



**UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
MAESTRIA EN INGENIERIA ESTRUCTURAL**

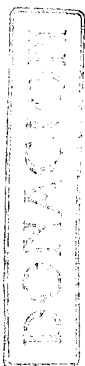


**PROCESO AUTOMATIZADO DE REFINAMIENTO H-ADAPTATIVO
BASADO EN INDICES DE ENERGIA DE DEFORMACION**

WWW.BDIGITAL.ULA.VE

Trabajo de grado presentado para optar al título de
Magíster Scientiae en Ingeniería Estructural

Ing. María Elisa Elberg Rojas



Mérida, Venezuela
Marzo de 2002

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

RESUMEN

El Método de Elementos Finitos y sus aplicaciones a la Ingeniería en cualquiera de sus ramas, es utilizado mundialmente para análisis de problemas de soluciones complejas. Al ser un método numérico aproximado, su solución se aproximará a la exacta dependiendo en gran parte del apropiado refinamiento de la malla, ya que la variación de los resultados puede ser afectada por la acumulación de errores de redondeo al tener una malla muy densa o al tener un número de elementos muy bajo obteniendo también resultados erróneos. Entonces, la escogencia del tamaño de los elementos finitos y su número determina la calidad de la solución numérica para el problema estudiado. En este trabajo se muestra el proceso automatizado del refinamiento de mallas del tipo H-adaptativo con base a Índices de Energía de Deformación obtenidos mediante un análisis por Elementos Finitos con valores conseguidos por el programa comercial ABAQUS. Se presentan resultados logrados mediante la creación y aplicación de un software desarrollado para lograr el refinamiento de mallas de elementos tridimensionales de cualquier complejidad. El programa que se utiliza está realizado en Fortran 90 que procesa los valores de energía de deformación, indicadores del refinamiento, logrando un nuevo mallado de mayor refinamiento sólo en la zona del modelo que presenta singularidades. Los valores obtenidos mediante mallado denso y uniforme para encontrar una solución apropiada, muestra elevado tiempo computacional. El conjunto formado por PlusRun, PlusOne y ABAQUS puede ser utilizado fácilmente para cualquier modelo, siguiendo la ayuda dada por una Guía de Usuario creado específicamente para ayudar en la utilización del proceso automatizado de refinamiento tipo H-adaptativo.

No es que no nos atrevemos
porque las cosas son difíciles,
en realidad son difíciles
porque no nos atrevemos.
Séneca

WWW.BDIGITAL.ULA.VE

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

AGRADECIMIENTOS

Mi especial gratitud al Prof. Héctor Febres Cedillo por su constante estímulo y orientación para la realización de este Trabajo de Grado.

Quiero agradecer también a los profesores Kay Tucci, Julio Florez López, Pether Inglessis, Rafael Febres C., Alexis López, José Alvarado, Carlos Quintero, Fernando Sarmiento, Eduard Thomsom y Francisco Hidrobo.

Y por supuesto el agradecimiento individual a cada uno de los miembros de mi familia que me brindaron el apoyo y tiempo necesarios para la culminación de la Maestría en Ingeniería Estructural.

CONTENIDO

RESUMEN

INTRODUCCIÓN

CAPITULO I. ASPECTOS GENERALES

- I.1 Formulación del Problema
- I.2 Objetivos de la Investigación
 - I.2.1 Objetivo Principal
 - I.2.2 Objetivos Secundarios
- I.3 Justificación e importancia de la Investigación

CAPITULO II. MARCO REFERENCIAL

- II.1 Antecedentes
- II.2 Bases teóricas
 - II.2.1 Método de Elementos Finitos
 - II.2.2 Principio del Trabajo Virtual
 - II.2.3 Estrategias de Refinamiento de mallas
 - II.2.4 ABAQUS

CAPITULO III. METODOLOGÍA

- III.1 Etapas de la Investigación
- III.2 Algoritmos
 - III.2.1 Diagrama General
 - III.2.2 PlusRun
 - III.2.3 PlusOne

CAPITULO IV. RESULTADOS

- IV.1 Resultados numéricos
 - IV.1.1 Modelo original
 - IV.1.2 Primer ciclo
 - IV.1.3 Segundo ciclo
 - IV.1.4 Tercer ciclo

- IV.2 Ejemplos Adicionales
 - IV.2.1 Ejemplo No. 1
 - IV.2.2 Ejemplo No. 2

- IV.3 Guía de Usuario

CAPITULO V. CONCLUSIONES Y RECOMENDACIONES

WWW.BDIGITAL.ULA.VE
LITERATURA CONSULTADA

APENDICES

Apéndice A

- Listado de archivos del primer ciclo
- Listado de archivos del segundo ciclo
- Listado de archivos del tercer ciclo

Apéndice B

- Listado del programa PlusRun

Apéndice C

- Listado del programa PlusOne



INTRODUCCIÓN

El presente trabajo tiene la finalidad de mostrar el proceso de automatización del refinamiento H-adaptativo basado en Índices de Energía de Deformación obtenidos mediante un análisis por Elementos Finitos con resultados logrados por el programa comercial ABAQUS.

Este método de refinamiento ha sido estudiado por diversos autores entre los cuales se pueden citar H. Febres, A. Buenaño y C. Meléndez (1997), quienes han desarrollado software para el proceso antes mencionado.

Los programas desarrollados no presentan interacción automática con los indicadores de refinamiento, es por esto que el proceso puede llegar a ser engorroso y lento para lograr un apropiado grado de refinamiento en elementos complejos con geometrías compuestas por elevado número de nodos y elementos.

Esta investigación presenta un software desarrollado para lograr la automatización del proceso para elementos tridimensionales con interacción automática a ABAQUS.

El programa principal que se muestra, realizado en lenguaje FORTAN 90, logra automáticamente la lectura de los valores de energía de deformación, indicadores de refinamiento, logrando un nuevo mallado refinado en la zona donde se presentan elevados gradientes.



La ciencia computacional ha avanzado enormemente, logrando lenguajes de programación y sistemas operativos de alto nivel. Es por esto que en esta investigación se codifican dos programas que facilitan la interacción buscada, donde cada uno de ellos ejecuta una parte del proceso de automatización llevando al grado máximo la optimización del proceso, en tiempo de ejecución y aplicación apropiada de los lenguajes y sistemas operativos. Por esto se utiliza UNIX y FORTRAN 90. UNIX, como sistema operativo, usado mediante el intérprete de comandos SHELL, es decir, el programa que interpreta las peticiones del usuario, y FORTRAN 90 como lenguaje de programación elegido por ser un lenguaje científico y de alto nivel.

En esta investigación son creados los programas PlusRun en UNIX (sh) y PlusOne en Fortran 90.

- PlusRun como programa Master que dirige y activa la ejecución de PlusOne y la activación de ABAQUS de forma automática.
- PlusOne como programa encargado de realizar el proceso de refinamiento obteniendo un nuevo mallado para diversos ciclos del proceso.
- ABAQUS, programa comercial, que ejecuta y genera archivos utilizados por PlusOne.

En las páginas siguientes se muestran las especificaciones de cada uno de ellos, las características del equipo necesarias para su ejecución y una guía de usuario.

El trabajo se encuentra dividido en cinco (5) capítulos, que introducen progresivamente a los conceptos y procedimientos empleados para la realización de los programas y su implementación.

El Capítulo I presenta los aspectos generales de la investigación donde se detalla la formulación del problema a estudiar, presenta los objetivos de la investigación, su justificación e importancia.

El Capítulo II muestra el marco referencial donde se incluyen las antecedentes y las bases teóricas aplicables al trabajo. Se incluyen conceptos fundamentales sobre el Método de Elementos Finitos y energía de deformación,

Licencia Creative Commons:

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



junto con estrategias de refinamiento y una breve reseña sobre el programa comercial ABAQUS.

El Capítulo III presenta la metodología aplicada así como las diversas etapas que comprenden el presente trabajo y los algoritmos de los programas realizados.

El Capítulo IV muestra los resultados numéricos obtenidos para ejemplos encontrados en la literatura, como comprobación del método, y de nuevos ejemplos donde se especifican ciertas singularidades en el modelo para orientar el grado de discretización. Se muestra también en este capítulo una Guía de Usuario para que el software creado pueda ser utilizado por usuarios interesados en encontrar un mallado apropiado para problemas a resolver por medio de la Teoría de Elementos Finitos.

Por último, el Capítulo V muestra las conclusiones y recomendaciones generadas a partir de los resultados obtenidos del proceso automatizado de refinamiento tipo H-adaptativo.

La literatura consultada se encuentra al final de trabajo junto con los apéndices: el Apéndice A contiene el listado de los archivos generados por PlusOne en cada ciclo de refinamiento, el Apéndice B proporciona el listado del programa PlusRun y, el apéndice C tiene el listado del programa PlusOne.



CAPITULO I

ASPECTOS GENERALES

I. Formulación del Problema

Todo análisis realizado con el Método de Elementos Finitos (MEF) exige conocimiento del problema a estudiar y manejo del Método en sí.

Es por esto que el procedimiento para realizar satisfactoriamente un análisis utilizando MEF, puede resumirse en lo siguiente:

- Conocer las características geométricas y mecánicas del problema a estudiar
- Generar un modelo apropiado al problema real en estudio
- Seleccionar una malla inicial
- Formular las propiedades de cada elemento
- Aplicar las condiciones de borde del modelo
- Aplicar las cargas al modelo
- Resolver las ecuaciones algebraicas lineales para determinar el valor de las incógnitas nodales
- Analizar el comportamiento del modelo
- Seleccionar una nueva malla de mayor refinamiento, generalmente.

Licencia Creative Commons:

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

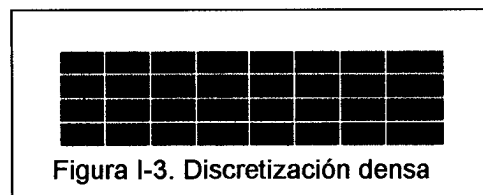
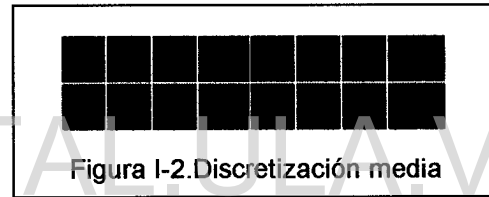
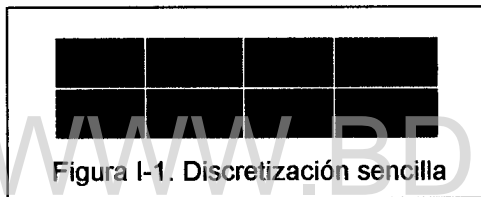


- Repetir el proceso hasta que la diferencia de dos análisis sucesivos este dentro de una tolerancia permitida.

Es por esto, que para un adecuado análisis por elementos finitos, un punto fundamental es la escogencia del mallado apropiado.

La escogencia del tamaño de los elementos finitos determina la calidad de la solución numérica para el problema estudiado.

La Figura I-1 muestra un elemento bidimensional con una discretización sencilla, las Figuras I-2 y I-3 presentan discretizaciones algo mas complejas.



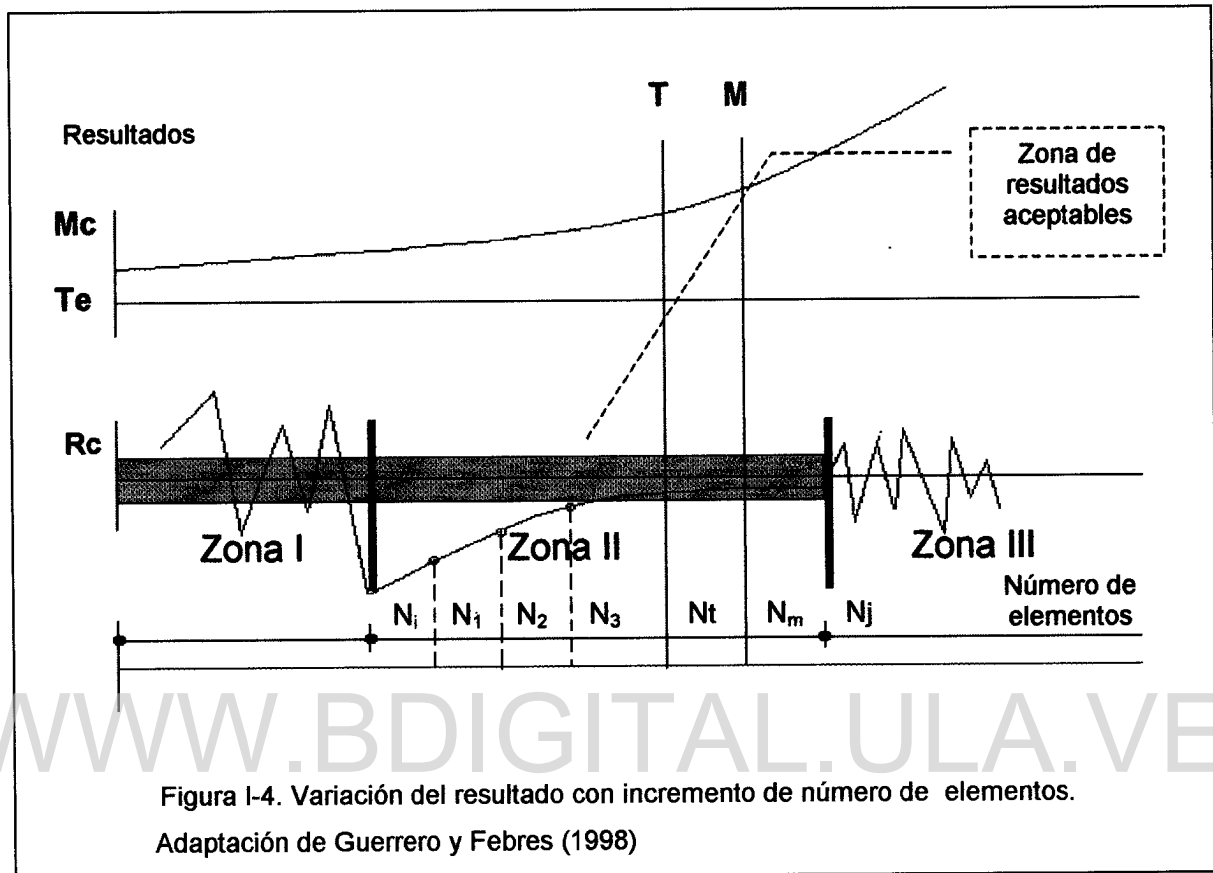
Mallas muy sencillas no arrojaran buenos resultados, pero si se realizan discretizaciones muy densas o mallas muy finas, se genera el problema de tener el aumento del número de grados de libertad a medida que las mallas se van refinando.



El sistema de ecuaciones lineales se incrementa a medida que la malla es refinada. Aún con una óptima numeración de las incógnitas nodales, la malla de elementos finitos pronto tendrá un tamaño impráctico desde el punto de vista computacional en términos de tiempo y almacenamiento.

Las variaciones de los resultados con el incremento del número de elementos se presentan en la Figura I-4 y muestra la existencia de tres zonas:

- ✓ Zona I: Cuando el número de elementos es pequeño ($N_e < N_i$) los resultados no son confiables, esta zona se ha denominado zona de representación impropia.
- ✓ Zona II: El incremento en el número de elementos por encima de N_i lleva a una gradual convergencia de los resultados. Esta zona se conoce como zona de representación propia
- ✓ Zona III: El incremento del número de elementos por encima de N_j conlleva la acumulación de errores por redondeo, y en consecuencia los resultados pierden la convergencia.



Donde:

Mc: memoria del computador

Te: tiempo de ejecución

Rc: resultado correcto

T: restricción por tiempo del computador

M: restricción de memoria del computador.

Ni: Número de elementos en el cual comienza la convergencia hacia el resultado correcto

N₁, N₂, N₃: Número de elementos en cada discretización

N_t: Número máximo de elementos debido a tiempo de ejecución

N_m: Número máximo de elementos debido a memoria del computador

N_j: Número de elementos en el cual comienzan la acumulación de errores por redondeo.



Lo anterior evidencia claramente que el número de nodos, entendiendo esto como el grado de refinamiento de la malla y el tipo de elemento es un punto central en el correcto análisis por MEF. Es por esto que esta investigación se centra la creación de refinamiento denso en zonas exclusivamente necesarias, esto es, donde existan singularidades y por consiguiente concentraciones de energía, logrando resultados en rangos aceptables con menor número de elementos que mallas altamente refinadas y uniformes.

I. 2 Objetivos de la Investigación

I.2.1 Objetivo Principal

El objetivo principal de este trabajo consiste en codificar un software para la automatización de la interacción con el programa comercial ABAQUS, y, el proceso de refinamiento H-adaptativo basado en Índices de Energía de Deformación (SED) utilizando la técnica de macro elementos, con el propósito de encontrar la discretización óptima del modelo. Por tanto se intenta buscar la mayor exactitud numérica y el menor costo computacional y la malla mas adecuada.

I.2.2 Objetivos Secundarios

- Verificar y modificar las subrutinas existentes en Fortran para el Proceso de Refinamiento tipo H.
- Realizar un programa en Fortran 90 que tome en cuenta las subrutinas existentes modificadas y las nuevas necesarias el proceso automatizado. Programa PlusOne.

Licencia Creative Commons:

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



- Direccionar las activaciones de PlusOne y ABAQUS mediante un programa de comandos en el C-SHELL de UNIX. Programa PlusRun.
- Comparar concentraciones de energía de deformación y esfuerzos principales en las zonas de singularidades y altos gradientes.
- Comparar resultados obtenidos por el proceso automatizado y ejemplos de la literatura consultada.

I.3 Justificación e importancia de la Investigación

El actual desarrollo computacional hace que cualquier problema de índole matemático se simplifique grandemente. Anteriormente, para realizar cálculos importantes la ayuda fundamental era un Regla de Cálculo y los problemas a resolver, según su magnitud, podían tomar un tiempo considerable. Al crecer la potencialidad de las herramientas de cálculo, crecieron también las expectativas de los resultados a obtener soluciones esperadas de análisis aún mayores. El desarrollo de algunas Teorías y Métodos fue prácticamente a la par con el desarrollo computacional. Las máquinas actuales con las que cuenta la Universidad de Los Andes (ULA) en el Centro de Cálculo de la ULA (CeCalcULA), IBM SP2 y SGI Origin 2000, por ejemplo, permiten análisis por MEF en tiempos prudencialmente cortos. Estableciendo comparaciones con épocas anteriores, al aparecer la Regla de Calculo, tal vez se pensó que nada podía mejorar tal velocidad para lograr un cálculo y análisis cualquiera. Es por esto que esta investigación pretende aportar en el mejoramiento y optimización del uso de computadoras de gran potencia. Reduciendo los tiempos de ejecución, reduciendo el uso de memoria computacional y logrando resultados de igual o mejor calidad que en modelos de mallas densas. La automatización del refinamiento tipo H-adaptativo, en modelos tridimensionales permitirá dar un paso adelante en la línea de investigación del MEF y servirá como inicio en estudios de problemas complejos que necesiten un mallado altamente ajustado a singularidades importantes.

Licencia Creative Commons:

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



CAPITULO II

MARCO REFERENCIAL

II.1 Antecedentes

Lakshmanam y otros (1997) afirman que con el Método de Elementos finitos no es posible diseñar una malla perfecta la primera vez, especialmente si se trata de modelos complejos. Por su parte, A. Umar y otros (1996) establecen que 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.

El desarrollo de procesos adaptativos ha recibido en la última década especial atención tanto por parte de los investigadores 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 lo problemas de ingeniería de una manera eficiente, confiable y económica, a un costo computacional bajo y que además requiera la menor intervención posible del usuario. [Meléndez y Febres, 1999]

Lee y Lo (1992) presentan un esquema generalizado donde un proceso adaptativo debería constar de los siguientes módulos:

Licencia Creative Commons:



- ◆ Módulo para el análisis por elementos finitos
- ◆ Módulo indicador de refinamiento y estimación del error
- ◆ Módulo para generación de mallas adaptativas.

Selman, Hinton y Bicanic (1997) consideran que un proceso de refinamiento debe constar de:

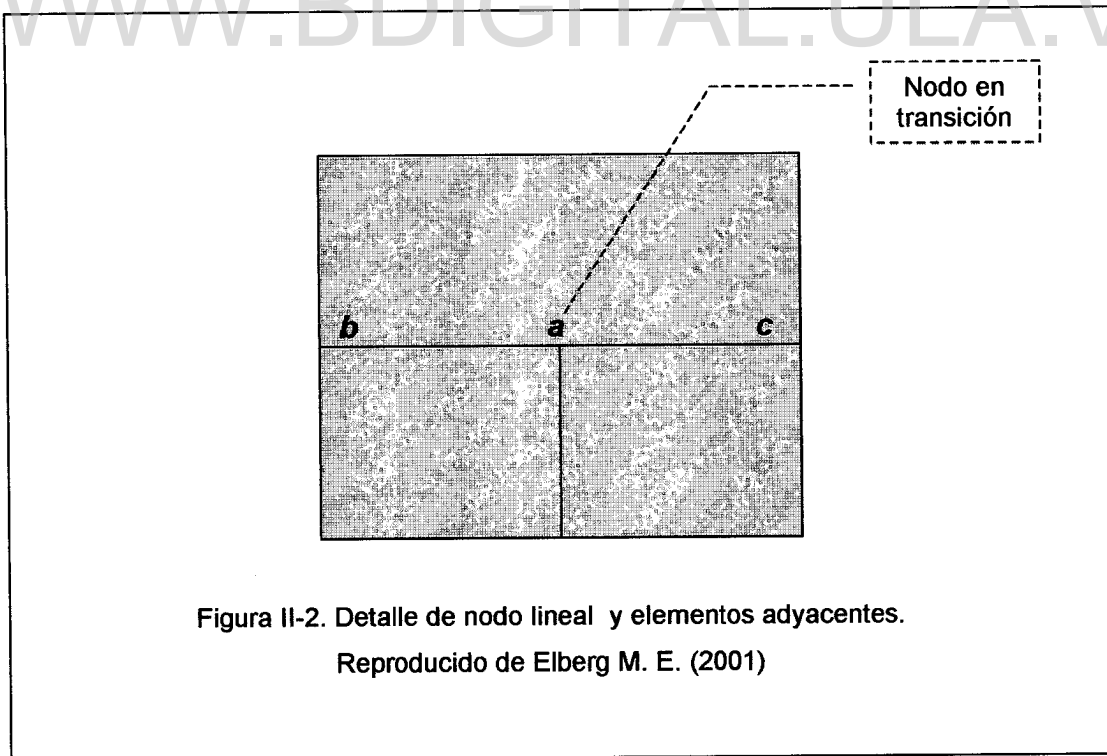
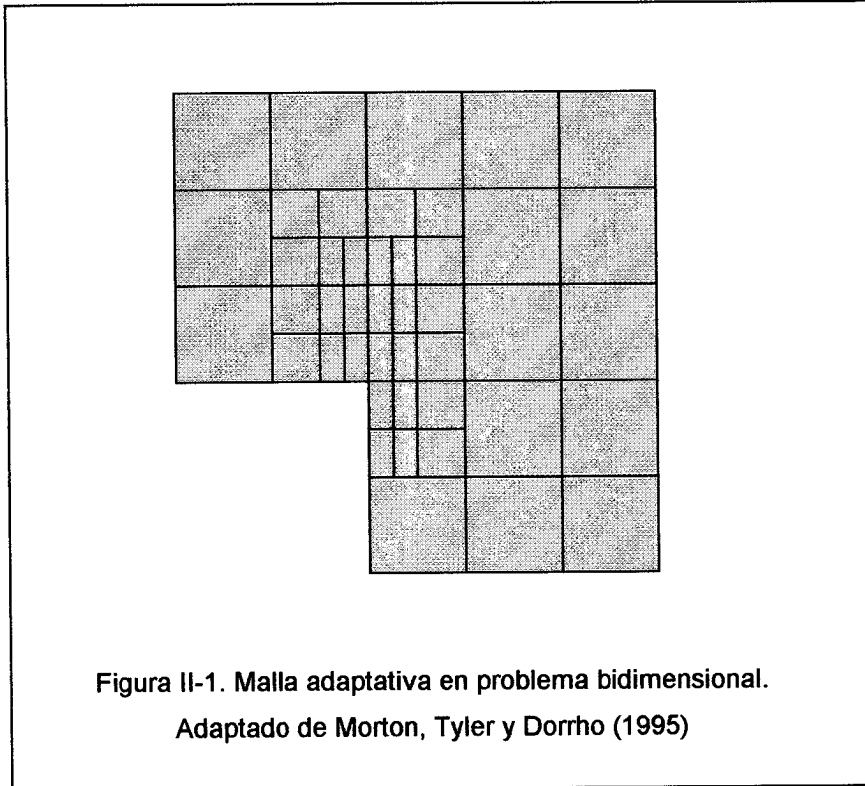
- ◆ Módulo para indicador de refinamiento local
- ◆ Módulo convertidor de la información del refinamiento de la malla
- ◆ Módulo generador del mallado automático facilitador de la producción de mallas con densidad establecida.

Choy y Lee (1996), afirman que el objetivo del diseño de cualquier proceso adaptativo es obtener automáticamente una malla óptima, de tal manera que el error pueda ser minimizado para un número de grados de libertad y alcanzar económicamente la relación máxima de convergencia de las aproximaciones de elementos finitos.

Morton, Tyler y Dorroh (1995) muestra un ejemplo de malla adaptativa en un problema bidimensional (Figura II.1), donde se utiliza el H-refinamiento en una malla irregular en el que se tiene uno o dos elementos adyacentes en cada lado.

La Figura II.2 muestra en detalle un nodo lineal y sus correspondientes elementos vecinos. Al punto mostrado **a** le son signadas las mismas restricciones que poseen los puntos **b** y **c**. Este nodo es considerado lineal.

Meléndez y Febres (1999) hacen referencia a los nodos bilineales, los cuales son característicos de problemas tridimensionales. La Figura II.3 muestra nodos bilineales y elementos definen sus restricciones.





Los mismos autores detallan la forma de una n -ridimensi irregular, donde citan a Devloo, Oden y Strouboulis (1987) que indican que la solución del nodo de n -ridimensi es condicionada a la solución de los nodos cercanos.

Las Figuras II-3 y II-4, muestran un modelo n -ridimensional donde Meléndez y Febres (1999) realizaron un refinamiento tipo H-adaptativo.

Por otro lado, diversos autores refieren sus trabajos de estrategias de refinamiento a creación de software para realizar mallados apropiados a partir de n -ridimensional específicas.

H. Febres (1984) utiliza exitosamente por primera vez los índices de n -ridimen de energía de n -ridimensio (SED) como un indicador de refinamiento en problemas de elasticidad plana.

Buenaño y Febres (1995) aplica la misma base teórica para lograr procesos de refinamiento en placas deflectadas y Meléndez (1999) realiza procesos para refinamiento en problemas n -ridimensionales.

WWW.BDIGITAL.ULA.VE

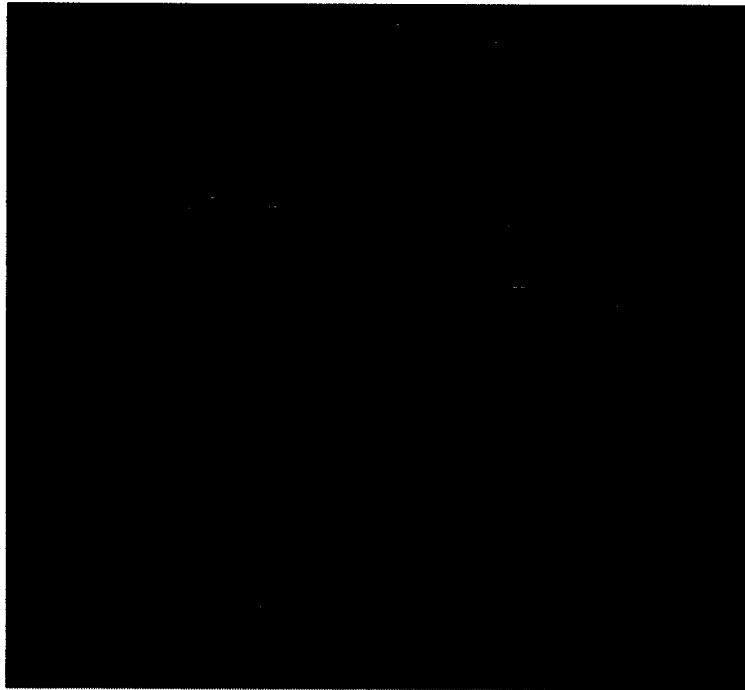


Figura II-3. Refinamiento H-adaptativo de unión viga columna.

