GOTO 22
      IDIFF=IPOIN-NREP
      IF (IPOIN.GT.LREP(NREP)) GOTO 24
      DO 25 IREP=1,NREP
      KREP=NREP-IREP+1
      IF (IPOIN.LT.LREP(KREP)) IDIFF=IPOIN-KREP+1
25    CONTINUE
24    CONTINUE
      DO 26 IDIME=1,NDIME
      COORD(IDIFF,IDIME)=COORD(IPOIN,IDIME)
26    CONTINUE
22    CONTINUE
      DO 27 IELEM=1,NELEM
      DO 27 INODE=1,NNODE
      NPOSI=LNODS(IELEM,INODE)
      DO 28 IREP=1,NREP
      IF (NPOSI.EQ.LREP(IREP)) GOTO 27
28    CONTINUE
      IF (NPOSI.LT.LREP(1)) GOTO 27

```

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

IDIFF=NPOSI-NREPN
IF (NPOSI.GT.LREPN(NREPN)) GOTO 29
DO 30 IREP=1,NREPN
KREP=NREPN-IREP+1
IF (NPOSI.LT.LREPN(KREP)) IDIFF=NPOSI-KREP+1
30 CONTINUE
29 LNODS(IELEM,INODE)=IDIFF
27 CONTINUE
C
C   NUMERACION DE LOS PUNTOS NODALES DESPUES DE LA ELIMINACION DE
C   LOS NODOS REPETIDOS
C
C   NPOIN=NPOIN-NREPN
C
C   IDENTIFICACION DE LOS NODOS LINEALES
C
C
35 DO 521 I=1,IG
DO 531 IPOIN=1,NPOIN
NL=0
KPOIN=IPOIN
IF (NRENM.EQ.1.AND.ORIGN.EQ.1) KPOIN=NNCNEC(IPOIN)
IF (PLIL(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (PLIL(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (PLIL(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN
IPOLI(I)=KPOIN
ENDIF
NL=0.
IF (CALII(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CALII(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CALII(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN
NODII(I)=KPOIN
ENDIF
NL=0.
IF (CALID(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CALID(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CALID(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN
NODIS(I)=KPOIN
ENDIF
531 CONTINUE
521 CONTINUE
WRITE(*,140)'ESTOS SON LINEALES Y SUS NODOS VECINOS'
DO 331 I=1,IG
WRITE(*,100)'NODO LINEAL(' ,I,')=' ,IPOLI(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODII(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODIS(I)
READ(*,*)
331 CONTINUE

C
C   IDENTIFICACION DE LOS NODOS BILINEALES
C
C
DO 52 I=1,ICON
DO 53 IPOIN=1,NPOIN
NL=0
KPOIN=IPOIN
IF (NRENM.EQ.1.AND.ORIGN.EQ.1) KPOIN=NNCNEC(IPOIN)
IF (CBILI(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CBILI(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CBILI(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN
IPOL(I)=KPOIN
ENDIF
NL=0.
IF (CODASII(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASII(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASII(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN
NODOII(I)=KPOIN
ENDIF
NL=0.
IF (CODASIS(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASIS(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASIS(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3) THEN

```

WWW.BDIGITAL.ULA.VE

```

NODOIS(I)=KPOIN
ENDIF
NL=0.
IF (CODASDI(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASDI(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASDI(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
NODODI(I)=KPOIN
ENDIF
NL=0.
IF (CODASDS(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASDS(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASDS(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
NODODS(I)=KPOIN
ENDIF
53 CONTINUE
52 CONTINUE
WRITE(*,140)'ESTOS SON BILINEALES Y SUS NODOS VECINOS'
DO 33 I=1,ICON
WRITE(*,100)'NODO BILINEAR(' ,I,')=' ,IPOL(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODOIS(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODODI(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODODS(I)
WRITE(*,130)'NODO INFERIOR IZQUIERDO=' ,NODOII(I)
READ(*,*)
33 CONTINUE
100 FORMAT(2X,A,I3,A,I5)
130 FORMAT(1X,A,I3)
140 FORMAT(//,2X,A)
RETURN
END

SUBROUTINE INCOMP (NBLOC,NDIVX,CORDB,POBIL,LNODE,NCOR,TOTX,
1 TOTY,TOTZ,MPONT,NM,MBLOC,ICON,INCON,PNOBIL,
2 NNCOR,CBILI,CODAS,CODASIS,CODASII,
3 CODASDS,CODASDI,COLMBD,COLMBR,COLMBU,COLMBL,
4 COBIL,COBIR,COBRD,COBRU,COBUL,COBUR,COBLU,
5 COBLD,PLIL,CALII,CALID,CHEQ,SALTO,IG,ICONTROL1
6 ,ICONTROL2,ICONTROL3,ICONTROL4)
COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION NDI VX(MBLOC),CORDB(MPONT,NDIME),POBIL(MPONT,NDIME),
1 LNODB(MBLOC,LNODE),NCOR(MPONT,NDIME),NNCOR(MPONT,NDIME)
2 ,PNOBIL(MPONT,NDIME),CBILI(MPONT,NDIME),
3 CODAS(MPONT,NNODE,NDIME),CODASIS(MPONT,NDIME),
4 CODASII(MPONT,NDIME),CODASDS(MPONT,NDIME),
5 CODASDI(MPONT,NDIME),COLMBD(MPONT,NDIME)
6 ,COLMBR(MPONT,NDIME),COLMBU(MPONT,NDIME)
DIMENSION COLMBL(MPONT,NDIME),COBIL(MPONT,NDIME),COBIR(MPONT,
1 NDIME),COBRD(MPONT,NDIME),COBRU(MPONT,NDIME),COBUL
2 (MPONT,NDIME),COBUR(MPONT,NDIME),COBLU(MPONT,NDIME)
3 ,COBLD(MPONT,NDIME),PLIL(MPONT,NDIME),CALII(MPONT,
4 NDIME),CALID(MPONT,NDIME),CHEQ(NDIME),SALTO(4,4)
REAL TOTX,TOTY,TOTZ
INTEGER NM,ICON,INCON,ICONTROL1(MPONT),ICONTROL2(MPONT),
1 ICONTROL3(MPONT),ICONTROL4(MPONT)
C -----
C IDENTIFICACION DE LAS CARAS
C -----
SALTO(1,1)=1
SALTO(1,2)=2
SALTO(1,3)=6
SALTO(1,4)=5
SALTO(2,1)=3
SALTO(2,2)=7
SALTO(2,3)=6
SALTO(2,4)=2
SALTO(3,1)=4
SALTO(3,2)=8
SALTO(3,3)=7
SALTO(3,4)=3
SALTO(4,1)=1
SALTO(4,2)=5
SALTO(4,3)=8
SALTO(4,4)=4
C *****

```

```

C
C      ESTA SUBROUTINA LOCALIZA LOS NODOS CON POSIBILIDAD DE SER
C      INCOMPATIBLES EN ELEMENTOS DIVIDIDOS Y NO DIVIDIDOS
C      *****
IM=0
DO 22 INODE=1, LNODE
DO 23 IDIME=1, NDIME
NCOR(INODE, IDIME)=0.0
23 CONTINUE
22 CONTINUE
DO 1 IBLOC=1, NBLOC
JBLOC=IBLOC
NM=NDIVX(JBLOC)
C
C      -----
C      CALCULO DE LAS COORDENADAS DE LOS NODOS EN EL CENTRO DE CADA CARA
C      SOLO DE LOS ELEMENTOS QUE SERAN DIVIDIDOS
C      -----
IF (NM.EQ.1) GOTO 31
DO 8 INODE=1, 8
NM=LNODB(IBLOC, INODE)
NCOR(INODE, 1)=CORDB(NM, 1)
NCOR(INODE, 2)=CORDB(NM, 2)
NCOR(INODE, 3)=CORDB(NM, 3)
8 CONTINUE
C      COORDENADA NODO CENTROIDAL CARA UNO
TOTX=0.
TOTY=0.
TOTZ=0.
DO 2 INODE=1, 4
TOTX=TOTX+NCOR(INODE, 1)
TOTY=TOTY+NCOR(INODE, 2)
TOTZ=TOTZ+NCOR(INODE, 3)
2 CONTINUE
IM=IM+1
POBIL(IM, 1)=TOTX/4.
POBIL(IM, 2)=TOTY/4.
POBIL(IM, 3)=TOTZ/4.
IR=0.
DO 70 INODE=1, 4
IR=IR+1
CODAS(IM, IR, 1)=NCOR(INODE, 1)
CODAS(IM, IR, 2)=NCOR(INODE, 2)
CODAS(IM, IR, 3)=NCOR(INODE, 3)
70 CONTINUE
C      COORDENADA NODO CENTROIDAL CARA DOS
TOTX=0.
TOTY=0.
TOTZ=0.
DO 3 INODE=5, 8
TOTX=TOTX+NCOR(INODE, 1)
TOTY=TOTY+NCOR(INODE, 2)
TOTZ=TOTZ+NCOR(INODE, 3)
3 CONTINUE
IM=IM+1
POBIL(IM, 1)=TOTX/4.
POBIL(IM, 2)=TOTY/4.
POBIL(IM, 3)=TOTZ/4.
IR=0.
DO 72 INODE=5, 8
IR=IR+1
CODAS(IM, IR, 1)=NCOR(INODE, 1)
CODAS(IM, IR, 2)=NCOR(INODE, 2)
CODAS(IM, IR, 3)=NCOR(INODE, 3)
72 CONTINUE
C      COORDENADA NODO CENTROIDAL CARA TRES
TOTX=0.
TOTY=0.
TOTZ=0.
J=1
DO 4 INODE=1, 4
NM=SALTO(J, INODE)
TOTX=TOTX+NCOR(NM, 1)
TOTY=TOTY+NCOR(NM, 2)
TOTZ=TOTZ+NCOR(NM, 3)
4 CONTINUE
IM=IM+1

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

POBIL(IM,1)=TOTX/4.
POBIL(IM,2)=TOTY/4.
POBIL(IM,3)=TOTZ/4.
IR=0.
DO 74 INODE=1,4
NM=SALTO(J,INODE)
IR=IR+1
CODAS(IM,IR,1)=NCOR(NM,1)
CODAS(IM,IR,2)=NCOR(NM,2)
CODAS(IM,IR,3)=NCOR(NM,3)
74 CONTINUE
C COORDENADA NODO CENTROIDAL CARA CUATRO
TOTX=0.
TOTY=0.
TOTZ=0.
J=2
DO 5 INODE=1,4
NM=SALTO(J,INODE)
TOTX=TOTX+NCOR(NM,1)
TOTY=TOTY+NCOR(NM,2)
TOTZ=TOTZ+NCOR(NM,3)
5 CONTINUE
IM=IM+1
POBIL(IM,1)=TOTX/4.
POBIL(IM,2)=TOTY/4.
POBIL(IM,3)=TOTZ/4.
IR=0.
DO 76 INODE=1,4
NM=SALTO(J,INODE)
IR=IR+1
CODAS(IM,IR,1)=NCOR(NM,1)
CODAS(IM,IR,2)=NCOR(NM,2)
CODAS(IM,IR,3)=NCOR(NM,3)
76 CONTINUE
C COORDENADA NODO CENTROIDAL CARA CINCO
TOTX=0.
TOTY=0.
TOTZ=0.
J=3
DO 6 INODE=1,4
NM=SALTO(J,INODE)
TOTX=TOTX+NCOR(NM,1)
TOTY=TOTY+NCOR(NM,2)
TOTZ=TOTZ+NCOR(NM,3)
6 CONTINUE
IM=IM+1
POBIL(IM,1)=TOTX/4.
POBIL(IM,2)=TOTY/4.
POBIL(IM,3)=TOTZ/4.
IR=0.
DO 78 INODE=1,4
NM=SALTO(J,INODE)
IR=IR+1
CODAS(IM,IR,1)=NCOR(NM,1)
CODAS(IM,IR,2)=NCOR(NM,2)
CODAS(IM,IR,3)=NCOR(NM,3)
78 CONTINUE
C COORDENADA NODO CENTROIDAL CARA SEIS
TOTX=0.
TOTY=0.
TOTZ=0.
J=4
DO 7 INODE=1,4
NM=SALTO(J,INODE)
TOTX=TOTX+NCOR(NM,1)
TOTY=TOTY+NCOR(NM,2)
TOTZ=TOTZ+NCOR(NM,3)
7 CONTINUE
IM=IM+1
POBIL(IM,1)=TOTX/4.
POBIL(IM,2)=TOTY/4.
POBIL(IM,3)=TOTZ/4.
IR=0.
DO 80 INODE=1,4
NM=SALTO(J,INODE)
IR=IR+1

