

CUPEX, PROGRAMA DE ELEMENTOS FINITOS P-ADAPTATIVOS PARA CALCULOS ESTRUCTURALES CON MODULOS EXPERTOS DE AYUDA

MIGUEL GALANTE

y

JAVIER CASTRO

*E.T.S. Ingenieros Industriales
Tafira Baja s/n
35017 Las Palmas de Gran Canaria
España*

RESUMEN

Se ha creado un código para el cálculo de estructuras elásticas bidimensionales con las características necesarias para abarcar el mayor número de usuarios posibles. Para ello, se ha desarrollado en un micro computador, se ha elegido un lenguaje interactivo y de gran capacidad gráfica, y se le ha implementado la experiencia existente en este campo.

En este artículo se definen los conceptos básicos empleados, se expone la configuración general del código, se comentan las bases conceptuales de los algoritmos expertos necesarios para crear una malla ingenieril y de los necesarios para pronosticar la precisión de la solución antes de su cálculo.

Se justifica el empleo de elementos finitos jerarquizados p por su alta velocidad de convergencia, de indicadores de error por su bajo costo computacional en refinamientos p , se muestra la estrategia adaptativa utilizada en cada etapa de refinamiento y el control que se da al usuario del proceso de análisis a través del conocimiento de la velocidad de convergencia y grado de selectividad.

SUMMARY

The goal has been to make a code to work out bidimensional elastic structures which can be used by a broad field of user. In order to get it, the code has been used. Also, the human experience in this area of the engineering has been implemented.

The paper states the basic concepts, the code and the expert modules necessary to make an engineering mesh and to forecast the error in the solution.

p -Hierarchic finite element has been used because of its high rate of convergence, error indicators because of their low computational cost and their big help to select the adaptative strategy. Also is displayed the necessary graphics to give the user the process control.

Recibido: Febrero 1989

INTRODUCCION

El desarrollo espectacular de los ordenadores ha hecho posible obtener de las ecuaciones diferenciales de gobierno de problemas reales una respuesta que antes de la aparición de estas potentes máquinas de calcular era imposible en la mayoría de los casos.

Como es conocido, el paso para su conversión en un método numérico implementable en un computador se consigue a través de un proceso por diferencias finitas o por métodos que transforman las ecuaciones diferenciales en ecuaciones integrales. Uno de estos métodos, es el de los Elementos Finitos (EF), que hoy en día ha adquirido un gran desarrollo y popularidad y es el que se va a utilizar como base de análisis.

El problema de éste y el de todos estos métodos numéricos, como ya dijo Courant¹ es la carencia de poder estimar la precisión de la respuesta que suministran.

Hasta ahora la garantía de una solución razonablemente precisa en EF se ha basado en la experiencia e intuición del usuario, que elabora varias mallas de la forma que considera más adecuada, obtiene las soluciones correspondientes y mediante una extrapolación de Richardson² intenta acercarse a la solución "verdadera". Esta forma de proceder requiere bastante mano de obra, encareciendo el costo del problema. La entrada de datos en un código de elementos finitos es siempre una labor pesada que puede llegar a ocupar el 90 % del tiempo total empleado en la resolución. Parece pues necesario, buscar otra vía para solucionar el inconveniente de los métodos numéricos.

El primer paso dado en este sentido, dentro del cálculo estructural, ha sido obtener una cota superior del error a posteriori de la solución, basada en la norma energética del error³. Esto ha representado un avance importante en cuanto al control de la precisión de la respuesta y es un módulo que en un futuro no lejano veremos en los códigos comerciales de cálculos estructurales por EF.

El siguiente paso y el definitivo ha sido la llegada de los sistemas expertos. Si se implementa en un código de análisis, módulos que almacenan la experiencia humana, se puede reducir notablemente el costo de ejecución. En el cálculo estructural los objetivos de estos módulos son el aconsejar para el diseño de la malla en función de la precisión deseada y vigilar durante el proceso de análisis de esa precisión, deteniendo el cálculo y aconsejando qué hacer en caso contrario.

En CUPEX se ha implementado el cálculo del estimador de error en la primera etapa del análisis junto con un módulo experto que predice el error final. Y de la experiencia existente en los problemas elastoestáticos bidimensionales⁴ se ha creado un módulo que aconseja y participa en el diseño de la malla.

El deseo de abarcar el mayor número de usuarios ha conducido al empleo de micro ordenadores. El tipo de refinamiento elegido es el proporcionado por los EF jerarquizados p auto-adaptativos, dada su mayor velocidad de convergencia⁵.

Con esta base de partida se ha desarrollado el programa CUPEX. El tipo de elemento geométrico empleado ha sido el cuadrilátero lineal con posibilidad de curvar los lados⁶ si el contorno del dominio en análisis lo requiere.

E.F. JERARQUIZADOS P CONFORMES

El concepto de jerarquizado en los polinomios (p), asocia el comportamiento analítico de la variable (u) dentro de cada elemento a un desarrollo similar a una serie de Fourier, es por ello que al incrementar las funciones de forma (N_i) no se alteran las existentes⁷, situación que no se da en la formulación clásica⁸:

$$u \simeq \hat{u} = \sum_1^n N_i u_i \quad (1)$$

$$u \simeq \hat{\hat{u}} = \sum_1^n N'_i u_i + N_{n+1} u_{n+1} \quad (2)$$

Formulación clásica : $N'_i \neq N_i$

Formulación jerarquizada : $N'_i = N_i$

En la Figura 1 se muestra para el elemento cuadrilátero, la diferencia entre ambas formas de expresar la variable.

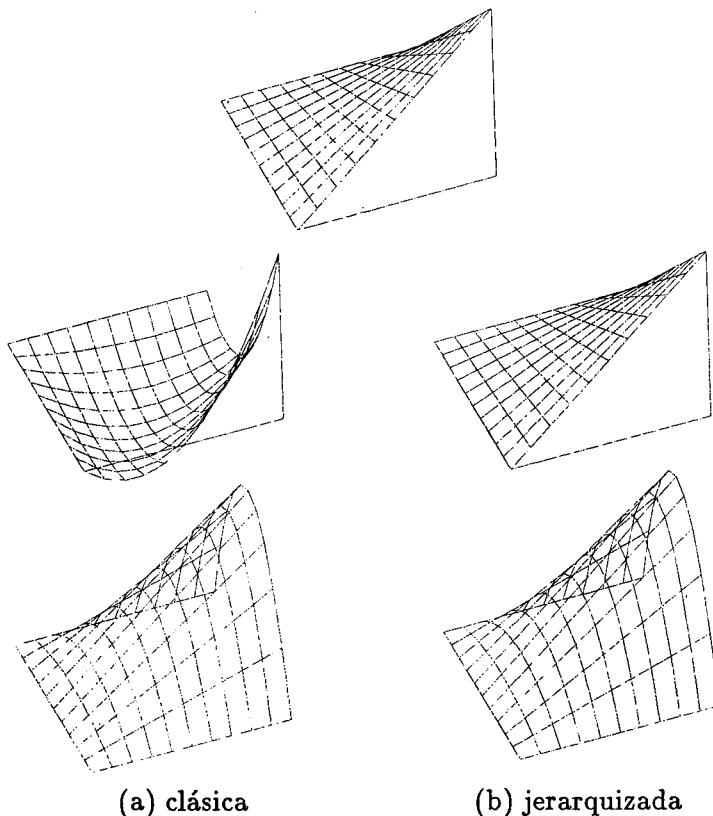


Figura 1. Formas clásica y jerarquizada para funciones de forma polinomiales cuadráticas.