Adaptada de Meléndez (1999)

WWW.BDIGITAL.ULA.VE

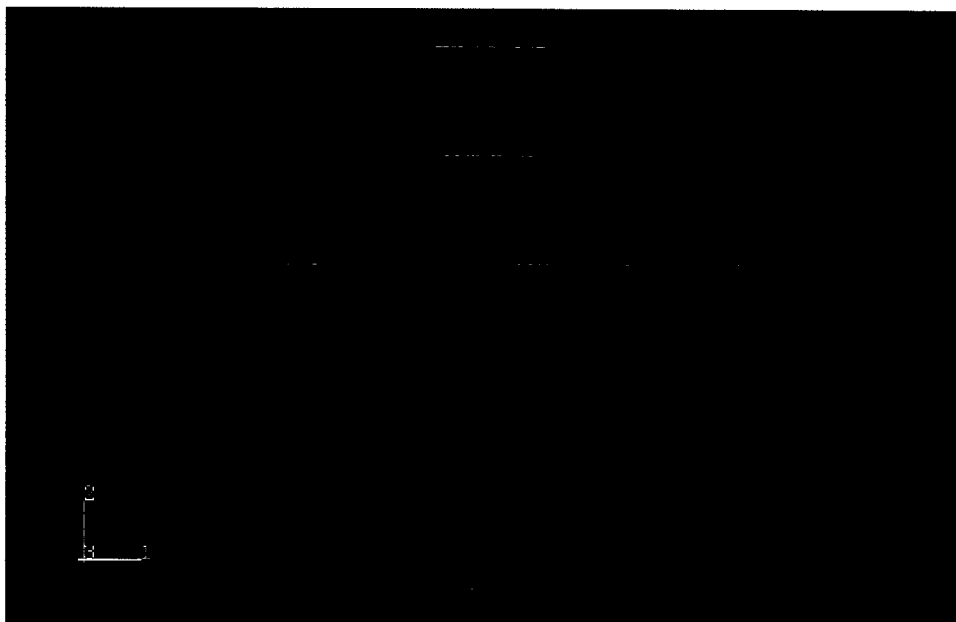


Figura II-4. Vista bidimensional de mallado de junta viga-columna

Adaptada de Meléndez (1999)



II.2. Bases Teóricas

Las bases teóricas de este trabajo han sido definidas en orden de presentar las consideraciones fundamentales para puntualizar y revisar conceptos y relaciones importantes que definen el Método de Elementos Finitos, las estrategias de refinamiento y el generalidades del programa comercial ABAQUS ya que es una herramienta importante en el desarrollo de esta investigación.

II.2.1 Método de Elementos Finitos

El Método de Elementos Finitos es utilizado mundialmente para análisis de problemas de geometrías y condiciones de contorno muy complejas. Se usa principalmente para resolver problemas para los cuales no existe una solución exacta. En Ingeniería se usa para análisis de esfuerzos y deformaciones, simulación de estratos de terreno, conducción de calor, fluidos en movimiento y modelados para bioingeniería, entre otros.

Un análisis típico se basa en la creación de la geometría de la estructura, de la aplicación de las condiciones de borde y de las cargas y de establecer un mallado o subdivisión del continuo escogiendo cuidadosamente los elementos a utilizar y su tamaño ya que podrían generarse resultados erróneos debido a un modelo defectuoso.

Actualmente, el análisis por Elementos Finitos de problemas complejos en cualquier rama de la Ingeniería se encuentra en su apogeo debido la versatilidad del método y a la gran cantidad de programas comerciales que se han creado.

El procedimiento numérico de cálculo implica ecuaciones algebraicas simultáneas las cuales son generadas y resueltas generalmente por estos programas comerciales.

La interpretación de los resultados es la parte fundamental, pues los valores obtenidos pueden ser lógicos o tremendamente absurdos.



El Método de basa en el principio de que cualquier problema físico puede ser dividido en un cierto número de áreas o volúmenes. Estas divisiones son llamadas Elementos Finitos.

La Figura II.5 muestra la geometría de una estructura sencilla. La calidad de detalle con que se diseñe la geometría del modelo es decisión del analista.

La Figura II.6 presenta el mallado compuesto de elementos tridimensionales.

Con la Teoría de Elementos Finitos se puede convertir un problema de un número infinito de grados de libertad a uno con un número finito.

Un Elemento Finito es entonces una subregión de un continuo discretizado y, la malla es, por lo tanto, el ensamblaje de los Elementos Finitos conectados entre si por puntos comunes o nodos.

Los elementos finitos pueden ser unidimensionales, bidimensionales y tridimensionales. Teniendo cada uno de ellos una definición específica para cada tipo de **problema** a resolver.

Podemos encontrar problemas de una (1), dos (2) y tres (3) dimensiones.

- ◆ Un Problema Unidimensional (1D) puede ser definido cuando una de las tres dimensiones es representativamente mayor que las otras dos. Es por esto que la representación de una viga suele ser una línea recta.

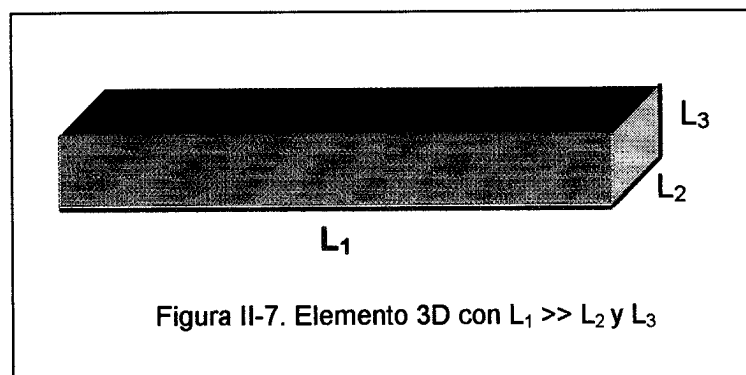
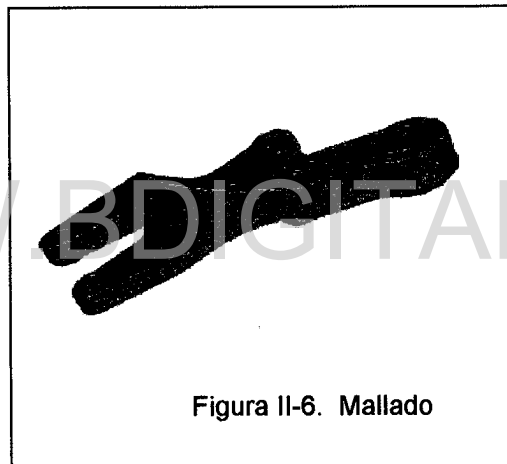
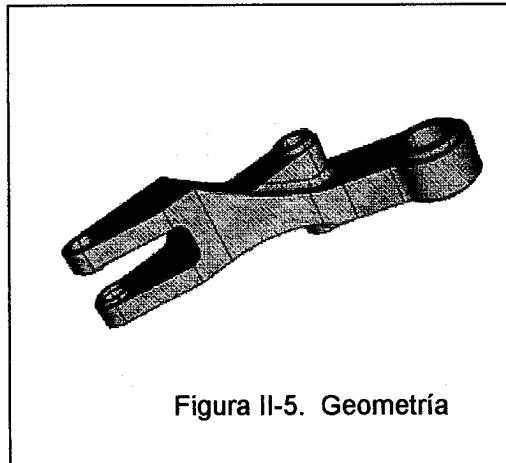
La Figura II-7 muestra un elemento tridimensional con $L_1 \gg L_2$ y L_3 y la Figura II-8 presenta la representación unidimensional.

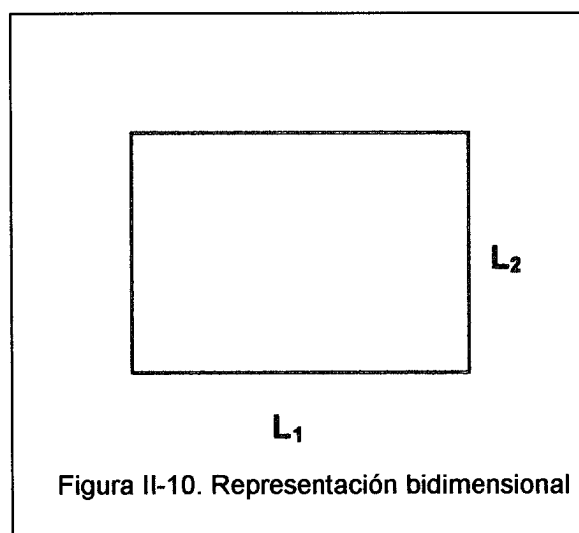
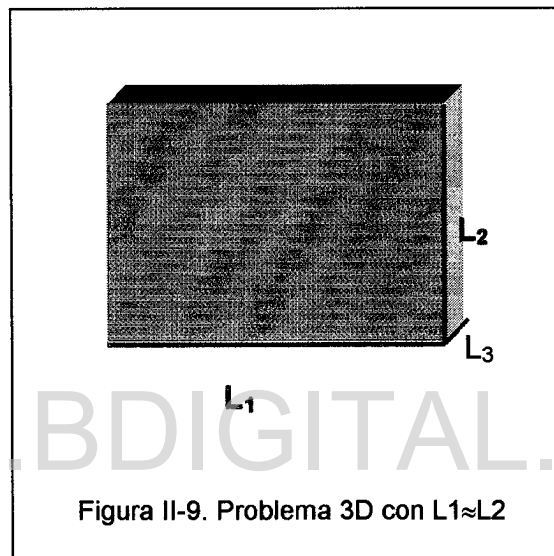
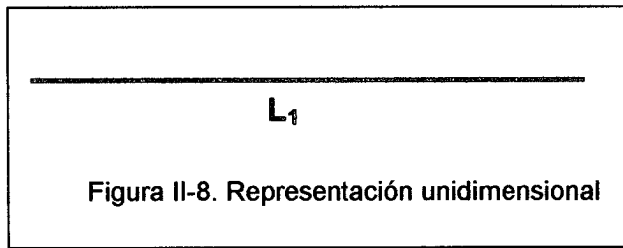
- ◆ Un Problema Bidimensional (2D) es considerado cuando dos dimensiones son similares entre si. En las Figuras II.9 y II.10 se muestra un elemento tridimensional con $L_1 \approx L_2$ y su representación bidimensional, respectivamente.

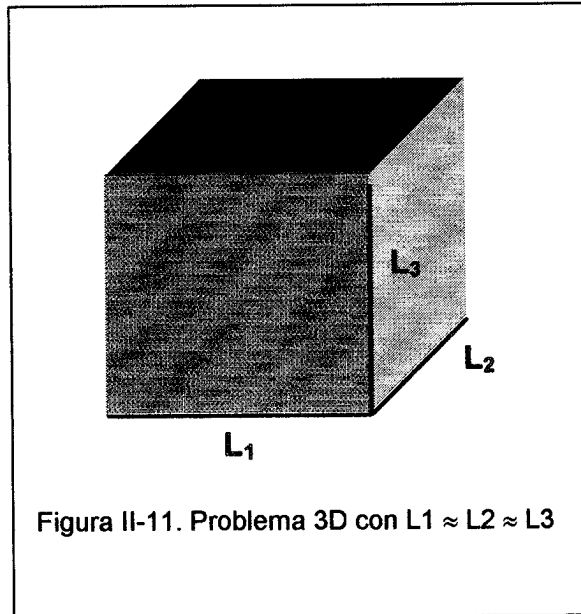
- ◆ Un problema Tridimensional (3D) se presenta cuando las tres dimensiones son aproximadamente iguales. Esto puede observarse en las Figuras II.11 y II.12.

Licencia Creative Commons:

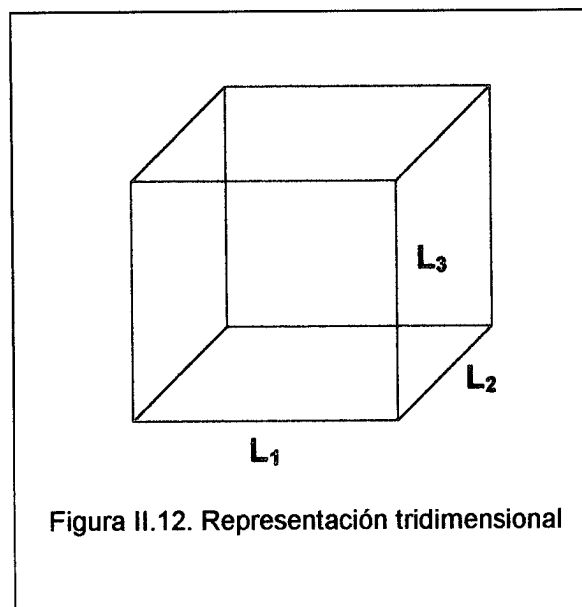
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)







WWW.BDIGITAL.ULA.VE





La consideración de estos tres tipos de problema depende exclusivamente del usuario y la optimización que requiera del modelo que realiza.

Problemas unidimensionales pueden perfectamente ser modelados como problemas de dos o tres dimensiones, dependiendo del tipo de análisis y de resultados que deseen obtenerse.

De igual manera problemas 3D pueden ser modelados como 1D y/o 2D. El uso de simetría y condiciones de contorno (cargas, desplazamientos) son fundamentales para realizar un modelo apropiado.

Luego de categorizar el problema a resolver, es importante el uso correcto del tipo de Elemento Finito apropiado para cada caso.

- **Elementos Unidimensionales:**

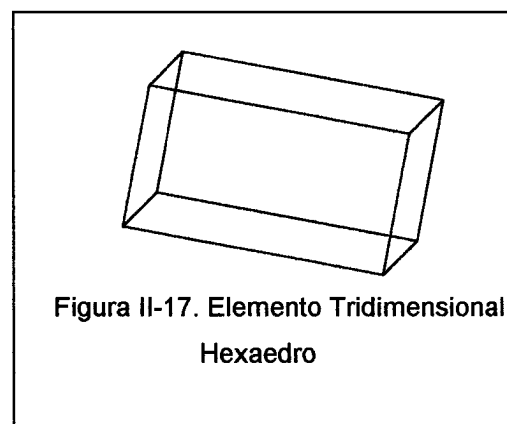
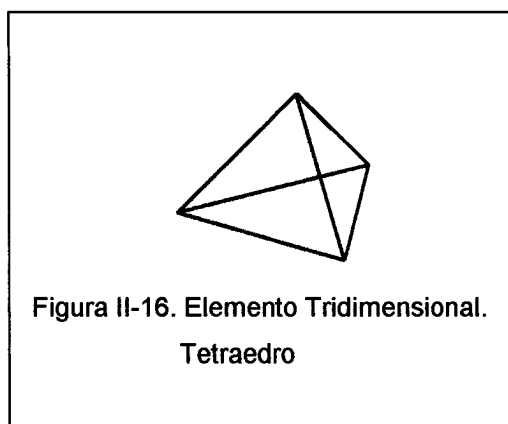
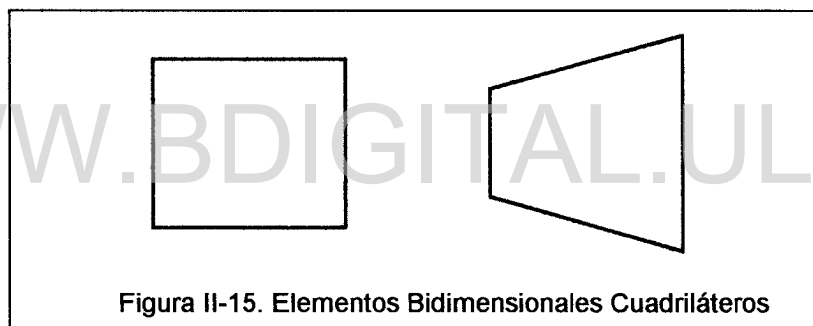
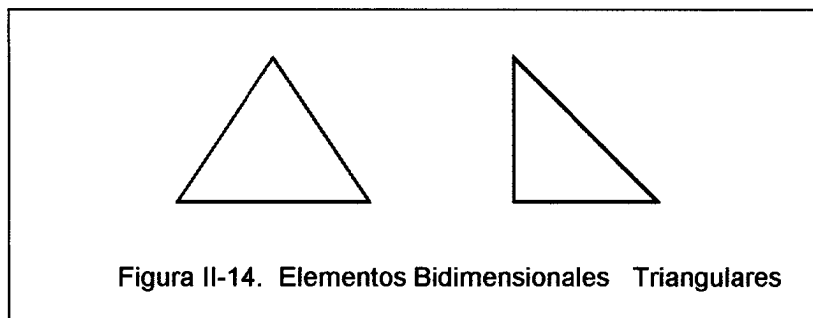
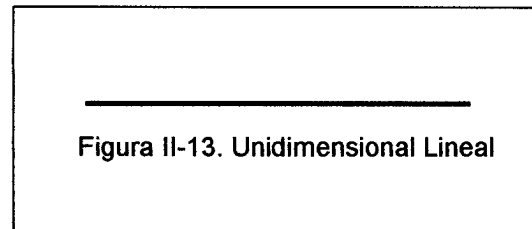
Son elementos cuya forma geométrica es lineal. Es utilizado principalmente para barras y vigas. (Figura II-13)

- **Elementos Bidimensionales:**

Pueden ser Triangulares o Cuadriláteros. Son utilizados para el modelado de placas o membranas, principalmente. (Figura II-14)

- **Elementos Tridimensionales:**

El mas utilizado es el Hexaedro, pero también se tiene el Tetraedro. Se usan para sólidos 3D y placas gruesas, entre muchas otras. (Figuras II-16 y II-17)





Los Nodos, no sólo son definidos en los puntos de conexión de los Elementos Finitos, existen también los Nodos Internos que son utilizados matemáticamente para mejorar la solución del problema. Un Cuadrilátero no solo se define con cuatro (4) nodos, ya que puede tener cualquier número de ellos. A esto se le denomina el Orden del Elemento.

El análisis de este ensamblaje, provee de los desplazamientos 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 corresponden a este elemento. Cada nodo, tiene un número determinado de posibles movimientos llamados Grados de Libertad. Pueden considerarse tres (3) desplazamientos (en las direcciones X, Y y Z) y tres (3) rotaciones (alrededor de los ejes X, Y y Z), por ejemplo, como también otros grados de libertad: temperatura, presión de poros, potencial eléctrico entre muchos otros.

Conocidos entonces los desplazamientos, pueden conocerse las reacciones, las deformaciones y los esfuerzos. Este procedimiento de cálculo requiere gran cantidad de esfuerzo computacional, es por esto que los modelos deben ser optimizados antes de comenzar el proceso de análisis computacional.

La Figura II.18 muestra un elemento representativo de volumen para el caso tridimensional. Entonces, considerando un problema en 3D, en general, pudiendo hacer simplificaciones para casos 2D y 1D, se tiene para cada elemento:

- | | |
|--------------------------------|--|
| Los vectores de desplazamiento | $\{U\}^t = (U_1, U_2, U_3),$ |
| La matriz de deformaciones | $\{\varepsilon\}^t = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{23}, \varepsilon_{31})$ y |
| La matriz de esfuerzos | $\{\sigma\}^t = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{31})$ |

Que son las incógnitas de cada elemento representativo del volumen considerado.

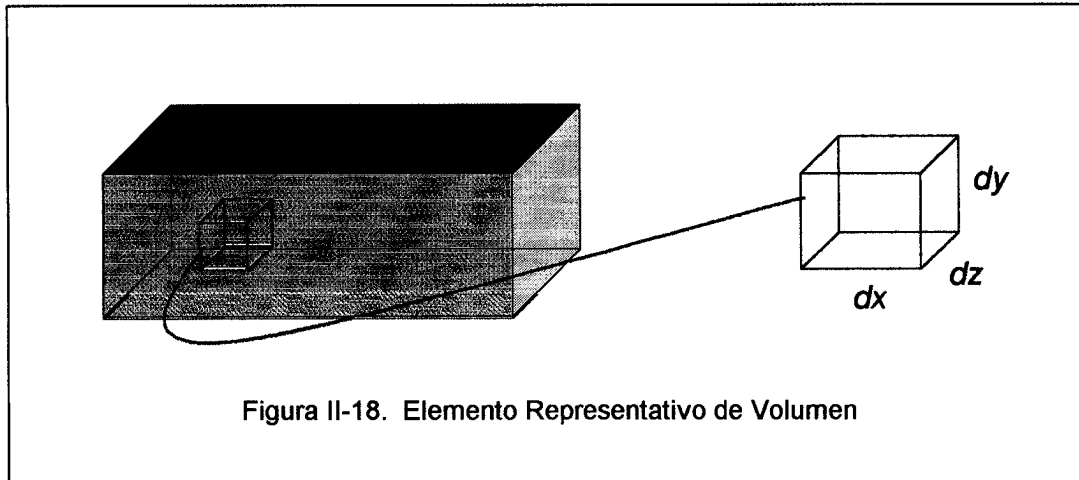


Figura II-18. Elemento Representativo de Volumen

WWW.BDIGITAL.ULA.VE

Ahora bien, para el problema completo es necesario considerar

El campo de desplazamientos: $U_i = U_i(x_1, x_2, x_3)$,

El campo de deformaciones: $\varepsilon_{ij} = \varepsilon_{ij}(x_1, x_2, x_3)$ y

El campo de esfuerzos: $\sigma_{ij} = \sigma_{ij}(x_1, x_2, x_3)$

Los datos que se deben incluir en el modelo son:

- Geometría del sólido
- Apoyos y sus desplazamientos $\{d\}$
- Fuerzas externas $\{F_s\}$ aplicadas sobre la superficie libre
- Fuerzas externas $\{F_d\}$ aplicadas sobre el dominio
- Material del sólido (Coeficientes E, ν , etc.)



Para la resolución, se cuenta con las ecuaciones del modelo, mostradas en este caso para análisis en rango elástico, pudiendo aumentar la complejidad hasta incluir plasticidad en cualquiera de los modelos conocidos, también fractura o daño, por ejemplo.

- Ecuaciones cinemáticas: relaciones entre desplazamientos y deformaciones, considerando que no hay cambio de forma sin movimiento.

Tensor de deformaciones de Green-Lagrange:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} + \frac{\partial U_1}{\partial x_i} \frac{\partial U_1}{\partial x_j} + \frac{\partial U_2}{\partial x_i} \frac{\partial U_2}{\partial x_j} + \frac{\partial U_3}{\partial x_i} \frac{\partial U_3}{\partial x_j} \right)$$

Tensor de deformaciones infinitesimales:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

- Teorema de trabajos virtuales, ecuaciones de equilibrio:

$\{F_1\}^t\{U^*\} + \{F_2\}^t\{U^*\} + \dots + \{F_n\}^t\{U^*\} = 0$ interpretado como el trabajo realizado por la fuerza F sobre el desplazamiento virtual U*

$\{F_1\} + \{F_2\} + \dots + \{F_n\} = 0$ interpretado como el equilibrio de todas las fuerzas actuantes.

- Leyes de comportamiento elásticas considerando que los esfuerzos producen deformaciones, esto es:

Ley de Hooke generalizada: $\{\sigma\} = [K] \{\varepsilon\}$ donde [K]= matriz de coeficientes del material

Por otro lado, el mallado apropiado es de gran importancia llegando a diferir muchas veces los resultados para discretizaciones diferentes.

Licencia Creative Commons:



El proceso matemático que realiza el computador se refiere al análisis del ensamblaje de la estructura, y a la obtención de los desplazamientos para cada nodo dependiendo de las cargas aplicadas y condiciones de borde del modelo. Los desplazamientos para otros puntos son interpolados de los valores del nodo que corresponden al elemento.

Con esto, pueden ser calculadas reacciones, desplazamientos, deformaciones y esfuerzos para nodos y/o elementos del modelo.

II.2.2 Energía de Deformación

El uso de los teoremas energéticos es planteado por Zienkiewicz (1994), donde se considera el Teorema de la Energía Potencial, esto es:

$$\Pi = U + W$$

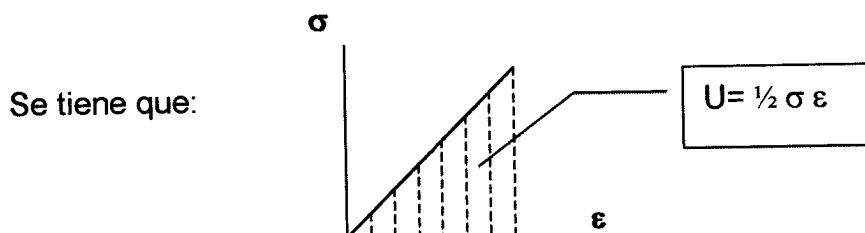
Donde:

Π : Energía Potencial

U : Energía de Deformación Interna

W : Energía Potencial de las Cargas

La Energía de Deformación (U) es definida de la siguiente manera:



Con base a la Ley de Hooke:

$$\sigma = \epsilon E$$



Donde:

σ : esfuerzos

ε : deformaciones

E: módulo de Young

Se obtiene:

$U = \frac{1}{2} \sigma \varepsilon = \frac{1}{2} E \varepsilon^2$, que generalizando para todo el volumen:

$$U = \frac{1}{2} \iiint E \varepsilon^2 dv \quad \text{por lo tanto} \quad U = \frac{1}{2} E A \int_0^L \varepsilon^2 dx$$

Y por otro lado $\varepsilon = du / dx$ se obtiene $U = \frac{1}{2} E A \int_0^L (du / dx)^2 dx$

Es decir que la Energía Potencial es función de los desplazamientos: $\Pi = \Pi(u)$

Tomando en cuenta el principio de la Mínima Energía Potencial, el cual establece que,

“ De todos los posibles campos de desplazamientos cinemáticamente admisibles, aquellos que satisfacen las ecuaciones de equilibrio minimizan la energía potencial total. “

Entonces, cuando la energía es minimizada con respecto a los desplazamientos, se definen directamente las ecuaciones de equilibrio:

$$\partial \Pi / \partial d_i = 0$$



El Índice de Densidad de Energía de Deformación (SED) lo define H. Febres (1984) para un *elemento*(ϵ) como:

$$\text{SED } \epsilon = U \epsilon / V \epsilon$$

Donde:

$U \epsilon$ = energía de deformación del *elemento*(ϵ)

$V \epsilon$ = volumen del *elemento*(ϵ)

Por otro lado, el Índice de Concentración de Energía Global (SEDG) se obtiene mediante la relación:

$$\text{SEDG} = \text{ETD} / \text{VT}$$

Donde :

ETD = Energía Total de Deformación

VT = Volumen Total

Se establece también un indicador (ϕ) para establecer comparaciones entre elementos, el cual se define de la siguiente manera:

$$\phi \epsilon = \text{SED } \epsilon / \text{SEDG}$$

Donde los elementos de refinamiento serán adicionados donde se cumpla que:

$$\phi \epsilon \geq \beta$$



Mantiene el número de elementos fijos e incrementa el grado del polinomio. Esto es, incrementa el orden de interpolación de los elementos.

- Refinamiento tipo H
Refinamiento local de la malla, incrementando el número de elementos. Se reduce el tamaño del elemento y se mantiene constante el grado del polinomio.
- Refinamiento tipo H-P
Generalmente se usa H-refinamiento hasta alcanzar una exactitud deseada y luego P-refinamiento sobre todo el dominio.
- Refinamiento tipo H-Method
Modificación del refinamiento H reduciendo el tamaño del elemento pero condicionado a las componentes de discretización del error.

II.2.4 ABAQUS

El análisis por Elementos finitos requiere gran potencialidad de hardware y software. En esto último, son muchos los programas creados para agilizar, mejorar y obtener resultados adecuados en tiempos prudencialmente cortos.

ABAQUS es un programa comercial creado para propósitos generales en análisis no lineal por Elementos Finitos (FEA, Finite Element Analysis). Ha sido desarrollado por Hibbitt, Karlsson & Sorensen, Inc. Este software es utilizado a nivel mundial para simular la respuesta física de estructuras y cuerpos ante cargas, temperatura, contacto, impacto y otras condiciones ambientales. También análisis de esfuerzos en diversos tipos de análisis estructurales, civiles, biomédicos, y muchas otras aplicaciones de la ingeniería.

Este programa de alto nivel, cuenta con diversos módulos altamente especializados para pre procesamiento, procesamiento en sí y post procesamiento. Estos son: ABAQUS/ Explicit, ABAQUS/ Standard, ABAQUS/ Aqua, ABAQUS/ Design, ABAQUS/ Safe, ABAQUS/ CAE, ABAQUS/ Viewer, ABAQUS/ ADAMS, ABAQUS/ CAT, ABAQUS/ C-MOLD, y ABAQUS/ MOLDFLOW.

Licencia Creative Commons:

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



Cada uno de ellos puede ser estudiado detenidamente en el Manual de Usuario de ABAQUS. ABAQUS (2001).

Para la investigación que se presenta es importante el conocimiento de este programa ya que el proceso automatizado creado utiliza a ABAQUS como subrutina generadora de valores de Energía de Deformación, llamados SENER.

Los SENER son los Índices de Energía de Deformación dados por ABAQUS. Corresponden a "*Elastic strain energy density, per unit volume*", ABAQUS (2001). Son obtenidos 8 valores de SENER por cada elemento C3D8, uno (1) por cada nodo.

