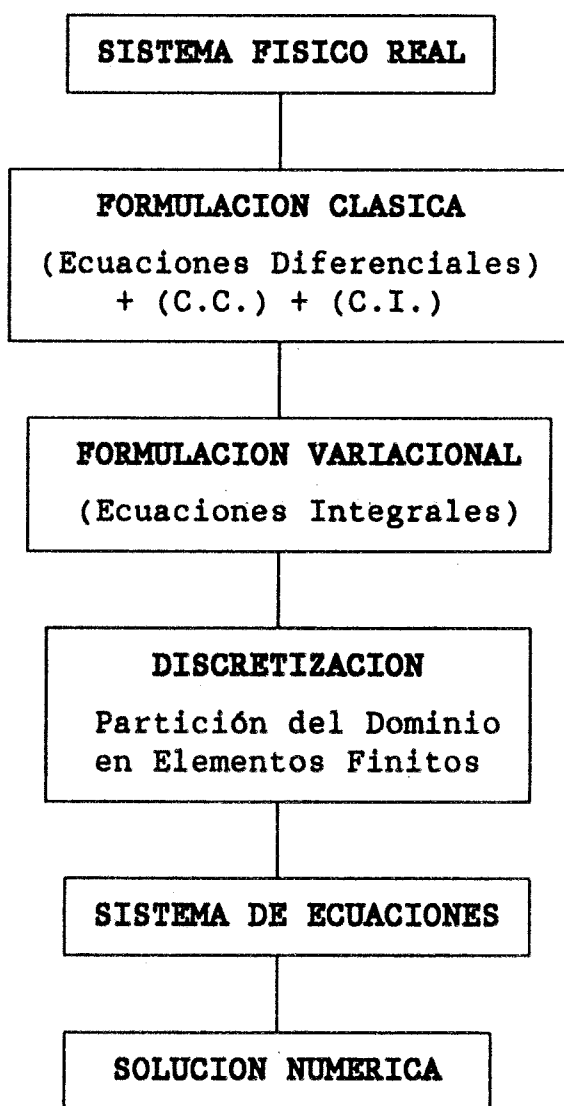


*** ETAPAS FUNDAMENTALES DEL METODO DE LOS ELEMENTOS FINITOS ***



* FORMULACION ABSTRACTA DEL METODO DE LOS ELEMENTOS FINITOS *

* FORMULACION CLASICA:

Hallar $u \in X$ en Ω tal que:

$$\begin{aligned} Au &= f \\ + \text{C.C.} \\ + \text{C.I.} \end{aligned}$$

* FORMULACION VARIACIONAL:

Hallar $u \in X'$ en Ω tal que:

$$\int_{\Omega} (Au) v \, d\Omega = \int_{\Omega} f v \, d\Omega \quad \forall v \in X'$$

6

$$a(u, v) = b(v) \quad \forall v \in X'$$

* FORMULACION EN DIMENSION FINITA:

Hallar $u_h \in X'_h \subset X'$ en Ω tal que:

$$a(u_h, v_h) = b(v_h) \quad \forall v_h \in X'_h$$

Tomando una base de X'_h ; $\{ \varphi_1, \varphi_2, \dots, \varphi_n \}$,

$$a(u_h, \varphi_j) = b(\varphi_j) \quad (j=1, 2, \dots, n)$$

Siendo $u_h = \sum_{i=1}^n u_h^i \varphi_i$, tenemos que

$$\sum_{i=1}^n u_h^i a(\varphi_i, \varphi_j) = b(\varphi_j) \quad (j=1, 2, \dots, n)$$

Resultando el sistema de ecuaciones:

$$A u_h = f$$

siendo,

$$(A)_{ij} = a(\varphi_i, \varphi_j)$$

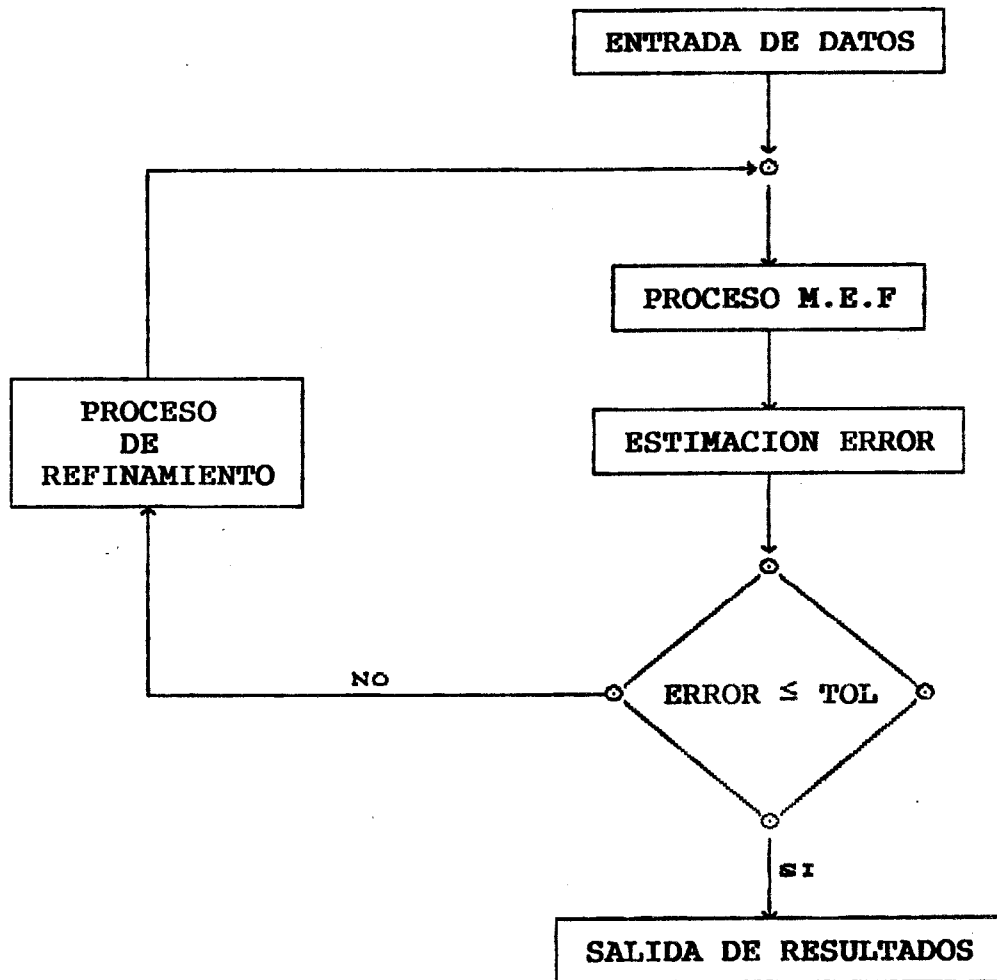
$$u_h = \begin{bmatrix} u_h^1 \\ u_h^2 \\ \vdots \\ u_h^n \end{bmatrix} \quad f = \begin{bmatrix} b(\varphi_1) \\ b(\varphi_2) \\ \vdots \\ b(\varphi_n) \end{bmatrix}$$