```

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

      CODAS(IM,IR,1)=NCOR(NM,1)
      CODAS(IM,IR,2)=NCOR(NM,2)
      CODAS(IM,IR,3)=NCOR(NM,3)
80  CONTINUE
      GOTO 1
C -----
C          CALCULO DE LAS COORDENADAS DE LOS NODOS EN EL CENTRO DE CADA CARA
C          SOLO DE LOS ELEMENTOS QUE NO SERAN DIVIDIDOS
C -----
31  DO 11 INODE=1,8
      NM=LNOB( IBLOC, INODE)
      NNCOR(INODE,1)=CORDB(NM,1)
      NNCOR(INODE,2)=CORDB(NM,2)
      NNCOR(INODE,3)=CORDB(NM,3)
11  CONTINUE
      INK=INK+1
C          COORDENADA NODO CENTROIDAL CARA UNO
      TOTX=0.
      TOTY=0.
      TOTZ=0.
      DO 12 INODE=1,4
      TOTX=TOTX+NNCOR(INODE,1)
      TOTY=TOTY+NNCOR(INODE,2)
      TOTZ=TOTZ+NNCOR(INODE,3)
12  CONTINUE
      IIM=IIM+1
      PNOBIL(IIM,1)=TOTX/4.
      PNOBIL(IIM,2)=TOTY/4.
      PNOBIL(IIM,3)=TOTZ/4.
C          COORDENADA NODO CENTROIDAL CARA DOS
      TOTX=0.
      TOTY=0.
      TOTZ=0.
      DO 13 INODE=5,8
      TOTX=TOTX+NNCOR(INODE,1)
      TOTY=TOTY+NNCOR(INODE,2)
      TOTZ=TOTZ+NNCOR(INODE,3)
13  CONTINUE
      IIM=IIM+1
      PNOBIL(IIM,1)=TOTX/4.
      PNOBIL(IIM,2)=TOTY/4.
      PNOBIL(IIM,3)=TOTZ/4.
C          COORDENADA NODO CENTROIDAL CARA TRES
      TOTX=0.
      TOTY=0.
      TOTZ=0.
      J=1
      DO 14 INODE=1,4
      NM=SALTO(J, INODE)
      TOTX=TOTX+NNCOR(NM,1)
      TOTY=TOTY+NNCOR(NM,2)
      TOTZ=TOTZ+NNCOR(NM,3)
14  CONTINUE
      IIM=IIM+1
      PNOBIL(IIM,1)=TOTX/4.
      PNOBIL(IIM,2)=TOTY/4.
      PNOBIL(IIM,3)=TOTZ/4.
C          COORDENADA NODO CENTROIDAL CARA CUATRO
      TOTX=0.
      TOTY=0.
      TOTZ=0.
      J=2
      DO 15 INODE=1,4
      NM=SALTO(J, INODE)
      TOTX=TOTX+NNCOR(NM,1)
      TOTY=TOTY+NNCOR(NM,2)
      TOTZ=TOTZ+NNCOR(NM,3)
15  CONTINUE
      IIM=IIM+1
      PNOBIL(IIM,1)=TOTX/4.
      PNOBIL(IIM,2)=TOTY/4.
      PNOBIL(IIM,3)=TOTZ/4.
C          COORDENADA NODO CENTROIDAL CARA CINCO
      TOTX=0.
      TOTY=0.
      TOTZ=0.

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

J=3
DO 16 INODE=1,4
NM=SALTO(J,INODE)
TOTX=TOTX+NNCOR(NM,1)
TOTY=TOTY+NNCOR(NM,2)
TOTZ=TOTZ+NNCOR(NM,3)
16 CONTINUE
IIM=IIM+1
PNOBIL(IIM,1)=TOTX/4.
PNOBIL(IIM,2)=TOTY/4.
PNOBIL(IIM,3)=TOTZ/4.
C COORDENADA NODO CENTROIDAL CARA SEIS
TOTX=0.
TOTY=0.
TOTZ=0.
J=4
DO 17 INODE=1,4
NM=SALTO(J,INODE)
TOTX=TOTX+NNCOR(NM,1)
TOTY=TOTY+NNCOR(NM,2)
TOTZ=TOTZ+NNCOR(NM,3)
17 CONTINUE
IIM=IIM+1
PNOBIL(IIM,1)=TOTX/4.
PNOBIL(IIM,2)=TOTY/4.
PNOBIL(IIM,3)=TOTZ/4.
1 CONTINUE
ICON=IM
INCON=IIM
IM=0
DO 113 I=1,ICON
INODE=0.
DO 51 J=1,INCON
NL=0
IF (POBIL(I,1).EQ.PNOBIL(J,1)) NL=NL+1
IF (POBIL(I,2).EQ.PNOBIL(J,2)) NL=NL+1
IF (POBIL(I,3).EQ.PNOBIL(J,3)) NL=NL+1
IF (NL.EQ.3) THEN
IM=IM+1
CBILI(IM,1)=POBIL(I,1)
CBILI(IM,2)=POBIL(I,2)
CBILI(IM,3)=POBIL(I,3)
WRITE(*,*)'*****NODO BILINEAL*****'
WRITE(*,5000)'CBILI(',IM,',1)=' ,CBILI(IM,1)
WRITE(*,5000)'CBILI(',IM,',2)=' ,CBILI(IM,2)
WRITE(*,5000)'CBILI(',IM,',3)=' ,CBILI(IM,3)
5000 FORMAT(4X,A,I3,A,F12.4)
C NODO ASOCIADO UNO
WRITE(*,*)'****NODOS EXTREMOS DEL BILINEAL****'
DO 90 IDIME=1,NDIME
CODASII(IM,IDIME)=CODAS(I,1,IDIME)
WRITE(*,6000)'CODASII(',IM,',',IDIME,')=' ,CODASII(IM,IDIME)
90 CONTINUE
DO 91 IDIME=1,NDIME
CODASIS(IM,IDIME)=CODAS(I,2,IDIME)
WRITE(*,6000)'CODASIS(',IM,',',IDIME,')=' ,CODASIS(IM,IDIME)
91 CONTINUE
DO 92 IDIME=1,NDIME
CODASDI(IM,IDIME)=CODAS(I,3,IDIME)
WRITE(*,6000)'CODASDI(',IM,',',IDIME,')=' ,CODASDI(IM,IDIME)
92 CONTINUE
DO 93 IDIME=1,NDIME
CODASDS(IM,IDIME)=CODAS(I,4,IDIME)
WRITE(*,6000)'CODASDS(',IM,',',IDIME,')=' ,CODASDS(IM,IDIME)
93 CONTINUE
ENDIF
51 CONTINUE
113 CONTINUE
READ(*,*)
6000 FORMAT(3X,A,I3,A,I3,A,F12.4)
100 FORMAT(2X,A,I3,A,F12.4)
C
C -----
C BUSQUEDA DE LAS COORDENADAS DE LOS NODOS LINEALES
C -----
DO 63 I=1,ICON
IF (CODASII(I,1).EQ.CODASIS(I,1).AND.CODASDI(I,1).EQ.CODASDS(I,1))

```