ABAQUS genera diversos tipos de archivos, que dependiendo de lo deseado por el usuario pueden ser solicitados y revisados. Para esta investigación son de suma importancia los archivos de extensión *.inp* y *.dat*, siendo el primero un archivo ejecutable por ABAQUS y el contenedor de la información de la geometría, el mallado y las condiciones propias y principales del modelo. El segundo es generado según las especificaciones del primero y contiene las salidas numéricas de los resultados solicitados. Es en este archivo donde se obtienen los valores de los SENER.



CAPITULO III

METODOLOGIA

III.1 Etapas de la Investigación

El trabajo que se presenta utiliza una metodología que puede ser dividida en tres (3) etapas diferentes:

ETAPA I. Recopilación de información.

En esta fase, se realiza el estudio y la definición del problema a analizar, se recopila la información teórica y los antecedentes de investigaciones anteriores relacionadas con el tema.

Se realiza la verificación y modificación de subrutinas existentes en Fortran 90 para el Proceso de Refinamiento tipo H-adaptativo.



ETAPA II. Creación y uso del Software

Esta etapa consiste en la elaboración y prueba de los programas PlusRun y PlusOne encargados de automatizar el proceso de refinamiento H-adaptativo.

ETAPA III. Análisis de Resultados

Esta parte de la investigación consiste en la comparación de resultados obtenidos por el proceso automatizado creado y ejemplos de la literatura consultada.

III. 2 Algoritmos

El proceso automatizado de refinamiento H que se presenta, cuenta con tres herramientas fundamentales:

PlusRun: programa creado en CSH ejecutado bajo UNÍX. Es el encargado de direccionar y activar PlusRun y ABAQUS.

PlusOne: programa creado en Fortran. Se encarga de leer valores de los SENER dados en el archivo **.dat** y generar un nuevo archivo ejecutable por ABAQUS, el archivo **.inp**.

ABAQUS: programa comercial mencionado en la sección II.2.5.

III.2.1 Diagrama General

El proceso automatizado de refinamiento H-adaptativo, es descrito con el diagrama general siguiente que se muestra en la Figura III.1.



III.2.2 PlusRun

PlusRun es un programa creado para realizar el control y direccionamiento del proceso de refinamiento H-adaptativo automatizado.

Está realizado en el Shell de UNIX. Cuenta con tres (3) Bloques o unidades donde cada uno de ellos está realizado para cumplir específicas funciones del proceso automatizado de nueva discretización de malla que se realiza.

Unidad I. Verificación

Este bloque es el encargado de la confirmación de la presencia del programa **PlusOne** y del archivo **.inp** indispensables para iniciar el proceso de refinamiento. Todos deben estar en el mismo directorio de trabajo para poder iniciar el proceso. Verifica también la posibilidad de errores de sintaxis, ya que el archivo ejecutable por ABAQUS debe ser dado sin la extensión **.inp**. Esto es, el archivo de llama **plus.inp** debe ser escrito solamente **plus** en el línea de ejecución para cuando PlusRun es activado por el usuario. Por otro lado, si es cometido un error y se introduce **pluss** en lugar de **plus**, el error es detectado y envía un mensaje indicando que el archivo no existe.

Unidad II. Control de ciclos de refinamiento

PlusRun esta capacitado para activar automáticamente n ciclos de refinamiento y uno adicional inicial para activar la generación del archivo **.dat** producto del archivo **.inp** original. Dicho de otra forma, luego de las verificaciones originales, el programa activa ABAQUS que lee el **.inp** para dar como resultado un archivo **.dat** con la información de los SENER requeridos par realizar el refinamiento H.



El número de repeticiones de los ciclos de refinamiento puede ser modificado si el usuario así lo requiere.

Unidad III. Recopilación de archivos:

PlusRun guarda los archivos principales *.inp* y *.dat* generados en el proceso de refinamiento, entregando no solo un archivo final con el nuevo mallado, sino los archivos con los mallados obtenidos en los ciclos de proceso intermedios, esto con la finalidad de poder comparar resultados. Respalda los siete (7) archivos generados en cada ciclo:

plus.inp, plus.dat, plus.gen, plus.sen, plus.sed, plud.inp, plud.odb

Las unidades descritas anteriormente, se muestran en el esquema que se presenta a continuación por definición de colores: amarillo para Unidad I, azul para Unidad II y anaranjado para Unidad III.

WWW.BDIGITAL.ULA.VE

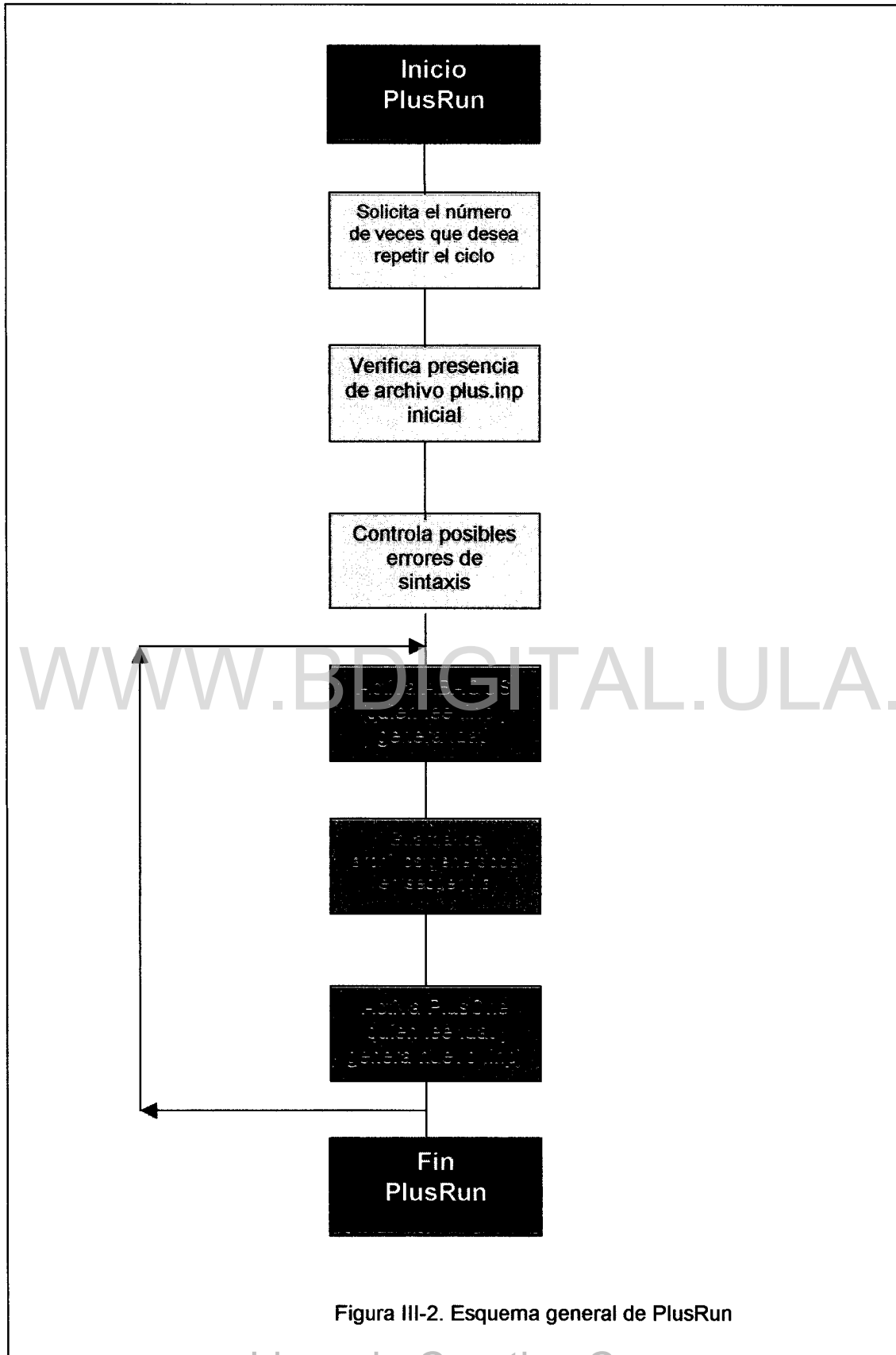


Figura III-2. Esquema general de PlusRun



III.2.3 PlusOne

El programa PlusOne es realizado en Fortran 90 y es el encargado de realizar el proceso de refinamiento de la malla, del tipo H-adaptativo.

PlusOne cuenta con tres (3) Unidades que se encargan de diversas funciones. Estas son:

UNIDAD I. Lectura de datos

Corresponde a la introducción de datos para la posterior realización del proceso de discretización definido con anterioridad.

Como primer paso PlusOne lee el archivo **plus.inp** para la determinación de las características mas importantes del modelo creando el archivo **plus.gen**, seguidamente lee el archivo **plud.dat** generado por ABAQUS, busca los valores de los Índices de Energía de Deformación, SENER, y genera un nuevo archivo **plus.sen** contenedor de esta información.

Luego es generado el archivo **plus.sed** en el cual se especifican todos los valores calculados de SED, SEDG, el valor del índice de refinamiento β y el numero de divisiones obtenido para cada elemento. Este es un archivo sólo de carácter informativo para el analista.

UNIDAD II. Refinamiento tipo H-adaptativo

En este módulo, PlusOne genera un nuevo mallado según los valores de los SENER. Utiliza el archivo **plus.sen** y **plus.gen** para el proceso de refinamiento.

UNIDAD III. Resultados

Los resultados obtenidos por PlusOne se resumen en archivos **plus.inp**, **plus.gen**, **plus.dat**, **plus.sen**, **plus.sed**, **plud.inp** y **plus.odb** generados y guardados en cada ciclo.

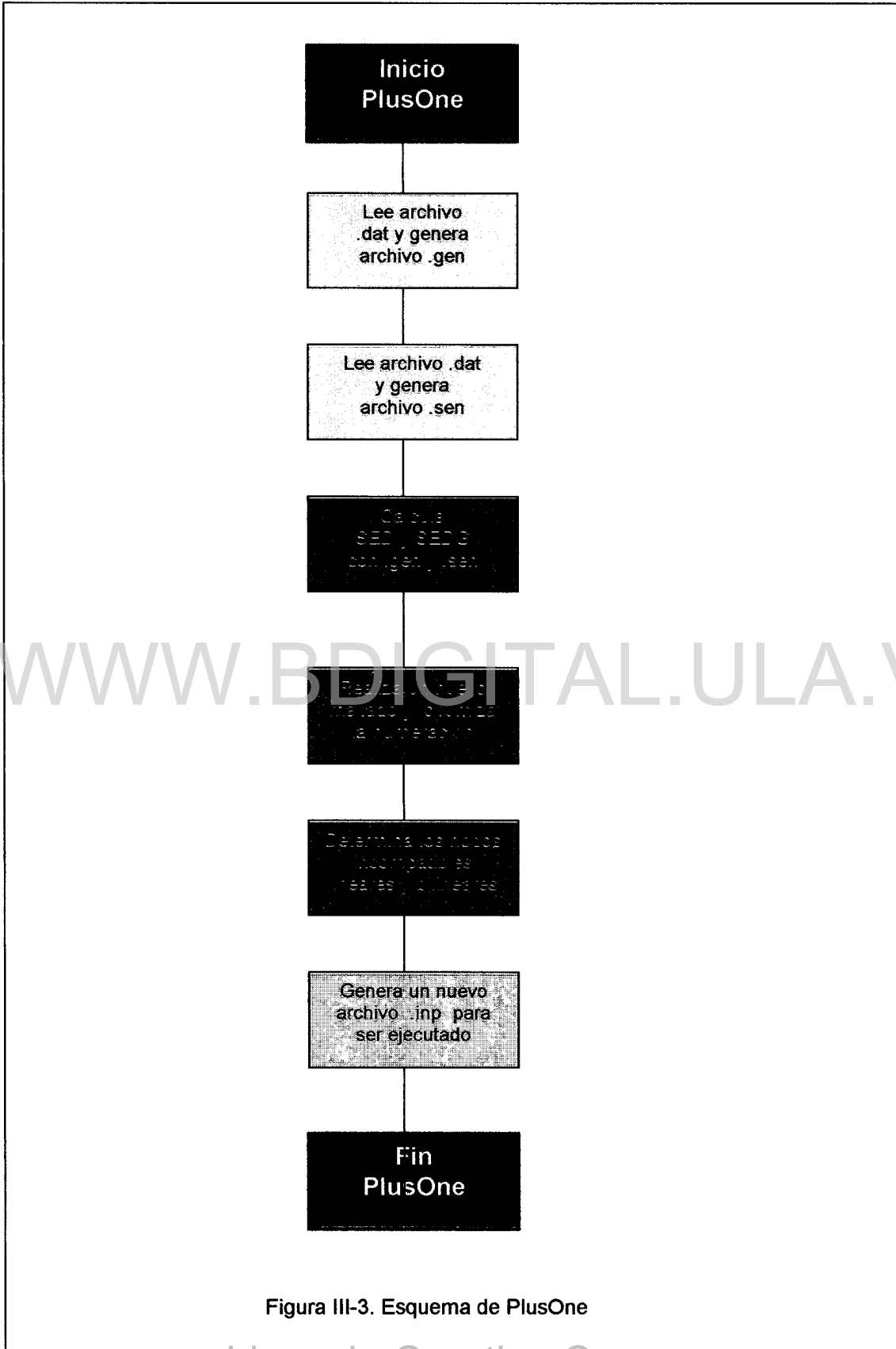


Figura III-3. Esquema de PlusOne



PlusOne se compone de 13 subrutinas, estas son:

DIMEN, GENIN, GENSEN, CHECK, ADAPTIVE, MDSIDE, OPTNUM, SHP20, INCOMP, REORDR, GENMSH, UBILSD y OUTPUT .

Las subrutinas principales de PlusOne son GENIN y GENSEN creadas específicamente para realizar el redireccionamiento de la información de los modelos a analizar y la creación de archivos indispensables para los ciclos de ejecución.

Todas las demás subrutinas fueron modificadas de su forma original a excepción de CHEK, SHP20 y REORDR para permitir el proceso de refinamiento de manera totalmente automático.

Subrutina DIMEN

La subrutina DIMEN contiene las dimensiones principales de arreglos principales. Su llamada se realiza al inicio de PlusOne para utilizar estos valores como constantes comunes dentro de la ejecución del programa. Se utilizan también para el dimensionamiento de arreglos de una, dos y tres dimensiones dentro de las subrutinas. Estos valores son utilizados también como límites de contadores máximos y comparaciones preestablecidas.

Subrutina GENIN

La subrutina GENIN es la encargada de iniciar el proceso automatizado. Está encargada de leer el archivo **plus.inp** en cada uno de los ciclos de refinamiento. Obtiene para cualquier extensión de archivo el número de nodos y el número de elementos, valores que son modificados en cada ciclo de refinamiento y que serán constantes a lo largo de la ejecución de cada ciclo en particular. Con estos valores es creado el archivo **plus.gen**. Este archivo contiene los características principales de la geometría del modelo el cual es utilizado para relacionar correctamente elementos y nodos en el proceso en si.

Licencia Creative Commons:

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



Subrutina GENSEN

La subrutina GENSEN lee el archivo *plus.dat*. Obtiene los valores de SENER generados por ABAQUS para cada nodo de cada elemento del modelo. Con estos datos genera el archivo *plus.sen* el cual es utilizado para calculo de SED y SEDG.

Subrutina ADAPTIVE

La subrutina ADAPTIVE lee el archivo *plus.gen* y el archivo *plus.sen*. Calcula los SED, SEDG y las divisiones en X, Y y Z según comparación al indicador Beta.

Subrutina MDSIDE

Esta subrutina interpola los nodos centrales sobre los lados localizados en el lado recto de los súper elementos hexaédricos.

Subrutina OPTNUM

La subrutina OPTNUM es la encargada de eliminar los nodos repetidos.

Subrutina INCOMP

La subrutina INCOMP determina las coordenadas de los posibles nodos incompatibles y asocia a ellos las restricciones de los nodos vecinos, lineares o bilineares, según sea el caso.

Subrutina REORDR

La subrutina REORDR renumera los elementos de acuerdo a la proximidad relativa entre ellos.



Subrutina GENMSH

La subrutina GENMSH genera una malla de elementos hexaédricos tridimensionales.

Subrutina UBILSD

La subrutina UBILSD localiza los nodos lineales incompatibles en elementos refinados y no refinados.

Subrutina RENMBR

La subrutina RENMBR calcula cuantos nodos están conectados a un nodo dado y almacena los números de los nodos

Subrutina OUTPUT

La subrutina OUTPUT genera el nuevo archivo **plus.inp** y **plud.inp**. Ambos archivos son ejecutados por ABAQUS, el principal producto de la interacción con **plus.inp** es el archivo **plus.dat** que contiene el listado de los SENER. El archivo **plud.inp** se relaciona igualmente con **plud.odb**, que tiene la información para que los resultados puedan ser visualizados por el ABAQUS/VIEWER.



CAPITULO IV

RESULTADOS

IV.1 Resultados numéricos

En este capítulo se muestran los resultados obtenidos luego de realizar el proceso automáticamente por tres (3) ciclos.

IV.1.1 Modelo original

Como ejemplo de aplicación, fue seleccionado un modelo que representa una viga empotrada con empotramiento defectuoso. La viga consta de cinco (5) elementos y veinticuatro (24) nodos. En la Figura IV-1, se muestra el modelo de la viga a analizar.



IV.1.2. Primer ciclo.

En este primer ciclo fueron obtenidos valores de Índice de Energía de Deformación (SED) y Energía de Deformación Global (SEDG) para el modelo los cuales se muestran en la Tabla IV-1

La Figura IV-2 presenta los valores de SENER obtenidos para el modelo analizado.

Tabla IV-1. Valores resumen del primer ciclo.

WWW.BDIGITAL.ULA.VE

Elementos	SEDG	β	Divisiones en el elemento
1	4.09	1	si
2	0.65	1	no
3	0.19	1	no
4	0.057	1	no
5	0.00	1	no

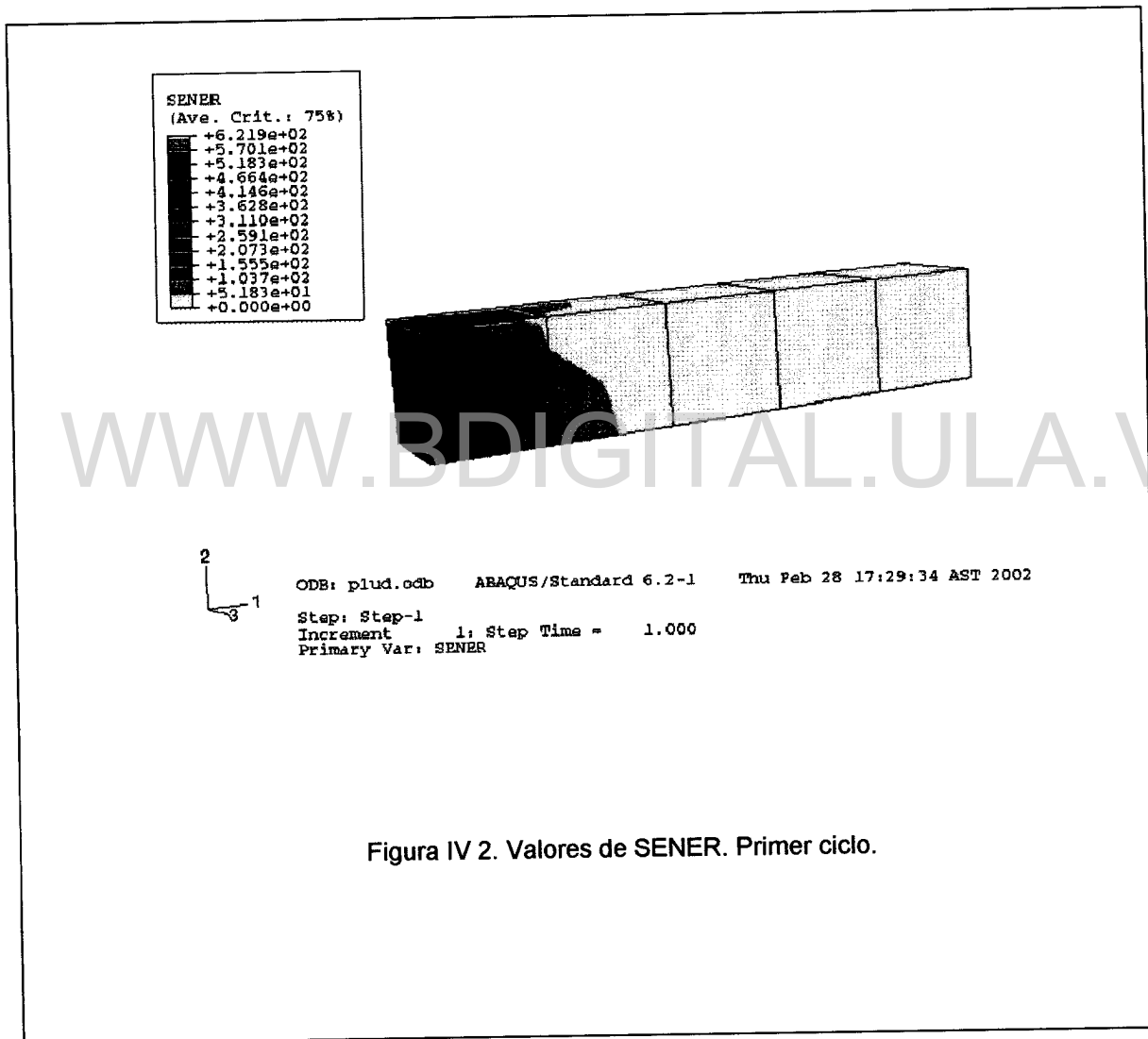
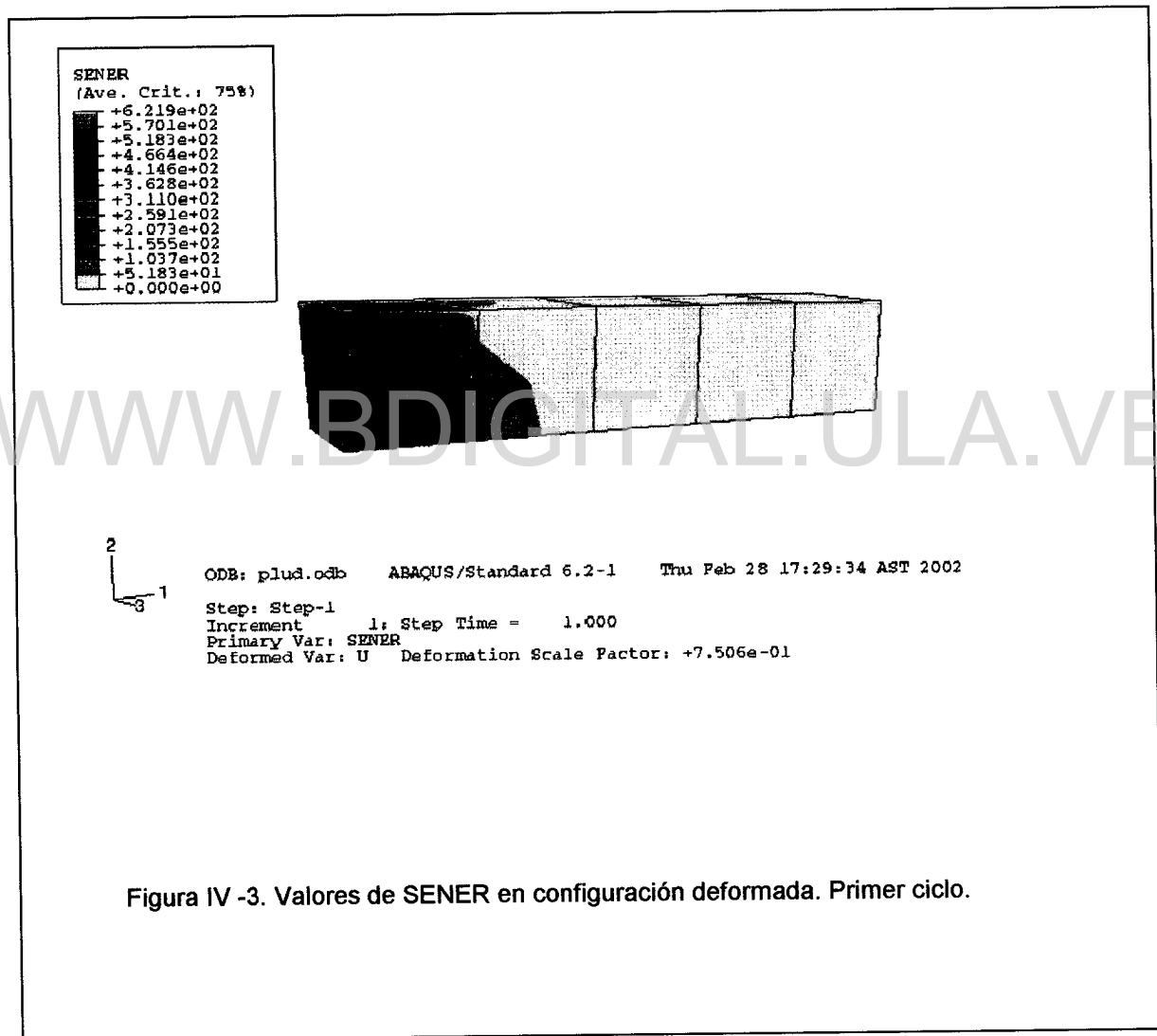
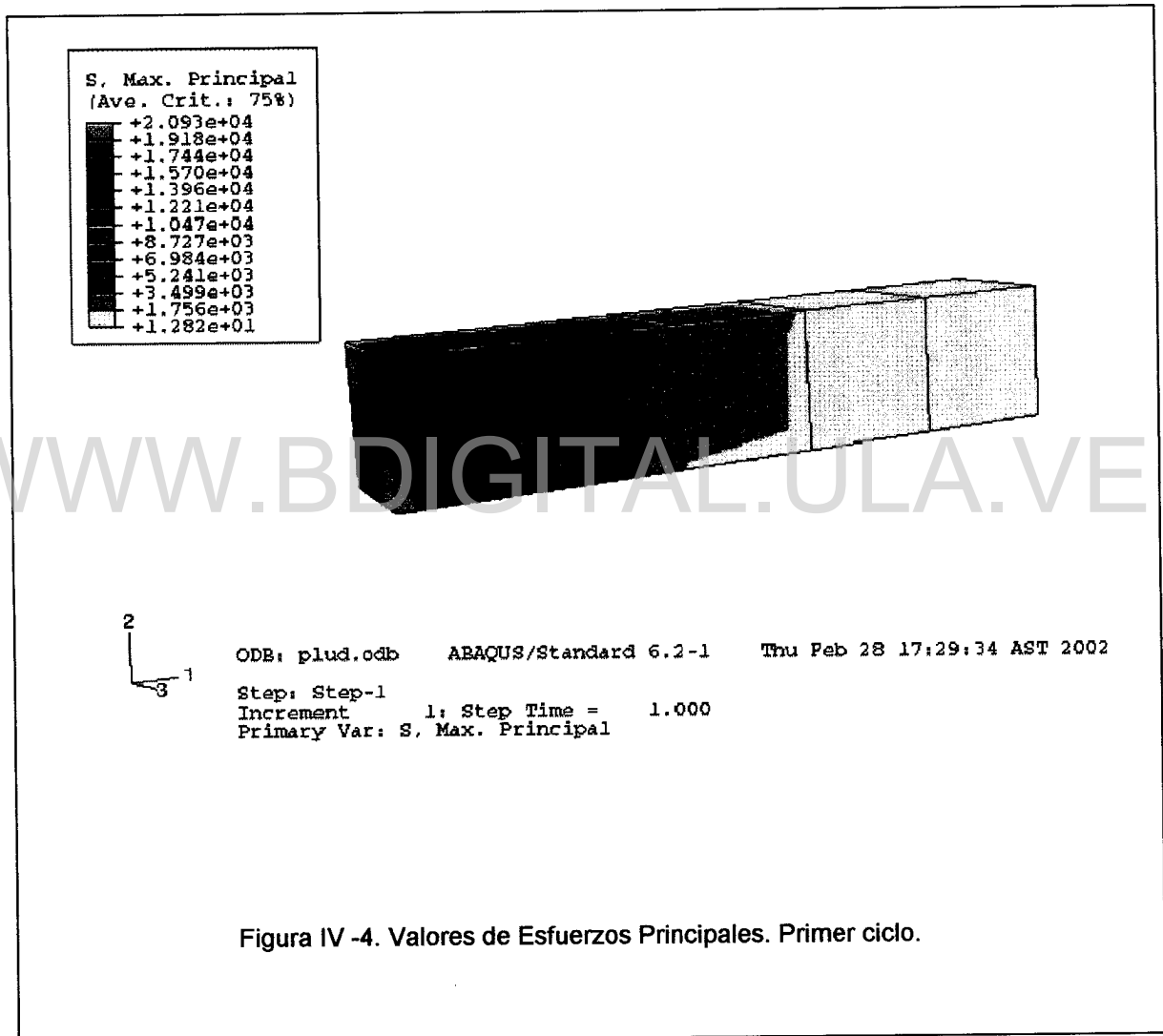
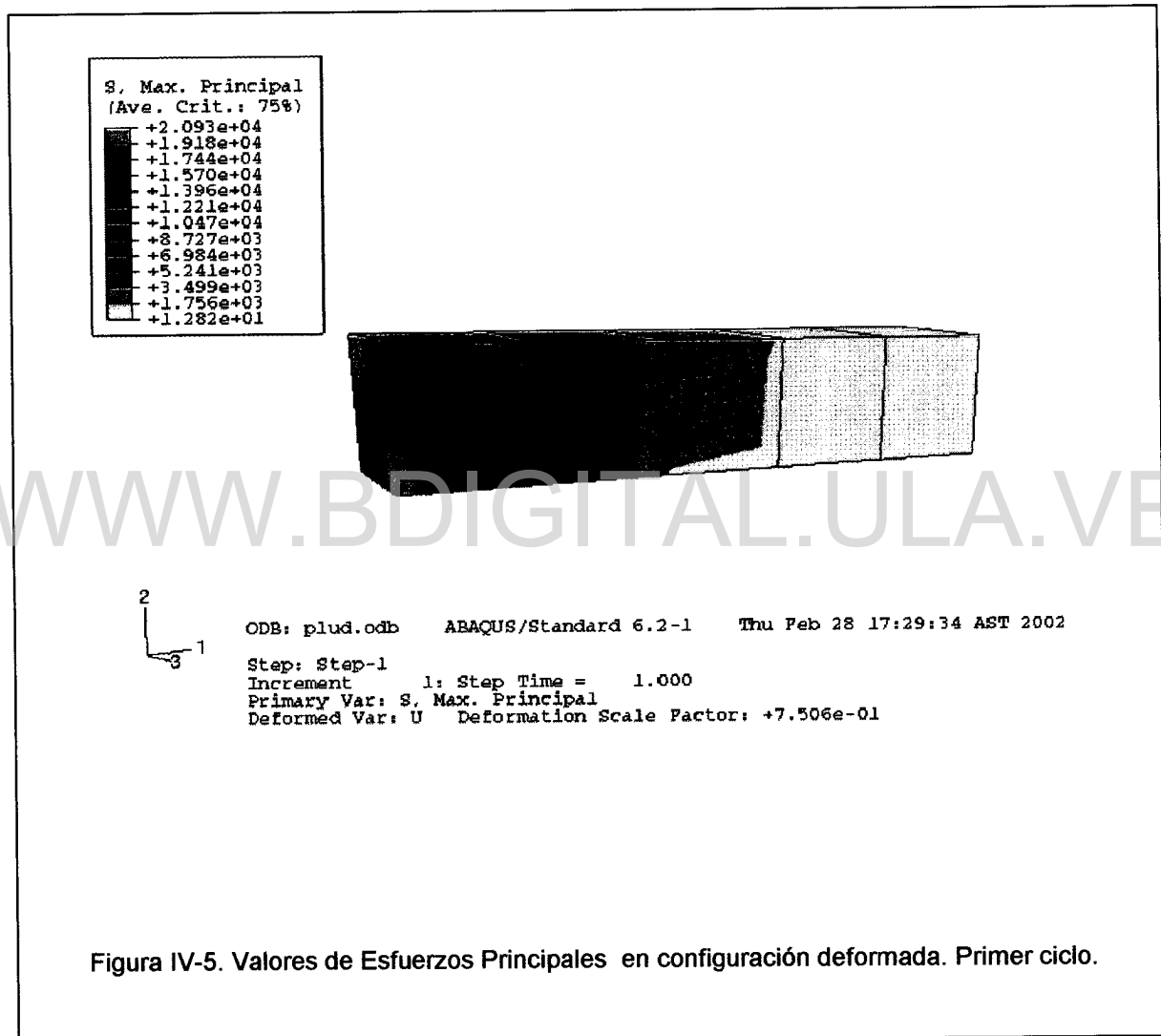


Figura IV 2. Valores de SENER. Primer ciclo.