* ERROR DEL METODO:

$$\| e \| = \| u - u_h \| \leq C h^\alpha$$

*** INTRODUCCION ***

*** ORGANIGRAMA GENERAL DE UN CODIGO ADAPTATIVO ***



* PROBLEMA PLANTEADO EN 1-D *

$$-\frac{d}{dx}\left[K(x)\frac{du(x)}{dx}\right] + v(x)\frac{du(x)}{dx} + b(x)u(x) = f(x) \quad \text{en } \Omega=(a,b)$$

CONDICIONES DE CONTORNO

$$u(a) = u_a$$

$$u(b) = u_b$$

$$K(a)\frac{du(a)}{dx} = \sigma_a$$

$$-K(b)\frac{du(b)}{dx} = \sigma_b$$

$$K(a)\frac{du(a)}{dx} = p_a[u(a)-u_a]$$

$$-K(b)\frac{du(b)}{dx} = p_b[u(b)-u_b]$$

* ESTIMACION E INDICACION DEL ERROR *

ESTIMACION DEL ERROR

$$\|e\|_{\Sigma}^2 \approx \|e_o\|_{\Sigma}^2 = \sum_{i=1}^N \varepsilon_i^2 = \sum_{i=1}^N \frac{h_i^2}{12 K_{\min}} \int_{\Omega_i} r^2 dx$$

RESIDUO DE LA ECUACION

$$r(x) = - \frac{d}{dx} \left[K(x) \frac{du_h(x)}{dx} \right] + v(x) \frac{du_h(x)}{dx} + b(x)u_h(x) - f(x)$$

INDICADORES DE ERROR

$$\varepsilon_i = \left[\frac{h_i^2}{12 K_{\min}} \int_{\Omega_i} r^2 dx \right]^{1/2}$$

COEFICIENTE DE EFECTIVIDAD DE LA ESTIMACION

$$\theta = \frac{\|e_o\|_{\Sigma}}{\|e\|_{\Sigma}}$$

* Propiedades:

a) $\theta \geq 1$

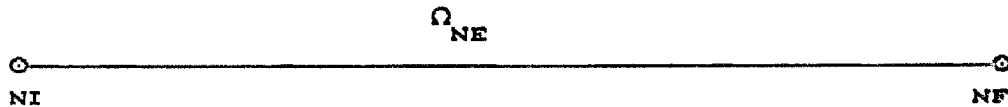
b) $\theta \rightarrow 1$ cuando $h_i \rightarrow 0$

* ESTRATEGIAS DE REFINAMIENTO *

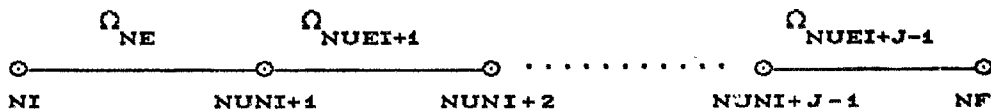
- a) ¿ Qué elementos se van a refinar ?.
- b) ¿Cuál es el grado de refinamiento máximo ?.
- c) ¿Cuál es el grado de refinamiento de cada elemento ?.

REFINAMIENTO DE UN ELEMENTO GENERICO EN J SUBELEMENTOS

* Malla M_n :



* Malla M_{n+1} :



- 1°) Definición del subelemento extremo izquierdo de Ω_{NE} .
- 2°) Definición del los subelementos interiores de Ω_{NE} .
- 3°) Definición del subelemento extremo derecho de Ω_{NE} .

* APLICACIONES NUMERICAS EN 1-D *

PROBLEMA I: DIFUSIÓN PURA

$$- \left[\left[\frac{1}{\alpha} + \alpha(x-\beta)^2 \right] u' \right]' = 2 \left[1 + \alpha(x-\beta) \left[\arctg \alpha(x-\beta) + \arctg \alpha\beta \right] \right]$$

para $x \in (0,1)$, y con condiciones de Dirichlet:

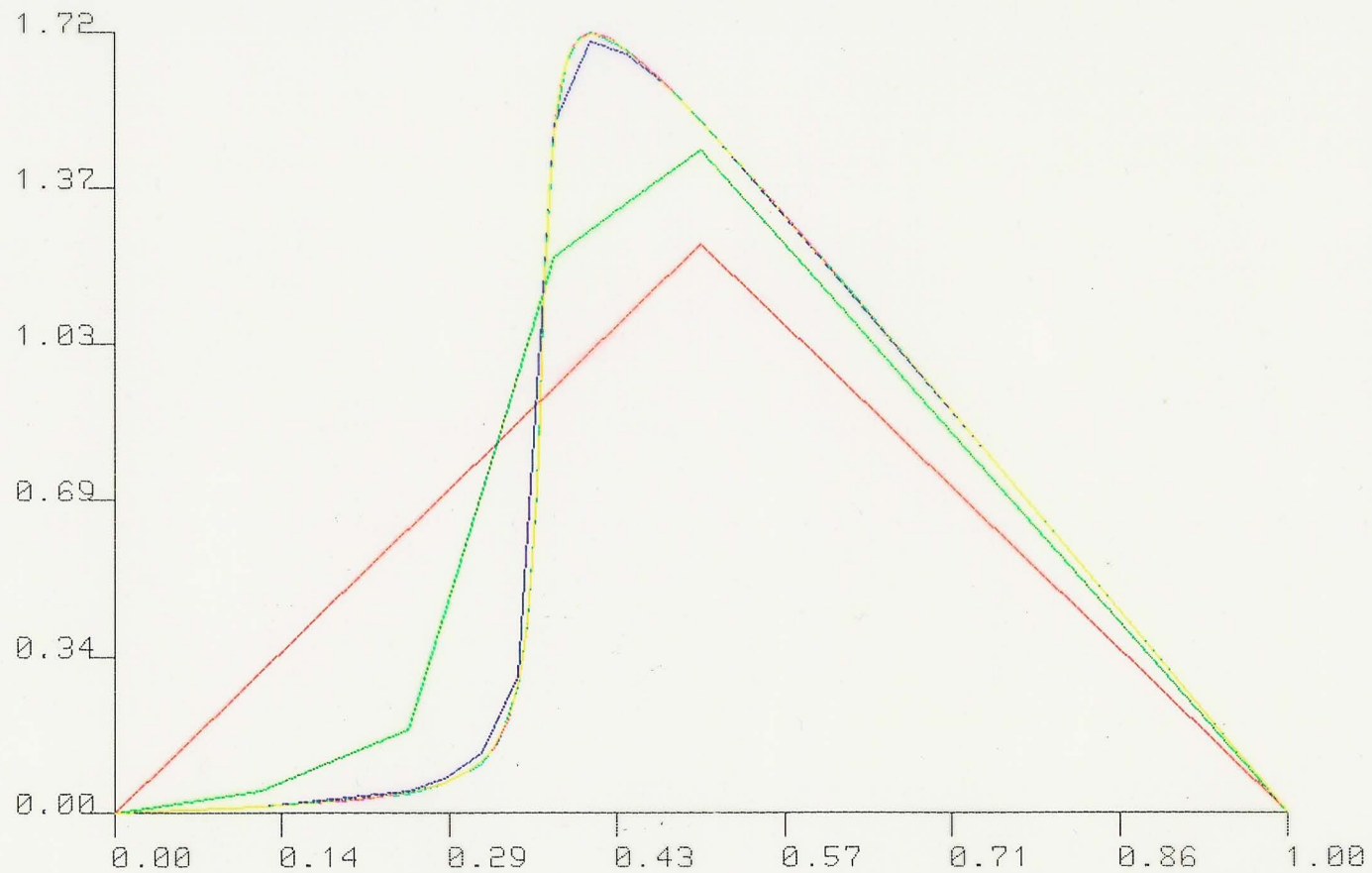
$$\begin{cases} u(0) = 0 \\ u(1) = 0 \end{cases}$$

cuya solución exacta es:

$$u(x) = (1-x) \left[\arctg \alpha(x-\beta) + \arctg \alpha\beta \right]$$

Trabajamos con $\alpha = 100$ y $\beta = 0.36388$

C: 4,1,4,2,0.01,0.1



--- : I0: 3 NODOS
--- : I1: 6 NODOS
--- : I2: 13 NODOS
--- : I3: 20 NODOS
--- : I4: 55 NODOS [AD-FFL]
--- : SOL. EXACT.