La consecuencia inmediata de la jerarquización p es el mejor condicionamiento de la matriz de rigidez del sistema de ecuaciones lo que permite una resolución iterativa en cada etapa de refinamiento de la solución anterior y no necesitando mantener en memoria central nada más que el bloque de jerarquización que se está resolviendo. En CUPEX se aprovecha esta propiedad, habiéndose empleado como funciones de jerarquización los siguientes polinomios de Legendre por su cuasi ortogonalidad en el campo bidimensional

$$N_p(\xi) = \frac{1}{(p-1)!} \frac{1}{2^{p-1}} \frac{d^p}{d\xi^p} [(\xi^2 - 1)^p] \quad (3)$$

Estimadores de error

Como es sabido, el error en la obtención de una variable a través de cualquier método aproximado, es la diferencia entre la solución exacta y la aproximación obtenida:

$$\mathbf{e} = \mathbf{u} - \hat{\mathbf{u}} \quad (4)$$

\mathbf{e} : error

\mathbf{u} : solución exacta

$\hat{\mathbf{u}}$: solución aproximada

La imposibilidad de su determinación, ya que eso sería tanto como conocer la solución exacta, ha conducido a la búsqueda de medidas alternativas. La más utilizada actualmente, es la norma energética del error¹⁰.

Siendo las ecuaciones de gobierno de la variable $\mathbf{u}(\mathbf{x})$

$$\begin{aligned} \mathcal{L}\mathbf{u} + \mathbf{q} &= 0 & \forall \mathbf{x} \in \Omega & \subset \mathbb{R}^n \\ \mathcal{N}\mathbf{u} + \mathbf{s} &= 0 & \forall \mathbf{x} \in \partial\Omega_n & \subset \mathbb{R}^{n-1} \\ \mathbf{u} - \mathbf{u}_0 &= 0 & \forall \mathbf{x} \in \partial\Omega_d & \subset \mathbb{R}^{n-1} \end{aligned} \quad (5)$$

$\partial\Omega$ contorno del dominio Ω : $\partial\Omega = \partial\Omega_n \cup \partial\Omega_d$

\mathcal{L} y \mathcal{N} operadores diferenciales lineales

\mathbf{q} y \mathbf{s} vectores independientes

Se define la norma energética del error como

$$\|e\|_E^2 = \int_{\Omega} \mathbf{e}^T \mathcal{L}\mathbf{e} \, d\Omega + \int_{\partial\Omega} \mathbf{e}^T \mathcal{N}\mathbf{e} \, d\Gamma \quad (6)$$

que se puede expresar a través de los residuos que aparecen en las ecuaciones de gobierno al sustituir la variable por su valor aproximado:

$$\begin{aligned} \mathcal{L}\hat{\mathbf{u}} + \mathbf{q} + \mathbf{r} &= 0 \\ \mathcal{N}\hat{\mathbf{u}} + \mathbf{s} + \mathbf{r}' &= 0 \end{aligned} \quad (7)$$

r y r' fuerzas residuales

al restar (7) de (6) y sustituir en la expresión de la norma energética queda:

$$\|e\|_E^2 = \int_{\Omega} \mathbf{e}^T \mathbf{r} \, d\Omega + \int_{\partial\Omega_n} \mathbf{e}^T \mathbf{r}' \, d\Gamma \quad (8)$$

Sobre esta expresión Gago¹¹ obtiene una sobrestimación del error para aplicar a dominios bidimensionales elásticos discretizados en elementos finitos. Siendo la expresión para cada elemento Ω_e de frontera Γ_e :

$$\varepsilon_e^2 = \frac{h_e^2}{24 p^2 k} \int_{\Omega_e} r^2 \, d\Omega + \frac{h_e}{24 p k} \int_{\Gamma_e} \Theta \, d\Gamma \quad (9)$$

h_e longitud característica del elemento

r residuo en el dominio del elemento

Θ salto de tensiones en la frontera

$$K = \frac{E}{1-\mu}$$

E módulo de Young

μ módulo de Poisson

p grado mayor de los polinomios de las funciones de forma

Este ha sido el estimador de error implementado en CUPEX. Para facilitar su interpretación el código presenta un solo escalar como medidor de la precisión. Este valor representa en tanto por ciento, la raíz cuadrada del valor relativo del error estimado con relación a la energía interna:

$$\varepsilon = \sqrt{\frac{\sum \varepsilon_e^2}{\|U\|_E^2 + \sum \varepsilon_e^2}} \cdot 100 \% \quad (10)$$

La obtención del estimador de error es cara debido a los tiempos computacionales de cálculo de los residuos. En los ejemplos efectuados el tiempo de ordenador ha sido del orden y a veces mayor que el necesario para calcular la matriz de rigidez y el sistema de ecuaciones. Esto hace prohibitivo su cálculo en cada pasada de jerarquización. El criterio empleado para evitar esta dificultad ha sido calcular el estimador de error de acuerdo con la expresión (9) en la primera etapa ($p = 1$) y estimarlo mediante los incrementos de energía elástica que se produce en las etapas sucesivas; no obstante, se da la opción al usuario de su cálculo a través del estimador.

Estimadores de error

Miden la importancia relativa entre los grados de libertad a introducir en una nueva etapa de refinamiento (h o p), permitiendo así realizar una selección entre ellos.

El empleo de elementos finitos jerarquizados p , permite utilizar una expresión para evaluar el indicador, fácil de implementar y a muy bajo costo computacional al no estar relacionada con el cálculo de residuos.

Si se introduce un nuevo grado de libertad (u_{n+1}):

$$\hat{u}_e = \sum_1^n N_i u_i + N_{n+1} u_{n+1} \quad (11)$$

y se hace la hipótesis de no alteración en los grados de libertad existente. Se define como indicador el incremento en la norma energética del error:

$$\eta_{n+1}^2 = u_{n+1} \left(\int_{\Omega} N_{n+1} r \, d\Omega + \int_{\partial\Omega_n} N_{n+1} r' \, d\Gamma \right) \quad (12)$$

El sistema de ecuaciones que resuelve el problema numérico, queda por la introducción del nuevo grado de libertad de la forma

$$\begin{bmatrix} K_{n,n} & \vdots & K_{n,n+1} \\ \dots & \vdots & \dots \\ K_{n+1,n} & \vdots & K_{n+1,n+1} \end{bmatrix} \begin{bmatrix} u_n \\ \dots \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} f_n \\ \dots \\ f_{n+1} \end{bmatrix} \quad (13)$$

Despejando u_{n+1} de (13) y teniendo en cuenta que en (12) las expresiones integrales representan la aplicación del método de Galerkin con la función de ponderación N_{n+1} , se obtiene para el indicador:

$$\eta_{n+1} = \sqrt{\frac{(f_{n+1} - K_{n+1,n} u_n)^2}{k_{n+1,n+1}}} \quad (14)$$

Esta expresión fue presentada por Peano¹² y es la que se ha implementado en CUPEX.

ALGORITMOS EXPERTOS

Los sistemas expertos constituyen una importante área en el campo de la inteligencia artificial. Se han desarrollado grandes sistemas expertos en el campo de la diagnosis médica (MYCIN, INTERNIST), de las exploraciones minereológicas (PROSPECTOR), en el diseño de configuraciones de sistemas de ordenadores (R1), etc. Recientemente está llegando al campo de la mecánica estructural donde existe un creciente interés en las posibilidades que las técnicas de sistemas expertos basadas en el conocimiento pueden ofrecer¹³.