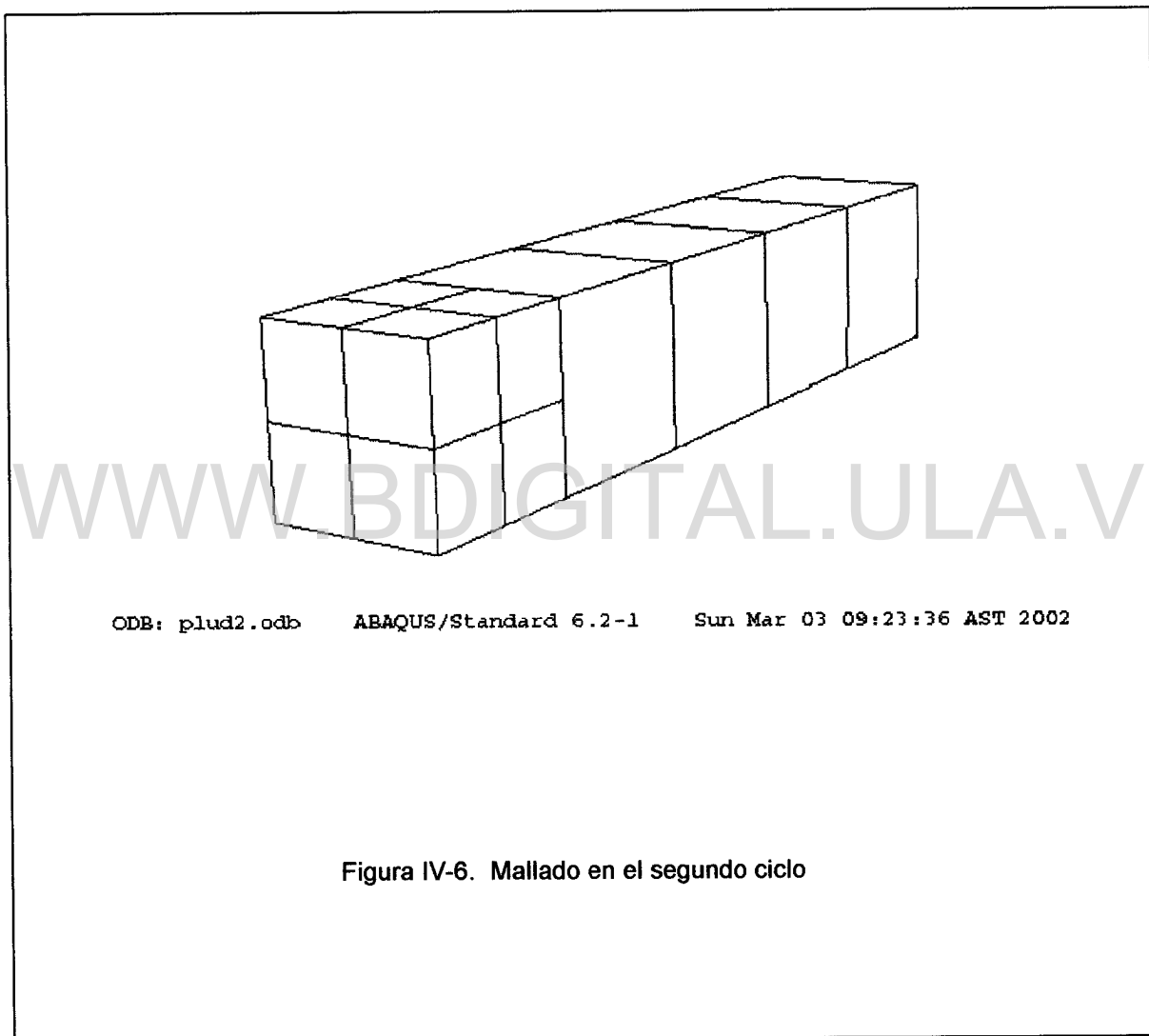


IV.1.3. Segundo ciclo

Los resultados obtenidos luego del primer ciclo se resumen en nuevos valores de energía de deformación que permiten a PlusOne obtener un nuevo mallado. Los resultados numéricos principales obtenidos en este ciclo se muestran en la Tabla IV-2.

Tabla IV.2 Valores resumen del segundo ciclo.

Elementos	SEDG	β	Divisiones en el elemento
1	0,54	1	no
2	0,24	1	no
3	0,24	1	no
4	0,28	1	no
5	5,6	1	si
6	1,97	1	si
7	1,97	1	si
8	1,12	1	si
9	0	1	no
10	0	1	no
11	0	1	no
12	0	1	no



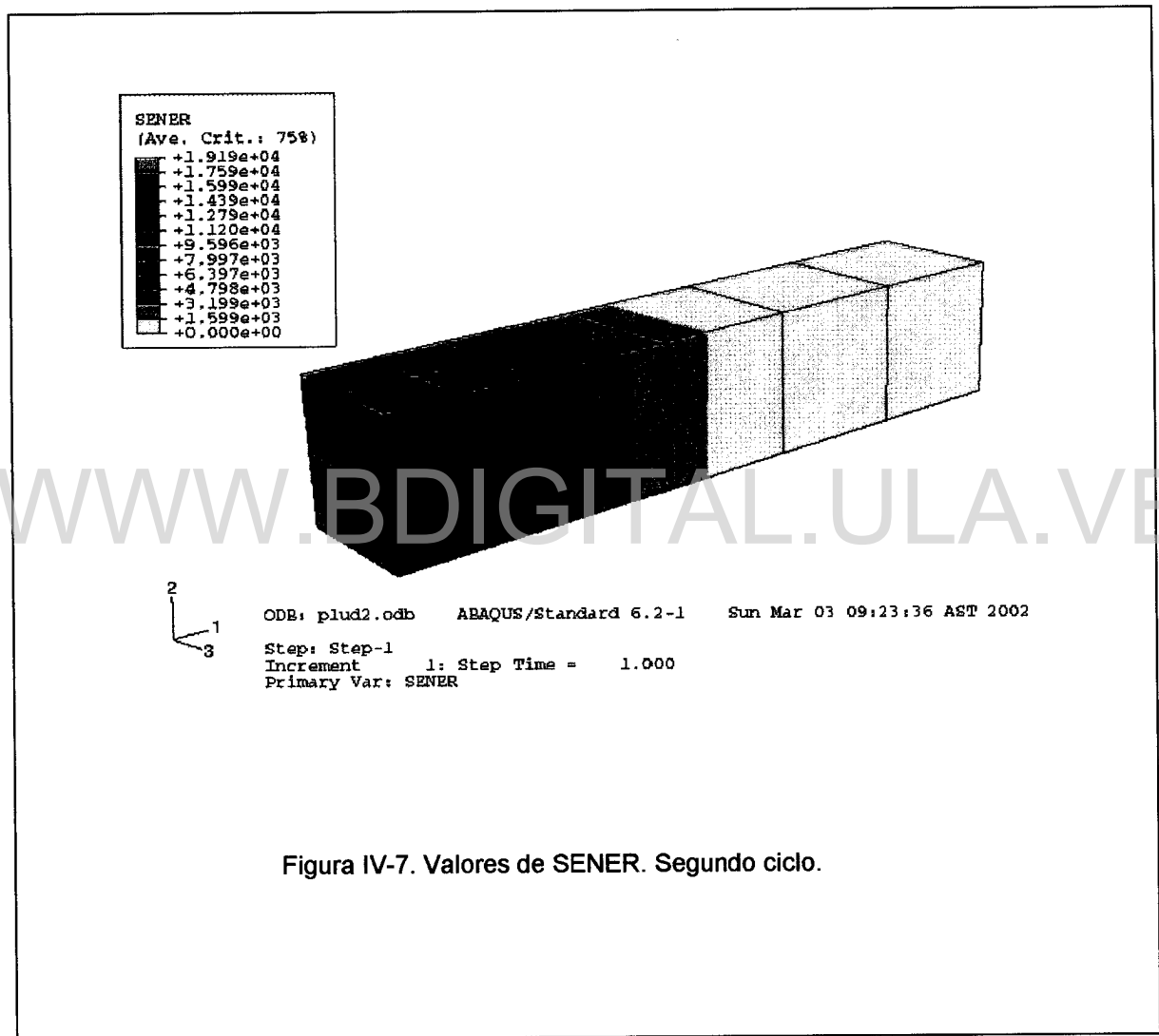


Figura IV-7. Valores de SENER. Segundo ciclo.

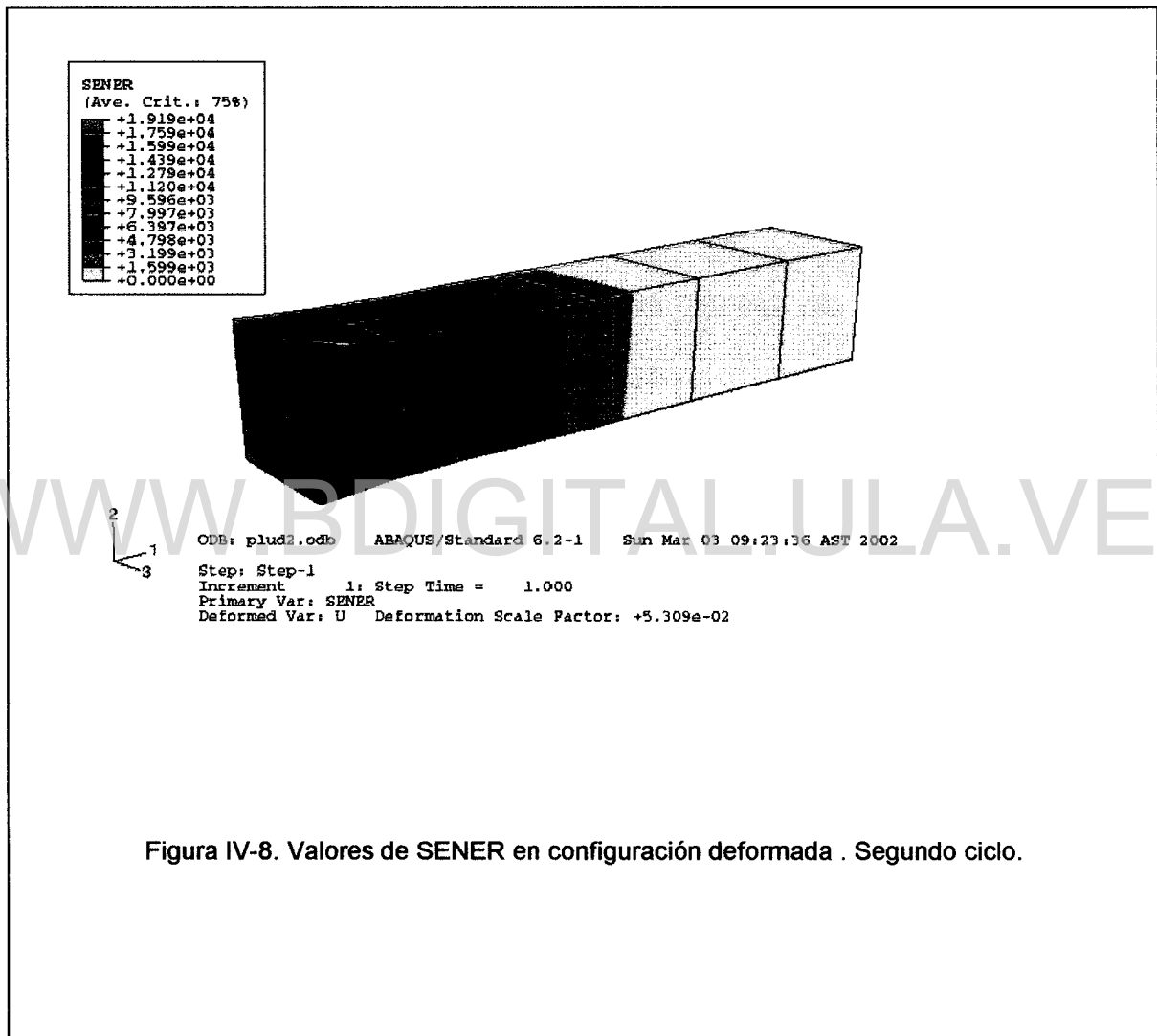


Figura IV-8. Valores de SENER en configuración deformada . Segundo ciclo.

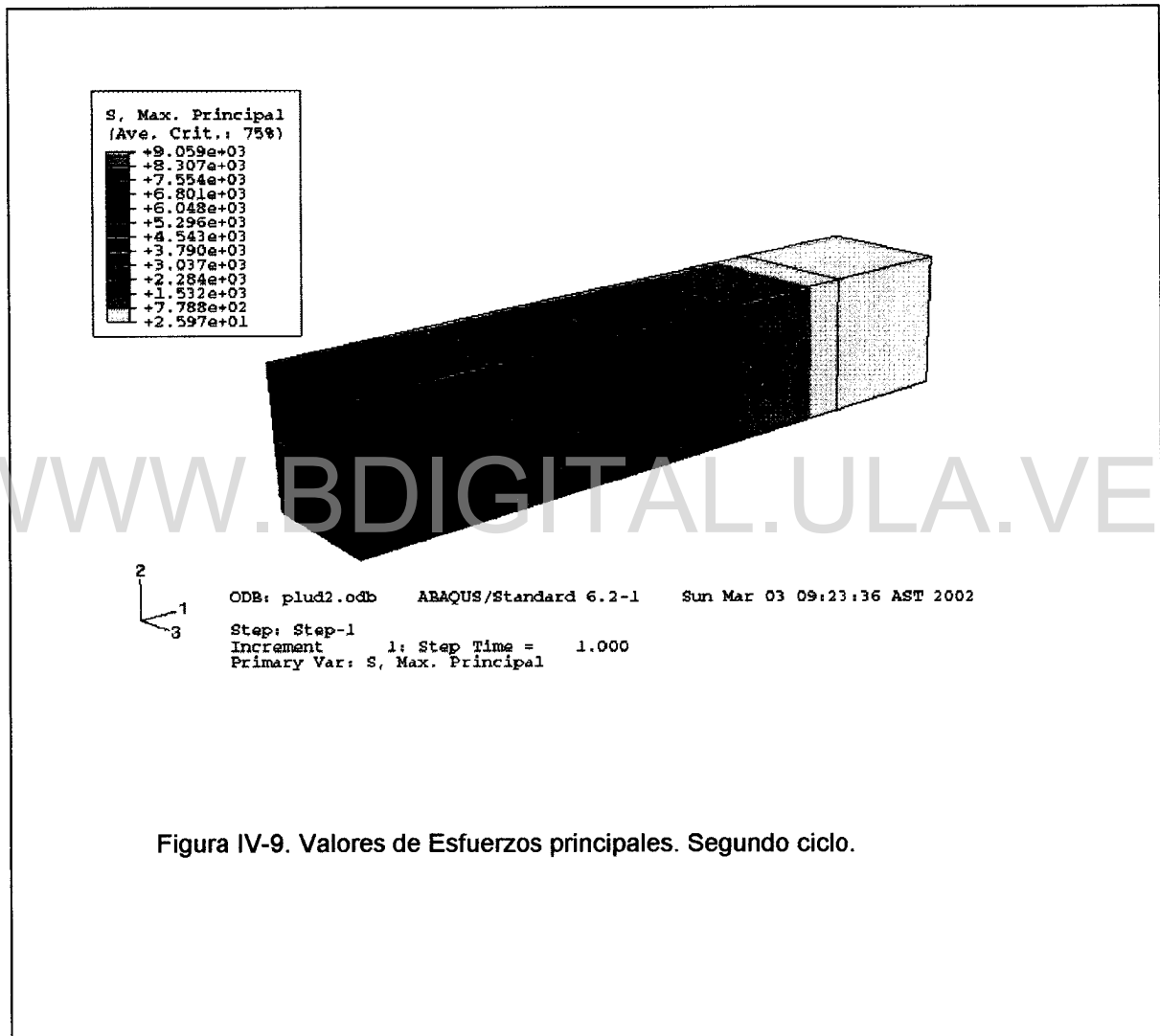


Figura IV-9. Valores de Esfuerzos principales. Segundo ciclo.

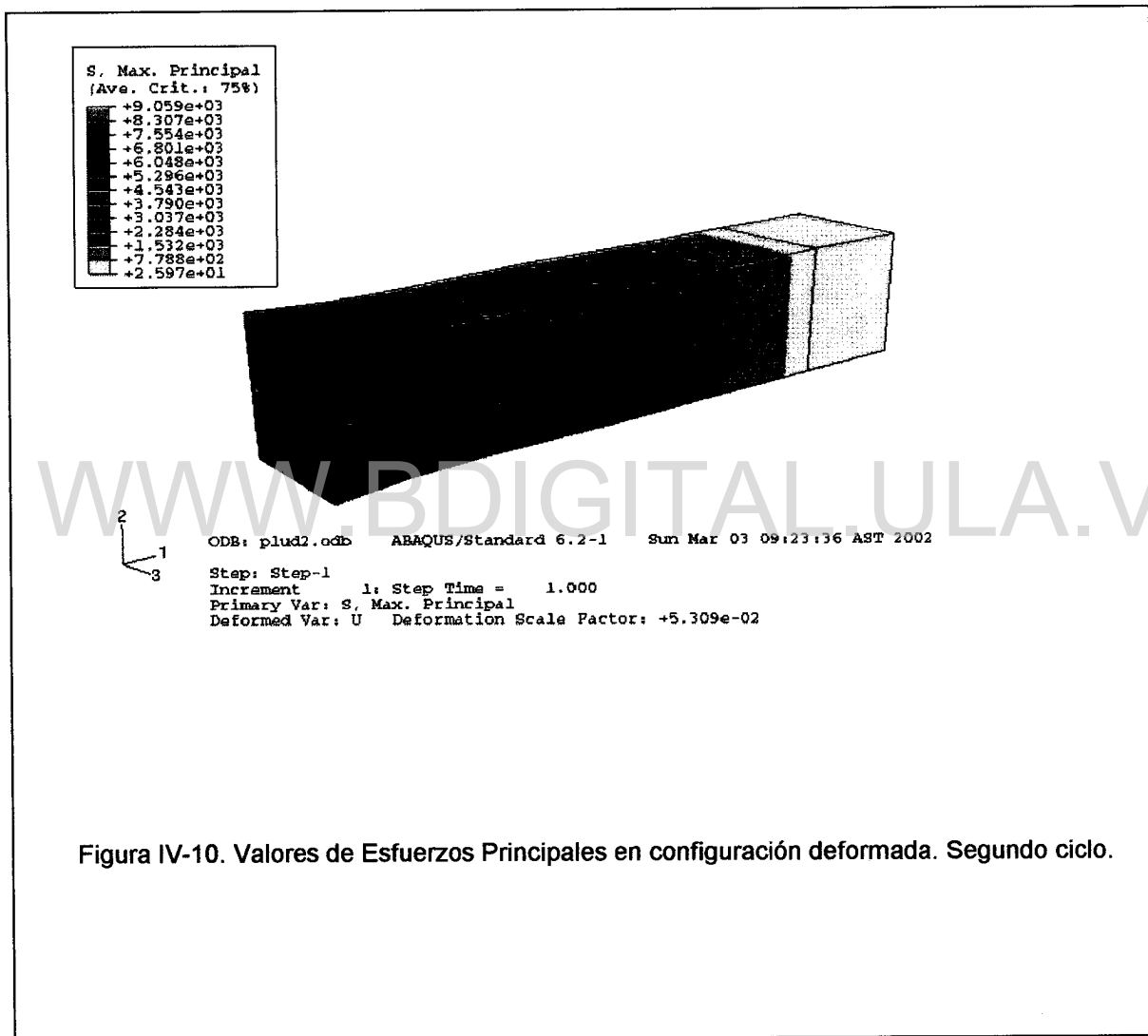


Figura IV-10. Valores de Esfuerzos Principales en configuración deformada. Segundo ciclo.



IV.1.4. Tercer ciclo

Los resultados de este tercer ciclo se resumen perfectamente en el nuevo mallado mostrado en la Figura IV-11.

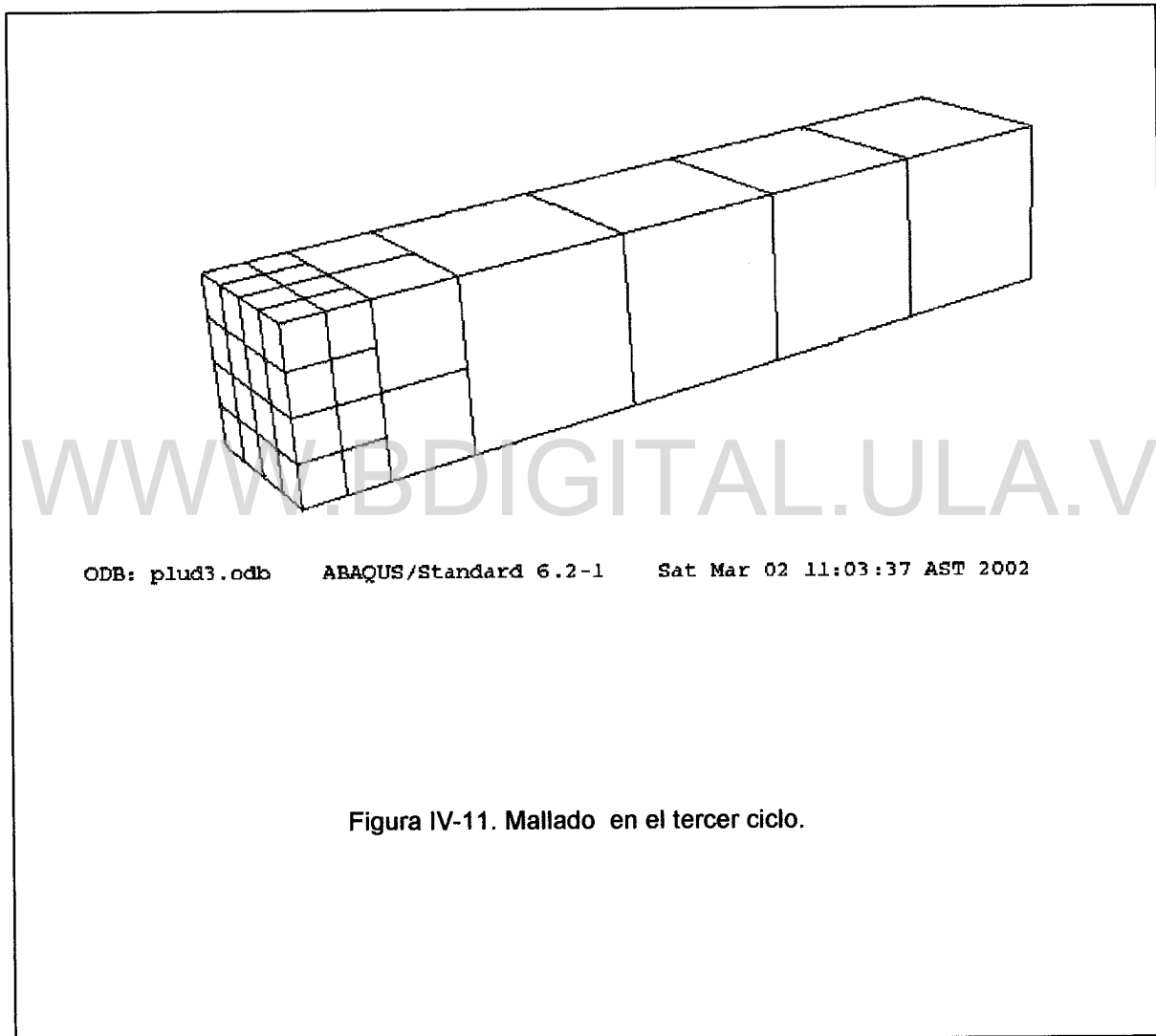


Figura IV-11. Mallado en el tercer ciclo.