PROBLEMA II: CONVECCIÓN-DIFUSIÓN HOMOGÉNEO

$$-\varepsilon u'' + u' = 0 \quad \text{para } x \in (0,1)$$

$$\begin{cases} u(0) = 0 \\ u(1) = 1 \end{cases}$$

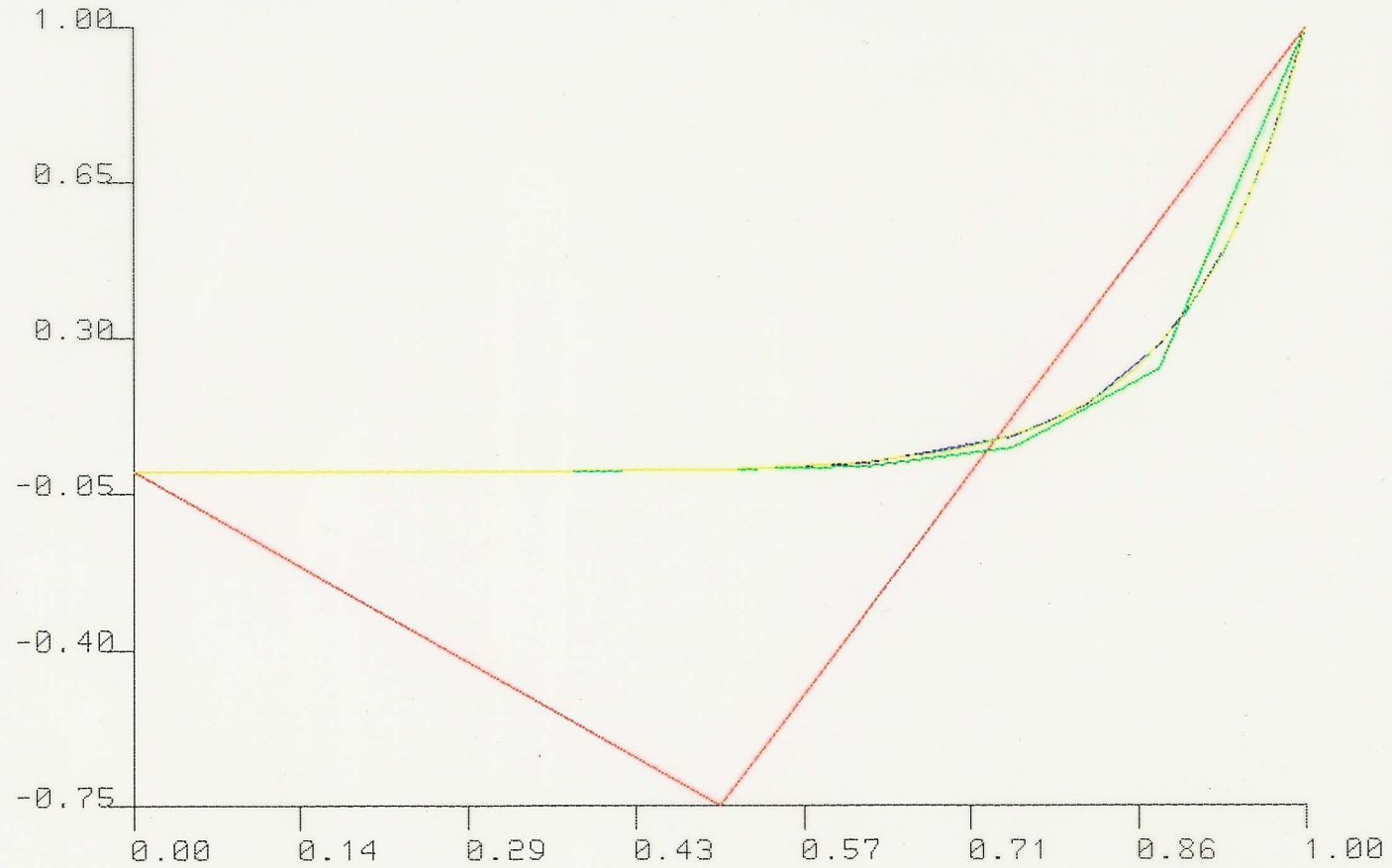
La solución exacta de este problema es:

$$u(x) = \frac{1 - e^{x/\varepsilon}}{1 - e^{1/\varepsilon}}$$

* Problema II.a.- ($\varepsilon = 0.1$) *

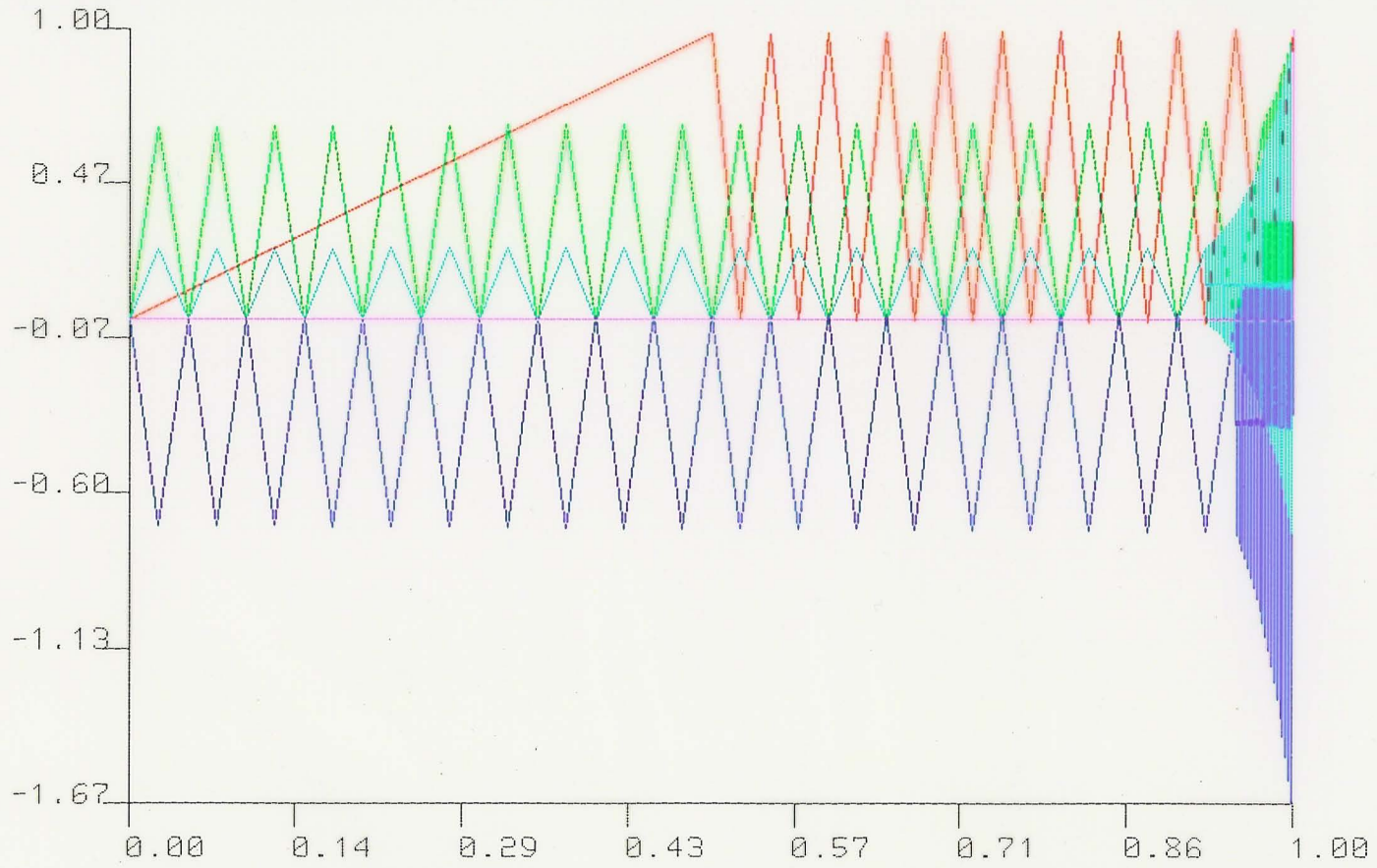
* Problema II.b.- ($\varepsilon = 10^{-5}$) *