La función manifiesta de los sistemas expertos, consiste en proporcionar experiencia humana a un ordenador. Se espera que un código experto ayude al usuario como un experto humano lo haría¹⁴⁻¹⁷.

Los elementos de un sistema experto son las "bases de conocimiento" y los "motores de inferencia" que resuelven el problema interpretando el conocimiento conforme se va adquiriendo. Existe también un "espacio de trabajo" donde se almacena el estado del problema en cada momento.

Las bases de conocimiento se clasifican en "superficiales" cuando se basan en reglas que buscan cuales algoritmos emplear en función de los datos de entrada recibidos, y

“profundos” cuando hacen uso de reglas matemáticas y conocimientos heurísticos para interpretar dichos datos de entrada y sacar conclusiones¹⁸.

En el cálculo estructural, un sistema experto busca alcanzar una mejor solución, en menos tiempo y a más bajo costo. A esta meta final se llega tras superar metas intermedias que se consiguen en base a “tareas internas” que preguntan por datos de entrada y “tareas externas” que se encargan de ejecutar algoritmos con estos datos, este proceso se define formalmente mediante reglas como: SI A y/o B ENTONCES E, SI C y/o D ENTONCES F, SI E y/o F ENTONCES MF. La aplicación sucesiva de estas reglas da lugar a un árbol de búsqueda como el de la Figura 2.

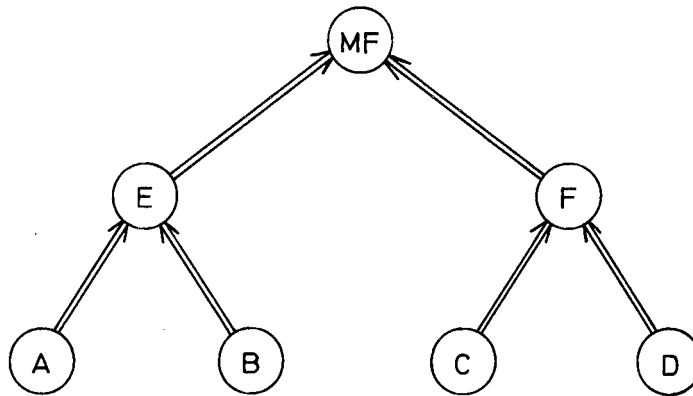


Figura 2. Árbol de sucesos de un sistema experto.

En aplicaciones, como medicina y otros campos más probabilísticos, los árboles como consecuencia de la cantidad de conocimientos superficiales que necesitan adquieren grandes magnitudes y complejidad. En la mecánica estructural la gran parte del conocimiento es profundo¹⁹⁻²², es por ello que estamos dentro de un campo de expertos más simple y de árboles menos complicados. En particular, el conocimiento profundo que ha de implementarse al experto es el comportamiento de los elementos finitos jerarquizados p , elegidos para la resolución del problema, y la estimación del error de las soluciones obtenidas.

El árbol de objetivos y tareas que da lugar al experto presentado es mostrado en la Figura 3, en el que cada objetivo se consigue cuando se realizan las tareas u objetivos que de él cuelgan. El motor de inferencia que se ha empleado es de encadenamiento inverso, en el que se parte del objetivo al que se trata de llegar intentando verificar los subobjetivos o tareas que dan lugar a él. Cuando un objetivo o tarea no puede ser verificado debe retrocederse en el árbol de búsqueda hasta un punto donde pueda ser escogida otra opción que pueda dar lugar a la validación de dicho objetivo. En el experto desarrollado los únicos puntos de opción están en la creación de la malla, opción que se deja en poder del usuario bajo consejos del experto, y la selección de grados de libertad adicionales, que es realizada automáticamente mediante un módulo experto de conocimiento profundo.

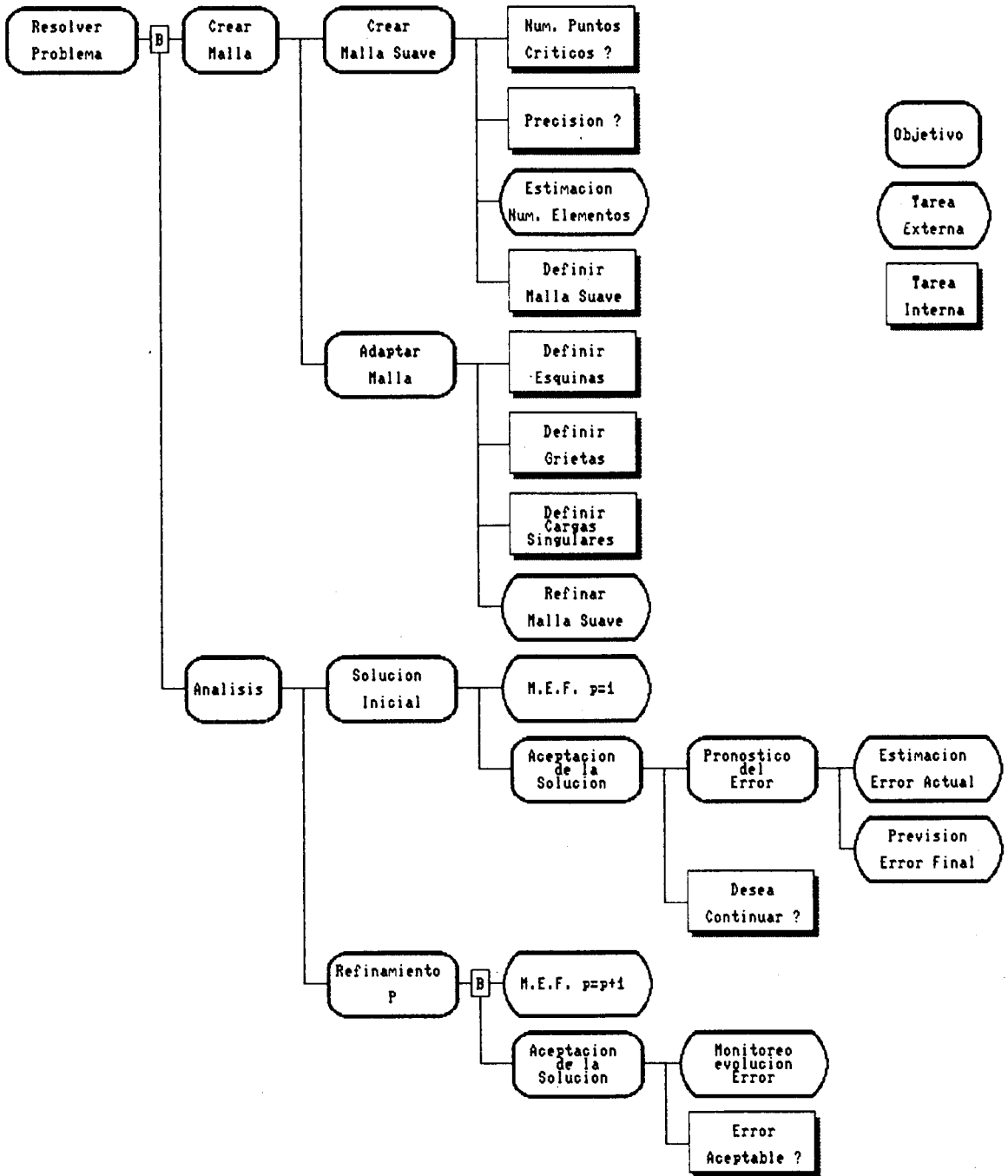


Figura 3. Arbol de tareas de CUPEX.