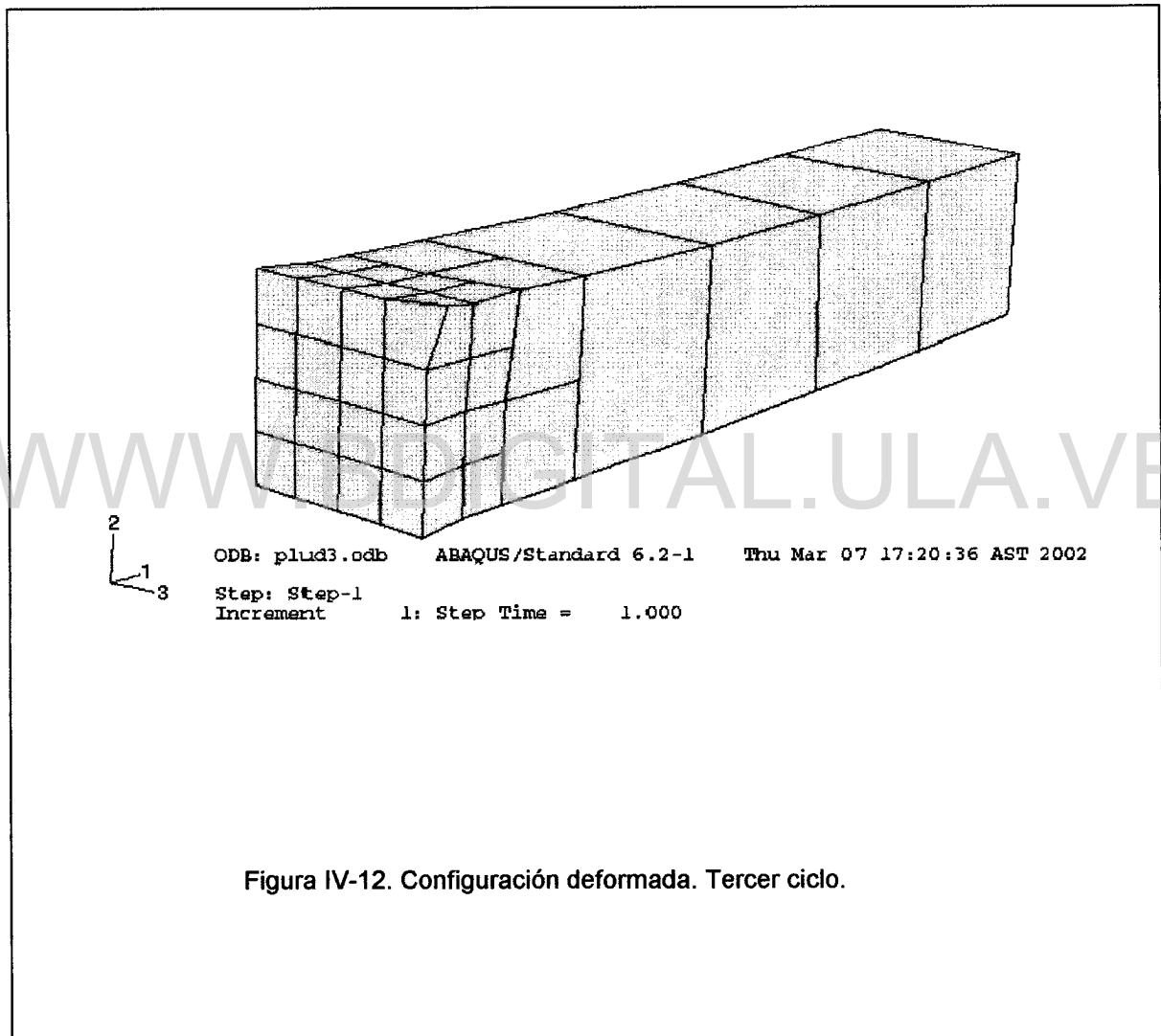


Figura IV-12. Configuración deformada. Tercer ciclo.

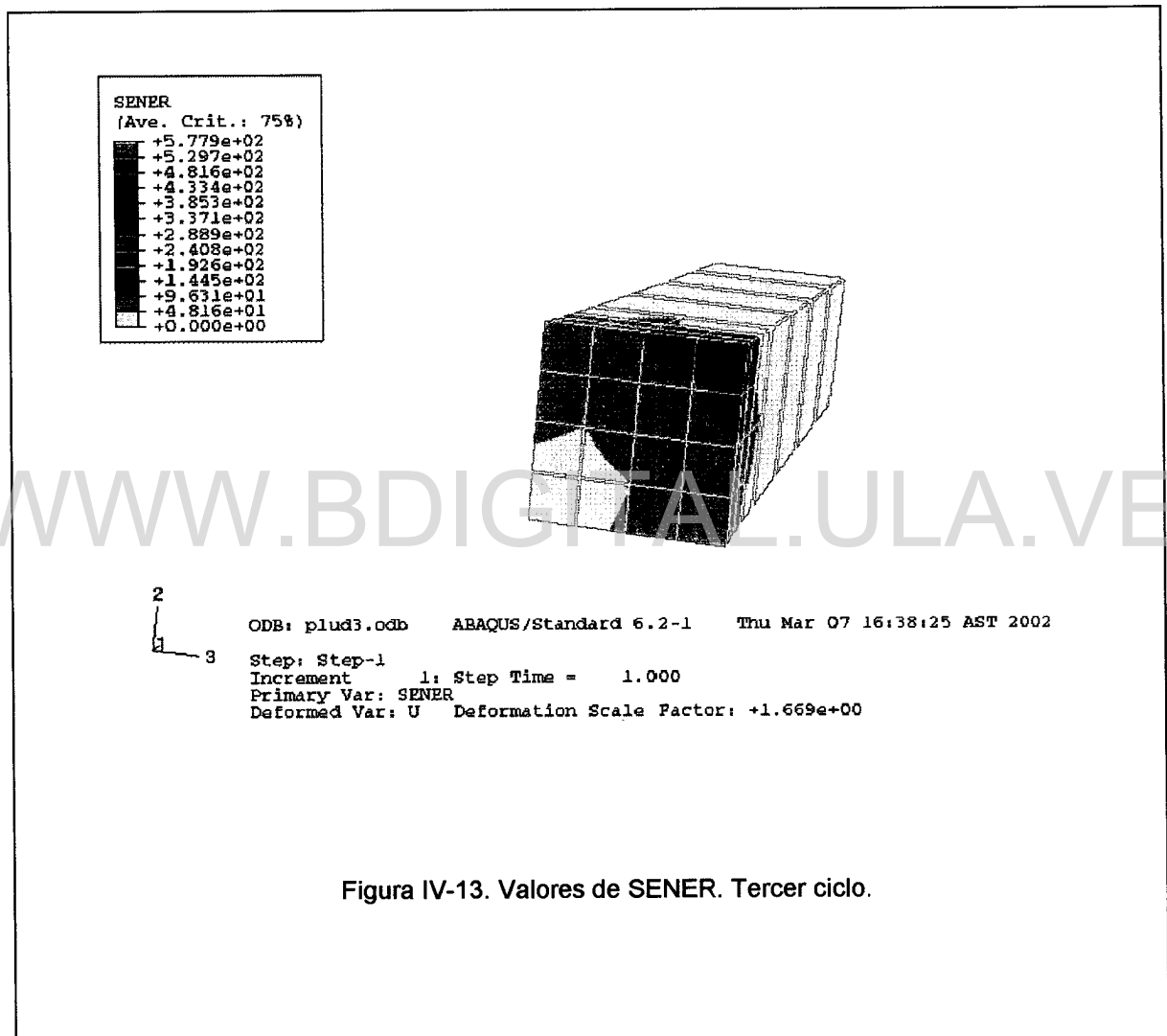
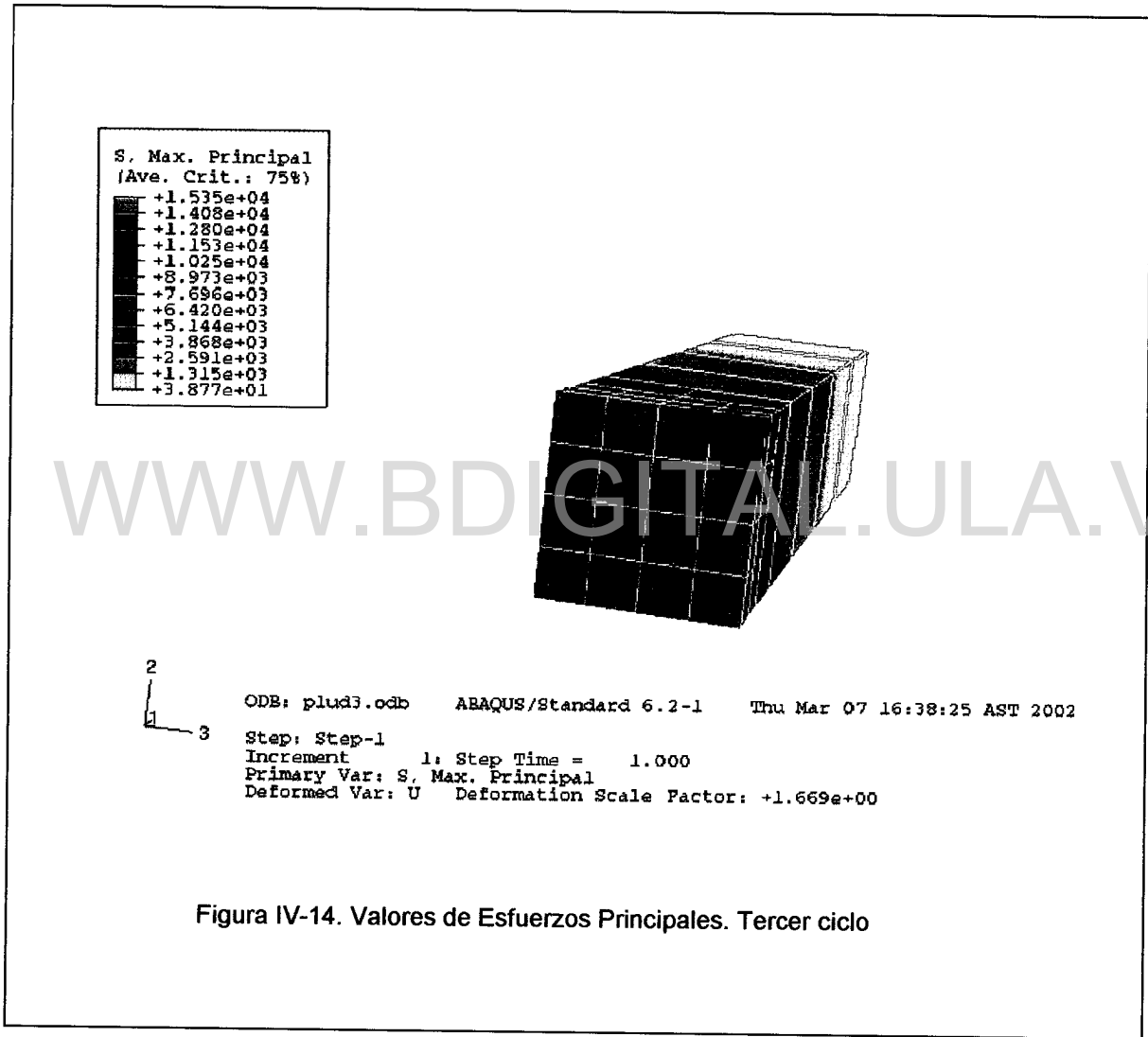


Figura IV-13. Valores de SENER. Tercer ciclo.





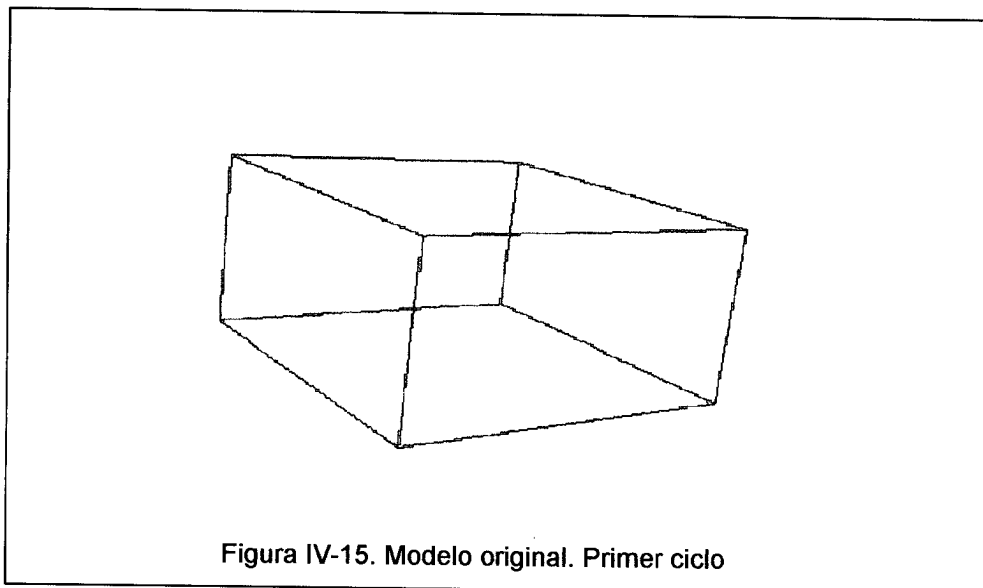
IV. 2. Ejemplos adicionales

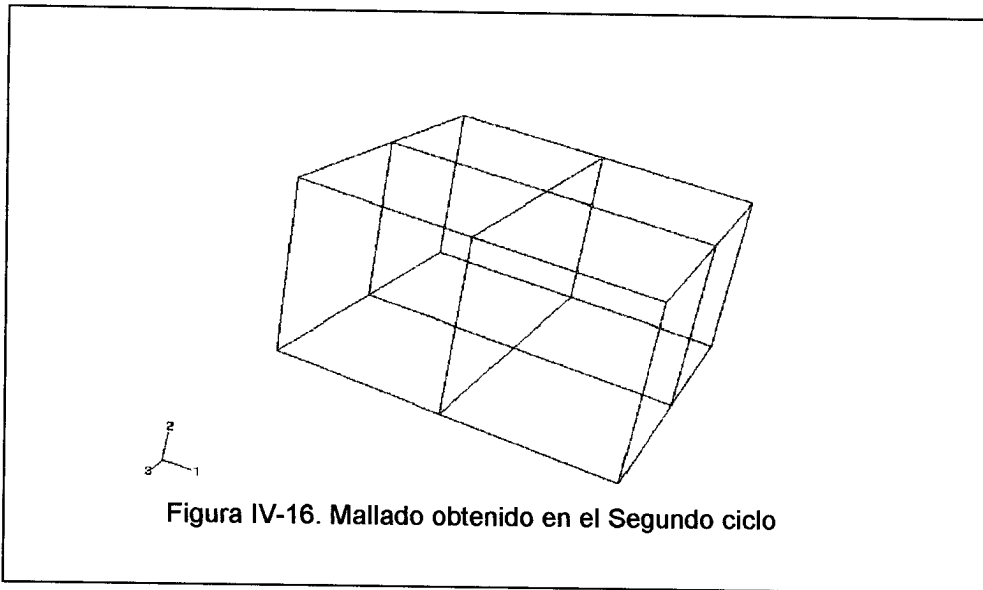
Fueron corridos dos (2) ejemplos adicionales. Los resultados de los mallados obtenidos en cada ciclo de ejecución se muestran junto con la distribución de SENER.

IV.2.1 Ejemplo No.1

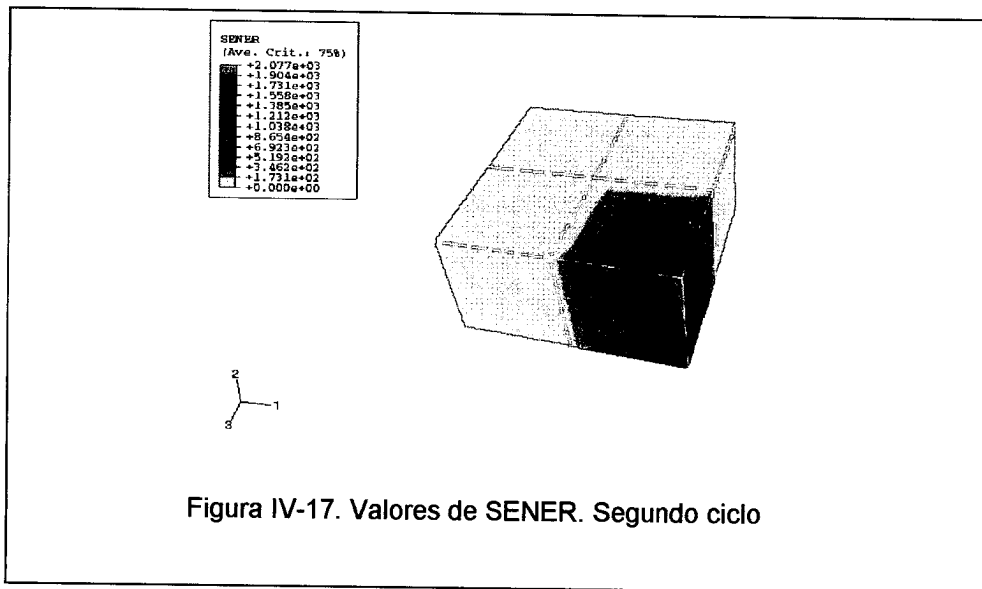
El primer ejemplo adicional corresponde al modelo de una fundación con una carga en el centro. Por condiciones de simetría, se representa $\frac{1}{4}$ de la fundación con una carga puntual en el extremo. La Figura IV-15 muestra el mallado correspondiente a un (1) sólo elemento.

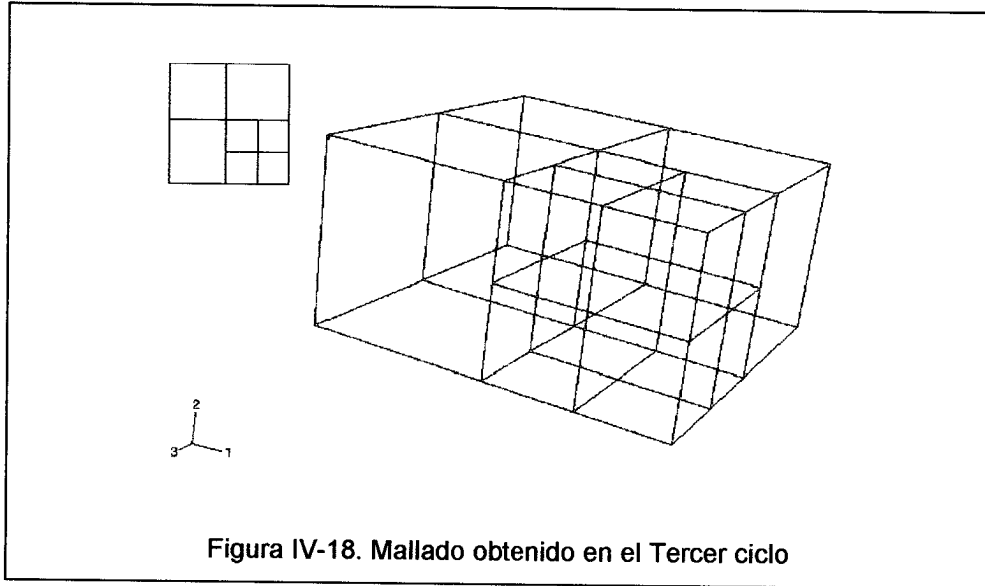
Fueron ejecutados cuatro (4) ciclos para este modelo.



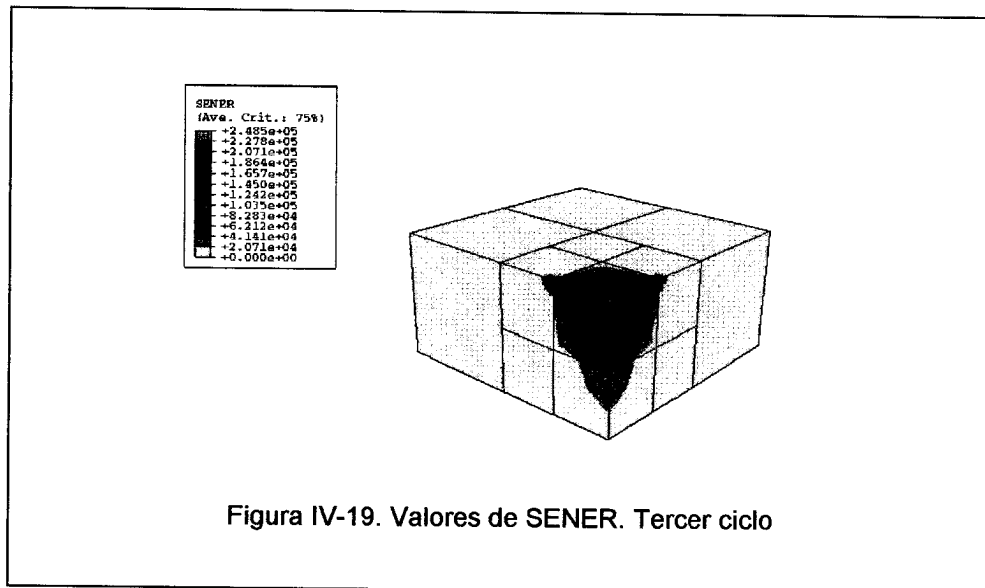


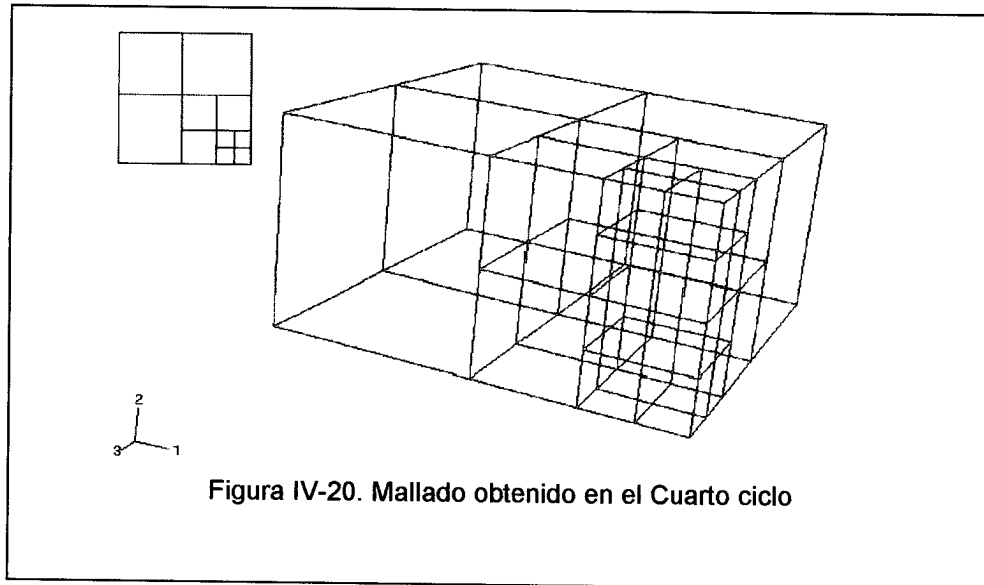
WWW.BDIGITAL.ULA.VE





WWW.BDIGITAL.ULA.VE



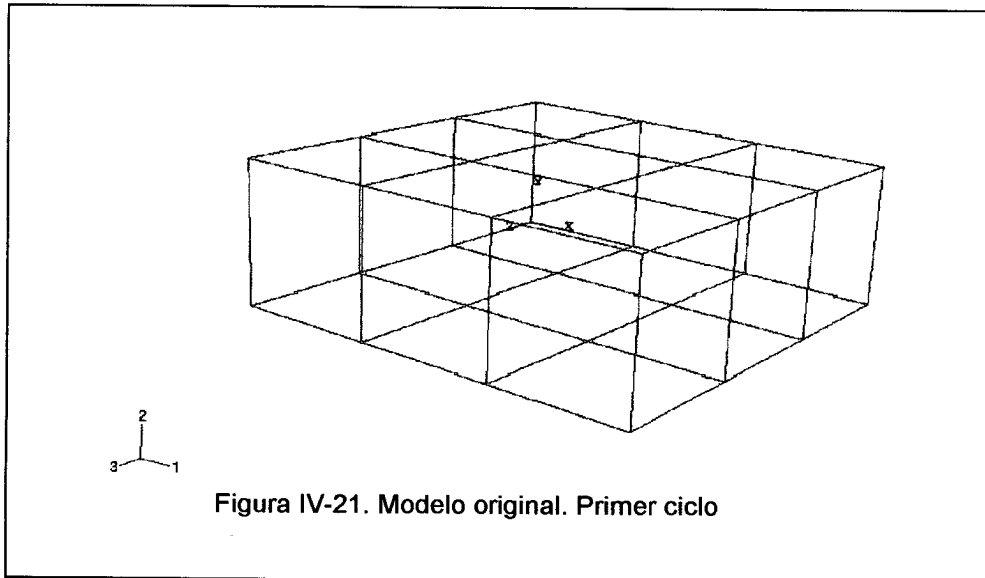


WWW.BDIGITAL.ULA.VE

IV.2.2 Ejemplo No.2

El segundo ejemplo adicional corresponde un bloque con un agujero en el centro y una carga concentrada en una esquina interior.

Fueron ejecutados dos (2) ciclos para este modelo.



WWW.BDIGITAL.ULA.VE

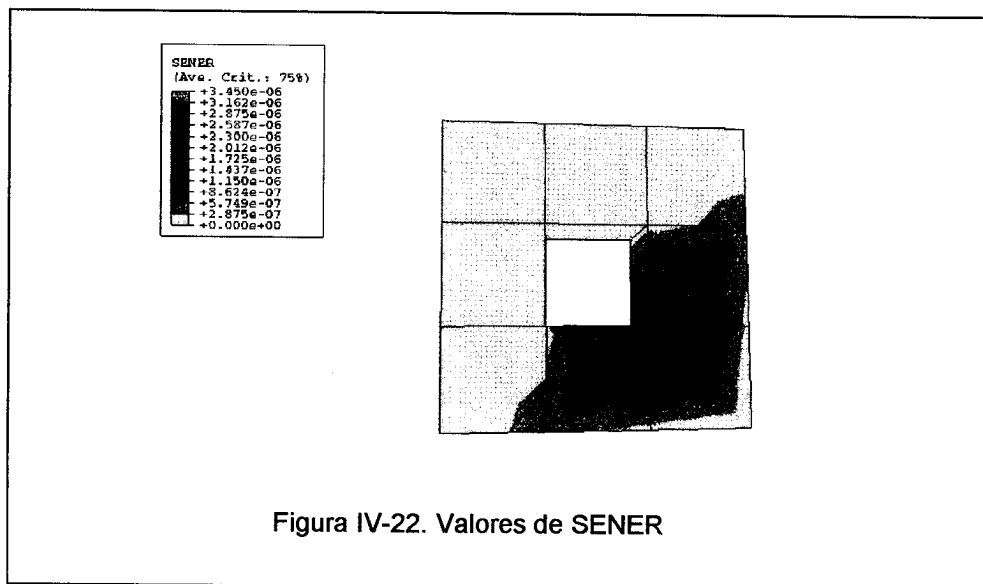


Figura IV-22. Valores de SENER

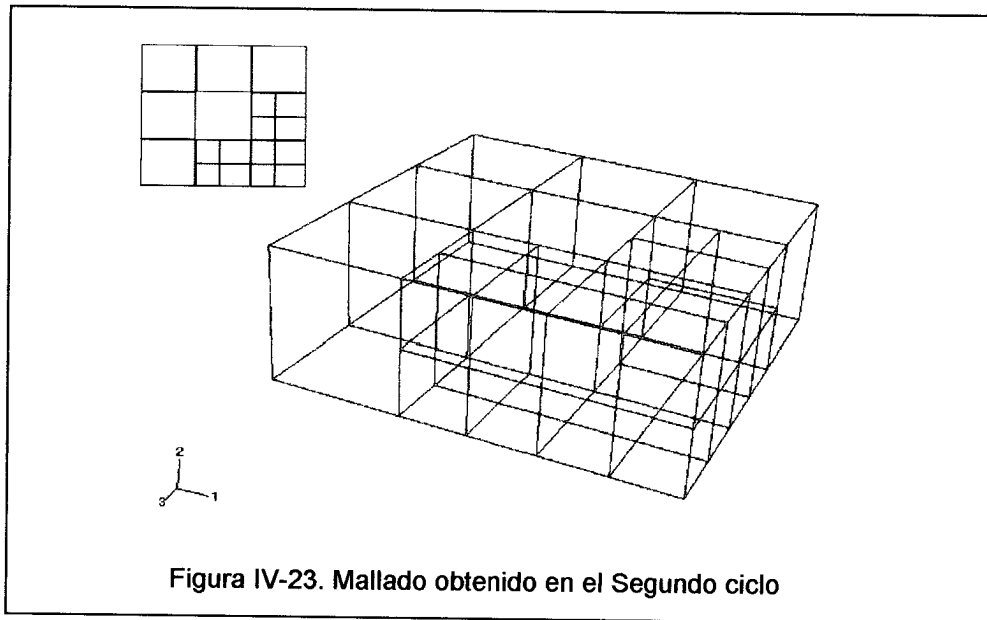


Figura IV-23. Mallado obtenido en el Segundo ciclo

WWW.BDIGITAL.ULA.VE



IV. 3. Guía de Usuario.

La utilización del software creado es sencilla si se siguen las siguientes instrucciones.