C: 4,1,4,2,0.01,0.1



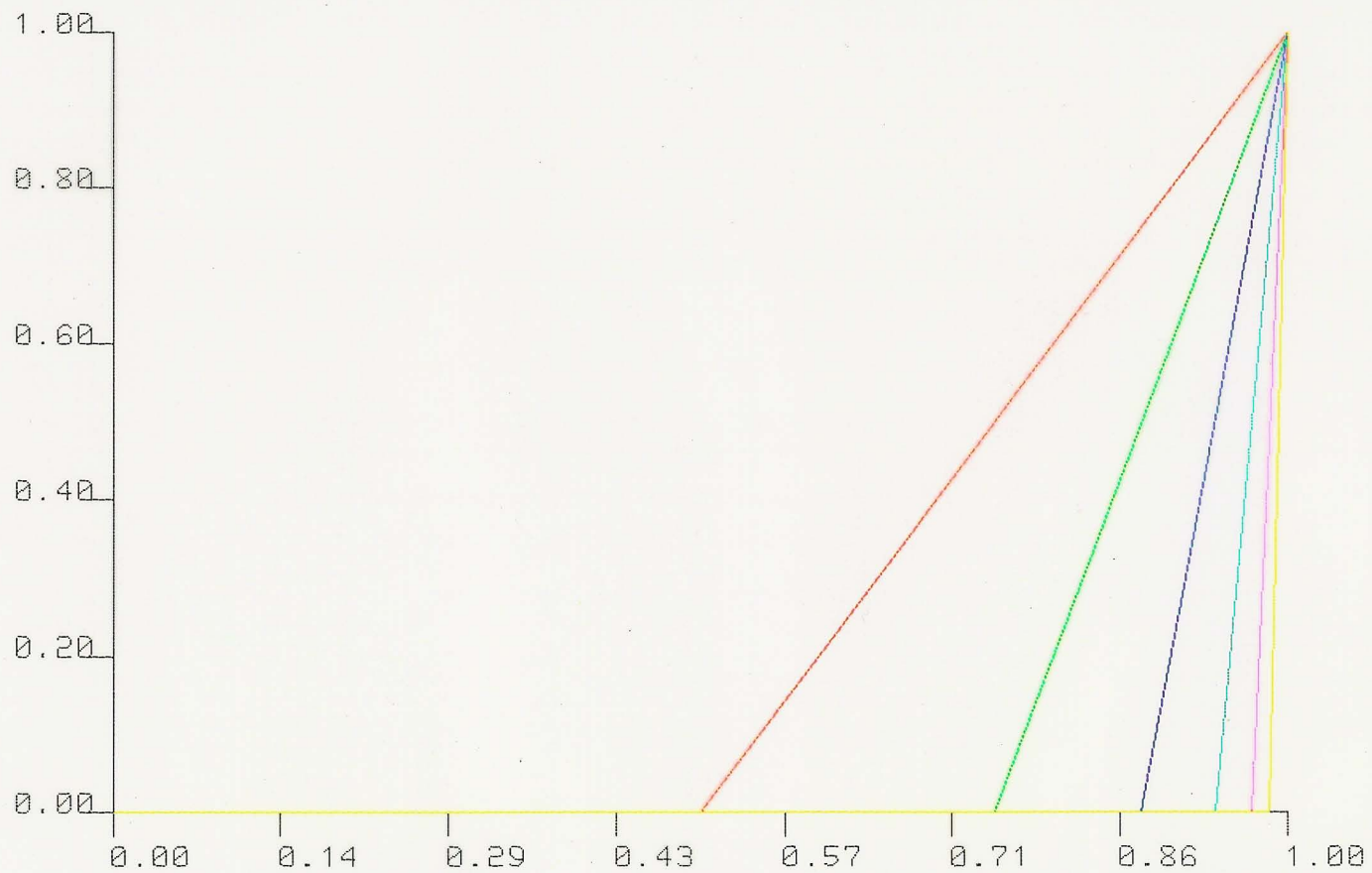
- : I0: 3 NODOS
- : I1: 8 NODOS
- : I2: 12 NODOS
- : I3: 28 NODOS
- : I4: 81 NODOS [AD-FFL]
- : SOL. EXACT.

C: 4,1,20,1,0.1,1.



— : I1: 22 NODOS
— : I3: 60 NODOS
— : I4: 79 NODOS
— : I5: 98 NODOS
— : I7: 136 NODOS [AD-FFL]

C: 3,2,4,2,0.1,0



--- : C0: 3 NODOS
--- : C1: 4 NODOS
--- : C2: 5 NODOS
--- : C3: 6 NODOS

--- : C4: 7 NODOS
--- : C5: 8 NODOS

[AD-DA]