Diseño de malla

Una malla se considera óptima, si para obtener una determinada precisión se utilizan el menor número de elementos finitos posibles. Esto hoy día es inalcanzable, pero se puede mediante un módulo experto diseñar una malla, que en cierta medida, posea la garantía de encontrarse en un entorno a ese punto óptimo.

Para conseguir este objetivo, el módulo experto debe garantizar la alta velocidad de convergencia que los refinamientos p adaptativos pueden mantener²³ y que se ve truncada en cuanto en el dominio existen zonas no suaves en tensiones. Babuska ha demostrado⁴ que realizando refinamientos h en capas de progresión geométrica 0.15 en estas zonas (ver Figura 4) se consigue mantener la alta velocidad de convergencia.

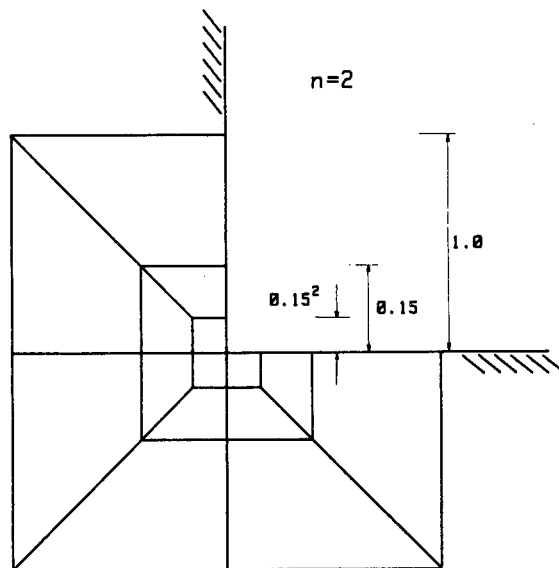


Figura 4. Refinamiento h en capas.

Planteada la situación, el módulo experto la lleva a cabo adquiriendo el conocimiento necesario para construir una malla regular adaptada al contorno que se ha denominado "malla suave". Una vez construida recaba información de las zonas no suaves en tensión, para automáticamente realizar refinamientos h en ellas y en el número de capas aconsejados (en el árbol de búsqueda de la Figura 3, ver la subtask malla).

Para construir la malla suave; como base de conocimiento superficial, el código pide al usuario la precisión que desea alcanzar. El hecho de que en los problemas de cálculos estructurales se introduzcan coeficientes de seguridad que escalan la respuesta en tensiones al menos en un 150 %, y la razón de calcular el estimador sobreestimándolo, indican que solicitar una precisión excesiva en la respuesta elevaría innecesariamente el costo del problema, e incluso podría llegar a no ser alcanzada por defecto del estimador. Es por eso que en este módulo, la primera tarea externa comienza solicitando el error que el usuario considera admisible (Ead), aconsejando para ello dos tipos de precisiones:

- 1) Precisión normal: $5\% \text{ Ead} \leq 10\%$
- 2) Alta precisión: $\text{Ead} \leq 5\%$ (casos excepcionales)

A continuación el motor de inferencia selecciona otra tarea externa consistente en un algoritmo de estimación del número de elementos de la malla suave. Para lo cual se hace la hipótesis de alcanzar Ead con un refinamiento jerarquizado parabólico completo, con lo que se está del lado de la seguridad ya que en la resolución posterior se puede llegar a mayores refinamientos y de forma adaptativa.

Heurísticamente para fijar el orden de magnitud de la malla se ha considerado que el punto de diseño (P_1 en Figura 5) está sobre la recta de refinamiento h completo con elementos finitos bilineales.

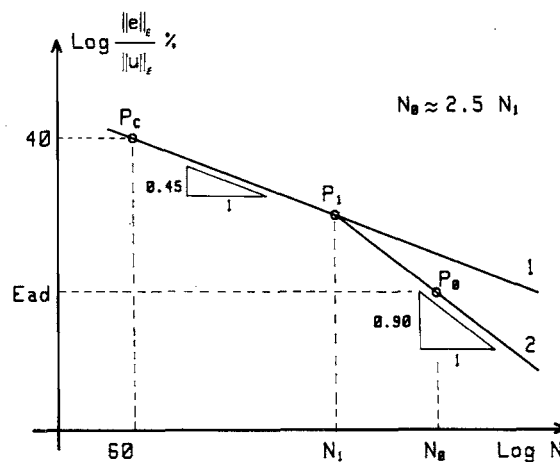


Figura 5. Estimación de los grados de libertad N_1 de la malla suave.

($p = 1$) determinada al considerar muy aceptable una aproximación del 40 % de error con 60 grados de libertad (punto P_c en la Figura 5) y estimar una pendiente de 0.45 para la misma, y del doble para el refinamiento jerarquizado parabólico completo. Con estas consideraciones y teniendo en cuenta que el número de grados de libertad correspondientes al refinamiento parabólico es dos veces la suma de nodos y lados de la malla se ha obtenido la expresión heurística:

$$N_{el} \approx 2.5 N_1 \approx 50 \text{ Ead}^{-0.45} * \lambda \quad (15)$$

$$\lambda = 1 + 0.1 * N_{pc} \quad (16)$$

donde N_{pc} = Número de puntos críticos
 Punto crítico es el representativo
 de una zona no suave en tensiones

El factor λ trata de que los puntos críticos tengan una ligera influencia sobre esta malla. El computador presenta como número de elementos aceptables de la malla el valor obtenido de (15) y muestra gráficamente las medidas del elemento medio, con lo cual el usuario construye la malla.

Realizada la malla suave el motor de inferencia selecciona la segunda rama del árbol de búsqueda (ver Figura 3) que se compone de tres tareas internas y una externa. Las internas son:

- 1) Introducción de datos de esquinas reentrantes y determinación del número de capas en cada una.
- 2) Introducción de datos de vértices de grietas y determinación del número de capas en cada uno.
- 3) Introducción de cambios en las condiciones de contorno (cargas puntuales) y número de capas en cada uno.

Para la determinación del número de capas en cada punto crítico, CUPEX aconseja al usuario en función de sus características.

La tarea externa consiste en un algoritmo que recibe como datos de entrada la malla suave y los puntos críticos con las capas definidas en ellos, realizando los correspondientes refinamientos h .

Al final queda como malla a introducir en los datos de entrada que analizan el problema, la malla suave con la introducción automática en las zonas críticas de las capas de refinamiento h consideradas convenientes por el usuario.

Pronóstico de error

Babuska y Rank²⁴ sugieren para estimar el error para la versión p completa en una malla fija:

$$\| e \|_E = C N^{-\delta} \quad (16)$$

donde δ , por la experiencia para bajos valores de p y mallas groseras toma valores $\delta = 0.7$ a $\delta = 0.75$, dando resultados razonables en la mayoría de los casos prácticos. En CUPEX se ha utilizado para el análisis una versión p adaptativa, cuya velocidad de convergencia es superior a la completa. Se comenzó utilizando $\delta = 0.75$, pero en todos los casos prácticos resueltos se alcanzó mayor precisión que la pronosticada. Se han obtenido resultados razonables tomando para valor del parámetro $\delta = 0.9$.

Iniciado en CUPEX el análisis del problema y una vez obtenida la solución lineal se calcula la norma energética del error, y en base a este conocimiento profundo y al heurístico de la expresión (16) se pronostica el error final (Efore), presentándose al usuario en pantalla y dándole opción a abortar el cálculo o a continuar las etapas de refinamiento si es aceptado.