```

1   NL=NL+1
   IF (CODASII(I,1).EQ.CODASDI(I,1).AND.CODASIS(I,1).EQ.CODASDS(I,1))
1   NL=NL+1
   IF (CODASII(I,1).EQ.CODASDS(I,1).AND.CODASIS(I,1).EQ.CODASDI(I,1))
1   NL=NL+1
   IF (NL.NE.3) GOTO 1233
   IF (CODASII(I,2).NE.CODASIS(I,2)) THEN
   ICONTROL1(I)=4
   ICONTROL2(I)=2
   ICONTROL3(I)=5
   ICONTROL4(I)=6
   GOTO 63
   ENDIF
   ICONTROL1(I)=1
   ICONTROL2(I)=2
   ICONTROL3(I)=3
   ICONTROL4(I)=7
   GOTO 63
1233 IF (CODASII(I,2).EQ.CODASIS(I,2).AND.CODASII(I,2).EQ.CODASII(I,2))
1   NL=NL+1
   IF (CODASII(I,2).EQ.CODASDI(I,2).AND.CODASIS(I,2).EQ.CODASDS(I,2))
1   NL=NL+1
   IF (CODASII(I,2).EQ.CODASDS(I,2).AND.CODASIS(I,2).EQ.CODASDI(I,2))
1   NL=NL+1
   IF (NL.NE.3) GOTO 1234
   IF (CODASII(I,2).NE.CODASIS(I,2)) THEN
   ICONTROL1(I)=1
   ICONTROL2(I)=2
   ICONTROL3(I)=8
   ICONTROL4(I)=9
   GOTO 63
   ENDIF
   ICONTROL1(I)=1
   ICONTROL2(I)=2
   ICONTROL3(I)=3
   ICONTROL4(I)=10
   GOTO 63
1234 ICONTROL1(I)=1
   ICONTROL2(I)=2
   ICONTROL3(I)=8
   ICONTROL4(I)=9
63  CONTINUE
C   LINEAL DE BORDE INFERIOR
   ICON=IM
   DO 53 I=1,ICON
   IF (ICONTROL1(I).EQ.1) THEN
   COLMBD(I,1)=(CODASII(I,1)+CODASDS(I,1))/2.
   COLMBD(I,2)=(CODASII(I,2)+CODASDS(I,2))/2.
   COLMBD(I,3)=(CODASII(I,3)+CODASDS(I,3))/2.
   COBIL(I,1)=CODASII(I,1)
   COBIL(I,2)=CODASII(I,2)
   COBIL(I,3)=CODASII(I,3)
   COBIR(I,1)=CODASDS(I,1)
   COBIR(I,2)=CODASDS(I,2)
   COBIR(I,3)=CODASDS(I,3)
   GOTO 153
   ENDIF
   COLMBD(I,1)=(CODASII(I,1)+CODASIS(I,1))/2.
   COLMBD(I,2)=(CODASII(I,2)+CODASIS(I,2))/2.
   COLMBD(I,3)=(CODASII(I,3)+CODASIS(I,3))/2.
   COBIL(I,1)=CODASII(I,1)
   COBIL(I,2)=CODASII(I,2)
   COBIL(I,3)=CODASII(I,3)
   COBIR(I,1)=CODASIS(I,1)
   COBIR(I,2)=CODASIS(I,2)
   COBIR(I,3)=CODASIS(I,3)
153 WRITE(*,*)'*****LINEAL BORDE INFERIOR*****'
   WRITE(*,5000)'COLMBD(' ,I, '1)'=',COLMBD(I,1)
   WRITE(*,5000)'COLMBD(' ,I, '2)'=',COLMBD(I,2)
   WRITE(*,5000)'COLMBD(' ,I, '3)'=',COLMBD(I,3)
53  CONTINUE
C   LINEAL DE BORDE DERECHO
   DO 54 J=1,ICON
   IF (ICONTROL3(J).EQ.8) THEN
   COLMBR(J,1)=(CODASIS(J,1)+CODASII(J,1))/2.
   COLMBR(J,2)=(CODASIS(J,2)+CODASII(J,2))/2.

```

```

COLMBR(J,3)=(CODASIS(J,3)+CODASII(J,3))/2.
COBRD(J,1)=CODASIS(J,1)
COBRD(J,2)=CODASIS(J,2)
COBRD(J,3)=CODASIS(J,3)
COBRU(J,1)=CODASII(J,1)
COBRU(J,2)=CODASII(J,2)
COBRU(J,3)=CODASII(J,3)
GOTO 154
ENDIF
IF (ICONTROL3(J).EQ.5) THEN
COLMBR(J,1)=(CODASDI(J,1)+CODASIS(J,1))/2.
COLMBR(J,2)=(CODASDI(J,2)+CODASIS(J,2))/2.
COLMBR(J,3)=(CODASDI(J,3)+CODASIS(J,3))/2.
COBRD(J,1)=CODASDI(J,1)
COBRD(J,2)=CODASDI(J,2)
COBRD(J,3)=CODASDI(J,3)
COBRU(J,1)=CODASIS(J,1)
COBRU(J,2)=CODASIS(J,2)
COBRU(J,3)=CODASIS(J,3)
GOTO 154
ENDIF
COLMBR(J,1)=(CODASDI(J,1)+CODASDS(J,1))/2.
COLMBR(J,2)=(CODASDI(J,2)+CODASDS(J,2))/2.
COLMBR(J,3)=(CODASDI(J,3)+CODASDS(J,3))/2.
COBRD(J,1)=CODASDI(J,1)
COBRD(J,2)=CODASDI(J,2)
COBRD(J,3)=CODASDI(J,3)
COBRU(J,1)=CODASDS(J,1)
COBRU(J,2)=CODASDS(J,2)
COBRU(J,3)=CODASDS(J,3)
154 WRITE(*,*)'*****LINEAL BORDE DERECHO*****'
WRITE(*,5000)'COLMBR(' ,J,'1)'=',COLMBR(J,1)
WRITE(*,5000)'COLMBR(' ,J,'2)'=',COLMBR(J,2)
WRITE(*,5000)'COLMBR(' ,J,'3)'=',COLMBR(J,3)
54 CONTINUE
C
LINEAL DE BORDE SUPERIOR
DO 55 I=1,ICON
IF (ICONTROL2(I).EQ.2) THEN
COLMBU(I,1)=(CODASDI(I,1)+CODASIS(I,1))/2.
COLMBU(I,2)=(CODASDI(I,2)+CODASIS(I,2))/2.
COLMBU(I,3)=(CODASDI(I,3)+CODASIS(I,3))/2.
COBUL(I,1)=CODASDI(I,1)
COBUL(I,2)=CODASDI(I,2)
COBUL(I,3)=CODASDI(I,3)
COBUR(I,1)=CODASIS(I,1)
COBUR(I,2)=CODASIS(I,2)
COBUR(I,3)=CODASIS(I,3)
GOTO 155
ENDIF
COLMBU(I,1)=(CODASDI(I,1)+CODASDS(I,1))/2.
COLMBU(I,2)=(CODASDI(I,2)+CODASDS(I,2))/2.
COLMBU(I,3)=(CODASDI(I,3)+CODASDS(I,3))/2.
COBUL(I,1)=CODASDI(I,1)
COBUL(I,2)=CODASDI(I,2)
COBUL(I,3)=CODASDI(I,3)
COBUR(I,1)=CODASDS(I,1)
COBUR(I,2)=CODASDS(I,2)
COBUR(I,3)=CODASDS(I,3)
155 WRITE(*,*)'*****LINEAL BORDE SUPERIOR*****'
WRITE(*,5000)'COLMBU(' ,I,'1)'=',COLMBU(I,1)
WRITE(*,5000)'COLMBU(' ,I,'2)'=',COLMBU(I,2)
WRITE(*,5000)'COLMBU(' ,I,'3)'=',COLMBU(I,3)
55 CONTINUE
C
LINEAL DE BORDE IZQUIERDO
DO 56 I=1,ICON
IF (ICONTROL4(I).EQ.9) THEN
COLMBL(I,1)=(CODASDS(I,1)+CODASDI(I,1))/2.
COLMBL(I,2)=(CODASDS(I,2)+CODASDI(I,2))/2.
COLMBL(I,3)=(CODASDS(I,3)+CODASDI(I,3))/2.
COBLU(I,1)=CODASDS(I,1)
COBLU(I,2)=CODASDS(I,2)
COBLU(I,3)=CODASDS(I,3)
COBLD(I,1)=CODASDI(I,1)
COBLD(I,2)=CODASDI(I,2)
COBLD(I,3)=CODASDI(I,3)
GOTO 156

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

ENDIF
IF (ICONTROL4(I).EQ.6)THEN
COLMBL(I,1)=(CODASII(I,1)+CODASDS(I,1))/2.
COLMBL(I,2)=(CODASII(I,2)+CODASDS(I,2))/2.
COLMBL(I,3)=(CODASII(I,3)+CODASDS(I,3))/2.
COBLU(I,1)=CODASII(I,1)
COBLU(I,2)=CODASII(I,2)
COBLU(I,3)=CODASII(I,3)
COBLD(I,1)=CODASDS(I,1)
COBLD(I,2)=CODASDS(I,2)
COBLD(I,3)=CODASDS(I,3)
GOTO 156
ENDIF
COLMBL(I,1)=(CODASII(I,1)+CODASIS(I,1))/2.
COLMBL(I,2)=(CODASII(I,2)+CODASIS(I,2))/2.
COLMBL(I,3)=(CODASII(I,3)+CODASIS(I,3))/2.
COBLU(I,1)=CODASII(I,1)
COBLU(I,2)=CODASII(I,2)
COBLU(I,3)=CODASII(I,3)
COBLD(I,1)=CODASIS(I,1)
COBLD(I,2)=CODASIS(I,2)
COBLD(I,3)=CODASIS(I,3)
156 WRITE(*,*)'*****LINEAL BORDE IZQUIERDO*****'
WRITE(*,5000)'COLMBL(' ,I, '1)'=',COLMBL(I,1)
WRITE(*,5000)'COLMBL(' ,I, '2)'=',COLMBL(I,2)
WRITE(*,5000)'COLMBL(' ,I, '3)'=',COLMBL(I,3)
56 CONTINUE
C *****
C ELIMINACION DE NODOS REPETIDOS
C *****
C -----
C BORDE INFERIOR
C -----
DO 342 K=1,MPONT
DO 343 L=1,NDIME
PLIL(K,L)=0.0
343 CONTINUE
342 CONTINUE
PLIL(1,1)=COLMBD(1,1)
PLIL(1,2)=COLMBD(1,2)
PLIL(1,3)=COLMBD(1,3)
CALII(1,1)=COBIL(1,1)
CALII(1,2)=COBIL(1,2)
CALII(1,3)=COBIL(1,3)
CALID(1,1)=COBIR(1,1)
CALID(1,2)=COBIR(1,2)
CALID(1,3)=COBIR(1,3)
IG=1
DO 46 I=2,IM
CHEQ(1)=COLMBD(I,1)
CHEQ(2)=COLMBD(I,2)
CHEQ(3)=COLMBD(I,3)
DO 50 K=1,IG
NL=0.
456 FORMAT(2X,A,F12.4)
IF (PLIL(K,1).EQ.CHEQ(1)) NL=NL+1
IF (PLIL(K,2).EQ.CHEQ(2)) NL=NL+1
IF (PLIL(K,3).EQ.CHEQ(3)) NL=NL+1
IF (NL.EQ.3) GOTO 46
457 FORMAT(2X,A,I3)
50 CONTINUE
IG=IG+1
PLIL(IG,1)=CHEQ(1)
PLIL(IG,2)=CHEQ(2)
PLIL(IG,3)=CHEQ(3)
CALII(IG,1)=COBIL(I,1)
CALII(IG,2)=COBIL(I,2)
CALII(IG,3)=COBIL(I,3)
CALID(IG,1)=COBIR(I,1)
CALID(IG,2)=COBIR(I,2)
CALID(IG,3)=COBIR(I,3)
46 CONTINUE
C -----
C BORDE DERECHO
C -----
DO 47 I=1,IM