PROBLEMA III: CONVECCIÓN-DESIÓN NO HOMOGÉNEO

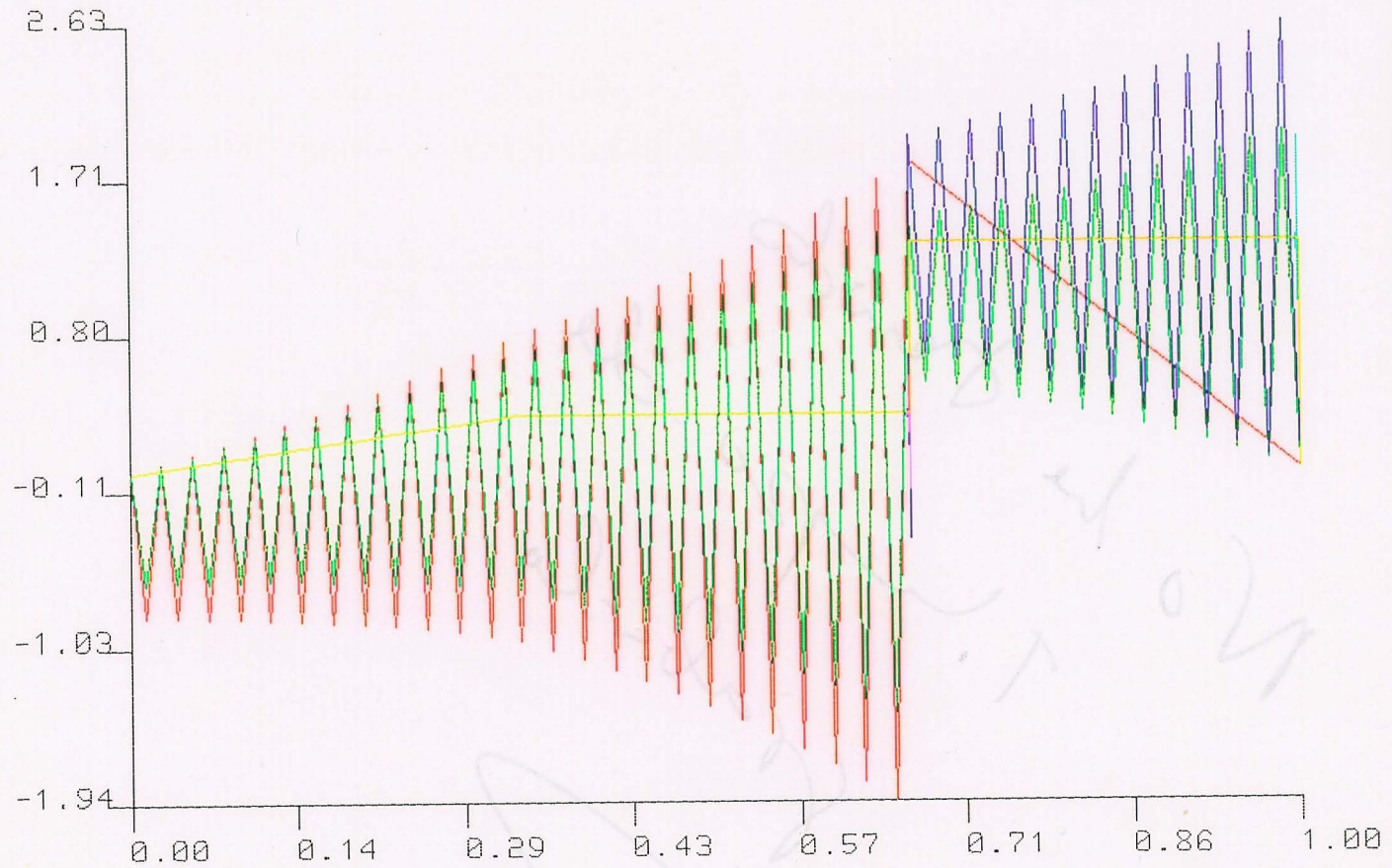
$$\begin{cases} -\epsilon u'' + u' = f & \text{en } (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

Tomando:

$$\epsilon = 0.0001$$

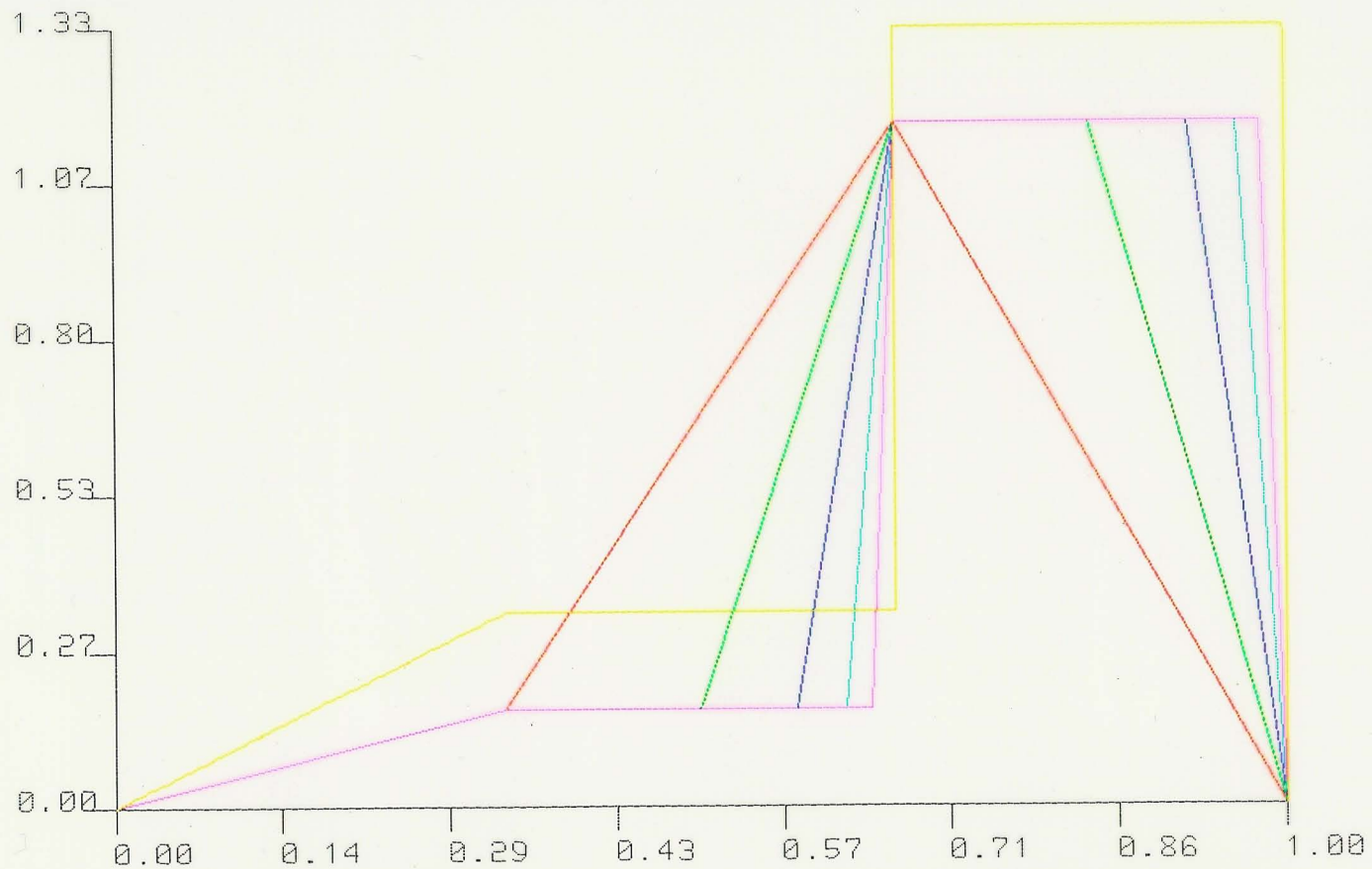
$$f(x) = \begin{cases} 1 & \text{si } 0 < x < \frac{1}{3} \\ 0 & \text{si } \frac{1}{3} < x < 1 \text{ ; } x \neq \frac{2}{3} \\ \delta\left(x - \frac{2}{3}\right) & \text{si } x = \frac{2}{3} \end{cases}$$

C: 4,1,25,1,0.1,1.