1. Debe contarse con acceso al programa comercial **ABAQUS**. Se utilizarán dos licencias de ejecución en el proceso: para los archivos **plus.inp** y **plud.inp** que serán corridos en paralelo. El primero generará el archivo **plus.dat** que contiene los valores SENER a ser leídos por PlusOne y el segundo dará origen al archivo **plud.odb** para que el usuario tenga la facilidad de visualizar el mallado resultado de cada ciclo con ABAQUS/VIEWER, si así lo desea.
2. Los programas **PlusRun** y **PlusOne**, y el archivo **plus.inp** deben estar en el mismo directorio.
3. La sintaxis de ejecución para la activación del proceso es la siguiente:

```
./PlusRun.sh plus 3
```
4. El proceso queda activado y se detendrá, sin interacción con el usuario, luego de **3** ciclos, dando como resultado para cada ciclo los archivos: **plus.inp**, **plus.gen**, **plus.sen**, **plus.sed**, **plus.dat**, **plud.inp** y **plud.odb**.
5. Los archivos nombrados son guardados en cada proceso, esto es:

Primer ciclo:



plus.inp, *plus.gen*, *plus.sen*, *plus.sed*, *plus.dat*, *plud.inp* y
plud.odt.

Segundo ciclo:

plus2.inp, *plus2.gen*, *plus2.sen*, *plus2.sed*, *plus2.dat*, *plud2.inp* y
plud2.odt.

Tercer ciclo:

plus3.inp, *plus3.gen*, *plus3.sen*, *plus3.sed*, *plus3.dat*, *plud3.inp* y
plud3.odt.

6. Concluido el proceso el usuario podrá analizar toda la información obtenida en los siete (7) archivos generados.

plus.inp: tiene la información del modelo, mallado, cargas, apoyos. Contiene la sintaxis para la creación de *plus.dat*.

plus.gen: contiene la información depurada de *plus.inp*, para iniciar el proceso.

plus.sen: su función es depurar el archivo *plus.dat* caracterizado generalmente por su gran extensión. Contiene exclusivamente los valores de SENER de cada nodo de cada elemento del modelo.

plus.sed: reproduce la información fundamental generada por PlusOne donde se especifican los valores de SEG, SEDG y el número de divisiones de cada elemento en cada eje. Muestra también el valor del indicador β entre otros.

plus.dat: contiene una extensa información relacionada con la ejecución de *plus.inp*. Incluye también los valores de SENER, utilizados por *plus.sen*.

Licencia Creative Commons:

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



plud.inp: al igual que *plus.inp*, contiene características del modelo, mallado, apoyos y cargas, pero su función es generar, luego de su ejecución, el archivo *plud.odb*.

plud.odb: es originado luego de la ejecución de *plud.inp*, y su objetivo es visualizar los resultados en ABAQUS/VIEWER.

7. La culminación del proceso en sí corresponde al análisis y generación de conclusiones, por parte del usuario, de los resultados obtenidos en cada ciclo.

WWW.BDIGITAL.ULA.VE



CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

El logro de la automatización del proceso de Refinamiento Tipo H-adaptativo ha sido llevado a cabo gracias a la utilización de herramientas de avanzada como es el lenguaje de programación Fortran 90 y el C-Shell en UNIX. Además el uso del programa comercial ABAQUS es de suma importancia en el desarrollo del proceso.

Las conclusiones obtenidas en esta investigación se presentan enumeradas a continuación:

1. El conjunto formado por PlusRun, PlusOne y ABAQUS realiza el proceso de refinamiento H-adaptativo en forma automática encontrando la discretización mas apropiada del modelo.
2. La concentración del Índice de Densidad de Energía de Deformación es elevada en las zonas de singularidad y con tendencia a cero (0) en las zonas de menor esfuerzo.



3. La distribución de los Esfuerzos Principales es similar a la distribución de los SENER
4. .No debe excederse el número de ciclos de refinamiento pues en el proceso de ejecución, podría iniciarse la incursión en la Zona III de error.
5. Para esta investigación fueron suficientes tres (3) ciclos de ejecución, ya que en este punto se obtuvo mallado similar a estudios anteriores.
6. El número de ciclos dependerá de la pericia del analista para determinar en base a la complejidad del modelo y a la exactitud de los resultados buscados.
7. El valor del Indicador de Refinamiento b debe ser elegido dependiendo de la densidad de malla deseada. $b= 0$ indica que todos los elementos serán refinados y, $b= 1$ indica que sólo los elementos con SED_i mayores que el promedio SED_G serán refinados.
8. Para el inicio del proceso automatizado de refinamiento H-adaptativo con modelos de un (1) elemento, el indicador de refinamiento b debe ser menor a la unidad.
9. Los resultados obtenidos en esta investigación se resumen en nuevas mallas generadas automáticamente con la casi nula intervención del usuario, adaptadas a singularidades importantes del modelo definidas por las concentraciones de Energía de Deformación.



Por otro lado, las principales recomendaciones surgidas de este trabajo van dirigidas a la continuación de esta línea de investigación y se reusumen en lo siguiente:

1. Realizar el proceso de refinamiento tipo H-adaptativo para cualquier elemento diferente al C3D8 utilizado por ABAQUS.
2. Lograr la detención del proceso con un indicador de error.
3. Estudiar la sensibilidad al indicador de refinamiento β , con un modelo de referencia analizando resultados numéricos y tiempos de ejecución.

WWW.BDIGITAL.ULA.VE



LITERATURA CONSULTADA

1. **ABAQUS, User's Manual**, (2001). Versión 6.2. Hibbitt, Karlsson & Sorensen, Inc.
2. Buenaño A. y Febres H. E. (1995), **Generador de malla H – adaptativo Aplicado a Problemas de Flexión de Placas Usando Índices de Densidad de Energía de Deformación**, Boletín Técnico IMME.
3. Buenaño A. y Febres H. E. (1996), **Proceso Adaptativo Basado en Índices de Energía de Deformación Aplicado al Problema de Flexión de Placas**, Tesis de Maestría en Ingeniería Estructural. Universidad de Los Andes. Mérida, Venezuela.
4. 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.
5. 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. U.S.A.
6. Cook R. D. (1995), **Finite Element Modeling for Stress Analysis**, Editorial Jhon Wiley & Sons, Inc. U.S.A.



7. Davis G. – Hoffmann T. **FORTRAN 77 , un estilo estructurado y disciplinado.** Segunda Edición. Mc Graw Hill, México D.F, México.
8. Elberg, M.E. (2001). **Simulación de estructuras sometidas a cargas estáticas y dinámicas realizadas con Teoría de Elementos Finitos.** III Jornadas A & C Ingeniería de Avanzada. IUPSM. Mérida, Venezuela.
9. Elberg M.E. y Febres H. (2002). **Proceso automatizado de refinamiento H-Adaptativo en base a Índices de Energía de Deformación.** V Jornadas Científico Técnicas. Universidad de Los Andes, Mérida, Venezuela.
10. Elberg M.E. y Febres H. (2002). **Proceso automatizado de refinamiento H-Adaptativo en Análisis por Elementos Finitos.** XXX Jornadas Sul-Americanas de Engenharia Estrutural. Brasília, Brasil.
11. Elberg M.E. y Febres H. (2002). **Automatización del proceso de refinamiento H-Adaptativo en problemas tridimensionales.** V Congreso de Métodos Numéricos en Ingeniería. Madrid, España.
12. Febres – Cedillo H. E. y Bhatti M. A. (1988) **A Simple Stain Energy Bases Finite Element Mesh Refinement Scheme,** Computers & Structures.
13. Febres – Cedillo H. E. (1984). **An Energy – Based Adaptive Element Analysis of Two – Dimensional Elasticity Problems,** Thesis for Degree of Science in Civil and Environmental Engineering, University of Iowa, U.S.A.
14. Guerrero N. **Aplicación de técnicas de extrapolación en el cálculo del error de discretización.** Tesis de Maestría en Ingeniería Estructural. Universidad de Los Andes. Mérida, Venezuela.



23. Hahn H. (1995). **UNIX sin fronteras**. Mc. Graw Hill. Interamericana de México. Edo. de México, México.
24. Weaver W. y Johnston P. **Finite Elements for Structural Analysis**, Editorial Prentice Hall, Inc., Englewood Clif, New Jersey.
25. Zienkiewicz O. C. y Taylor R. L. (1994), **El Método de Los Elementos Finitos**, 4^{ta} Edición, Editorial McGraw – Hill – Interamericana de España, S.A. España.

WWW.BDIGITAL.ULA.VE



APENDICE A

WWW.BDIGITAL.ULA.VE



ARCHIVO plus.inp

PRIMER CICLO

*Heading

**

*Node

1,	0.,	40.,	0.
2,	0.,	0.,	0.
3,	0.,	40.,	40.
4,	0.,	0.,	40.
5,	40.,	40.,	0.
6,	40.,	0.,	0.
7,	40.,	40.,	40.
8,	40.,	0.,	40.
9,	80.,	40.,	0.
10,	80.,	0.,	0.
11,	80.,	40.,	40.
12,	80.,	0.,	40.
13,	120.,	40.,	0.
14,	120.,	0.,	0.
15,	120.,	40.,	40.
16,	120.,	0.,	40.
17,	160.,	40.,	0.
18,	160.,	0.,	0.
19,	160.,	40.,	40.
20,	160.,	0.,	40.
21,	200.,	40.,	0.
22,	200.,	0.,	0.
23,	200.,	40.,	40.
24,	200.,	0.,	40.

*NSET,NSET=APOYO

1,2,4

**

*Element, type=C3D8, Elset= TODOS

1, 5, 6, 8, 7, 1, 2, 4, 3

Licencia Creative Commons:

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



```
2, 9, 10, 12, 11, 5, 6, 8, 7
3, 13, 14, 16, 15, 9, 10, 12, 11
4, 17, 18, 20, 19, 13, 14, 16, 15
5, 21, 22, 24, 23, 17, 18, 20, 19
**
*Solid section, Elset= TODOS, Material= MAT1
*Material, name= MAT1
*Elastic
210000., 0.30
**
*Restart, Write
**
*Step
*Static
*Dload
TODOS, P2, 1000
**
*Boundary
APOYO, 1, 3
**
**
*EL FILE, ELSET= TODOS
S, E
*EL PRINT
SENER
*ENERGY PRINT
SENER
*ENERGY FILE
SENER
**
**
**
*End Step
```

WWW.BDIGITAL.ULA.VE



ARCHIVO plus.gen

PRIMER CICLO

Archivo plus.gen

Primera linea:Nelem,Nnodos,CoordSAuxS

Primer grupo:Elem,NCood,8 Nodos

Segundo grupo: SisRef, Nodo,3 Coord

5,	24,	0,	0						
1,	1,	5,	6,	8,	7,	1,	2,	4,	3
2,	1,	9,	10,	12,	11,	5,	6,	8,	7
3,	1,	13,	14,	16,	15,	9,	10,	12,	11
4,	1,	17,	18,	20,	19,	13,	14,	16,	15
5,	1,	21,	22,	24,	23,	17,	18,	20,	19
0,	1,	.00,	40.00,	.00					
0,	2,	.00,	.00,	.00					
0,	3,	.00,	40.00,	40.00					
0,	4,	.00,	.00,	40.00					
0,	5,	40.00,	40.00,	.00					
0,	6,	40.00,	.00,	.00					
0,	7,	40.00,	40.00,	40.00					
0,	8,	40.00,	.00,	40.00					
0,	9,	80.00,	40.00,	.00					
0,	10,	80.00,	.00,	.00					
0,	11,	80.00,	40.00,	40.00					
0,	12,	80.00,	.00,	40.00					
0,	13,	120.00,	40.00,	.00					
0,	14,	120.00,	.00,	.00					
0,	15,	120.00,	40.00,	40.00					
0,	16,	120.00,	.00,	40.00					
0,	17,	160.00,	40.00,	.00					
0,	18,	160.00,	.00,	.00					
0,	19,	160.00,	40.00,	40.00					
0,	20,	160.00,	.00,	40.00					
0,	21,	200.00,	40.00,	.00					
0,	22,	200.00,	.00,	.00					

Licencia Creative Commons:

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



0,	23,	200.00,	40.00,	40.00
0,	24,	200.00,	.00,	40.00

ARCHIVO plus.sen

PRIMER CICLO

Archivo plus.sen

Contiene los valores de SENER
generados por ABAQUS en plus.dat

Orden: ELEMENTO NODO SENER

1	1	148.2000
1	2	16.7700
1	3	15.8400
1	4	148.2000
1	5	376.9000
1	6	230.2000
1	7	216.6000
1	8	376.9000
2	1	14.7600
2	2	27.8100
2	3	27.6100
2	4	14.7600
2	5	34.6000
2	6	44.9000
2	7	47.0800
2	8	34.6000
3	1	9.0820
3	2	9.5480
3	3	9.5790

Licencia Creative Commons:

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



3	4	9.0820
3	5	8.5050
3	6	9.0060
3	7	9.0000
3	8	8.5050
4	1	2.1520
4	2	1.9500
4	3	1.9490
4	4	2.1520
4	5	3.4140
4	6	3.2410
4	7	3.2530
4	8	3.4140
5	1	.0007
5	2	.0004
5	3	.0004
5	4	.0007
5	5	.0026
5	6	.0016
5	7	.0016
5	8	.0026

WWW.BDIGITAL.ULA.VE

ARCHIVO plus.sed

PRIMER CICLO

Archivo plus.sed

Contiene los valores de SED, SEDG

Numero de divisiones por elemento

Indicador de refinamiento BETA

Valores de SED por cada elemento

SED(1) 1529.6100

SED(2) 246.1200

Licencia Creative Commons:

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



SED(3) 72.3070
SED(4) 21.5250
SED(5) .0106

Valor de SEDG total: 1869.573000

Valor de SEDG= SEDG / N elem: 373.914600

SED= SED/SEDG (1)= 4.0908

SED= SED/SEDG (2)= .6582

SED= SED/SEDG (3)= .1934

SED= SED/SEDG (4)= .0576

SED= SED/SEDG (5)= .0000

Calculo de SED como SED/SEDG

Valor del indicador BETA seleccionado: 1.000000

Elemento	N div X	N div Y	N div Z
1	2	2	2
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1

~~WWW.BDIGITAL.ULA.VE~~

ARCHIVO plus2.inp

SEGUNDO CICLO

*HEADING

**

*NODE

1,	40.00,	40.00,	.00
2,	40.00,	20.00,	.00
3,	40.00,	.00,	.00
4,	40.00,	40.00,	20.00
5,	40.00,	20.00,	20.00
6,	40.00,	.00,	20.00
7,	40.00,	40.00,	40.00
8,	40.00,	20.00,	40.00
9,	40.00,	.00,	40.00
10,	20.00,	40.00,	.00

Licencia Creative Commons:

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



11,	20.00,	20.00,	.00
12,	20.00,	.00,	.00
13,	20.00,	40.00,	20.00
14,	20.00,	20.00,	20.00
15,	20.00,	.00,	20.00
16,	20.00,	40.00,	40.00
17,	20.00,	20.00,	40.00
18,	20.00,	.00,	40.00
19,	.00,	40.00,	.00
20,	.00,	20.00,	.00
21,	.00,	.00,	.00
22,	.00,	40.00,	20.00
23,	.00,	20.00,	20.00
24,	.00,	.00,	20.00
25,	.00,	40.00,	40.00
26,	.00,	20.00,	40.00
27,	.00,	.00,	40.00
28,	80.00,	40.00,	.00
29,	80.00,	.00,	.00
30,	80.00,	40.00,	40.00
31,	80.00,	.00,	40.00
32,	120.00,	40.00,	.00
33,	120.00,	.00,	.00
34,	120.00,	40.00,	40.00
35,	120.00,	.00,	40.00
36,	160.00,	40.00,	.00
37,	160.00,	.00,	.00
38,	160.00,	40.00,	40.00
39,	160.00,	.00,	40.00
40,	200.00,	40.00,	.00
41,	200.00,	.00,	.00
42,	200.00,	40.00,	40.00
43,	200.00,	.00,	40.00

**

*ELEMENT, TYPE=C3D8, ELSET= TODOS

1,	1,	2,	5,	4,	10,	11,	14,	13
2,	2,	3,	6,	5,	11,	12,	15,	14
3,	4,	5,	8,	7,	13,	14,	17,	16

Licencia Creative Commons:

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



4,	5,	6,	9,	8,	14,	15,	18,	17
5,	10,	11,	14,	13,	19,	20,	23,	22
6,	11,	12,	15,	14,	20,	21,	24,	23
7,	13,	14,	17,	16,	22,	23,	26,	25
8,	14,	15,	18,	17,	23,	24,	27,	26
9,	28,	29,	31,	30,	1,	3,	9,	7
10,	32,	33,	35,	34,	28,	29,	31,	30
11,	36,	37,	39,	38,	32,	33,	35,	34
12,	40,	41,	43,	42,	36,	37,	39,	38

*SOLID SECTION,ELSET= TODOS,MATERIAL=MAT1

*MATERIAL,NAME=MAT1

*ELASTIC

210000,0.3

*RESTART,WRITE

**

*STEP

*STATIC

*DLOAD

TODOS,P2,1000

**

*BOUNDARY

APOYO,1,3

**

*NSET, NSET=APOYO

20,21,22,23,24,25,26,27

**

**MPC

**LINEAR, 2, 1, 3

**LINEAR, 4, 1, 7

**LINEAR, 6, 3, 9

**LINEAR, 8, 7, 9

**BILINEAR, 5, 2, 4, 8, 6

**

*EL FILE,ELSET= TODOS

S,E

*EL PRINT

SENER

*ENERGY PRINT



SENER
 *ENERGY FILE
 SENER
 *END STEP

ARCHIVO plus2.gen
SEGUNDO CICLO

Archivo plus.gen

Primera linea:Nelem,Nnodos,CoordSAuxS

Primer grupo:Elem,NCood,8 Nodos

Segundo grupo: SisRef, Nodo,3 Coord

12,	43,	0,	0						
1,	1,	1,	2,	5,	4,	10,	11,	14,	13
2,	1,	2,	3,	6,	5,	11,	12,	15,	14
3,	1,	4,	5,	8,	7,	13,	14,	17,	16
4,	1,	5,	6,	9,	8,	14,	15,	18,	17
5,	1,	10,	11,	14,	13,	19,	20,	23,	22
6,	1,	11,	12,	15,	14,	20,	21,	24,	23
7,	1,	13,	14,	17,	16,	22,	23,	26,	25
8,	1,	14,	15,	18,	17,	23,	24,	27,	26
9,	1,	28,	29,	31,	30,	1,	3,	9,	7
10,	1,	32,	33,	35,	34,	28,	29,	31,	30
11,	1,	36,	37,	39,	38,	32,	33,	35,	34
12,	1,	40,	41,	43,	42,	36,	37,	39,	38
0,	1,	40.00,	40.00,			.00			
0,	2,	40.00,	20.00,			.00			
0,	3,	40.00,	.00,			.00			
0,	4,	40.00,	40.00,			20.00			
0,	5,	40.00,	20.00,			20.00			

Licencia Creative Commons:

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



0,	6,	40.00,	.00,	20.00
0,	7,	40.00,	40.00,	40.00
0,	8,	40.00,	20.00,	40.00
0,	9,	40.00,	.00,	40.00
0,	10,	20.00,	40.00,	.00
0,	11,	20.00,	20.00,	.00
0,	12,	20.00,	.00,	.00
0,	13,	20.00,	40.00,	20.00
0,	14,	20.00,	20.00,	20.00
0,	15,	20.00,	.00,	20.00
0,	16,	20.00,	40.00,	40.00
0,	17,	20.00,	20.00,	40.00
0,	18,	20.00,	.00,	40.00
0,	19,	.00,	40.00,	.00
0,	20,	.00,	20.00,	.00
0,	21,	.00,	.00,	.00
0,	22,	.00,	40.00,	20.00
0,	23,	.00,	20.00,	20.00
0,	24,	.00,	.00,	20.00
0,	25,	.00,	40.00,	40.00
0,	26,	.00,	20.00,	40.00
0,	27,	.00,	.00,	40.00
0,	28,	80.00,	40.00,	.00
0,	29,	80.00,	.00,	.00
0,	30,	80.00,	40.00,	40.00
0,	31,	80.00,	.00,	40.00
0,	32,	120.00,	40.00,	.00
0,	33,	120.00,	.00,	.00
0,	34,	120.00,	40.00,	40.00
0,	35,	120.00,	.00,	40.00
0,	36,	160.00,	40.00,	.00
0,	37,	160.00,	.00,	.00
0,	38,	160.00,	40.00,	40.00
0,	39,	160.00,	.00,	40.00
0,	40,	200.00,	40.00,	.00
0,	41,	200.00,	.00,	.00
0,	42,	200.00,	40.00,	40.00
0,	43,	200.00,	.00,	40.00

Licencia Creative Commons:

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



ARCHIVO plus2.sen
SEGUNDO CICLO

Archivo plus.sen

Contiene los valores de SENER
generados por ABAQUS en plus.dat

Orden: ELEMENTO NODO SENER

1	1	21.0400
1	2	6.5950
1	3	6.5950
1	4	4.1340
1	5	42.6400
1	6	17.5900
1	7	17.5900
1	8	9.3070
2	1	4.5490
2	2	7.1180
2	3	3.0330
2	4	2.9310
2	5	8.5890
2	6	14.9200
2	7	4.9140
2	8	9.8170
3	1	4.5490
3	2	3.0330
3	3	7.1180
3	4	2.9310
3	5	8.5890
3	6	4.9140
3	7	14.9200
3	8	9.8170
4	1	3.4640
4	2	4.7240
4	3	4.7240

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



4	4	8.9570
4	5	4.6300
4	6	10.6000
4	7	10.6000
4	8	18.5000
5	1	66.5500
5	2	177.0000
5	3	177.0000
5	4	114.4000
5	5	194.1000
5	6	237.7000
5	7	237.7000
5	8	83.9700
6	1	123.4000
6	2	40.9700
6	3	42.7000
6	4	14.1200
6	5	117.6000
6	6	35.5200
6	7	52.5800
6	8	26.7100
7	1	123.4000
7	2	42.7000
7	3	40.9700
7	4	14.1200
7	5	117.6000
7	6	52.5800
7	7	35.5200
7	8	26.7100
8	1	9.0830
8	2	17.0100
8	3	17.0100
8	4	51.4700
8	5	27.4900
8	6	33.6700
8	7	33.6700
8	8	67.9700
9	1	.0324

WWW.BDIGITAL.ULA.VE



9	2	.0548
9	3	.0548
9	4	.0350
9	5	.1858
9	6	.2564
9	7	.2564
9	8	.2701
10	1	.0058
10	2	.0055
10	3	.0055
10	4	.0048
10	5	.0220
10	6	.0203
10	7	.0203
10	8	.0131
11	1	.0011
11	2	.0009
11	3	.0009
11	4	.0010
11	5	.0034
11	6	.0030
11	7	.0030
11	8	.0024
12	1	.0003
12	2	.0003
12	3	.0003
12	4	.0003
12	5	.0006
12	6	.0006
12	7	.0006
12	8	.0006

WWW.BDIGITAL.ULA.VE



ARCHIVO plus2.sed
SEGUNDO CICLO

Archivo plus.sed

Contiene los valores de SED, SEDG
Numero de divisiones por elemento
Indicador de refinamiento BETA

Valores de SED por cada elemento

SED(1) 125.4910
SED(2) 55.8710
SED(3) 55.8710
SED(4) 66.1990
SED(5) 1288.4200
SED(6) 453.5999
SED(7) 453.6000
SED(8) 257.3730
SED(9) 1.1457
SED(10) .0973
SED(11) .0157
SED(12) .0036

Valor de SEDG total: 2757.688000

Valor de SEDG= SEDG / N elem: 229.807300

SED= SED/SEDG (1)= .5461
SED= SED/SEDG (2)= .2431
SED= SED/SEDG (3)= .2431
SED= SED/SEDG (4)= .2881
SED= SED/SEDG (5)= 5.6065
SED= SED/SEDG (6)= 1.9738
SED= SED/SEDG (7)= 1.9738
SED= SED/SEDG (8)= 1.1200
SED= SED/SEDG (9)= .0050
SED= SED/SEDG (10)= .0004
SED= SED/SEDG (11)= .0001
SED= SED/SEDG (12)= .0000

Calculo de SED como SED/SEDG



14,	40.00,	20.00,	40.00
15,	20.00,	40.00,	40.00
16,	20.00,	20.00,	40.00
17,	40.00,	.00,	40.00
18,	20.00,	.00,	40.00
19,	20.00,	30.00,	.00
20,	20.00,	40.00,	10.00
21,	20.00,	30.00,	10.00
22,	20.00,	20.00,	10.00
23,	20.00,	30.00,	20.00
24,	10.00,	40.00,	.00
25,	10.00,	30.00,	.00
26,	10.00,	20.00,	.00
27,	10.00,	40.00,	10.00
28,	10.00,	30.00,	10.00
29,	10.00,	20.00,	10.00
30,	10.00,	40.00,	20.00
31,	10.00,	30.00,	20.00
32,	10.00,	20.00,	20.00
33,	.00,	40.00,	.00
34,	.00,	30.00,	.00
35,	.00,	20.00,	.00
36,	.00,	40.00,	10.00
37,	.00,	30.00,	10.00
38,	.00,	20.00,	10.00
39,	.00,	40.00,	20.00
40,	.00,	30.00,	20.00
41,	.00,	20.00,	20.00
42,	20.00,	10.00,	.00
43,	20.00,	10.00,	10.00
44,	20.00,	.00,	10.00
45,	20.00,	10.00,	20.00
46,	10.00,	10.00,	.00
47,	10.00,	.00,	.00
48,	10.00,	10.00,	10.00
49,	10.00,	.00,	10.00
50,	10.00,	10.00,	20.00
51,	10.00,	.00,	20.00

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



52,	.00,	10.00,	.00
53,	.00,	.00,	.00
54,	.00,	10.00,	10.00
55,	.00,	.00,	10.00
56,	.00,	10.00,	20.00
57,	.00,	.00,	20.00
58,	20.00,	40.00,	30.00
59,	20.00,	30.00,	30.00
60,	20.00,	20.00,	30.00
61,	20.00,	30.00,	40.00
62,	10.00,	40.00,	30.00
63,	10.00,	30.00,	30.00
64,	10.00,	20.00,	30.00
65,	10.00,	40.00,	40.00
66,	10.00,	30.00,	40.00
67,	10.00,	20.00,	40.00
68,	.00,	40.00,	30.00
69,	.00,	30.00,	30.00
70,	.00,	20.00,	30.00
71,	.00,	40.00,	40.00
72,	.00,	30.00,	40.00
73,	.00,	20.00,	40.00
74,	20.00,	10.00,	30.00
75,	20.00,	.00,	30.00
76,	20.00,	10.00,	40.00
77,	10.00,	10.00,	30.00
78,	10.00,	.00,	30.00
79,	10.00,	10.00,	40.00
80,	10.00,	.00,	40.00
81,	.00,	10.00,	30.00
82,	.00,	.00,	30.00
83,	.00,	10.00,	40.00
84,	.00,	.00,	40.00
85,	80.00,	40.00,	.00
86,	80.00,	.00,	.00
87,	80.00,	40.00,	40.00
88,	80.00,	.00,	40.00
89,	120.00,	40.00,	.00

Licencia Creative Commons:

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



90,	120.00,	.00,	.00
91,	120.00,	40.00,	40.00
92,	120.00,	.00,	40.00
93,	160.00,	40.00,	.00
94,	160.00,	.00,	.00
95,	160.00,	40.00,	40.00
96,	160.00,	.00,	40.00
97,	200.00,	40.00,	.00
98,	200.00,	.00,	.00
99,	200.00,	40.00,	40.00
100,	200.00,	.00,	40.00

**

*ELEMENT,TYPE=C3D8,ELSET= TODOS

1,	1,	2,	4,	3,	5,	6,	8,	7
2,	2,	9,	10,	4,	6,	11,	12,	8
3,	3,	4,	14,	13,	7,	8,	16,	15
4,	4,	10,	17,	14,	8,	12,	18,	16
5,	5,	19,	21,	20,	24,	25,	28,	27
6,	19,	6,	22,	21,	25,	26,	29,	28
7,	20,	21,	23,	7,	27,	28,	31,	30
8,	21,	22,	8,	23,	28,	29,	32,	31
9,	24,	25,	28,	27,	33,	34,	37,	36
10,	25,	26,	29,	28,	34,	35,	38,	37
11,	27,	28,	31,	30,	36,	37,	40,	39
12,	28,	29,	32,	31,	37,	38,	41,	40
13,	6,	42,	43,	22,	26,	46,	48,	29
14,	42,	11,	44,	43,	46,	47,	49,	48
15,	22,	43,	45,	8,	29,	48,	50,	32
16,	43,	44,	12,	45,	48,	49,	51,	50
17,	26,	46,	48,	29,	35,	52,	54,	38
18,	46,	47,	49,	48,	52,	53,	55,	54
19,	29,	48,	50,	32,	38,	54,	56,	41
20,	48,	49,	51,	50,	54,	55,	57,	56
21,	7,	23,	59,	58,	30,	31,	63,	62
22,	23,	8,	60,	59,	31,	32,	64,	63
23,	58,	59,	61,	15,	62,	63,	66,	65
24,	59,	60,	16,	61,	63,	64,	67,	66
25,	30,	31,	63,	62,	39,	40,	69,	68