De continuar la resolución al final de cada etapa adaptativa de jerarquización se le muestran al usuario una serie de gráficos que le sirven de instrumentos de control para observar como va la singladura (análisis del problema) en las etapas cubiertas hasta ese momento. Estos gráficos representan el estimador de error por grados de libertad y por etapas, el número de g.l. seleccionados de los totales existentes y la energía interna que

representan los indicadores de error de los g.l. seleccionados y la realmente absorbida en cada etapa. Estos gráficos que serán presentados con los ejemplos numéricos, dan al usuario una visión clara de la situación de los mismos que actúan como bases de datos del motor de inferencia que resuelve mediante estos consejos. Así pues constituye una conveniente ayuda para decidir realizar una etapa más de análisis o abortar por considerar la respuesta satisfactoria o estimar se debe partir de una malla más fina.

Selección de grados de libertad

Para elegir entre los grados de libertad seleccionables en una etapa de un proceso de refinamiento, el criterio generalmente utilizado ha consistido en fijar el valor de un parámetro $\gamma \in (0, 1)$ y considerar como válidos los grados de libertad cuyo indicador de error cumplen:

$$\eta \geq \gamma \eta_{\max} \quad (17)$$

Al principio se utilizaron valores de γ comprendidos entre 0.25 y 0.5¹¹. Posteriormente se observó⁹ que la mayoría de los indicadores de error toman un valor menor que un décimo del indicador máximo por lo que la elección de $\gamma = 0.1$ debe dar la mayor velocidad de convergencia.

En un microcomputador hay que vigilar que no se dispare el tiempo de cálculo por lo que se debe intentar bajar el número de etapas de refinamiento. Por otro lado, las limitaciones de memoria RAM obligan a que los refinamientos sucesivos traten de incrementar lo menos posible los grados de libertad.

Con estas consideraciones se ha establecido un criterio para que absorba gran parte de la energía elástica al comienzo del proceso de análisis, manteniendo siempre un control del número de grados de libertad. La estrategia local energética se adapta a través del parámetro $\beta \in (0, 1)$ eligiendo los L grados de libertad de mayor indicador que cumplan:

$$\sum_1^L \eta_i^2 \geq \beta \sum_1^N \eta_j^2 \quad (18)$$

N = Número de g.l. seleccionables en la etapa

y la estrategia global de control de indicadores se realiza con el parámetro γ tomando η_{\max} fijo en todas las etapas e igual al mayor obtenido en $p = 2$ (introducción del refinamiento parabólico). Se han obtenido resultados muy aceptables con valores de los parámetros $(\beta, \gamma) = (.99, .01)$.

ORGANIZACION INTERNA DEL PROGRAMA

En el apartado anterior se ha presentado la base de conocimientos, tanto superficiales como profundos, que se ha implementado en el programa CUPEX, así como el motor de inferencia que conduce al objetivo propuesto. Este proceso de inferencia podría haberse llevado a cabo mediante un lenguaje de programación lógica

(PROLOG, LISP,...) pero un análisis de las secuencias de tareas que son necesarias para la consecución del objetivo permite implementar el experto en una estructura secuencial de programación, la cual es natural para los lenguajes tradicionales (BASIC, FORTRAN, PASCAL).

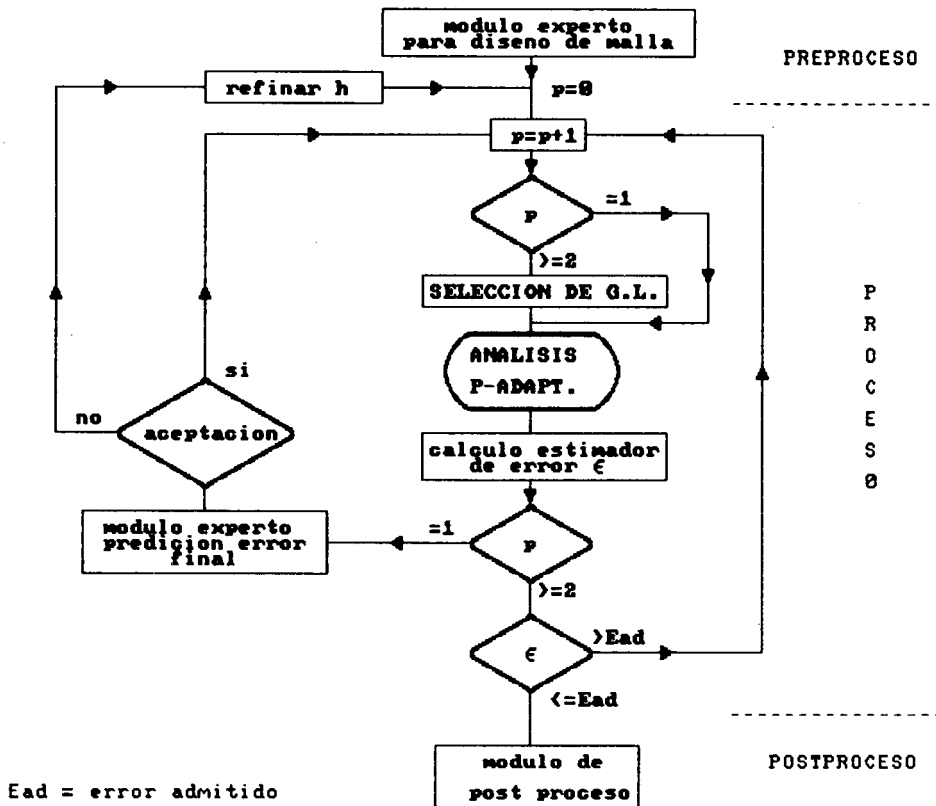


Figura 6. Diagrama de flujo de CUPEX.

El programa CUPEX se ha escrito en BASIC, el cual posee gran capacidad de graficación y facilidad de comunicación con el usuario. En la Figura 6 se muestra el diagrama de flujo, donde se observan tres partes diferenciadas: Preprocesado, Procesado y Postprocesado, a las que se accede mediante el Menú Principal de selección del programa presentado en pantalla.

Para ejecutar un problema previamente se han de crear los ficheros de datos de entrada, para lo cual se selecciona la opción DATOS DE ENTRADA del Menú Principal, apareciendo en pantalla el Menú correspondiente para la creación de la malla con ayuda del módulo experto e introducción del resto de datos necesarios.

Debido al empleo de E.F. jerarquizados p conformes las conectividades requieren un criterio de ordenación para mantener la conformidad C^0 en las fronteras entre elementos contiguos (ver Figura 7). Se ha provisto al programa de un algoritmo de chequeo que

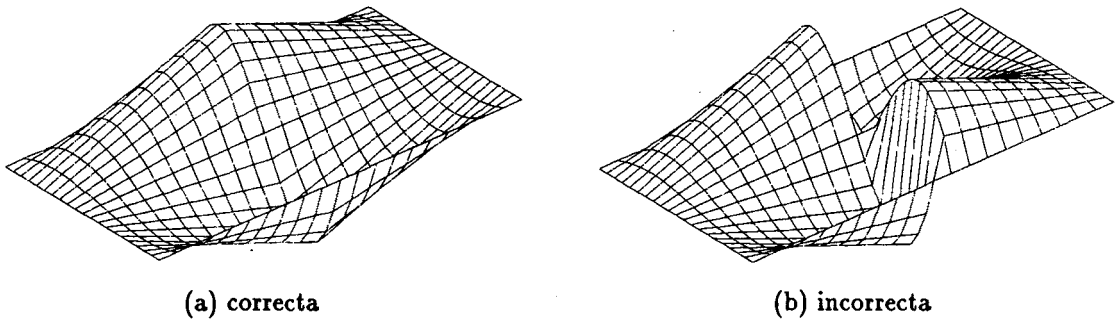


Figura 7. Ordenación de conectividades.

de un mensaje de error, indicando los elementos donde se rompe la conformidad, si así sucede.