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

CHEQ(1)=COLMBR(I,1)
CHEQ(2)=COLMBR(I,2)
CHEQ(3)=COLMBR(I,3)
DO 52 K=1,IG
NL=0.
IF (PLIL(K,1).EQ.CHEQ(1)) NL=NL+1
IF (PLIL(K,2).EQ.CHEQ(2)) NL=NL+1
IF (PLIL(K,3).EQ.CHEQ(3)) NL=NL+1
IF (NL.EQ.3) GOTO 47
52 CONTINUE
IG=IG+1
PLIL(IG,1)=CHEQ(1)
PLIL(IG,2)=CHEQ(2)
PLIL(IG,3)=CHEQ(3)
CALII(IG,1)=COBRD(I,1)
CALII(IG,2)=COBRD(I,2)
CALII(IG,3)=COBRD(I,3)
CALID(IG,1)=COBRU(I,1)
CALID(IG,2)=COBRU(I,2)
CALID(IG,3)=COBRU(I,3)
47 CONTINUE
C -----
C          BORDE SUPERIOR
C -----
DO 48 I=1,IM
CHEQ(1)=COLMBU(I,1)
CHEQ(2)=COLMBU(I,2)
CHEQ(3)=COLMBU(I,3)
DO 58 K=1,IG
NL=0.
IF (PLIL(K,1).EQ.CHEQ(1)) NL=NL+1
IF (PLIL(K,2).EQ.CHEQ(2)) NL=NL+1
IF (PLIL(K,3).EQ.CHEQ(3)) NL=NL+1
IF (NL.EQ.3) GOTO 48
58 CONTINUE
IG=IG+1
PLIL(IG,1)=CHEQ(1)
PLIL(IG,2)=CHEQ(2)
PLIL(IG,3)=CHEQ(3)
CALII(IG,1)=COBUL(I,1)
CALII(IG,2)=COBUL(I,2)
CALII(IG,3)=COBUL(I,3)
CALID(IG,1)=COBUR(I,1)
CALID(IG,2)=COBUR(I,2)
CALID(IG,3)=COBUR(I,3)
48 CONTINUE
C -----
C          BORDE IZQUIERDO
C -----
117 FORMAT(2X,A,F12.4)
DO 49 I=1,IM
CHEQ(1)=COLMBL(I,1)
CHEQ(2)=COLMBL(I,2)
CHEQ(3)=COLMBL(I,3)
DO 104 K=1,IG
NL=0.
IF (PLIL(K,1).EQ.CHEQ(1)) NL=NL+1
IF (PLIL(K,2).EQ.CHEQ(2)) NL=NL+1
IF (PLIL(K,3).EQ.CHEQ(3)) NL=NL+1
IF (NL.EQ.3) GOTO 49
104 CONTINUE
IG=IG+1
PLIL(IG,1)=CHEQ(1)
PLIL(IG,2)=CHEQ(2)
PLIL(IG,3)=CHEQ(3)
CALII(IG,1)=COBLU(I,1)
CALII(IG,2)=COBLU(I,2)
CALII(IG,3)=COBLU(I,3)
CALID(IG,1)=COBLD(I,1)
CALID(IG,2)=COBLD(I,2)
CALID(IG,3)=COBLD(I,3)
49 CONTINUE
RETURN
END

```

SUBROUTINE MDSIDE (ICORDB, LNODE, MBLOC, MPONT, NBLOC, NPONT)

```

C      COMMON/MODEL/NNODE, LNODE, NDIME
      DIMENSION CORDB(MPONT, NDIME), LNODB(MBLOC, LNODE)
C
C      -----
C      ESTA SUBROUTINA INTERPOLA LOS NODOS CENTRALES SOBRE LOS LADOS
C      LOCALIZADOS EN EL LADO RECTO DEL SUPERELEMENTOS HEXAHEDRICOS
C      DE 20 NODOS.
C      -----
      NN=NPONT
      DO 1 IBLOC=1, NBLOC
C
C      REALIZA LA INTERPOLACIÓN EN LA CARA MAS BAJA DEL SUPERELEMENTO
C
      DO 2 INODE=1, 4
C
C      CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
      NODST=LNODB(IBLOC, INODE)
      IGASH=INODE+1
      IF (IGASH.GT.4) IGASH=1
C
C      CALCULA EL NUMERO DE NODO DEL ULTIMO NODO
C
      NODFN=LNODB(IBLOC, IGASH)
      MIDPT=INODE+NNODE
C
C      CALCULA EL NUMERO DEL NODO DE LOS NODOS CENTRALES
C
      NODMD=LNODB(IBLOC, MIDPT)
      IF (NODMD.EQ.0) THEN
        NN=NN+1
        NODMD=NN
        LNODB(IBLOC, MIDPT)=NODMD
      ENDIF
      TOTAL=ABS(CORDB(NODMD, 1))+ABS(CORDB(NODMD, 2))+ABS(CORDB(NODMD, 3))
C
C      SI LAS COORDENADAS DE LOS NODOS CENTRALES SON CERO INTERPOLE
C      POR LINEA RECTA
C
      IF (TOTAL.GT.0.0) GOTO 2
      DO 3 IDIME=1, NDIME
        CORDB(NODMD, IDIME)=(CORDB(NODST, IDIME)+CORDB(NODFN, IDIME))/2.
      CONTINUE
      CONTINUE
C
C      REALIZA LA INTERPOLACIÓN EN LA CARA SUPERIOR DEL SUPERELEMENTO
C
      DO 4 INODE=5, 8
C
C      CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
      NODST=LNODB(IBLOC, INODE)
      IGASH=INODE+1
      IF (IGASH.GT.8) IGASH=5
C
C      CALCULA EL NUMERO DE NODO DEL ULTIMO NODO
C
      NODFN=LNODB(IBLOC, IGASH)
      MIDPT=INODE+12
C
C      CALCULA EL NUMERO DE NODO DE LOS NODOS CENTRALES
C
      NODMD=LNODB(IBLOC, MIDPT)
      IF (NODMD.EQ.0) THEN
        NN=NN+1
        NODMD=NN
        LNODB(IBLOC, MIDPT)=NODMD
      ENDIF
      TOTAL=ABS(CORDB(NODMD, 1))+ABS(CORDB(NODMD, 2))+ABS(CORDB(NODMD, 3))
C
C      SI LAS COORDENADAS DE LOS NODOS CENTRALES SON CERO INTERPOLE
C      POR LINEA RECTA
C
      IF (TOTAL.GT.0.0) GOTO 4
      DO 5 IDIME=1, NDIME
        CORDB(NODMD, IDIME)=(CORDB(NODST, IDIME)+CORDB(NODFN, IDIME))/2.

```

```

5     CONTINUE
4     CONTINUE
C
C     REALIZAR LA INTERPOLACION DE LOS EXTREMOS RESTANTES DEL SUPERELEMENTO
C
C     DO 6 INODE=1,4
C
C     CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
C     NODST=LNODB(IBLOC,INODE)
C     IGASH=INODE+4
C
C     CALCULA EL NUMERO DE NODO DEL ULTIMO NODO
C
C     NODFN=LNODB(IBLOC,IGASH)
C     MIDPT=INODE+12
C
C     CALCULA EL NUMERO DEL NODO DE LOS NODOS CENTRALES
C
C     NODMD=LNODB(IBLOC,MIDPT)
C     IF (NODMD.EQ.0.0) THEN
C     NN=NN+1
C     NODMD=NN
C     LNODB(IBLOC,MIDPT)=NODMD
C     ENDIF
C     TOTAL=ABS(CORDB(NODMD,1))+ABS(CORDB(NODMD,2))+ABS(CORDB(NODMD,3))
C
C     SI LAS COORDENADAS DE LOS NODOS CENTRALES SON CERO INTERPOLE
C     POR LINEA RECTA
C
C     IF (TOTAL.GT.0.0)GOTO 6
C     DO 7 IDIME=1,NDIME
C     CORDB(NODMD,IDIME)=(CORDB(NODST,IDIME)+CORDB(NODFN,IDIME))/2.
7     CONTINUE
6     CONTINUE
1     CONTINUE
C     RETURN
C     END
C
C     SUBROUTINE OPTNUM (NPOIN,MPOIN,IDIFF,ORIGN,MAXCOM,NNCNEC,
1     PONTRS,JNODC)
C
C     COMMON/MODEL/NNODE,LNODE,NDIME
C     DIMENSION NNCNEC(MPOIN),JNODC(4800*10),NEWNOD(4800)
C     INTEGER PONTRS(MPOIN),OLDNOD(4800)
C     INTEGER ORIGN,PJNOD,RNODE
C
C     -----
C     ESTA SUBRUTINA REALIZA LA RENUMERACION NODAL EN ORDEN
C     DE OBTENER UNA DIFERENCIA NODAL MINIMA
C     -----
C
C     ORIGN=1
C     MINDIF=IDIFF
C
C     ANIDAMIENTO PARA LAS POSIBLES COMBINACIONES DIFERENTES DE NPOIN
C     EN LA NUMERACION
C
C     DO 1 IPOIN=1,NPOIN
C     DO 2 JPOIN=1,NPOIN
C     OLDNOD(JPOIN)=0
C     NEWNOD(JPOIN)=0
2     CONTINUE
C     JDIFF=0
C     KPOIN=1
C     NEWNOD(1)=IPOIN
C     OLDNOD(IPOIN)=1
C     LPOIN=1
3     NTNOD=NNCNEC(NEWNOD(KPOIN))
C     IF (NTNOD.EQ.0) GOTO 4
C     PJNOD=(NEWNOD(KPOIN)-1)*MAXCOM
C     DO 5 ITNOD=1,NTNOD
C     RNODE=JNODC(PJNOD+ITNOD)
C     IF (OLDNOD(RNODE).GT.0) GOTO 5
C     LPOIN=LPOIN+1
C     NEWNOD(LPOIN)=RNODE
C     OLDNOD(RNODE)=LPOIN
C     NDIFF=IABS(KPOIN-LPOIN)

```