Licencia Creative Commons:

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



LINEAR, 44, 11, 12
LINEAR, 45, 12, 8
LINEAR, 75, 12, 18
LINEAR, 76, 18, 16
LINEAR, 60, 16, 8
LINEAR, 61, 16, 15
LINEAR, 58, 15, 7
BILINEAR, 4, 1, 9, 17, 13
BILINEAR, 21, 5, 6, 8, 7
BILINEAR, 43, 6, 11, 12, 8
BILINEAR, 74, 8, 12, 18, 16
BILINEAR, 59, 7, 8, 16, 15

**

*EL FILE,ELSET= TODOS

S,E

*EL PRINT

SENER

*ENERGY PRINT

SENER

*ENERGY FILE

SENER

*END STEP

ARCHIVO plus3.gen

TERCER CICLO

Archivo plus.gen

Primera linea:Nelem,Nodos,CoordSAuxS

Primer grupo:Elem,NCood,8 Nodos

Segundo grupo: SisRef, Nodo,3 Coord

40, 100, 0, 0

1, 1, 1, 2, 4, 3, 5, 6, 8, 7

Licencia Creative Commons:

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



2,	1,	2,	9,	10,	4,	6,	11,	12,	8
3,	1,	3,	4,	14,	13,	7,	8,	16,	15
4,	1,	4,	10,	17,	14,	8,	12,	18,	16
5,	1,	5,	19,	21,	20,	24,	25,	28,	27
6,	1,	19,	6,	22,	21,	25,	26,	29,	28
7,	1,	20,	21,	23,	7,	27,	28,	31,	30
8,	1,	21,	22,	8,	23,	28,	29,	32,	31
9,	1,	24,	25,	28,	27,	33,	34,	37,	36
10,	1,	25,	26,	29,	28,	34,	35,	38,	37
11,	1,	27,	28,	31,	30,	36,	37,	40,	39
12,	1,	28,	29,	32,	31,	37,	38,	41,	40
13,	1,	6,	42,	43,	22,	26,	46,	48,	29
14,	1,	42,	11,	44,	43,	46,	47,	49,	48
15,	1,	22,	43,	45,	8,	29,	48,	50,	32
16,	1,	43,	44,	12,	45,	48,	49,	51,	50
17,	1,	26,	46,	48,	29,	35,	52,	54,	38
18,	1,	46,	47,	49,	48,	52,	53,	55,	54
19,	1,	29,	48,	50,	32,	38,	54,	56,	41
20,	1,	48,	49,	51,	50,	54,	55,	57,	56
21,	1,	7,	23,	59,	58,	30,	31,	63,	62
22,	1,	23,	8,	60,	59,	31,	32,	64,	63
23,	1,	58,	59,	61,	15,	62,	63,	66,	65
24,	1,	59,	60,	16,	61,	63,	64,	67,	66
25,	1,	30,	31,	63,	62,	39,	40,	69,	68
26,	1,	31,	32,	64,	63,	40,	41,	70,	69
27,	1,	62,	63,	66,	65,	68,	69,	72,	71
28,	1,	63,	64,	67,	66,	69,	70,	73,	72
29,	1,	8,	45,	74,	60,	32,	50,	77,	64
30,	1,	45,	12,	75,	74,	50,	51,	78,	77
31,	1,	60,	74,	76,	16,	64,	77,	79,	67
32,	1,	74,	75,	18,	76,	77,	78,	80,	79
33,	1,	32,	50,	77,	64,	41,	56,	81,	70
34,	1,	50,	51,	78,	77,	56,	57,	82,	81
35,	1,	64,	77,	79,	67,	70,	81,	83,	73
36,	1,	77,	78,	80,	79,	81,	82,	84,	83
37,	1,	85,	86,	88,	87,	1,	9,	17,	13
38,	1,	89,	90,	92,	91,	85,	86,	88,	87
39,	1,	93,	94,	96,	95,	89,	90,	92,	91

Licencia Creative Commons:

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



	1,	97,	98,	100,	99,	93,	94,	96,	95
40,	1,	40.00,	40.00,			.00			
0,	2,	40.00,	20.00,			.00			
0,	3,	40.00,	40.00,		20.00				
0,	4,	40.00,	20.00,		20.00				
0,	5,	20.00,	40.00,		.00				
0,	6,	20.00,	20.00,		.00				
0,	7,	20.00,	40.00,		20.00				
0,	8,	20.00,	20.00,		20.00				
0,	9,	40.00,	.00,		.00				
0,	10,	40.00,	.00,		20.00				
0,	11,	20.00,	.00,		.00				
0,	12,	20.00,	.00,		20.00				
0,	13,	40.00,	40.00,		40.00				
0,	14,	40.00,	20.00,		40.00				
0,	15,	20.00,	40.00,		40.00				
0,	16,	20.00,	20.00,		40.00				
0,	17,	40.00,	.00,		40.00				
0,	18,	20.00,	.00,		40.00				
0,	19,	20.00,	30.00,		.00				
0,	20,	20.00,	40.00,		10.00				
0,	21,	20.00,	30.00,		10.00				
0,	22,	20.00,	20.00,		10.00				
0,	23,	20.00,	30.00,		20.00				
0,	24,	10.00,	40.00,		.00				
0,	25,	10.00,	30.00,		.00				
0,	26,	10.00,	20.00,		.00				
0,	27,	10.00,	40.00,		10.00				
0,	28,	10.00,	30.00,		10.00				
0,	29,	10.00,	20.00,		10.00				
0,	30,	10.00,	40.00,		20.00				
0,	31,	10.00,	30.00,		20.00				
0,	32,	10.00,	20.00,		20.00				
0,	33,	.00,	40.00,		.00				
0,	34,	.00,	30.00,		.00				
0,	35,	.00,	20.00,		.00				
0,	36,	.00,	40.00,		10.00				
0,	37,	.00,	30.00,		10.00				

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



0,	38,	.00,	20.00,	10.00
0,	39,	.00,	40.00,	20.00
0,	40,	.00,	30.00,	20.00
0,	41,	.00,	20.00,	20.00
0,	42,	20.00,	10.00,	.00
0,	43,	20.00,	10.00,	10.00
0,	44,	20.00,	.00,	10.00
0,	45,	20.00,	10.00,	20.00
0,	46,	10.00,	10.00,	.00
0,	47,	10.00,	.00,	.00
0,	48,	10.00,	10.00,	10.00
0,	49,	10.00,	.00,	10.00
0,	50,	10.00,	10.00,	20.00
0,	51,	10.00,	.00,	20.00
0,	52,	.00,	10.00,	.00
0,	53,	.00,	.00,	.00
0,	54,	.00,	10.00,	10.00
0,	55,	.00,	.00,	10.00
0,	56,	.00,	10.00,	20.00
0,	57,	.00,	.00,	20.00
0,	58,	20.00,	40.00,	30.00
0,	59,	20.00,	30.00,	30.00
0,	60,	20.00,	20.00,	30.00
0,	61,	20.00,	30.00,	40.00
0,	62,	10.00,	40.00,	30.00
0,	63,	10.00,	30.00,	30.00
0,	64,	10.00,	20.00,	30.00
0,	65,	10.00,	40.00,	40.00
0,	66,	10.00,	30.00,	40.00
0,	67,	10.00,	20.00,	40.00
0,	68,	.00,	40.00,	30.00
0,	69,	.00,	30.00,	30.00
0,	70,	.00,	20.00,	30.00
0,	71,	.00,	40.00,	40.00
0,	72,	.00,	30.00,	40.00
0,	73,	.00,	20.00,	40.00
0,	74,	20.00,	10.00,	30.00
0,	75,	20.00,	.00,	30.00

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



0,	76,	20.00,	10.00,	40.00
0,	77,	10.00,	10.00,	30.00
0,	78,	10.00,	.00,	30.00
0,	79,	10.00,	10.00,	40.00
0,	80,	10.00,	.00,	40.00
0,	81,	.00,	10.00,	30.00
0,	82,	.00,	.00,	30.00
0,	83,	.00,	10.00,	40.00
0,	84,	.00,	.00,	40.00
0,	85,	80.00,	40.00,	.00
0,	86,	80.00,	.00,	.00
0,	87,	80.00,	40.00,	40.00
0,	88,	80.00,	.00,	40.00
0,	89,	120.00,	40.00,	.00
0,	90,	120.00,	.00,	.00
0,	91,	120.00,	40.00,	40.00
0,	92,	120.00,	.00,	40.00
0,	93,	160.00,	40.00,	.00
0,	94,	160.00,	.00,	.00
0,	95,	160.00,	40.00,	40.00
0,	96,	160.00,	.00,	40.00
0,	97,	200.00,	40.00,	.00
0,	98,	200.00,	.00,	.00
0,	99,	200.00,	40.00,	40.00
0,	100,	200.00,	.00,	40.00

ARCHIVO plus3.sen

TERCER CICLO

Archivo plus.sen

Contiene los valores de SENER
generados por ABAQUS en plus.dat
Orden: ELEMENTO NODO SENER

Licencia Creative Commons:

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



5	7	101.7000
5	8	114.5000
6	1	81.5900
6	2	84.3200
6	3	125.7000
6	4	132.8000
6	5	79.7300
6	6	84.9500
6	7	126.8000
6	8	136.5000
7	1	141.5000
7	2	147.2000
7	3	178.5000
7	4	163.6000
7	5	166.9000
7	6	165.5000
7	7	209.6000
7	8	188.0000
8	1	155.9000
8	2	161.6000
8	3	158.1000
8	4	153.7000
8	5	170.4000
8	6	179.7000
8	7	171.4000
8	8	170.6000
9	1	19.9900
9	2	18.7300
9	3	56.3200
9	4	53.7100
9	5	35.8800
9	6	33.8800
9	7	69.8900
9	8	66.4200
10	1	21.6600
10	2	23.6900
10	3	56.6100
10	4	60.8700



10	5	35.0400
10	6	34.7800
10	7	67.8900
10	8	69.9300
11	1	189.9000
11	2	186.1000
11	3	310.7000
11	4	330.5000
11	5	178.9000
11	6	175.9000
11	7	288.6000
11	8	311.8000
12	1	230.4000
12	2	245.8000
12	3	374.4000
12	4	412.5000
12	5	219.4000
12	6	226.8000
12	7	370.1000
12	8	402.0000
13	1	92.8700
13	2	100.7000
13	3	143.4000
13	4	153.1000
13	5	96.0600
13	6	105.0000
13	7	147.6000
13	8	158.3000
14	1	94.9500
14	2	85.3200
14	3	149.7000
14	4	138.7000
14	5	100.9000
14	6	94.3200
14	7	163.5000
14	8	155.6000
15	1	170.0000
15	2	180.7000

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



15	3	160.3000
15	4	172.4000
15	5	190.1000
15	6	200.7000
15	7	181.8000
15	8	193.9000
16	1	177.4000
16	2	167.1000
16	3	170.4000
16	4	162.2000
16	5	205.5000
16	6	198.4000
16	7	199.8000
16	8	194.9000
17	1	29.2300
17	2	32.9700
17	3	66.8000
17	4	72.5500
17	5	38.3100
17	6	41.2700
17	7	74.8800
17	8	80.0400
18	1	42.6800
18	2	59.4400
18	3	93.3000
18	4	120.6000
18	5	51.3500
18	6	66.1000
18	7	95.9300
18	8	121.4000
19	1	267.5000
19	2	273.6000
19	3	451.7000
19	4	460.4000
19	5	244.6000
19	6	251.5000
19	7	432.6000
19	8	441.9000

WWW.BDIGITAL.ULA.VE



20	1	306.9000
20	2	350.5000
20	3	506.2000
20	4	565.9000
20	5	277.6000
20	6	314.3000
20	7	476.3000
20	8	529.6000
21	1	189.0000
21	2	145.9000
21	3	145.1000
21	4	106.7000
21	5	224.1000
21	6	178.3000
21	7	174.6000
21	8	133.8000
22	1	128.1000
22	2	100.6000
22	3	88.2800
22	4	55.2300
22	5	152.6000
22	6	134.3000
22	7	120.6000
22	8	97.3600
23	1	86.5500
23	2	37.8600
23	3	50.2800
23	4	13.1600
23	5	79.3500
23	6	55.1300
23	7	38.5400
23	8	22.3800
24	1	18.2600
24	2	10.3500
24	3	25.5500
24	4	25.2800
24	5	47.7800
24	6	43.3700



24	7	39.9400
24	8	45.3200
25	1	450.5000
25	2	336.4000
25	3	610.5000
25	4	159.6000
25	5	386.7000
25	6	383.8000
25	7	837.9000
25	8	461.3000
26	1	211.2000
26	2	250.6000
26	3	39.9000
26	4	43.1500
26	5	405.6000
26	6	432.4000
26	7	218.3000
26	8	207.7000
27	1	223.3000
27	2	12.7600
27	3	15.7100
27	4	8.1180
27	5	655.1000
27	6	239.8000
27	7	211.3000
27	8	37.4800
28	1	41.4500
28	2	23.9400
28	3	11.1600
28	4	11.5900
28	5	49.9800
28	6	21.3800
28	7	14.6900
28	8	8.7520
29	1	93.3600
29	2	106.0000
29	3	44.3400
29	4	49.8800

WWW.BPDIGITAL.ULA.VE

Licencia Creative Commons:

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



29	5	134.8000
29	6	145.7000
29	7	88.5100
29	8	91.8400
30	1	110.9000
30	2	113.1000
30	3	52.5500
30	4	59.6400
30	5	155.5000
30	6	157.2000
30	7	94.3100
30	8	100.7000
31	1	11.9100
31	2	10.8000
31	3	12.9700
31	4	8.7510
31	5	45.6200
31	6	44.7800
31	7	39.7100
31	8	35.0800
32	1	12.5700
32	2	17.1400
32	3	7.2490
32	4	5.9090
32	5	47.4800
32	6	53.8200
32	7	33.7000
32	8	33.9300
33	1	313.1000
33	2	318.5000
33	3	79.1200
33	4	77.1500
33	5	487.8000
33	6	524.0000
33	7	244.4000
33	8	271.0000
34	1	335.1000
34	2	369.8000

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



34	3	68.7400
34	4	66.5100
34	5	552.2000
34	6	598.3000
34	7	281.0000
34	8	286.6000
35	1	18.2400
35	2	22.0700
35	3	8.4440
35	4	9.5330
35	5	11.8600
35	6	15.6800
35	7	3.0310
35	8	3.2450
36	1	24.2600
36	2	23.7700
36	3	10.2600
36	4	9.3450
36	5	17.7300
36	6	23.2900
36	7	4.8620
36	8	8.4820
37	1	36.1200
37	2	39.3400
37	3	9.3400
37	4	11.6600
37	5	45.5400
37	6	41.9800
37	7	16.6400
37	8	12.5800
38	1	10.9200
38	2	14.1000
38	3	4.6350
38	4	7.4340
38	5	15.8100
38	6	14.7500
38	7	10.2900
38	8	8.1680

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



39	1	2.0460
39	2	2.2530
39	3	2.3910
39	4	1.8730
39	5	2.0220
39	6	2.7800
39	7	2.7830
39	8	2.9020
40	1	.0246
40	2	.0209
40	3	.0268
40	4	.0155
40	5	.0560
40	6	.0439
40	7	.0648
40	8	.0313

~~WWW.BDIGITAL.ULA.VE~~
ARCHIVO plus3.sed
TERCER CICLO

Archivo plus.sed

Contiene los valores de SED, SEDG

Numero de divisiones por elemento

Indicador de refinamiento BETA

Valores de SED por cada elemento

SED(1) 652.5601

SED(2) 652.3900

SED(3) 273.6250

SED(4) 152.7850

SED(5) 705.8800

SED(6) 852.3900

SED(7) 1360.8000

Licencia Creative Commons:

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



SED(8) 1321.4000
SED(9) 354.8200
SED(10) 370.4700
SED(11) 1972.4000
SED(12) 2481.4000
SED(13) 997.0300
SED(14) 982.9900
SED(15) 1449.9000
SED(16) 1475.7000
SED(17) 436.0500
SED(18) 650.8000
SED(19) 2823.8000
SED(20) 3327.3000
SED(21) 1297.5000
SED(22) 877.0700
SED(23) 383.2500
SED(24) 255.8500
SED(25) 3626.7000
SED(26) 1808.8500
SED(27) 1403.5680
SED(28) 182.9420
SED(29) 754.4301
SED(30) 843.9000
SED(31) 209.6210
SED(32) 211.7980
SED(33) 2315.0700
SED(34) 2558.2500
SED(35) 92.1030
SED(36) 121.9990
SED(37) 213.2000
SED(38) 86.1070
SED(39) 19.0500
SED(40) .2838

Valor de SEDG total: 40556.030000

Valor de SEDG= SEDG / N elem: 1013.901000

SED= SED/SEDG (1)= .6436

SED= SED/SEDG (2)= .6434

SED= SED/SEDG (3)= .2699

Licencia Creative Commons:

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



SED= SED/SEDG (4)=	.1507
SED= SED/SEDG (5)=	.6962
SED= SED/SEDG (6)=	.8407
SED= SED/SEDG (7)=	1.3421
SED= SED/SEDG (8)=	1.3033
SED= SED/SEDG (9)=	.3500
SED= SED/SEDG (10)=	.3654
SED= SED/SEDG (11)=	1.9454
SED= SED/SEDG (12)=	2.4474
SED= SED/SEDG (13)=	.9834
SED= SED/SEDG (14)=	.9695
SED= SED/SEDG (15)=	1.4300
SED= SED/SEDG (16)=	1.4555
SED= SED/SEDG (17)=	.4301
SED= SED/SEDG (18)=	.6419
SED= SED/SEDG (19)=	2.7851
SED= SED/SEDG (20)=	3.2817
SED= SED/SEDG (21)=	1.2797
SED= SED/SEDG (22)=	.8650
SED= SED/SEDG (23)=	.3780
SED= SED/SEDG (24)=	.2523
SED= SED/SEDG (25)=	3.5770
SED= SED/SEDG (26)=	1.7841
SED= SED/SEDG (27)=	1.3843
SED= SED/SEDG (28)=	.1804
SED= SED/SEDG (29)=	.7441
SED= SED/SEDG (30)=	.8323
SED= SED/SEDG (31)=	.2067
SED= SED/SEDG (32)=	.2089
SED= SED/SEDG (33)=	2.2833
SED= SED/SEDG (34)=	2.5232
SED= SED/SEDG (35)=	.0908
SED= SED/SEDG (36)=	.1203
SED= SED/SEDG (37)=	.2103
SED= SED/SEDG (38)=	.0849
SED= SED/SEDG (39)=	.0188
SED= SED/SEDG (40)=	.0003

Calculo de SED como SED/SEDG



Valor del indicador BETA seleccionado: 1.000000

Elemento	N div X	N div Y	N div Z
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	2	2	2
8	2	2	2
9	1	1	1
10	1	1	1
11	2	2	2
12	2	2	2
13	1	1	1
14	1	1	1
15	2	2	2
16	2	2	2
17	1	1	1
18	1	1	1
19	2	2	2
20	2	2	2
21	2	2	2
22	1	1	1
23	1	1	1
24	1	1	1
25	2	2	2
26	2	2	2
27	2	2	2
28	1	1	1
29	1	1	1
30	1	1	1
31	1	1	1
32	1	1	1
33	2	2	2
34	2	2	2
35	1	1	1
36	1	1	1

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



37	1	1	1
38	1	1	1
39	1	1	1
40	1	1	1

WWW.BDIGITAL.ULA.VE



PROGRAMA DE INTERPRETE DE COMANDOS EN SHELL DE UNIX PlusRun.sch

```
#!/bin/csh
if ( ! ( `expr $2 + 0` ) ) then
    echo Error de sintaxis....
    echo " " El segundo argumento debe ser numerico
    exit 1
endif
if ( !(-e plusone) ) then
    echo Error .... Es necesario el programa PlusOne en el directorio actual
    exit
endif
if ( !(-e plus.inp) ) then
    echo Error .... Es necesario el archivo plus.inp en el directorio actual
    exit
endif
if ( $1 == "" ) then
    echo Error de sintaxis....
    echo " " "${0}" "<archivo inp sin extension>"
    exit
endif
if ( !(-e $1.inp) ) then
    echo Error .... ${1}.inp no existe
    exit
endif
echo *****
```

Licencia Creative Commons:

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



```
echo **      PROGRAMA PLUS RUN      **
echo **
echo **      Ing. Maria Elisa Elberg  **
echo *****
echo Entrando al ciclo 0
abaqus job=${1}
set i=0
while ( ${i} != $2 )
    echo Entrando al ciclo ${i}
    plusone
    cp ${1}.inp ${1}_${i}.inp
    cp ${1}.dat ${1}_${i}.dat
    cp ${1}.gen ${1}_${i}.gen
    cp ${1}.sen ${1}_${i}.sen
    cp ${1}.sed ${1}_${i}.sed
    cp ${1}.odb ${1}_${i}.odb
    abaqus job=${i}
    set i='expr ${i} + 1'
end
```

WWW.BDIGITAL.ULA.VE



APENDICE C

WWW.BDIGITAL.ULA.VE



PROGRAMA PlusOne

PROGRAM PLUSONE

COMMON/MODEL/NNODE,LNODE,NDIME

DIMENSION CORDB(18000,3),MATNB(20),LNODB(18000,18000),ORIGB(10,3),
1 LNODR(20,20),NBUSD(20),NBNGH(20),NDIVX(20),NDIVY(20),
2 NDIVZ(20),WEITX(20,30),WEITY(20,30),WEITZ(20,30),
3 LREP(300),LASOC(300),LFINN(300),LFASC(300),COORD
4 (4800,3),MATNO(600),LNODS(600,8),NDUMY(20,4),DUMYX
5 (20,30),DUMYY(20,30),DUMYZ(20,30),NNCNEC(4800)

DIMENSION ELAST(20),CNABLA(20),POBIL(400,3),NNCOR(400,3),
1 PNOBIL(400,3),CBILI(400,3),CODAS(18000,8,3),
2 CODASII(400,3),CODASIS(400,3),CODASDI(400,3),
2 CODASDS(400,3),CLIN(400,3),LASO1(400,3),
3 LASO2(400,3)

DIMENSION CARAI(20,400,3),PLIL(400,3),CHEQ(3),CALII(400,3),
1 CALID(400,3),COLMBD(400,3),COLMBR(400,3),COLMBU(400,3)
2 ,COLMBL(400,3),COBIL(400,3),COBIR(400,3),COBRD(400,3)
3 ,COBRU(400,3),COBUL(400,3),COBUR(400,3),COBLU(400,3),
4 COBLD(400,3),SALTO(4,4),SED(20),XBLOC(20),XBLOC1(20)

INTEGER PONTRS(4800),ORIGN,POLA,CAMBIO,IPOL(4800),NODOIS(400),
1 NODOII(400),NODODI(400),NODODS(400),IPOLI(4800),
2 NODII(400),NODIS(400)

INTEGER COORD,SELEC,NM,NL,ICON,INCON,LINE(400),LINE1(400)
1 ,LINE2(400),LIRCI(3),LIRCD(3),IG,PILAS,K1,K2,K3,LNODB

DOUBLE PRECISION VALOR

REAL TOTX,TOTY,TOTZ,SEDG,BETA,NCOR,NNCOR,CORDB,CODAS,
2 CODASII,CODASIS,CODASDI,CODASDS

WRITE(*,*)'Plus One ACIVADO'
WRITE(*,*)

C
C
C

DATA NNODE,LNODE,NDIME/8,20,3/

CALL DIMEN (MBLOC,MPONT,MDIVI,MREP,MPOLA,MELEM,MPOIN)

CALL GENIN (NBLOC,NPONT)

CALL GENSEN (NBLOC,NPONT)

CALL ADAPTIVE (NBLOC,MPONT,MOPLA,MBLOC,MDIVI,SED,
1 CORDB,LNODB,NDIVX,NDIVY,NDIVZ,WEITX,WEITY,
2 WEITZ,SEDG,BETA,ORIGB,MATNB,PILAS)

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)

IF (PILAS.NE.1) THEN
CALL REORDR (NBLOC,MATNB,LNODB,LNODR,NBUSD,NBNGH,
1 NDIVX,NDIVY,NDIVZ,WEITX,WEITY,WEITZ,

Licencia Creative Commons:



```

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,MREPN,MELEM,
3      MPOIN,NL,ICON,NRENM,ORIGN,IPOL,NNCNEC,CBILI,CODASIS
4      ,CODASII,CODASDS,CODASDI,NODOIS,NODOII,NODODI,NODODS,
5      IPOLI,NODII,NODIS,PLIL,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      NODOII,NODOIS,NODODI,NODODS,IPOLI,NODII,NODIS,ICON,
6      IG)
    
```

```

    WRITE(*,*)Plus One DESACTIVADO ... ..
    WRITE (*,*)
    
```

END

C Subrutina GENIN

C

SUBROUTINE GENIN (NBLOC,NPONT)

C

COMMON/MODEL/NNODE,LNODE,NDIME

```

    DIMENSION ELEM(900),NOEL1(900),NOEL2(900),
1      NOEL3(900),NOEL4(900),NOEL5(900),
2      NOEL6(900),NOE7(900),NOEL8(900),NODE(900),
3      CORDB(900,4)
    REAL CORDB
    
```

```

    INTEGER I,NNODE,NELEM,JELEM,NUMEL,AUX,J, NNIMP, NNPAR,NODOS
1      NODOFIN,C,F,NOEL1,NOEL2,NOEL3,NOEL4,NOEL5,
2      NOEL6,NOE7,NOEL8
    
```

CHARACTER*15 A,M

```

    OPEN(UNIT=1,FILE="plus.inp",ACCESS='SEQUENTIAL',STATUS='OLD')
    
```

```

    DO 110 J=1,30
    
```

```

701      READ(1,701)M
          FORMAT (A5)
    
```

```

          IF (M.EQ.'*NODE') GOTO 576
          IF (M.EQ.'*Node') GOTO 576
    
```

Licencia Creative Commons:



```
110    CONTINUE

576    DO 345 I=1,900

123        READ (1,123) A
          FORMAT (A1)
          IF (A.EQ.***) THEN
              NPONT=I-1
              GOTO 112
          ENDIF

345    CONTINUE

112    DO 139 I=1,30

878        READ(1,878)A
          FORMAT (A8)
          IF (A.EQ.**ELEMENT') GOTO 333
          IF (A.EQ.**Element') GOTO 333

139    CONTINUE

333    DO 237 I=1,900

120        READ (1,120) A
          FORMAT (A1)
          IF (A.EQ.***) THEN
              NBLOC=I-1
              GOTO 212
          ENDIF

237    CONTINUE

212    CLOSE (UNIT=1)

      OPEN(UNIT=1,FILE="plus.inp",ACCESS='SEQUENTIAL',STATUS='OLD')

      DO 323 J=1,30
          READ(1,701)M
          IF (M.EQ.**NODE') GOTO 500
          IF (M.EQ.**Node') GOTO 500

323    CONTINUE

500    J=0
      DO 11 J=1, NPONT
```



READ(1,*) F,CORDB(J,1),CORDB(J,2),CORDB(J,3)

11 CONTINUE

DO 239 I=1,30

READ(1,878)A

IF (A.EQ.'*ELEMENT') GOTO 565
IF (A.EQ.'*Element') GOTO 565

239 CONTINUE

565 DO 313 JELEM=1, NBLOC

READ (1,*) F,NOEL1(JELEM),NOEL2(JELEM),
1
NOEL3(JELEM),NOEL4(JELEM),NOEL5(JELEM),
2 NOEL6(JELEM),NOE7(JELEM),NOEL8(JELEM)

313 CONTINUE

CLOSE (UNIT=1)

OPEN(1,FILE="plus.gen",ACCESS='SEQUENTIAL',STATUS='NEW')

C=0

WRITE(1,*)
Archivo plus.gen'
WRITE(1,*)

WRITE(1,*)
Nodos' WRITE(1,*) Primera linea:Nelem,Nnodos,CoordSAuxS'
Coord' WRITE(1,*) Primer grupo:Elem,NCood,8
WRITE(1,*) Segundo grupo: SisRef, Nodo,3

WRITE(1,102)NBLOC,NPONT,C,C

102 FORMAT (2X,I3,',',2X,I3,',',X,I2,',',X,I2)

C=1

DO 319 JELEM=1, NBLOC

WRITE (1,199)JELEM,C,NOEL1(JELEM),NOEL2(JELEM),
1
NOEL3(JELEM),NOEL4(JELEM),NOEL5(JELEM),
2 NOEL6(JELEM),NOE7(JELEM),NOEL8(JELEM)

199 FORMAT (1X,I4,',',2X,I2,',',7(X,I4,','),2X,I4)

Licencia Creative Commons:



319 CONTINUE

C=0
J=0

DO 99 J=1, NPONT

WRITE(1,101) C,J,CORDB(J,1),CORDB(J,2),CORDB(J,3)

101 FORMAT (1X,I4,';',X,I4,';',X,F7.2,';',X,F8.2,';',X,F8.2)

99 CONTINUE

CLOSE(UNIT=1)

NABLA=0.3
ELASTIC=1

RETURN

END

WWW.BDIGITAL.ULA.VE

C
C
C
C
C
C
C
C
C
C
C

Subrutina GENSEN

Esta subrutina lee el archivo plus.dat y genera el archivo plus.sen

SUBROUTINE GENSEN (NBLOC,NPONT)

DIMENSION ELSEN(900),NODSEN(900,10),SENER(900,10),ELSEN2(900)

INTEGER I,J,NELEM,NUMEL,AA,K,F,H

CHARACTER*20 S,A

C
C

OPEN(UNIT=1,FILE="plus.dat",ACCESS='SEQUENTIAL',STATUS='OLD')

DO 110 I=1,5000

701 READ(1,701)S
FORMAT (25X,A7)

IF (S.EQ.'SENER') GOTO 555



```
C
C
C *****
C Subrutina ADAPTIVE

SUBROUTINE ADAPTIVE (NBLOC,MPONT,MPOLA,MBLOC,MDIVI,SED,
1      CORDB,LNODB,NDIVX,NDIVY,NDIVZ,WEITX,WEITY,
2      WEITZ,SEDG,BETA,ORIGB,MATNB,PILAS)

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)

COMMON/MODEL/NNODE,LNODE,NDIME

REAL SEDG,BETA

INTEGER JBLOC,PILAS,F1,F2

CHARACTER*74 A

      BETA=1

      PILAS=1

      OPEN (UNIT=1,FILE='plus.gen',ACCESS='SEQUENTIAL',STATUS='OLD')
156      READ (1,156)A
          FORMAT (I,I,I,I,I,A1)

          READ (1,*)NBLOC,NPONT,NPOLA,NRENM

      CALL CHECK (NBLOC,NPONT,NPOLA,MBLOC,MPONT,MPOLA,ISTOP)
      IF (ISTOP.GT.0) STOP

      DO 777 I=1, NBLOC

          1      READ (1,*) F1,F2,LNODB(I,1),LNODB(I,2),LNODB(I,3),
2          LNODB(I,4),LNODB(I,5),LNODB(I,6),LNODB(I,7),
          LNODB(I,8)

777      CONTINUE

          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
```

WWW.BDIGITAL.ULA.VE



```

DO 64 IPONT=1, NPONT
DO 64 IDIME=1, NDIME
CORDB (IPONT, IDIME)=0.0
64 CONTINUE

I=0
DO 167 I=1, NPONT

READ (1, *) F1, F2, CORDB(I, 1), CORDB(I, 2), CORDB(I, 3)

167 CONTINUE

CLOSE (UNIT=1)

CALL MDSIDE (CORDB, LNODB, MBLOC, MPONT, NBLOC, NPONT)

OPEN (UNIT=2, FILE="plus.sen", ACCESS='SEQUENTIAL', STATUS='OLD')
OPEN (UNIT=1, FILE="plus.sed", ACCESS='SEQUENTIAL', STATUS='NEW')

WRITE(1, *)
  Archivo plus.sed'
WRITE(1, *)
  *****

WRITE(1, *)
  elemento'
  Contiene los valores de SED, SEDG'
  Numero de divisiones por
  Indicador de refinamiento BETA'
  *****

WRITE(1, *) 'Valores de SED por cada elemento'
416 READ (2, 416) A
  FORMAT (I, I, I, I, I, A1)

DO 3 I=1, NBLOC
  COUNT=0.
  DO 4 INODE=1, 8
  READ (2, *) A, B, SI
  COUNT=COUNT+SI
  4 CONTINUE
  SED(I)=COUNT

  WRITE(1, 4001) 'SED(' , I, ')', SED(I)
4001 FORMAT(3X, A, I3, A, F12.4)
3 CONTINUE

CLOSE (UNIT=2)

SEDG=0.

DO 5 I=1, NBLOC
  SEDG=SEDG+SED(I)
5 CONTINUE

WRITE(1, *) 'Valor de SEDG total: ', SEDG
4005 FORMAT(3X, A, F12.4)

SEDG=SEDG/NBLOC
    
```

WWW.BDIGITAL.ULA.VE



```
C
C
C *****
C   Subrutina CHECK

C
C   SUBROUTINE CHECK (NBLOC, NPONT, NPOLA, MBLOC, MPONT, MPOLA, ISTOP)
C
C   COMMON/MODEL/NNODE, LNODE, NDIME

      ISTOP=0

      IF (NBLOC.GT.MBLOC) ISTOP=1
      IF (NPONT.GT.MPONT) ISTOP=2
      IF (NPOLA.GT.MPOLA) ISTOP=3
      IF (NBLOC*LNODE.LT.NPONT) ISTOP=4
      IF (ISTOP.EQ.0)RETURN

1000 FORMAT (///,3X,'ERROR',I2,'WAS FOUND',
1          /,3X,'COMPUTATIONS WILL BE STOPPED')

      RETURN

      END
```

```
C
C
C *****
C
C *****
C   Subrutina DIMEN

      SUBROUTINE DIMEN (MBLOC, MPONT, MDIVI, MREPN, MPOLA, MELEM, MPOIN)
      COMMON/MODEL/NNODE, LNODE, NDIME

      MBLOC=900

      MPONT=MBLOC*LNODE

      MDIVI=30

      MREPN=300

      MPOLA=10

      MELEM=600

      MPOIN=MELEM*NNODE

      RETURN

      END
```

```
C
C
C *****
C
C *****
C   Subrutina GENMSH
```

Licencia Creative Commons:



```

C
SUBROUTINE GENMSH (NBLOC,NELEM,NPOIN,CORDB,MATNB,LNODEB,NDIVX,
1     NDIVY,NDIVZ,WEITX,WEITY,WEITZ,LREPN,LASOC,
2     LFINN,LFASC,COORD,MATNO,LNODS,MBLOC,MPONT,
3     MDIVI,MREPN,MELEM,MPOIN,NL,ICON,NRENM,ORIGN,
4     IPOL,NNCNEC,CBILI,CODASIS,CODASII,CODASDS,
5     CODASDI,NODOIS,NODOII,NODODI,NODODS,IPOLI,
6     NODII,NODIS,PLIL,CALII,CALID,IG)
C
COMMON/MODEL/NNODE,LNODE,NDIME

DIMENSION CORDB(MPONT,NDIME),MATNB(MBLOC),LNODB(MBLOC,LNODE),
1     NDIVX(MBLOC),NDIVY(MBLOC),NDIVZ(MBLOC),WEITX(MBLOC,
2     MDIVI),WEITY(MBLOC,MDIVI),WEITZ(MBLOC,MDIVI),SHAPE(20)
3     ,LREPN(MREPN),LASOC(MREPN),LFINN(MREPN),LFASC(MREPN),
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

NPOIN=0
NELEM=0

DO 1 IREPN=1,MREPN
    LREPN(IREPN)=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

DO 4 IBLOC=1,NBLOC
    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

```

Licencia Creative Commons:



```
TOTAL=TOTAL+WEITZ(IBLOC, IDIVZ)
7  CONTINUE

4002 FORMAT(3X,A,F12.4)

      ZNORM=2.0/TOTAL

      NXNNP=NDIVX(IBLOC)+1
      NYNNP=NDIVY(IBLOC)+1
      NZNNP=NDIVZ(IBLOC)+1

      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
          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)+
11      1  SHAPE(INODE)*CORDB(JTEMP, IDIME)
              CONTINUE

      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

      KWETX=KWETX+1
```

Licencia Creative Commons:

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



```
10 EXISP=EXISP+XNORM*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

NREPN=0
DO 13 IPOIN=1,NPOIN
    IF (NREPN.EQ.0)GOTO 14

    DO 15 IREPN=1,NREPN
        IF (IPOIN.EQ.LREPN(IREPN)) GOTO 13
15    CONTINUE
14 CONTINUE

LPOIN=IPOIN+1

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
    NREPN=NREPN+1

LREPN(NREPN)=JPONT
LASOC(NREPN)=IPOIN
16 CONTINUE
13 CONTINUE

IF (NREPN.EQ.0) GOTO 35

INDEX=0
DO 17 IPOIN=1,NPOIN

    DO 18 IREPN=1,NREPN
        IF (LREPN(IREPN).EQ.IPOIN) GOTO 19
18    CONTINUE

    GOTO 17

19    INDEX=INDEX+1
    LFINN(INDEX)=LREPN(IREPN)
    LFASC(INDEX)=LASOC(IREPN)
17 CONTINUE

DO 20 IREPN=1,NREPN
    LREPN(IREPN)=LFINN(IREPN)
    LASOC(IREPN)=LFASC(IREPN)
20 CONTINUE

DO 21 IREPN=1,NREPN
DO 21 IELEM=1,NELEM
DO 21 INODE=1,NNODE
    IF (LNODS(IELEM,INODE).EQ.LREPN(IREPN))
1    LNODS(IELEM,INODE)=LASOC(IREPN)
21 CONTINUE
```

WWW.BDIGITAL.ULA.VE



```
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
      NPOS=LNODS(IELEM,INODE)
      DO 28 IREP=1,NREP
      IF (NPOS.EQ.LREP(IREP)) GOTO 27
28    CONTINUE
      IF (NPOS.LT.LREP(1)) GOTO 27
      IDIFF=NPOS-NREP
      IF (NPOS.GT.LREP(NREP)) GOTO 29
      DO 30 IREP=1,NREP
      KREP=NREP-IREP+1
      IF (NPOS.LT.LREP(KREP)) IDIFF=NPOS-KREP+1
30    CONTINUE
29    LNODS(IELEM,INODE)=IDIFF
27    CONTINUE
      NPOIN=NPOIN-NREP
35 DO 521 I=1,IG
      DO 531 IPOIN=1,NPOIN
      NL=0
      KPOIN=IPOIN
      IF (NREN.EQ.1.AND.ORIG.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
```

Licencia Creative Commons:



```
ENDIF
NL=0.
IF (CALID(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CALID(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CALID(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    NODIS(I)=KPOIN
ENDIF
```

531 CONTINUE

521 CONTINUE

```
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(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CBILI(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CBILI(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    IPOL(I)=KPOIN
ENDIF
NL=0.
IF (CODASII(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASII(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASII(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    NODOII(I)=KPOIN
ENDIF
NL=0.
IF (CODASIS(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASIS(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASIS(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    NODOIS(I)=KPOIN
ENDIF
NL=0.
IF (CODASDI(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASDI(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASDI(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    NODODI(I)=KPOIN
ENDIF
NL=0.
IF (CODASDS(1,1).EQ.COORD(KPOIN,1)) NL=NL+1
IF (CODASDS(1,2).EQ.COORD(KPOIN,2)) NL=NL+1
IF (CODASDS(1,3).EQ.COORD(KPOIN,3)) NL=NL+1
IF (NL.EQ.3)THEN
    NODODS(I)=KPOIN
ENDIF
53 CONTINUE
52 CONTINUE
```

```
100 FORMAT(2X,A,13,A,15)
130 FORMAT(1X,A,13)
140 FORMAT(//,2X,A)
```

RETURN



```

END
C
C
C *****
C
C *****
C
C Subrutina INCOMP
C
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)

COMMON/MODEL/NNODE,LNODE,NDIME

DIMENSION NDIVX(MBLOC),CORDB(MPONT,NDIME),POBIL(MPONT,NDIME),
1      LNODE(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,CORDB,NCOR,NNCOR,CODAS,CODASII,CODASIS,
2      CODASDI,CODASDS

INTEGER NM,ICON,INCON,T,LNODE,KAY

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

IM=0

DO 22 INODE=1,LNODE
    DO 23 IDIME=1,NDIME
        NCOR(INODE,IDIME)=0.0
        NNCOR (INODE,IDIME)=0.0
    
```

Licencia Creative Commons:



23 CONTINUE
22 CONTINUE

C Inicio del gran lazo

```
DO 1 IBLOC=1,NBLOC
  JBLOC=IBLOC
  NM=NDIVX(JBLOC)
```

IF (NM.EQ.1) GOTO 31

```
DO 8 INODE=1,8
```

```
  KAY=0
  KAY=LNODB(IBLOC,INODE)
```

```
  NCOR(INODE,1)=CORDB(KAY,1)
  NCOR(INODE,2)=CORDB(KAY,2)
  NCOR(INODE,3)=CORDB(KAY,3)
```

8 CONTINUE

C COORDENADA NODO CENTROIDAL CARA UNO

TOTX=0.
TOTY=0.
TOTZ=0.

```
DO 99 I=1,4
```

```
  TOTX=TOTX+NCOR(I,1)
  TOTY=TOTY+NCOR(I,2)
  TOTZ=TOTZ+NCOR(I,3)
```

99 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

WWW.BDIGITAL.ULA.VE



```
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
```

```
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
```

WWW.BDIGITAL.ULA.VE



```
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
```

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 80 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)
80 CONTINUE

GOTO 1

C  Calculo de las coordenadas delos nodos en el centro de cada cara
C  solo en los elementos que no seran divididos

31 DO 11 INODE=1,8

    NM=LNODB(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
```

Licencia Creative Commons:



```
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.
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
```

WWW.DIGITAL.ULA.VE



```
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)

C  NODO ASOCIADO UNO

                DO 90 IDIME=1,NDIME
                    CODASII(IM,IDIME)=CODAS(I,1,IDIME)
90                CONTINUE
                DO 91 IDIME=1,NDIME
                    CODASIS(IM,IDIME)=CODAS(I,2,IDIME)
91                CONTINUE
                DO 92 IDIME=1,NDIME
                    CODASDI(IM,IDIME)=CODAS(I,3,IDIME)
92                CONTINUE
                DO 93 IDIME=1,NDIME
                    CODASDS(IM,IDIME)=CODAS(I,4,IDIME)
93                CONTINUE

                ENDIF

51                CONTINUE

113 CONTINUE

    ICON=IM

    DO 25 I=1,ICON

                DO 26 IDIME=1,NDIME
26                CONTINUE

25 CONTINUE

    DO 88 I=1,ICON
        DO 89 IDIME=1,NDIME
89            CONTINUE
88 CONTINUE

100 FORMAT(2X,A,I3,A,F12.4)
```



C Busqueda de las coordenadas de los nodos lineales

C LINEAL DE BORDE INFERIOR

DO 53 I=1,IM

```
IF ((CODASDS(I,2)-CODASII(I,2)).EQ.0.)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 53
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)
```

53 CONTINUE

C LINEAL DE BORDE DERECHO

DO 54 I=1,IM

```
IF ((CODASIS(I,1)-CODASII(I,1)).EQ.0.)THEN
  COLMBR(I,1)=(CODASIS(I,1)+CODASII(I,1))/2.
  COLMBR(I,2)=(CODASIS(I,2)+CODASII(I,2))/2.
  COLMBR(I,3)=(CODASIS(I,3)+CODASII(I,3))/2.
  COBRD(I,1)=CODASIS(I,1)
  COBRD(I,2)=CODASIS(I,2)
  COBRD(I,3)=CODASIS(I,3)
  COBRU(I,1)=CODASII(I,1)
  COBRU(I,2)=CODASII(I,2)
  COBRU(I,3)=CODASII(I,3)
  GOTO 54
ENDIF
```

```
COLMBR(I,1)=(CODASDI(I,1)+CODASIS(I,1))/2.
COLMBR(I,2)=(CODASDI(I,2)+CODASIS(I,2))/2.
COLMBR(I,3)=(CODASDI(I,3)+CODASIS(I,3))/2.
COBRD(I,1)=CODASDI(I,1)
COBRD(I,2)=CODASDI(I,2)
COBRD(I,3)=CODASDI(I,3)
COBRU(I,1)=CODASIS(I,1)
COBRU(I,2)=CODASIS(I,2)
COBRU(I,3)=CODASIS(I,3)
```

54 CONTINUE

C LINEAL DE BORDE SUPERIOR

DO 55 I=1,IM

```
IF ((CODASDI(I,2)-CODASIS(I,2)).EQ.0.)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)
```

Licencia Creative Commons:



```
COBUL(1,2)=CODASDI(1,2)
COBUL(1,3)=CODASDI(1,3)
COBUR(1,1)=CODASIS(1,1)
COBUR(1,2)=CODASIS(1,2)
COBUR(1,3)=CODASIS(1,3)
GOTO 55

ENDIF
COLMBU(1,1)=(CODASDI(1,1)+CODASDS(1,1))/2.
COLMBU(1,2)=(CODASDI(1,2)+CODASDS(1,2))/2.
COLMBU(1,3)=(CODASDI(1,3)+CODASDS(1,3))/2.
COBUL(1,1)=CODASDI(1,1)
COBUL(1,2)=CODASDI(1,2)
COBUL(1,3)=CODASDI(1,3)
COBUR(1,1)=CODASDS(1,1)
COBUR(1,2)=CODASDS(1,2)
COBUR(1,3)=CODASDS(1,3)

55 CONTINUE

C LINEAL DE BORDE IZQUIERDO
DO 56 I=1,IM
IF ((CODASDI(1,1)-CODASDS(1,1)).EQ.0.)THEN
COLMBL(1,1)=(CODASDS(1,1)+CODASDI(1,1))/2.
COLMBL(1,2)=(CODASDS(1,2)+CODASDI(1,2))/2.
COLMBL(1,3)=(CODASDS(1,3)+CODASDI(1,3))/2.
COBLU(1,1)=CODASDS(1,1)
COBLU(1,2)=CODASDS(1,2)
COBLU(1,3)=CODASDS(1,3)
COBLD(1,1)=CODASDI(1,1)
COBLD(1,2)=CODASDI(1,2)
COBLD(1,3)=CODASDI(1,3)
GOTO 56
ENDIF
COLMBL(1,1)=(CODASII(1,1)+CODASDS(1,1))/2.
COLMBL(1,2)=(CODASII(1,2)+CODASDS(1,2))/2.
COLMBL(1,3)=(CODASII(1,3)+CODASDS(1,3))/2.
COBLU(1,1)=CODASII(1,1)
COBLU(1,2)=CODASII(1,2)
COBLU(1,3)=CODASII(1,3)
COBLD(1,1)=CODASDS(1,1)
COBLD(1,2)=CODASDS(1,2)
COBLD(1,3)=CODASDS(1,3)

56 CONTINUE
C -----
C BORDE INFERIOR
C -----
DO 342 K=1,MPONT
DO 343 J=1,NDIME
PLIL(K,J)=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
```

Licencia Creative Commons:

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



DO 46 I=1,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)
      CALI(IG,1)=COBIL(I,1)
      CALI(IG,2)=COBIL(I,2)
      CALI(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 -----
```

WWW.BDIGITAL.ULA.VE

DO 47 I=1,IM

```
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)
      CALI(IG,1)=COBRD(I,1)
      CALI(IG,2)=COBRD(I,2)
      CALI(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

DO 243 I=1,IG
243 CONTINUE

RETURN

END

C
```

Licencia Creative Commons:

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



```

C *****
C
C
C
C
C *****
C Subrutina MDSIDE
C
C SUBROUTINE MDSIDE (CORDB, LNODB, MBLOC, MPONT, NBLOC, NPONT)
C
C COMMON/MODEL/NNODE, LNODE, NDIME
C
C DIMENSION CORDB(MPONT, NDIME), LNODB(MBLOC, LNODE)
C
C INTEGER LNODB, NODMD
C
C NODMD=0
C
C NN=NPONT
C
C DO 1 IBLOC=1, NBLOC
C
C REALIZA LA INTERPOLACIÓN EN LA CARA MAS BAJA DEL SUPERELEMENTO
C
C DO 2 INODE=1, 4
C
C CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
C NODST=LNODB(IBLOC, INODE)
C IGASH=INODE+1
C IF (IGASH.GT.4) IGASH=1
C
C CALCULA EL NUMERO DE NODO DEL ULTIMO NODO
C
C NODFN=LNODB(IBLOC, IGASH)
C MIDPT=INODE+NNODE
C
C CALCULA EL NUMERO DEL NODO DE LOS NODOS CENTRALES
C
C NODMD=LNODB(IBLOC, MIDPT)
C
C IF (NODMD.EQ.0) THEN
C     NN=NN+1
C     NODMD=NN
C     LNODB(IBLOC, MIDPT)=NODMD
C
C ENDIF
C
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 2
C
C DO 3 IDIME=1, NDIME
C     CORDB(NODMD, IDIME)=(CORDB(NODST, IDIME)+CORDB(NODFN, IDIME))/2.
C 3 CONTINUE
C
C 2 CONTINUE
C
C REALIZA LA INTERPOLACIÓN EN LA CARA SUPERIOR DEL SUPERELEMENTO
C
C DO 4 INODE=5, 8
C
C CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
C NODST=LNODB(IBLOC, INODE)
C IGASH=INODE+1
C IF (IGASH.GT.8) IGASH=5
    
```

Licencia Creative Commons:



```
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.0.0) 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
DO 6 INODE=1,4
C
C  CALCULA EL NUMERO DE NODO DEL PRIMER NODO
C
NODST=LNODB(IBLOC,INODE)
IGASH=INODE+4
C
C  CALCULA EL NUMERO DE NODO DEL ULTIMO NODO
C
NODFN=LNODB(IBLOC,IGASH)
MIDPT=INODE+12
C
C  CALCULA EL NUMERO DEL NODO DE LOS NODOS CENTRALES
C
NODMD=LNODB(IBLOC,MIDPT)
IF (NODMD.EQ.0.0) 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 6
DO 7 IDIME=1,NDIME
CORDB(NODMD,IDIME)=(CORDB(NODST,IDIME)+CORDB(NODFN,IDIME))/2.
7  CONTINUE
6  CONTINUE
1  CONTINUE
RETURN
END

C
C *****
C
C
C
C
C
C *****
```

Licencia Creative Commons:



```
C      Subrutina OPTNM
C
SUBROUTINE OPTNUM (NPOIN,MPOIN,IDIFF,ORIGN,MAXCOM,NNCNEC,
1      PONTRS,JNODC)
C
COMMON/MODEL/NNODE,LNODE,NDIME

DIMENSION NNCNEC(MPOIN),JNODC(4800*10),NEWNOD(4800)

INTEGER PONTRS(MPOIN),OLDNOD(4800)

INTEGER ORIGN,PJNOD,RNODE

ORIGN=1
MINDIF=IDIFF
C

DO 1 IPOIN=1,NPOIN
DO 2 JPOIN=1,NPOIN
      OLDNOD(JPOIN)=0
      NEWNOD(JPOIN)=0
2  CONTINUE

      JDIFF=0
      KPOIN=1
      NEWNOD(1)=IPOIN
      OLDNOD(IPOIN)=1
      LPOIN=1
3  NTNOD=NNCNEC(NEWNOD(KPOIN))

      IF (NTNOD.EQ.0) GOTO 4

      PJNOD=(NEWNOD(KPOIN)-1)*MAXCOM

      DO 5 ITNOD=1,NTNOD
      RNODE=JNODC(PJNOD+ITNOD)
      IF (OLDNOD(RNODE).GT.0) GOTO 5
      LPOIN=LPOIN+1
      NEWNOD(LPOIN)=RNODE
      OLDNOD(RNODE)=LPOIN
      NDIFF=IABS(KPOIN-LPOIN)
      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
```

WWW.BT.DIGITAL.ULA.VE

Licencia Creative Commons:

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



```

9      CONTINUE
C
      RETURN

      END

C
C *****
C
C
C
C
C *****
C      Subrutina OUTPUT
C
      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)
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)
      INTEGER ORIGN

      OPEN (6,FILE='plus.inp',ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      OPEN (3,FILE='plud.inp',ACCESS='SEQUENTIAL',STATUS='UNKNOWN')

      WRITE(6,1010)
      WRITE(3,1010)
      WRITE(6,1002)
      WRITE(3,1002)

      NPOLA=0.
      NRENM=0.
      MAT=1

      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

          GOTO 12
13          DO 15 INODE=1,NNODE
              IL(INODE)=LNODS(KELEM,INODE)
15          CONTINUE

12 CONTINUE

      JPOLA=0.
      DO 1 IPOIN=1,NPOIN
          KPOIN=IPOIN
          IF (NRENM.EQ.1.AND.ORIGN.EQ.1) KPOIN=NNCNEC(IPOIN)
          WRITE(6,1003) IPOIN,(COORD(KPOIN,IDIME),IDIME=1,NDIME)
          WRITE(3,1003) IPOIN,(COORD(KPOIN,IDIME),IDIME=1,NDIME)
1 CONTINUE
C
C SALIDA DE LA TOPOLOGIA DEL ELEMENTO
C
      WRITE(6,1004)
    
```

WWW.DIGITAL.ULA.VE



```
WRITE(3,1004)

DO 2 IELEM=1,NELEM
  KELEM=IELEM
  IF (NRENM.EQ.0.OR.ORIG.NEQ.0) GOTO 3
  DO 4 INODE=1,NNODE
  IL(INODE)=PONTRS(LNODS(KELEM,INODE))
4 CONTINUE

WRITE(6,1005) KELEM,
1 (IL(INODE),INODE=1,NNODE)
WRITE(3,1005) KELEM,
1 (IL(INODE),INODE=1,NNODE)

GOTO 2

3 DO 5 INODE=1,NNODE
  IL(INODE)=LNODS(KELEM,INODE)
5 CONTINUE

WRITE(6,1005) KELEM,
1 (IL(INODE),INODE=1,NNODE)
WRITE(3,1005) KELEM,
1 (IL(INODE),INODE=1,NNODE)

2 CONTINUE
WRITE(6,1006)
WRITE(6,1007)
WRITE(6,1008)
WRITE(6,1015)
WRITE(6,1012)
WRITE(3,1006)
WRITE(3,1007)
WRITE(3,1008)
WRITE(3,1015)
WRITE(3,1012)

IF ((IG.NE.0).OR.(ICON.NE.0))THEN
WRITE(6,1017)
WRITE(6,1017)

ENDIF
IF (IG.NE.0)THEN
DO 45 I=1,IG
WRITE(6,1018)IPOLI(I),NODII(I),NODIS(I)
WRITE(3,1018)IPOLI(I),NODII(I),NODIS(I)
45 CONTINUE
ENDIF
IF (ICON.NE.0)THEN
DO 46 I=1,ICON

WRITE(6,1019)IPOL(I),NODOII(I),NODOIS(I),NODODI(I),NODODS(I)
WRITE(3,1019)IPOL(I),NODOII(I),NODOIS(I),NODODI(I),NODODS(I)

46 CONTINUE
ENDIF

WRITE(6,1016)
WRITE(6,1014)
WRITE(3,1016)
WRITE(3,1014)

RETURN
```

200 FORMAT(5X,A,\$,1X)



```

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 VECINOS
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=NBNGH(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
    
```

Licencia Creative Commons:



```

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      ESTA SUBROUTINA LOCALIZA LOS NODOS LINEALES INCOMPATIBLES
C      TANTO EN ELEMENTOS NO REFINADOS COMO EN REFINADOS
C      *****
IML=0.
LDCI=LIRCD(1)
IRCI=LIRCI(1)
DO 63 I=1,IRCI,3
DO 64 J=1,LDCI,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
IML=IML+1
100 FORMAT(2X,A,I3)
CLIN(IML,1)=CARAI(IDI,1)
CLIN(IML,2)=CARAI(IDI,2)
CLIN(IML,3)=CARAI(IDI,3)
IMA=IDI-1
IME=IDI-2
LAS01(IML,1)=CARAI(IMA,1)
LAS01(IML,2)=CARAI(IMA,2)
LAS01(IML,3)=CARAI(IMA,3)
LAS02(IML,1)=CARAI(IME,1)
LAS02(IML,2)=CARAI(IME,2)
LAS02(IML,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)
LDCS=LIRCD(2)
IMS=0
DO 72 L=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
IML=IML+1
CLIN(IML,1)=CARAS(IDI,1)
CLIN(IML,2)=CARAS(IDI,2)
CLIN(IML,3)=CARAS(IDI,3)
IMA=IDI-1
IME=IDI-2
LAS01(IML,1)=CARAS(IMA,1)
LAS01(IML,2)=CARAS(IMA,2)
LAS01(IML,3)=CARAS(IMA,3)
LAS02(IML,1)=CARAS(IME,1)
LAS02(IML,2)=CARAS(IME,2)
LAS02(IML,3)=CARAS(IME,3)
ENDIF
74 CONTINUE
73 CONTINUE
72 CONTINUE
    
```

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

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



```

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 (NRENM.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 DO 85 I=1,IMI
105 FORMAT(2X,A,13,A,15)
130 FORMAT(1X,A,13)
140 FORMAT(/,2X,A)
RETURN
END
    
```

Licencia Creative Commons:

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