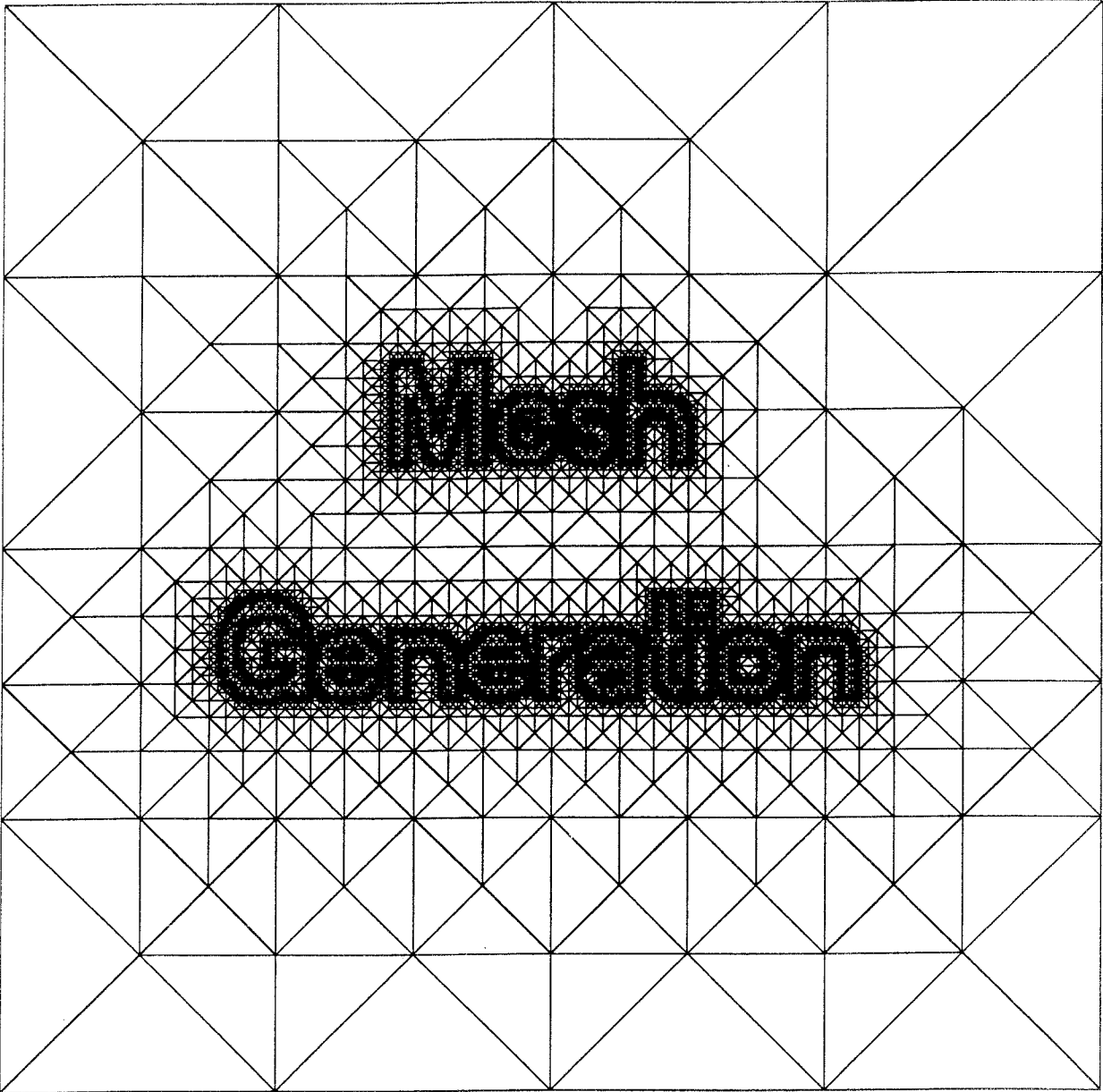
--- : I2: 52 NODOS
--- : I3: 76 NODOS
--- : I4: 100 NODOS
--- : I5: 124 NODOS
--- : I6: 148 NODOS [AD-FFL]
--- : I7: 172 NODOS

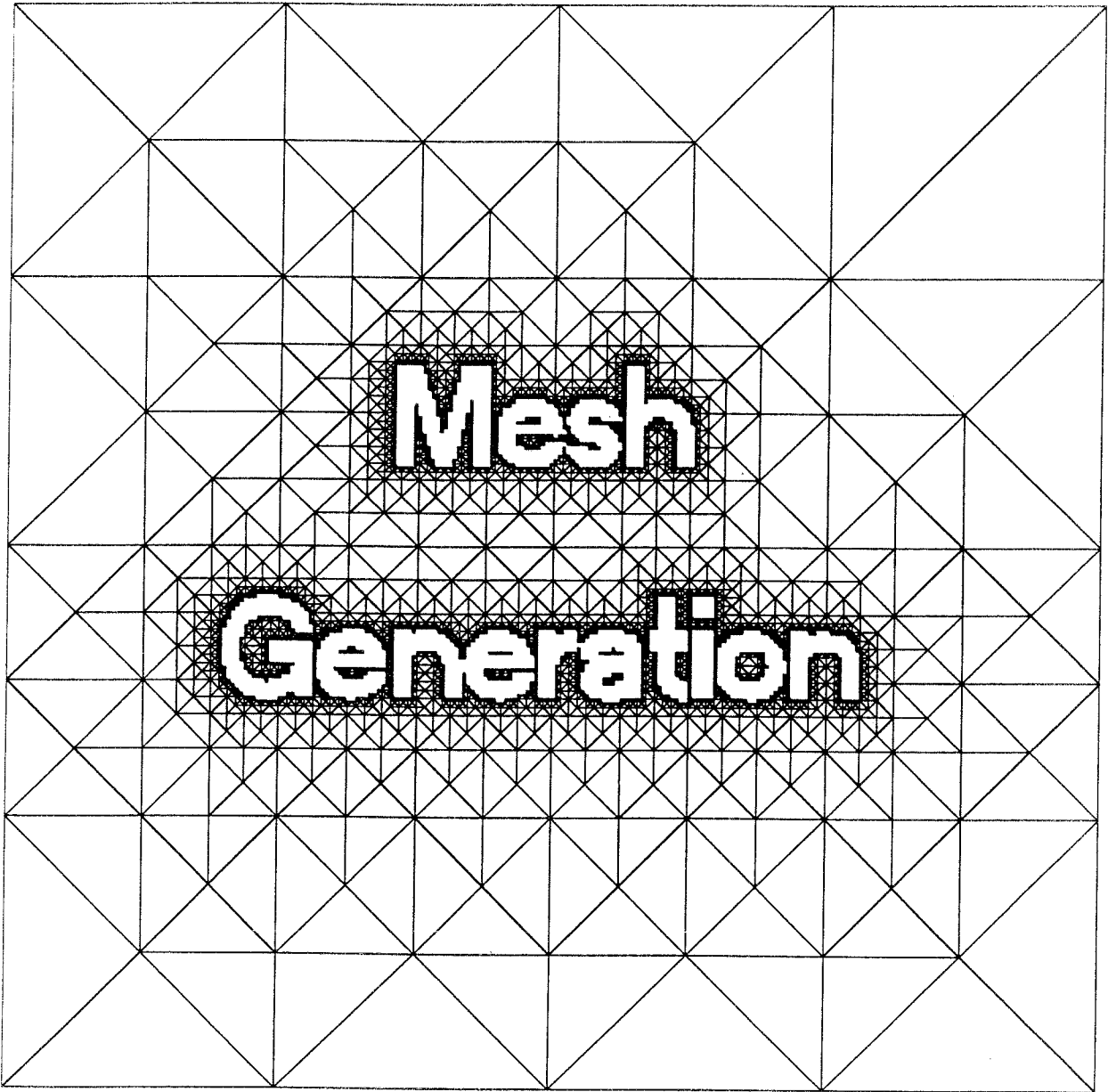
C: 3,2,4,2,0.1,0.