```

C
C      SI EL ANCHO DE BANDA CALCULADO PARA LA PRESENTE MALLA ES
C      MAS GRANDE QUE CUALQUIERA ANTES CALCULADO LA PRESENTE MALLA
C      ABANDONADA Y UNA NUEVA SERA EMPEZADA
C
      IF (NDIFF.GE.MINDIF) GOTO 1
      IF (NDIFF.GT.JDIFF) JDIFF=NDIFF
5     CONTINUE
      IF (LPOIN.NE.NPOIN) GOTO 6
4     KPOIN=KPOIN+1
      GOTO 3
6     MINDIF=JDIFF
      DO 7 JPOIN=1,NPOIN
      PONTRS(JPOIN)=OLDNOD(JPOIN)
7     CONTINUE
1     CONTINUE
      IF (MINDIF.LT.IDIFF) GOTO 8
      ORIGN=0
      RETURN
8     IDIFF=MINDIF
      DO 9 IPOIN=1,NPOIN
      NNCNEC(PONTRS(IPOIN))=IPOIN
9     CONTINUE
C
      RETURN
      END

      SUBROUTINE OUTPUT (NELEM,NPOIN,TITLE,COORD,MATNO, LNODS,MELEM,
1          MPOIN,ORIGN,NRENM,NNCNEC,PONTRS,NMATNB,ELAST
2          ,CNABLA,IPOL,NODOII,NODOIS,NODODI,NODODS,IPOLI
3          ,NODII,NODIS,ICON,IG,NODO)
C
      COMMON/MODEL/NNODE,LNODE,NDIME
      DIMENSION COORD(MPOIN,3),MATNO(MELEM),LNODS(MELEM,NNODE),
1          NNCNEC(MPOIN),IL(8),ELAST(NMATNB),CNABLA(NMATNB)
      INTEGER PONTRS(MPOIN),IPOL(MPOIN),NODOII(MPOIN),NODOIS(MPOIN),
1          NODODI(MPOIN),NODODS(MPOIN),IPOLI(MPOIN),NODII(MPOIN),
2          NODIS(MPOIN),NODO(MPOIN)
      INTEGER ORIGN
      CHARACTER*74 FILE1
      CHARACTER*74 FILE2
C
C      -----
C      ESTA SUBRUTINA GENERA LOS DATOS DE SALIDA
C      -----
      WRITE(*,*)TITLE
      READ(*,*)
      WRITE(*,*)'SE VA A GUARDAR EL ARCHIVO DE SALIDA PARA ABAQUS'
      WRITE(*,200)'ENTRAR DIRECTORIO,NOMBRE DE ARCHIVO(C:/A/MOD1.INP)='
      READ(*,'(A)')FILE1
      WRITE(*,*)'SE VA A GUARDAR EL ARCHIVO DE SALIDA DE DATOS'
      WRITE(*,200)'ENTRAR DIRECTORIO,NOMBRE DE ARCHIVO(C:/A/MOD1.GEN)='
      READ(*,'(A)')FILE2
C
      *****ALMACENAMIENTO DE RESULTADOS*****
      OPEN (6,FILE=FILE1,ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      OPEN (4,FILE=FILE2,ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
C      SALIDA DE LAS COORDENAS DE LOS PUNTOS NODALES
      WRITE(*,1010)
      WRITE(6,1010)
      WRITE(*,1002)
      WRITE(6,1002)
      NPOLA=0.
      NRENM=0.
      MAT=1
      WRITE(4,2008)NELEM,NPOIN,NPOLA,NRENM
      DO 12 IELEM=1,NELEM
      KELEM=IELEM
      IF (NRENM.EQ.0.OR.ORIGN.EQ.0) GOTO 13
      DO 14 INODE=1,NNODE
      IL(INODE)=PONTRS(LNODS(KELEM,INODE))
14     CONTINUE
      WRITE(4,2005) KELEM,MAT,
1          (IL(INODE),INODE=1,NNODE)
      GOTO 12
13     DO 15 INODE=1,NNODE
      IL(INODE)=LNODS(KELEM,INODE)

```

```

15  CONTINUE
    WRITE(4,2005) KELEM,MAT,
1   (IL(INODE),INODE=1,NNODE)
12  CONTINUE
    JPOLA=0.
    DO 1 IPOIN=1,NPOIN
      KPOIN=IPOIN
      IF (NREN.M.EQ.1.AND.ORIGN.EQ.1) KPOIN=NNCNEC(IPOIN)
      IF (IPOIN.EQ.15)READ(*,*)
      WRITE(*,1003) IPOIN,(COORD(KPOIN,IDIME),IDIME=1,NDIME)
      WRITE(6,1003) IPOIN,(COORD(KPOIN,IDIME),IDIME=1,NDIME)
      WRITE(4,2003) JPOLA,IPOIN,(COORD(KPOIN,IDIME),IDIME=1,NDIME)
1   CONTINUE
C
C   SALIDA DE LA TOPOLOGIA DEL ELEMENTO
C
    WRITE(*,200)'INTRODUCIR CONDICIONES DE APOYO (SI=1/NO=0) ='
    READ(*,*)DICE
    IF (DICE.EQ.1) THEN
      WRITE(*,200)'NUMERO DE NODOS RESTRINGIDOS ='
      READ(*,*)NR
      DO 34 IN=1,NR
        WRITE(*,210)'NODO (' ,IN, ')= '
        READ(*,*)NODO(IN)
34  CONTINUE
210 FORMAT(5X,A,I3,A,$)
    WRITE(6,2001)
    DO 36 IN=1,NR
      WRITE(6,2002)NODO(IN)
36  CONTINUE
    GOTO 35
  ENDIF
    WRITE(6,2004)
35  WRITE(*,1004)
    WRITE(6,1004)
    DO 2 IELEM=1,NELEM
      KELEM=IELEM
      IF (NREN.M.EQ.0.OR.ORIGN.EQ.0) GOTO 3
      DO 4 INODE=1,NNODE
        IL(INODE)=LNODS(KELEM,INODE)
4   CONTINUE
        WRITE(*,1005) KELEM,
1   (IL(INODE),INODE=1,NNODE)
        WRITE(6,1005) KELEM,
1   (IL(INODE),INODE=1,NNODE)
        GOTO 2
3   DO 5 INODE=1,NNODE
      IL(INODE)=LNODS(KELEM,INODE)
5   CONTINUE
      WRITE(*,1005) KELEM,
1   (IL(INODE),INODE=1,NNODE)
      WRITE(6,1005) KELEM,
1   (IL(INODE),INODE=1,NNODE)
2   CONTINUE
      WRITE(*,1006)
      WRITE(6,1006)
      WRITE(6,1007)
      WRITE(*,1008)
      WRITE(6,1008)
      WRITE(*,1015)
      WRITE(6,1015)
      WRITE(*,1012)
      WRITE(6,1012)
      IF (IPOL(1).EQ.0.AND.IPOLI(1).EQ.0.) GOTO 31
      WRITE(*,1017)
      WRITE(6,1017)
      IF (IPOL(1).EQ.0.)GOTO 32
      DO 45 I=1,IG
        WRITE(*,1018)IPOLI(I),NODII(I),NODIS(I)
        WRITE(6,1018)IPOLI(I),NODII(I),NODIS(I)
45  CONTINUE
      IF (IPOLI(1).EQ.0.)GOTO 31
32  DO 46 I=1,ICON
        WRITE(*,1019)IPOL(I),NODOIS(I),NODODI(I),NODODS(I),NODOII(I)
        WRITE(6,1019)IPOL(I),NODOIS(I),NODODI(I),NODODS(I),NODOII(I)
46  CONTINUE

```

Licencia Creative Commons:

```

31 WRITE(*,1016)
   WRITE(6,1016)
   WRITE(*,1014)
   WRITE(6,1014)
   RETURN
C
C   FORMATO DEL ARCHIVO DE SALIDA DE DATOS
C
200  FORMAT(5X,A,$,1X)
1002  FORMAT(45H*****//,27H**
1DEFINICION DE NODOS    //,45H*****
2*****//,5H*NODE)
1003  FORMAT(I5,1H,,F9.2,1H,,F9.2,1H,,F9.2)
2001  FORMAT(16H*NSET,NSET=APOYO)
2002  FORMAT(I3,1H,)
2004  FORMAT(16H*NSET,NSET=APOYO//,15H1,2,3,4,5,6,7,8)
2003  FORMAT(I5,1H,I5,1H,,F9.2,1H,,F9.2,1H,,F9.2)
1004  FORMAT(45H*****//,29H**DE
1FINICION DE ELEMENTOS //,45H*****
2*****//,34H*ELEMENT,TYPE=C3D8,ELSET= TODOS )
1005  FORMAT(I5,8(1H,I5))
2005  FORMAT(I5,1H,,I5,8(1H,I5))
2008  FORMAT(I5,1H,,I5,1H,,I3,1H,,I3)
1010  FORMAT(8H*HEADING//,3X,A70)
1006  FORMAT(40H*SOLID SECTION,ELSET=TODOS,MATERIAL=MAT1//,19H*MATERIAL,
1NAME=MAT1//,8H*ELASTIC)
1007  FORMAT(6H 1,0.3)
1008  FORMAT(14H*RESTART,WRITE)
1012  FORMAT(45H*****//,26H** CO
1NDICION DE APOYO    //,45H*****
2*****//,9H*BOUNDARY//,9HAPOYO,1,3)
1016  FORMAT(45H*****//,32H** SA
1LIDA PARA EL POSTPROCESADOR//,46H*****
2***** )
1014  FORMAT(20H*EL FILE,ELSET=TODOS//,3HS,E//,9H*EL PRINT//,5HSENER//,1
13H*ENERGY PRINT//,5HSENER//,12H*ENERGY FILE//,5HSENER//,9H*END STE
2P)
1015  FORMAT(45H*****//,28H**DEF
1INICION DE LA CARGA //,45H*****
2*****//,5H*STEP//,7H*STATIC//,6H*DLOAD//,9H1,P2,1000)
1017  FORMAT(4H*MPC)
1018  FORMAT(6HLINEAR,I3,1H,I3,1H,I3)
1019  FORMAT(8HBILINEAR,I3,1H,I3,1H,I3,1H,I3,1H,I3)

END

SUBROUTINE RENMBR (NELEM,NPOIN,LNODS,MELEM,MPOIN,
1          IDIFF,ORIGN,NNCNEC,PONTRS)
C
COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION NNCNEC(MPOIN),LNODS(MELEM,NNODE),JNODC(4800*10)
INTEGER PONTRS(MPOIN)
INTEGER ORIGN,PJNOD,RNODE
C
C-----
C   ESTA SUBRUTINA CALCULA CUANTOS NODOS ESTAN DIRECTAMENTE CONECTADOS
C   A UN NODO DADO, Y ALMACENA LOS NUMEROS DE LOS NODOS
C-----
C
C   MAXIMO NUMERO DE ELEMENTOS QUE UN SIMPLE NODO PUEDE COMPARTIR
C
MAXCOM=10
MCNEC=NPOIN*MAXCOM
C
C   INICIO
C
DO 1 IPOIN=1,NPOIN
NNCNEC(IPOIN)=0
1 CONTINUE
DO 2 ICNEC=1,MCNEC
JNODC(ICNEC)=0
2 CONTINUE
C
C   PROCESO SOBRE EL NUMERO DE ELEMENTOS
C
DO 3 IELEM=1,NELEM
DO 4 INODE=1,NNODE

```

```

      KNODE=LNODS(IELEM,INODE)
C
C      PARAR EL TRABAJO DESPUES QUE TODOS LOS NODOS SEA PROCESADOS
C
      IF (KNODE.EQ.0) GOTO 3
C
      PJNOD ES UN SEÑALADOR QUE INDICA EN LA VARIABLE JNODC EL
      COMIENZO DE LOS PUNTOS NODALES DIRECTAMENTE RELACIONADO A
      EL ELEMENTO EN ESTUDIO
C
      PJNOD=(KNODE-1)*MAXCOM
      DO 5 JNODE=1,NNODE
      IF (JNODE.EQ.INODE) GOTO 5
      RNODE=LNODS(IELEM,JNODE)
      IF (RNODE.EQ.0) GOTO 4
      NTNOD=NNCNEC(KNODE)
      IF (NTNOD.EQ.0) GOTO 6
      DO 7 ITNOD=1,NTNOD
      IF (JNODC(PJNOD+ITNOD).EQ.RNODE) GOTO 5
7      CONTINUE
6      NNCNEC(KNODE)=NNCNEC(KNODE)+1
      JNODC(PJNOD+NNCNEC(KNODE))=RNODE
C
C      CALCULA LA MAXIMA DIFERENCIA NODAL
C
      IF (IABS(KNODE-RNODE).GT.IDIFF) IDIFF=IABS(KNODE-RNODE)
5      CONTINUE
4      CONTINUE
3      CONTINUE
C
      INTENTA UNA RENUMERACION DIFERENTE DE LA MALLA EN ORDEN DE
      DETERMINAR UNA DIFERENCIA NODAL MEJOR
C
      CALL OPTNUM (NPOIN,MPOIN,
1          IDIFF,ORIGN,MAXCOM,
2          NNCNEC,PONTRS,JNODC)
C
      RETURN
      END
SUBROUTINE REORDR (NBLOC,MATNB,LNODB,LNODR,NBUSD,NBNGH,
1          NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,
2          NDUMY,DUMYX,DUMYY,DUMYZ,MBLOC,MDIVI)
C
      COMMON/MODEL/NNODE, LNODE, NDIME
      DIMENSION MATNB(MBLOC),LNODB(MBLOC,LNODE),
1          LNODR(MBLOC,LNODE),NBUSD(MBLOC),NBNGH(MBLOC),
2          NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ(MBLOC),
3          WEITX(MBLOC,MDIVI),WEITY(MBLOC,MDIVI),
4          WEITZ(MBLOC,MDIVI),NODE(8),NDUMY(MBLOC,4),
5          DUMYX(MBLOC,MDIVI),DUMYY(MBLOC,MDIVI),
6          DUMYZ(MBLOC,MDIVI)
-----
C      ESTA SUBROUTINA RENUMERA LOS SUPERELEMENTOS DE ACUERDO A LA
C      PROXIMIDAD RELATIVA ENTRE ELLOS
C      -----
      DO 1 IBLOC=1,NBLOC
      NBUSD(IBLOC)=1
1      CONTINUE
C
C      INICIO DEL REORDENAMIENTO A PARTIR DEL NUMERO 1
C
      NBUSD(1)=-1
      NBNGH(1)=1
      KOUNT=2
      DO 2 IBLOC=1,NBLOC
      NODE(1)=LNODB(NBNGH(IBLOC),1)
      NODE(2)=LNODB(NBNGH(IBLOC),2)
      NODE(3)=LNODB(NBNGH(IBLOC),3)
      NODE(4)=LNODB(NBNGH(IBLOC),4)
      NODE(5)=LNODB(NBNGH(IBLOC),5)
      NODE(6)=LNODB(NBNGH(IBLOC),6)
      NODE(7)=LNODB(NBNGH(IBLOC),7)
      NODE(8)=LNODB(NBNGH(IBLOC),8)
      IC=NBNGH(IBLOC)
C
C      EN BUSCA DE

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

C
DO 3 JBLOC=1,NBLOC
IF (IBLOC.EQ.JBLOC)GOTO 3
IF (NBUSD(JBLOC).LT.0) GOTO 3
M=0
DO 4 INODE=1,8
DO 4 JNODE=1,8
IF (LNODB(JBLOC,INODE).EQ.NODE(JNODE).AND.NBUSD(JBLOC).GT.0)
1M=M+1
4 CONTINUE
IF (M.EQ.0.OR.JBLOC.EQ.IC) GOTO 3
NBNGH(KOUNT)=JBLOC
KOUNT=KOUNT+1
NBUSD(JBLOC)=-1
3 CONTINUE
2 CONTINUE
C
C FINALIZAR REORDENAMIENTO DE TODAS LAS VARIABLE DEL SUPERELEMENTO
C
DO 5 IBLOC=1,NBLOC
DO 5 INODE=1,20
LNODR (IBLOC,INODE)=LNODB(NBNGH (IBLOC),INODE)
5 CONTINUE
DO 6 IBLOC=1,NBLOC
L=NENGH (IBLOC)
NDUMY (IBLOC,1)=NDIVX (L)
NDUMY (IBLOC,2)=NDIVY (L)
NDUMY (IBLOC,3)=NDIVZ (L)
NDUMY (IBLOC,4)=MATNB (L)
DO 7 IDIVX=1,NDIVX (L)
DUMYX (IBLOC,IDIVX)=WEITX (L,IDIVX)
7 CONTINUE
DO 8 IDIVY=1,NDIVY (L)
DUMYY (IBLOC,IDIVY)=WEITY (L,IDIVY)
8 CONTINUE
DO 9 IDIVZ=1,NDIVZ (L)
DUMYZ (IBLOC,IDIVZ)=WEITZ (L,IDIVZ)
9 CONTINUE
6 CONTINUE
DO 10 IBLOC=1,NBLOC
NDIVX (IBLOC)=NDUMY (IBLOC,1)
NDIVY (IBLOC)=NDUMY (IBLOC,2)
NDIVZ (IBLOC)=NDUMY (IBLOC,3)
MATNB (IBLOC)=NDUMY (IBLOC,4)
DO 11 IDIVX=1,NDUMY (IBLOC,1)
WEITX (IBLOC,IDIVX)=DUMYX (IBLOC,IDIVX)
11 CONTINUE
DO 12 IDIVY=1,NDUMY (IBLOC,2)
WEITY (IBLOC,IDIVY)=DUMYY (IBLOC,IDIVY)
12 CONTINUE
DO 13 IDIVZ=1,NDUMY (IBLOC,3)
WEITZ (IBLOC,IDIVZ)=DUMYZ (IBLOC,IDIVZ)
13 CONTINUE
10 CONTINUE
DO 14 IBLOC=1,NBLOC
DO 14 INODE=1,20
LNODB (IBLOC,INODE)=LNODR (IBLOC,INODE)
14 CONTINUE
RETURN
END

SUBROUTINE SHP20 (EXISP,ETASP,EPSIP,SHAPE)