Una vez introducidos los datos, se vuelve al Menú Principal y si se desea ejecutar el problema en ese momento se selecciona Análisis. Este se inicia (ver Figura 6) realizando un primer cálculo mediante EF bilineales, después se llama a un módulo que calcula el estimador de error de esta solución, y a continuación a un módulo experto que pronostica el error esperado de la respuesta final. Si ésta es aceptada por el usuario, continúa el análisis, comenzando las etapas ($p = 2, 2, \dots$) de cálculo de la solución con E.F. jerarquizados p conformes auto-adaptativos.

Si el error pronosticado no se acepta, el código aconseja introducir una capa más de refinamiento h en las zonas críticas. A continuación el programa vuelve al Menú Principal para que en la sección de Pre-procesado variemos los ficheros que sean necesarios.

Si el usuario acepta el error pronosticado, el programa realiza el cálculo p -autoadaptativo deteniéndose en la jerarquización que se alcanza el error aceptado, volviendo de nuevo al Menú Principal para si el usuario lo desea seleccionar la sección de Post-procesado donde puede escribir o leer en pantalla datos de desplazamientos y tensiones principales en el dominio, tensiones en la sección que se desee, deformaciones, etc.

RESULTADOS NUMERICOS

Para ilustrar el comportamiento de CUPEX se ha tomado como ejemplo de cálculo el alojamiento de máquina mostrado en la Figura 8, este problema ha sido usado ya por varios autores. Se ha pretendido obtener una precisión normal, así cuando CUPEX preguntó sobre el error admitido se le indicó de un 6 %. A las preguntas de números de esquinas reentrantes se le indicaron dos (puntos A y B de la Figura 8), cero vértices de grietas y cero cargas puntuales. Con estos datos CUPEX aconsejó una malla suave (sin refinamientos por capas) de unos 26 elementos.

Realizada esta malla, CUPEX pregunta para que se identifique el nodo de cada esquina reentrante, y en este caso aconseja realizar un refinamiento de una capa en cada una de ellas. Introducidos estos datos, realiza automáticamente el refinamiento h en el número de capas indicado por el usuario y que en este caso se ha hecho coincidir con el aconsejado, en la Figura 9 se muestra la malla obtenida. A continuación se

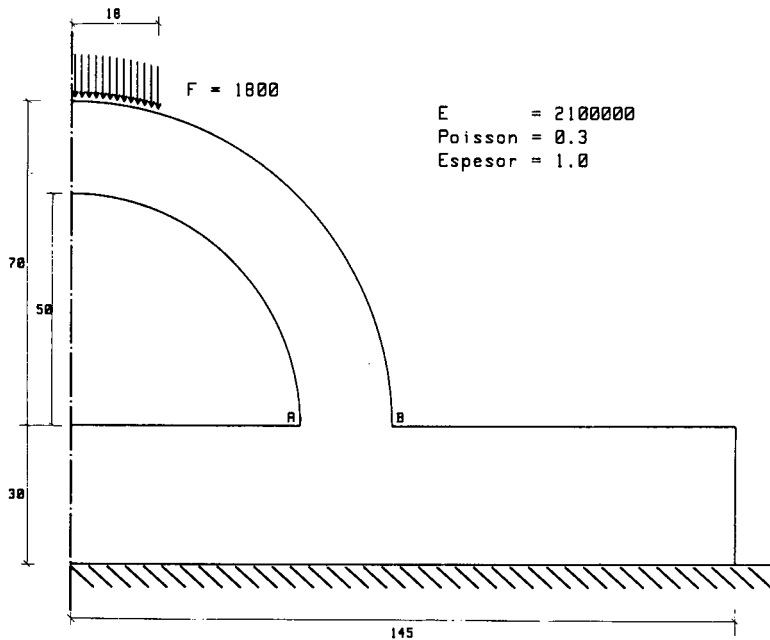


Figura 8. Alojamiento de máquina. Tensión plana.

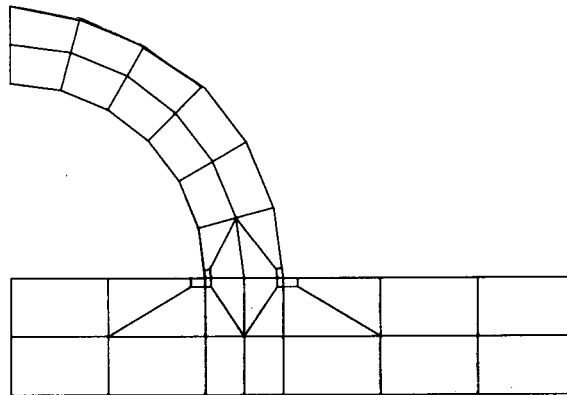


Figura 9. Malla definitiva para el análisis del alojamiento de máquina.

introduce el resto de datos numéricos pasándose a la fase de procesado o análisis.

CUPEX indica se encuentra en la primera etapa ($p = 1$), y va indicando cada paso realizado (cálculo de matrices de rigidez, vector de carga, etc.), después del cálculo del estimador de error presenta en pantalla el siguiente mensaje:

“CON LA MALLA ACTUAL TE PUEDO ASEGURAR MENOS DEL 9 % DE ERROR”

¿DESEAS CONTINUAR O PREFIERES USAR UN MALLADO MAS FINO?”

Se consideró aceptable y se le ordenó que continuase. Inmediatamente CUPEX

indica el inicio de la segunda etapa ($p = 2$) repitiendo el proceso realizado en la primera, finalizado el cual presenta en pantalla el gráfico indicado en la Figura 10, pero con sólo las dos etapas realizadas representadas. Se observa ha habido una reducción drástica en el error, pasando a ser ahora del 12 % y con una velocidad de convergencia de 2.3, muy superior a la media (0.9). En todos los ejemplos tratados esta etapa de introducción de las funciones de forma parabólica y de manera adaptativa es la que realiza la reducción más espectacular del error.

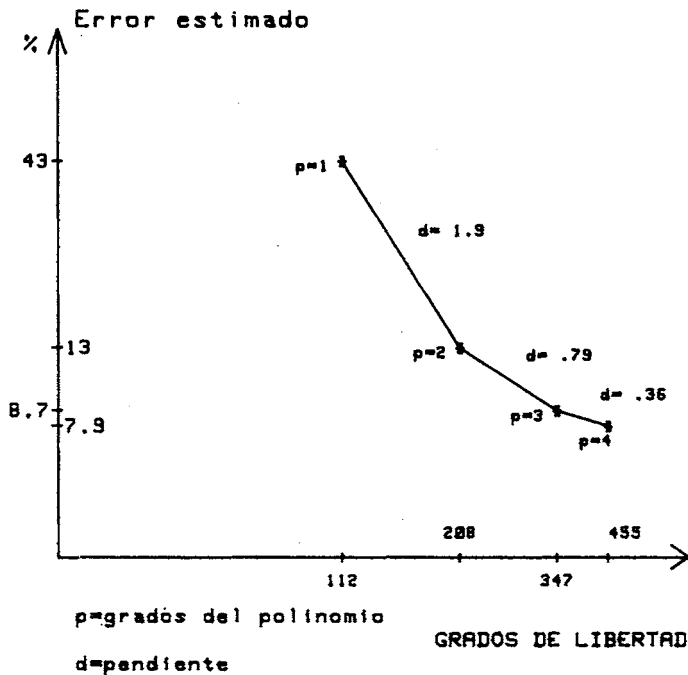


Figura 10. Gráfico de control. Evolución del error frente al número de grados de libertad.