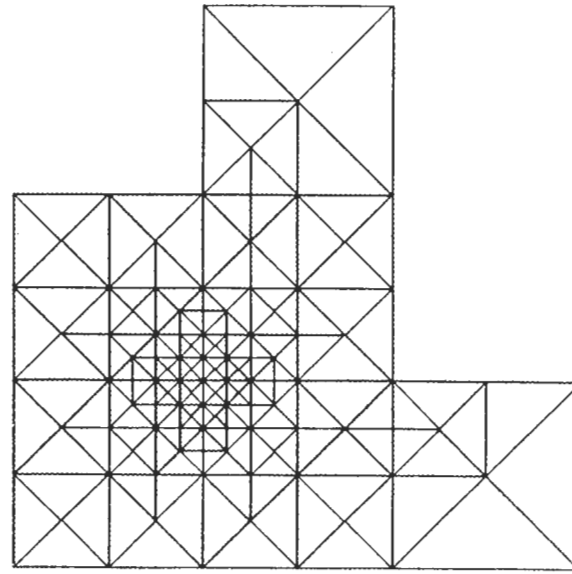
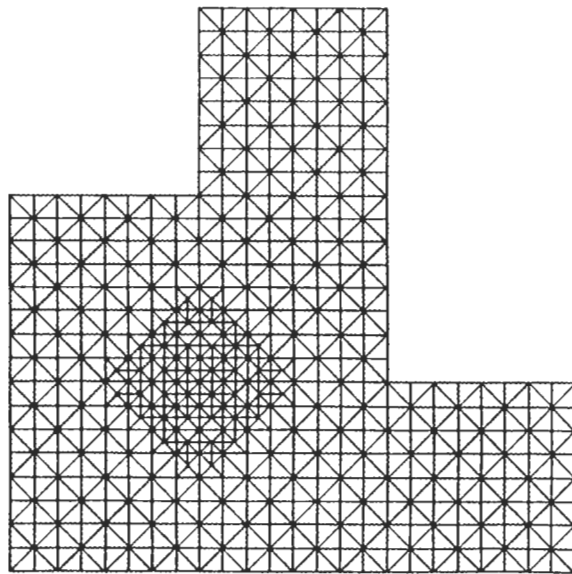
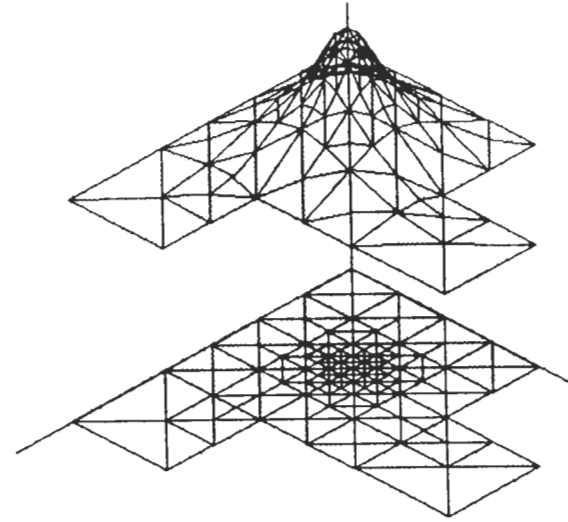
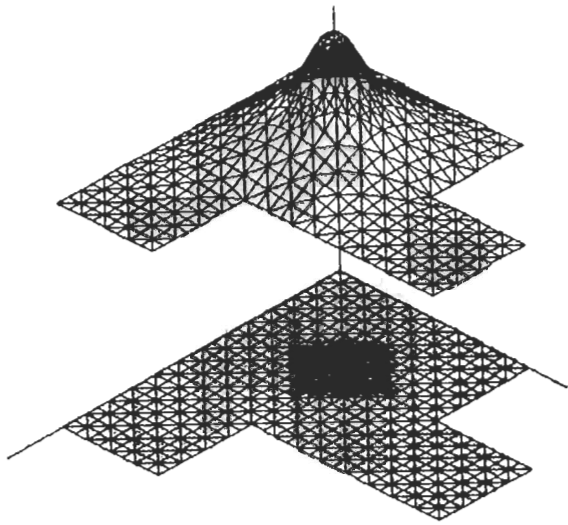
— : I0: 4 NODOS
— : I1: 6 NODOS
— : I2: 8 NODOS
— : I3: 10 NODOS