DIMENSION SHAPE (20)
-----
C
C ESTA SUBRUTINA EVALUA LAS FUNCIONES DE FORMA PARA EL ELEMENTO
C TRIDIMENSIONAL HEXAHEDRICO DE 20 NODOS
C -----
C
S=EXISP
T=ETASP
U=EPSIP
C
SS=S*S
TT=T*T
UU=U*U
C

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

C      NODOS DE LAS ESQUINAS
C
      SHAPE(1)=(1.-S)*(1.-T)*(1.-U)*(-S-T-U-2.)*0.125
      SHAPE(2)=(1.+S)*(1.-T)*(1.-U)*(+S-T-U-2.)*0.125
      SHAPE(3)=(1.+S)*(1.+T)*(1.-U)*(+S+T-U-2.)*0.125
      SHAPE(4)=(1.-S)*(1.+T)*(1.-U)*(-S+T-U-2.)*0.125
      SHAPE(5)=(1.-S)*(1.-T)*(1.+U)*(-S-T+U-2.)*0.125
      SHAPE(6)=(1.+S)*(1.-T)*(1.+U)*(+S-T+U-2.)*0.125
      SHAPE(7)=(1.+S)*(1.+T)*(1.+U)*(+S+T+U-2.)*0.125
      SHAPE(8)=(1.-S)*(1.+T)*(1.+U)*(-S+T+U-2.)*0.125
C
C      NODOS CENTRALES
C
      SHAPE(9)=(1.-SS)*(1.-T)*(1.-U)*0.25
      SHAPE(10)=(1.+S)*(1.-TT)*(1.-U)*0.25
      SHAPE(11)=(1.-SS)*(1.+T)*(1.-U)*0.25
      SHAPE(12)=(1.-S)*(1.-TT)*(1.-U)*0.25
C
      SHAPE(13)=(1.-S)*(1.-T)*(1.-UU)*0.25
      SHAPE(14)=(1.+S)*(1.-T)*(1.-UU)*0.25
      SHAPE(15)=(1.+S)*(1.+T)*(1.-UU)*0.25
      SHAPE(16)=(1.-S)*(1.+T)*(1.-UU)*0.25
C
      SHAPE(17)=(1.-SS)*(1.-T)*(1.+U)*0.25
      SHAPE(18)=(1.+S)*(1.-TT)*(1.+U)*0.25
      SHAPE(19)=(1.-SS)*(1.+T)*(1.+U)*0.25
      SHAPE(20)=(1.-S)*(1.-TT)*(1.+U)*0.25
      RETURN
      END

      SUBROUTINE TECLA (NBLOC,NPONT,CORDB,MATNB,LNODE,ORIGB,
1          NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,MBLOC,
2          MPONT,MDIVI,MPOLA,ELAST,CNABLA,NMATNB,NJPOLA,
3          CAMBIO,COORD,VALOR,NABLA,ELASTIC,PILAS)
C
      COMMON/MODEL/NNODE,LNODE,NDIME
      DIMENSION CORDB(MPONT,NDIME),MATNB(MBLOC),LNODB(MBLOC,LNODE)
      DIMENSION ORIGB(MPOLA,NDIME)
      DIMENSION NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ(MBLOC)
      DIMENSION WEITX(MBLOC,MDIVI),ELAST(MBLOC),CNABLA(MBLOC)
      DIMENSION WEITY(MBLOC,MDIVI),WEITZ(MBLOC,MDIVI)
      INTEGER PILAS
      REAL ELASTIC,NABLA
      CHARACTER*70 TITLE
      CHARACTER*70 FILENAME
C
      -----
C          ESTA SUBROUTINA CREA LOS DATOS DE ENTRADA
C          Y LOS ALMACENA EN UN ARCHIVO .IN
C      -----
      PILAS=0
      WRITE(*,100)
      PILAS=0
      WRITE(*,200)'INTRODUZCA DIRECTORIO Y NOMBRE DEL ARCHIVO(C:.IN)='
      READ(*,'(A)')FILENAME
      WRITE(*,190)
      WRITE(*,210)'TITULO DEL PROBLEMA='
      READ(*,'(A70)')TITLE
      WRITE(*,210)'NUMERO DE BLOQUES (NBLOC)='
      READ(*,*)NBLOC
      WRITE(*,210)'NUMERO DE NODOS DEL MODELO(NPONT)='
      READ(*,*)NPONT
      WRITE(*,210)'NUMERO DE SISTEMAS AUXILIARES (NPOLA)='
      READ(*,*)NPOLA
C
C      *****LECTURA DE CONECTIVIDAD DE LOS ELEMENTOS*****
C
C      -----NUMERO DE MATERIALES-----
101  WRITE(*,210)'EXISTE UN SOLO TIPO DE MATERIAL (SI=0/NO=1)='
      READ(*,*)I
C
C      -----UN SOLO TIPO DE MATERIAL-----
C
      IF (I.EQ.0) THEN
          NMATNB=1.0
          WRITE(*,210)'MODULO DE ELASTICIDAD='
          READ(*,*)ELASTIC

```

```

WRITE(*,210)'MODULO DE POISSON='
READ(*,*)NABLA
DO 333 IBLOC=1,NBLOC
MATNB(IBLOC)=NMATNB
ELAST(IBLOC)=ELASTIC
CNABLA(IBLOC)=NABLA
333 CONTINUE
C
C -----DIFERENTES TIPOS DE MATERIALES-----
C
ELSE IF (I.EQ.1) THEN
WRITE(*,210)'CANTIDAD DE MATERIALES='
READ(*,*)NMATNB
DO 1 J=1,NMATNB
WRITE(*,220)'CARACTERISTICAS DEL MATERIAL TIPO('',J,')'
WRITE(*,230)'NUMERO DEL BLOQUE ASOCIADO AL MATERIAL TIPO('',J,')='
READ(*,*)IBLOC
WRITE(*,210)'MODULO DE ELASTICIDAD='
READ(*,*)elast(MATNB(IBLOC))
WRITE(*,210)'MODULO DE POISSON='
READ(*,*)CNABLA(MATNB(IBLOC))
1 CONTINUE
ELSE IF (I.GT.1) THEN
GOTO 101
ENDIF
C
C -----TOPOLOGIA DEL BLOQUE-----
C
DO 2 IBLOC=1,NBLOC
WRITE(*,240)'TOPOLOGIA DEL BLOQUE'
DO 3 INODE=1,8
WRITE(*,250)'BLOQUE('',IBLOC,')----->NODO('',INODE,')='
READ(*,*)LNODB(IBLOC,INODE)
3 CONTINUE
DO 76 INODE=9,20
LNODB(IBLOC,INODE)=0
76 CONTINUE
2 CONTINUE
C
C *****TABLA DE CORRECCION TOPOLOGIA*****
C
111 DO 4 IBLOC=1,NBLOC
WRITE(*,260)
WRITE(*,220)'TOPOLOGIA DEL BLOQUE('',IBLOC,')'
WRITE(*,270)'NODO VALOR '
DO 5 INODE=1,8
WRITE(*,280) INODE,(LNODB(IBLOC,INODE))
5 CONTINUE
WRITE(*,210)'DESEA CAMBIAR ALGUN VALOR (NO=0/SI=1)=?'
READ(*,*)CAMBIO
IF (CAMBIO.EQ.1) THEN
WRITE(*,210)'INTRODUZCA VALOR DE (NODO,VALOR)='
READ(*,*)NODO,VALOR
LNODB(IBLOC,NODO)=VALOR
GOTO 111
ELSE IF (CAMBIO.EQ.0) THEN
GOTO 4
ELSE
GOTO 111
ENDIF
4 CONTINUE
C
C -----INIZIALIZACION DEL SISTEMA ORIGEN COORDENADO-----
C
DO 6 IPOLA=1,MPOLA
DO 7 IDIME=1,NDIME
ORIGB(IPOLA,IDIME)=0.0
7 CONTINUE
6 CONTINUE
C
C -----ORIGEN DE SISTEMA COORDENADOS-----
C
IF (NPOLA.EQ.0) THEN
NJPOLA=0
GOTO 121
ELSE

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

WRITE(*,290)'SISTEMAS AUXILIARES DE COORDENADAS'
DO 8 IPOLA=1,NPOLA
DO 13 IDIME=1,NDIME
WRITE(*,210)'# DE SISTEMA AUXILIAR ='
READ(*,*)NJPOLA
WRITE(*,300)'ORIGEN DEL SISTEMA(' ,NJPOLA, ' )'
WRITE(*,310)'COORD. X= '
READ(*,*)ORIGB(NJPOLA,NDIME)
WRITE(*,310)'COORD. Y= '
READ(*,*)ORIGB(NJPOLA,NDIME)
WRITE(*,310)'COORD. Z= '
READ(*,*)ORIGB(NJPOLA,NDIME)
13 CONTINUE
8 CONTINUE
ENDIF
C
C INICIALIZACION DE TODAS LAS COORDENADAS ANTES DE LEER
C ALGUNA DE ELLAS
C
121 DO 10 IPONT=1,MPONT
DO 10 IDIME=1,NDIME
CORDB(IPONT,IDIME)=0.0
10 CONTINUE
C
C -----COORDENADAS NODALES-----
C
WRITE(*,260)
WRITE(*,290)'COORDENADAS NODALES'
DO 11 JPONT=1,NPONT
WRITE(*,330)'NODO(' ,JPONT, ' )'
WRITE(*,310)'COORD. X= '
READ(*,*)CORDB(JPONT,1)
WRITE(*,310)'COORD. Y= '
READ(*,*)CORDB(JPONT,2)
WRITE(*,310)'COORD. Z= '
READ(*,*)CORDB(JPONT,3)
11 CONTINUE
C
C *****CHEQUEO DE LOS DATOS DE ENTRADA DEL MODELO*****
C CALL CHECK (NBLOC,NPONT,NPOLA,MBLOC,MPONT,MPOLA,ISTOP)
C
C *****PARAR SI SE CONSIGUE UN ENDO*****
C
IF (ISTOP.GT.0) THEN
WRITE(*,260)
WRITE(*,240)'ES UN CRAZO ERROR'
READ(*,*)IBLOC
STOP
ENDIF
C
C *****TABLA DE CORRECCION DE COORDENANDAS*****
C
131 WRITE (*,1004)
WRITE(*,350)
DO 17 IPONT=1,NPONT
WRITE (*,1005) IPONT, (CORDB(IPONT, IDIME), IDIME=1, NDIME)
17 CONTINUE
WRITE(*,210)'DESEA CAMBIAR ALGUN VALOR (NO=0/SI=1)='
READ(*,*)CAMBIO
IF (CAMBIO.EQ.1) THEN
WRITE(*,210)'INTRODUZCA VALOR DE (NODO,COORD,VALOR)='
READ(*,*)INODE,COORD,VALOR
CORDB(INODE,COORD)=VALOR
GOTO 131
ENDIF
C
C *****EVITA USAR TRANSFORMACION SI SE USA SISTEMA DE COORDENADAS*****
C CARTESIANAS
C
DO 14 JPONT=1,NPONT
IF (NJPOLA.GT.0.AND.NJPOLA.LE.5) THEN
RADIUS=CORDB(JPONT,1)
THETA =CORDB(JPONT,2)*3.141592654/180.
CORDB(JPONT,1)=RADIUS*COS(THETA)+ORIGB(NJPOLA,1)
CORDB(JPONT,2)=RADIUS*SIN(THETA)+ORIGB(NJPOLA,2)

```

WWW.BDIGITAL.ULA.VE