CUPEX indica al usuario que continúe el proceso, apareciendo en pantalla el gráfico mostrado de la Figura 11 (sólo hasta $p = 2$), que se ha pensado es una forma más sencilla de interpretar, por parte de un usuario, el proceso que va realizando CUPEX, y es por tanto la única gráfica que aparecerá en la versión comercial del programa.

A continuación se presenta en pantalla la Figura 12 (hasta $p = 2$) que va haciendo una valoración entre la energía elástica que absorbe el dominio por la introducción de nuevos grados de libertad y la energía elástica que los indicadores de dichos grados de libertad dicen que tienen. Se ha hecho una valoración relativa supuesto 100 la energía introducida en la etapa bilineal. La escala vertical se ha tomado logarítmicamente para poder representar valores de $p = 3$ en adelante. Es digno de observar que debido al efecto de acoplamiento la energía interna elástica introducida es mayor que la que los indicadores dicen que van a suministrar.

A continuación se muestra en pantalla el gráfico de la Figura 13 (hasta $p = 2$) que muestra dos líneas poligonales: la superior corresponde al número total de grados

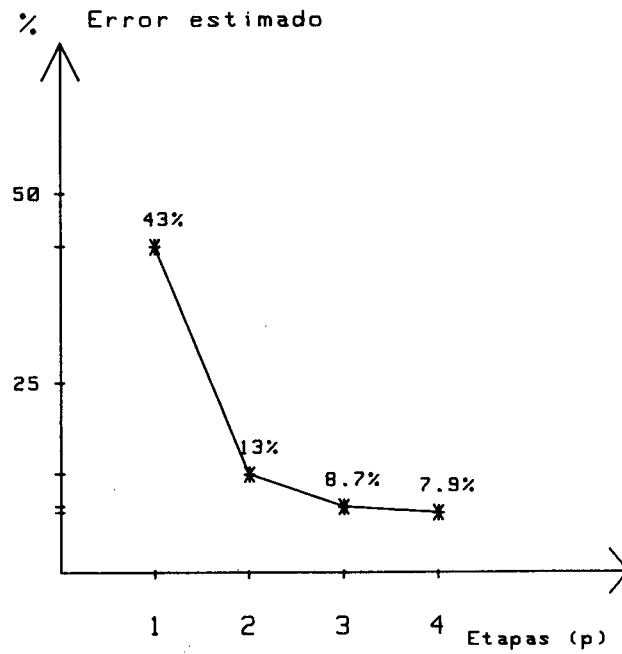


Figura 11. Gráfico de control. Evolución del error frente a las pasadas de jerarquización.

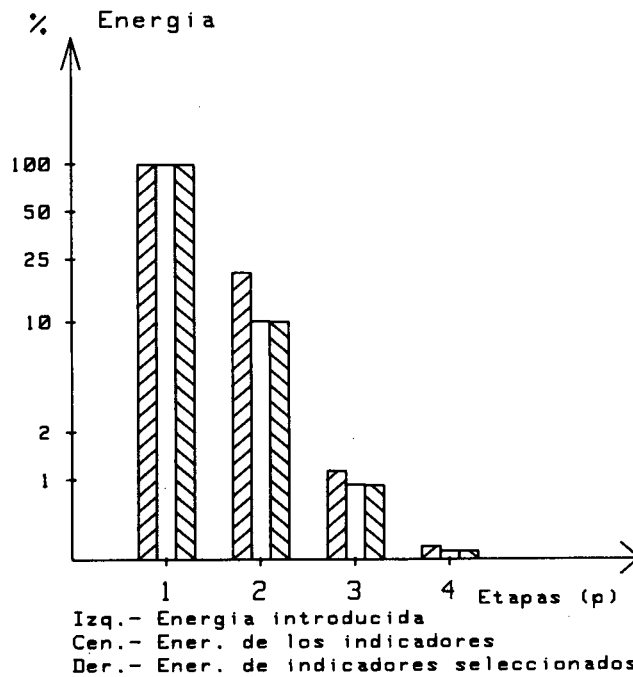


Figura 12. Gráfico de control. Evolución del proceso de selección: Energía.

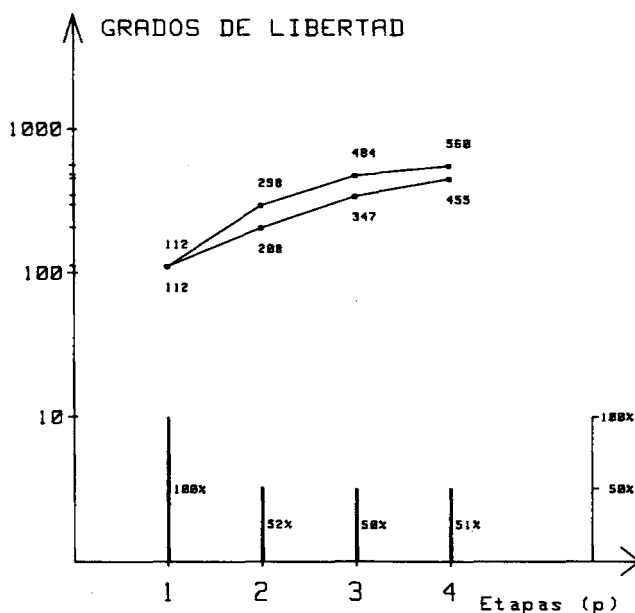


Figura 13. Gráfico de control. Evolución del proceso de selección: Grados de libertad.

de libertad posible en cada etapa y en la inferior el número de grados de libertad totales que se van seleccionando. Las barras de la parte inferior muestran el % de g.l. seleccionados en cada etapa. Este gráfico ha servido para medir la bondad de la estrategia adaptativa y adaptarla a los valores indicados.

A continuación de este gráfico pregunta si realiza una etapa más. Como se ha indicado, para el usuario del programa que no le interesa conocer el comportamiento interno del mismo, la gráfica de la Figura 11 es la que le va indicando la pauta a seguir. Se observa que en la etapa 3 el error cae al 7.5 % y en la etapa 4 al 6.8 %. Por la evolución de la línea quebrada se observa que en la próxima etapa el error apenas va a bajar, unido ésto a que el tiempo de ejecución de cada etapa va creciendo con p , se decidió no continuar el cálculo y aceptar esta solución como aceptable.

En la fase de post-proceso se ha desarrollado un módulo para calcular tensiones en secciones de corte. En la Figura 14 se representa la tensión normal correspondiente a la sección AB del dominio en estudio (ver Figura 8). Se observa en la gráfica pequeños saltos correspondientes a la interfrontera entre elementos.

Debido al elemento capa en los puntos A y B , la distribución de tensiones manifiesta las zonas críticas desde $p = 1$; no obstante, es a partir de $p = 2$ cuando la solución es aceptable, no mejorando ostensiblemente por la introducción de nuevos polinomios de jerarquización.