```

290  FORMAT (/,15X,A)
300  FORMAT (A,I2,A)
310  FORMAT (/,15X,A,$)
330  FORMAT (5X,/,5X,A,I2,A)
350  FORMAT (3X,'
370  FORMAT (1X,I2)
410  FORMAT (5X,I2,5X,I2,5X,I2)
999  FORMAT (60(/),79('-',)32X,'PROGRAMA GEMATRI',
1//,23X,'GENERADOR DE MALLA TRIDIMENSIONAL',
2//,13X,'USANDO SUPERELEMENTOS E INDICES DE DENSIDAD DE ENERGIA',
3/,79('-',))
1000 FORMAT (//,3X,'**TITULO DEL PROBLEMA',//,3X,A70)
1001 FORMAT (//,3X,'**DEFINICION DE LOS SUPERELEMENTOS PRINCIPALES',
1//,5X,'NBLOC=',I2,5X,'NPOLA=',I2)
1002 FORMAT (//,3X,'** DEFINICION DE TOPOLOGIA DE LOS SUPERELEMENTOS')
1004 FORMAT (3X,'** COORDENADAS PUNTOS NODALES SUPERELEMENTO',
1/,3X,'NODO X COORDENADA Y COORDENADA Z COORDENADA')
1005 FORMAT (1X,I4,6X,E12.4,5X,E12.4,5X,E12.4)
RETURN
END

SUBROUTINE UBILSD (NBLOC,NPOIN,NNCNEC,COORD,DARAI,DARAS,
1 DARAB,MPOIN,NRENM,ORIGN,CLIN,LAS01,LAS02
2 ,LINE,LINE1,LINE2,LIRCI,CARAI,CARAS,CARAB,
3 LIRCD,MPONT,IDRB,IDCS)
COMMON/MODEL/NNODE,LNODE,NDIME
DIMENSION DARAI (MPONT,NDIME),DARAS (MPONT,NDIME),DARAB (MPONT,NDIME)
1 ,NNCNEC (NPOIN),COORD (MPOIN,NDIME),CLIN (MPOIN,NDIME),
2 LAS01 (NPOIN,NDIME),LAS02 (NPOIN,NDIME),CARAI (MPONT,NDIME)
3 ,CARAS (MPONT,NDIME),CARAB (MPONT,NDIME)
INTEGER LINE (MPOIN),LINE1 (MPOIN),LINE2 (MPOIN),
1 LIRCI (NDIME),LIRCD (NDIME)
C *****
C ESTA SUBROUTINA LOCALIZA LOS NODOS LINEALES INCOMPATIBLES
C TANTO EN ELEMENTOS NO REFINADOS COMO EN REFINADOS
C *****
IMI=0.
LDGI=LIRCD(1)
IRCI=LIRCI(1)
DO 63 I=1,IRCI,3
DO 64 J=1,LDGI,3
NL=0.
IF (CARAI (IDI,1).EQ.DARAI (JID,1)) NL=NL+1
IF (CARAI (IDI,2).EQ.DARAI (JID,2)) NL=NL+1
IF (CARAI (IDI,3).EQ.DARAI (JID,3)) NL=NL+1
IF (NL.EQ.3) THEN
IMI=IMI+1
WRITE (*,100)'IMI=',IMI
100 FORMAT (2X,A,I3)
CLIN (IMI,1)=CARAI (IDI,1)
CLIN (IMI,2)=CARAI (IDI,2)
CLIN (IMI,3)=CARAI (IDI,3)
IMA=IDI-1
IME=IDI-2
LAS01 (IMI,1)=CARAI (IMA,1)
LAS01 (IMI,2)=CARAI (IMA,2)
LAS01 (IMI,3)=CARAI (IMA,3)
LAS02 (IMI,1)=CARAI (IME,1)
LAS02 (IMI,2)=CARAI (IME,2)
LAS02 (IMI,3)=CARAI (IME,3)
ENDIF
64 CONTINUE
63 CONTINUE
C -----
C NODOS UBICADOS EN LOS BORDES DE LA CARA SUPERIOR DE CADA ELEMENTO
C -----
IRCI=LIRCI(2)
LDGS=LIRCD(2)
IMS=0
DO 72 I=1,NBLOC
DO 73 I=1,IRCI
IDI=I*3
DO 74 J=1,IDCS
JID=J*3
NL=0
IF (CARAS (IDI,1).EQ.DARAS (JID,1)) NL=NL+1

```

```

IF (CARAS(IDI,2).EQ.DARAS(JID,2)) NL=NL+1
IF (CARAS(IDI,3).EQ.DARAS(JID,3)) NL=NL+1
IF (NL.EQ.3) THEN
  IMI=IMI+1
  CLIN(IMI,1)=CARAS(IDI,1)
  CLIN(IMI,2)=CARAS(IDI,2)
  CLIN(IMI,3)=CARAS(IDI,3)
  IMA=IDI-1
  IME=IDI-2
  LAS01(IMI,1)=CARAS(IMA,1)
  LAS01(IMI,2)=CARAS(IMA,2)
  LAS01(IMI,3)=CARAS(IMA,3)
  LAS02(IMI,1)=CARAS(IME,1)
  LAS02(IMI,2)=CARAS(IME,2)
  LAS02(IMI,3)=CARAS(IME,3)
ENDIF
74 CONTINUE
73 CONTINUE
72 CONTINUE
C
C -----
C          NODOS UBICADOS EN LOS BORDES RESTANTES DE CADA ELEMENTO
C -----
IRCI=LIRCI(3)
LDRB=LIRCD(3)
IMB=0
DO 82 L=1,NBLOC
DO 83 I=1,IRCI
  IDI=I*3
  DO 84 J=1,IDRB
    JID=J*3
    NL=0
    IF (CARAB(IDI,1).EQ.DARAB(JID,1)) NL=NL+1
    IF (CARAB(IDI,2).EQ.DARAB(JID,2)) NL=NL+1
    IF (CARAB(IDI,3).EQ.DARAB(JID,3)) NL=NL+1
    IF (NL.EQ.3) THEN
      IMI=IMI+1
      CLIN(IMI,1)=CARAB(IDI,1)
      CLIN(IMI,2)=CARAB(IDI,2)
      CLIN(IMI,3)=CARAB(IDI,3)
      IMA=IDI-1
      IME=IDI-2
      LAS01(IMI,1)=CARAB(IMA,1)
      LAS01(IMI,2)=CARAB(IMA,2)
      LAS01(IMI,3)=CARAB(IMA,3)
      LAS02(IMI,1)=CARAB(IME,1)
      LAS02(IMI,2)=CARAB(IME,2)
      LAS02(IMI,3)=CARAB(IME,3)
    ENDIF
84 CONTINUE
83 CONTINUE
82 CONTINUE
C *****
C          IDENTIFICACION DEL NUMERO DE LOS NODOS INCOMPATIBLES LINEALES
C *****
DO 93 I=1,IMI
DO 94 IPOIN=1,NPOIN
  NL=0
  KPOIN=IPOIN
  IF (NREN.M.EQ.1.AND.ORIGN.EQ.1) KPOIN=NNCNEC(IPOIN)
  IF (CLIN(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
  IF (CLIN(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
  IF (CLIN(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
  IF (NL.EQ.3) THEN
    LINE(I)=KPOIN
  ENDIF
  NL=0.
  IF (LAS01(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
  IF (LAS01(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
  IF (LAS01(I,3).EQ.COORD(KPOIN,3)) NL=NL+1
  IF (NL.EQ.3) THEN
    LINE1(I)=KPOIN
  ENDIF
  NL=0.
  IF (LAS02(I,1).EQ.COORD(KPOIN,1)) NL=NL+1
  IF (LAS02(I,2).EQ.COORD(KPOIN,2)) NL=NL+1
  IF (LAS02(I,3).EQ.COORD(KPOIN,3)) NL=NL+1

```

```
IF (NL.EQ.3) THEN
LINE2(I)=KPOIN
ENDIF
94 CONTINUE
93 CONTINUE
C WRITE(*,140)'ESTOS SON LOS NODOS LINEALES Y SUS NODOS VECINOS'
C READ(*,*)
C DO 85 I=1,IMI
C WRITE(*,105)'NODO LINEAL(' ,I,')=',LINE(I)
C READ(*,*)
C WRITE(*,130)'NODO IZQUIERDO=',LINE1(I)
C READ(*,*)
C WRITE(*,130)'NODO DERECHO=',LINE2(I)
C READ(*,*)
85 CONTINUE
C READ(*,*)
105 FORMAT(2X,A,I3,A,I5)
130 FORMAT(1X,A,I3)
140 FORMAT(//,2X,A)
RETURN
END
```

WWW.BDIGITAL.ULA.VE