CONCLUSIONES

Las pretensiones de crear un programa muy sencillo de utilizar, a bajo costo y con la adecuada precisión, en general han sido satisfechas. El uso de un sistema experto con un motor de inferencia por encadenamiento inverso ha hecho posible que el programa

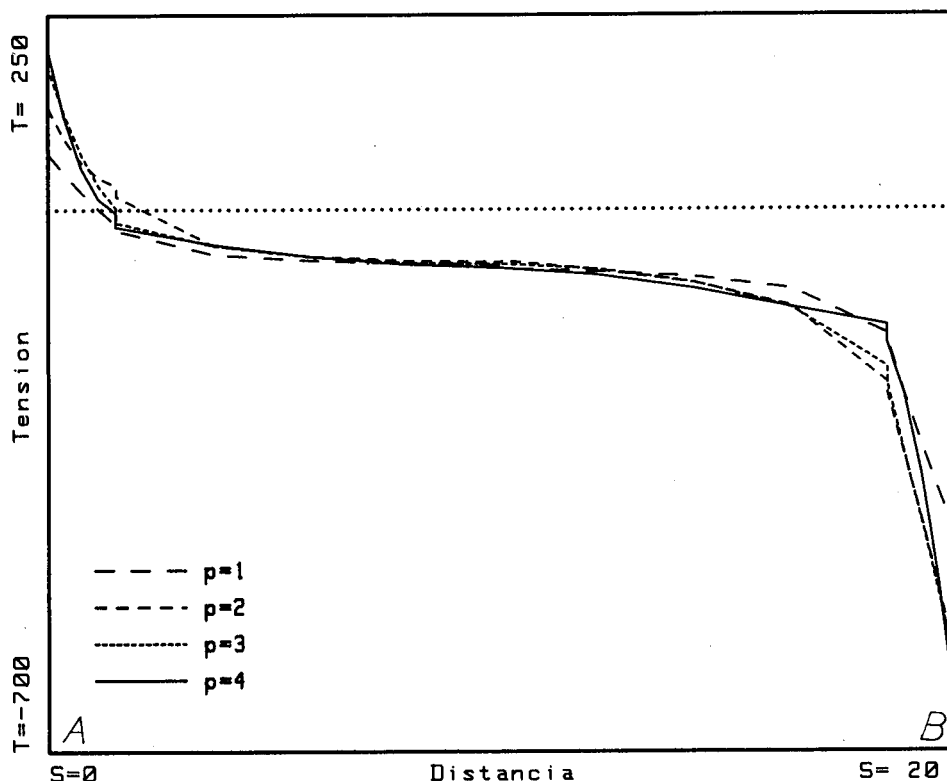


Figura 14. Tensiones en la sección A-B del alojamiento de máquina.

vaya pidiendo al usuario los datos necesarios de una forma lógica y ordenada.

CUPEX siempre ha presentado buenos resultados con la ejecución del adecuado número de etapas. Sin embargo cuando en la fase de post proceso se ha representado las tensiones en las zonas no suaves en tensiones se ha observado que no hubiera hecho falta algunas etapas finales que por otro lado encarecen el costo y tiempo de ejecución innecesariamente. Se piensa que el estimador de error no es representativo para el ingeniero calculista, no siendo rentable el alto costo que conlleva su cálculo. En un futuro, habrá que buscar un parámetro que evite el realizar etapas posteriores cuando ya el campo de tensiones esté suficientemente ajustado.

REFERENCIAS

1. R. Courant, "Variational methods for the solution of problems of equilibrium and vibration", *Bull. Am. Math.*, Vol. 49, pp. 1-23, (1943).
2. L.F. Richardson, "The approximate arithmetical solution by finite differences of physical problems", *Trans. Roy. Soc., London*, A210, (1978).
3. I. Babuska y W.C. Rheinboldt, "A posteriori error estimates for the FEM". *Int. J. Num. Meth. Eng.*, Vol. 12, pp. 1597-1615, (1978).
4. I. Babuska, "The p and $h - p$ version of the FEM. The state of the art", *Tech. Note BN-1156*, Univ. of Maryland, (1986).
5. A. Peano, "Hierarchics of conforming FE for plane elasticity and plate bending", *Comp.*

- Math. Appls.*, Vol. 2, pp. 211-224, (1976).
6. K-J. Bathe y E.L. Wilson, "Numerical Methods in FE Analysis", pp. 127-133, Prentice Hall, Inc., (1976).
 7. O.C. Zienkiewicz, J.P. Gago y D.W. Kelly, "The Hierarchical concept in finite element analysis", *Comp. & Struct.*, Vol. 16, pp. 53-65, (1983).
 8. O.C. Zienkiewicz, "*El Método de los Elementos Finitos*", Edit. Reverté, S.A.
 9. O.C. Zienkiewicz y A.W. Craig, "A posteriori error estimation and adaptative mesh refinement in the finite element method", *The mathematical basis of finite element method* Clarendon Press, Oxford, (1984).
 10. O.C. Zienkiewicz y K. Morgan, "Finite Elements and Approximation", Cap. 8, John Wiley & Sons, (1983).
 11. J.P. Gago, "A posteriori error analysis and adaptativity for the FEM", *Thesis*, University of Wales, (1982).
 12. A. Peano, M. Favelli, R. Riccioni y R. Sardella, "Self adaptive convergence at the crack tip of a dam buttress", *Int. Conf. on Fracture Mechanics*, Swansea, (1979).
 13. J. Mackerle y K. Orsborn, "Expert systems for finite element analysis and design optimization-a review", *Eng. Comput.*, Vol. 5, (1988).
 14. C. Naylor, "*Construya su propio sistema experto*", Ed. Díaz de Santos, S.A., (1986).
 15. J. Cuenca, G. Fernandez, R. Lopez y M.F. Verdejo, "*Inteligencia artificial: Sistemas expertos*", Alianza editorial, S.A., (1986).
 16. F. Hayes-Roth, D. Waterman y D. Lenat, "Building expert systems", Addison-Wesley publishing company, Inc., (1983).
 17. P. Harmon, "Overview of small expert systems building tools", *Expert Systems Strategies*, Vol. 1, pp. 1-10, (1985).
 18. M.H.J. Romanyca y F.J. Pelletier, "What is heuristic?", *Comp. Intell.*, Vol. 1, pp. 47-58, (1985).
 19. D. Sriam et al., "Knowledge-based expert systems in structural design", *Comput. Struct.*, Vol. 20, pp. 1-9, (1985).
 20. M.L. Mather y S.J. Fenves, "HI-RISE: an expert system for the preliminary structural design of high rise buildings", *Proc. Int. Fed. Inform. Process., WG5.2, Work. Conf. on Knowl. Eng. in Computer-Aided Design*, Budapest, (1984).
 21. H. Adeli, "Artificial intelligence in structural engineering", *Eng. Anal.*, Vol. 3, pp. 154-160, (1986).
 22. R.H. Holt y U.V.L. Narayana, "Adding intelligence to finite element modelling", CH2349-9/86, IEEE, New York, pp. 326-337, (1986).
 23. O.C. Zienkiewicz, D.W. Kelly, J. Gago y I. Babuska, "Hierarchical finite element approaches, error estimates and adaptative refinement", MAFELAP 1981, Brunel University, (1981).
 24. E. Rank e I. Babuska, "An expert system for the optimal mesh design in the hp-version of the finite element method", *Int. J. Num. Meth. in Eng.*, Vol. 24, pp. 2087-2106, (1987).