— : I4: 12 NODOS [AD-DA]
— : I24: 420 NODOS

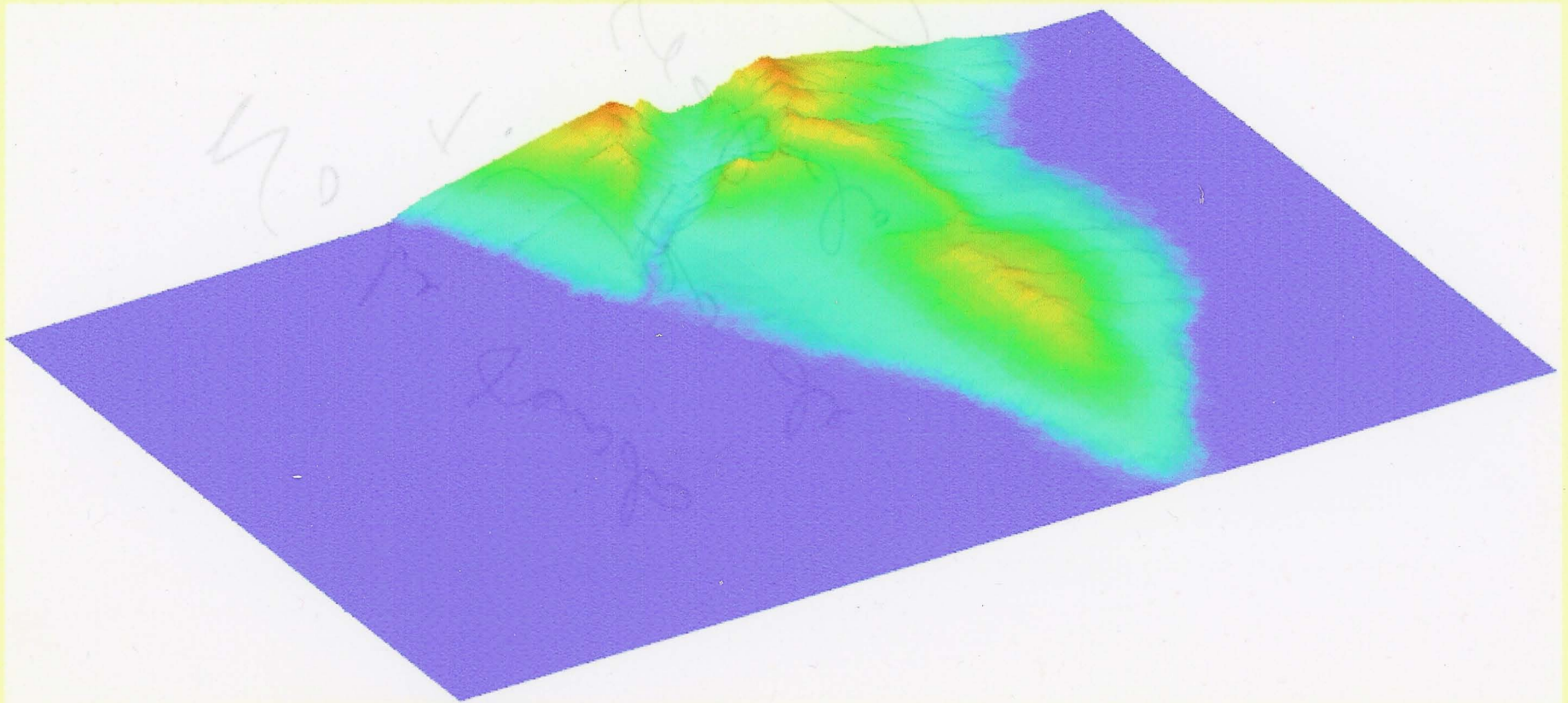




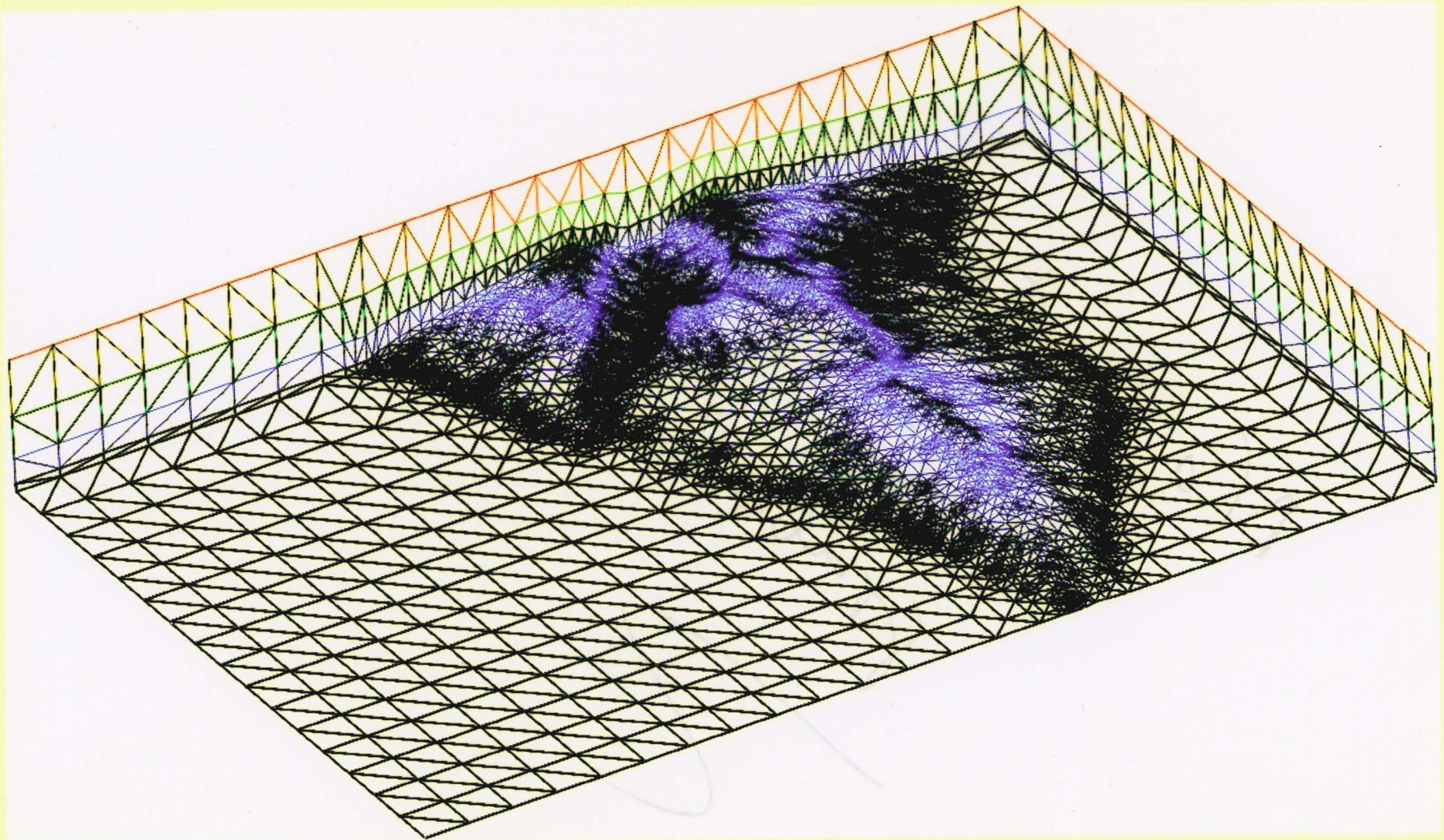
Mesh Generation



Introducción



Introducción



CONTRIBUCIÓN EN GENERACIÓN DE MALLAS NO ESTRUCTURADAS PARA EL MÉTODO DE ELEMENTOS FINITOS

1.- INTRODUCCIÓN.

2.- PROCESO DE REFINAMIENTO/DESREFINAMIENTO.

2.1.- Aspectos generales.

2.2.- Definiciones, propiedades y estructura de datos.

2.3.- El algoritmo de desrefinamiento.

3.- PROBLEMA DE CONVECCIÓN-DIFUSIÓN.

3.1.- Formulación semi-implícita.

3.2.- Estabilidad.

3.3.- Consistencia.

3.4.- Estrategia adaptable.

4.- RESULTADOS NUMÉRICOS EN 2-D.

4.1.- Un problema de convección-difusión.

4.2.- Un problema quasi-estacionario.

4.3.- Un problema estacionario.

4.4.- Aproximación de una imagen bidimensional.

5.- DIAGRAMA DE VORONOI Y TRIANGULACIÓN DE DELAUNAY.

5.1.- Algoritmo incremental para construir la triangulación de Delaunay.

5.2.- Problemas debidos a errores de redondeo.

5.3.- Construcción de $D(X_{i+1})$.

5.3.1.- Búsqueda del núcleo.

5.3.2.- Formación de D_1^i .

5.3.3.- Formación de D_2^i .

6.- APLICACIONES DEL GENERADOR DE MALLAS EN 3-D.

7.- CONCLUSIONES Y LÍNEAS FUTURAS.

8.- REFERENCIAS.

Ecuaciones en Derivadas Parciales

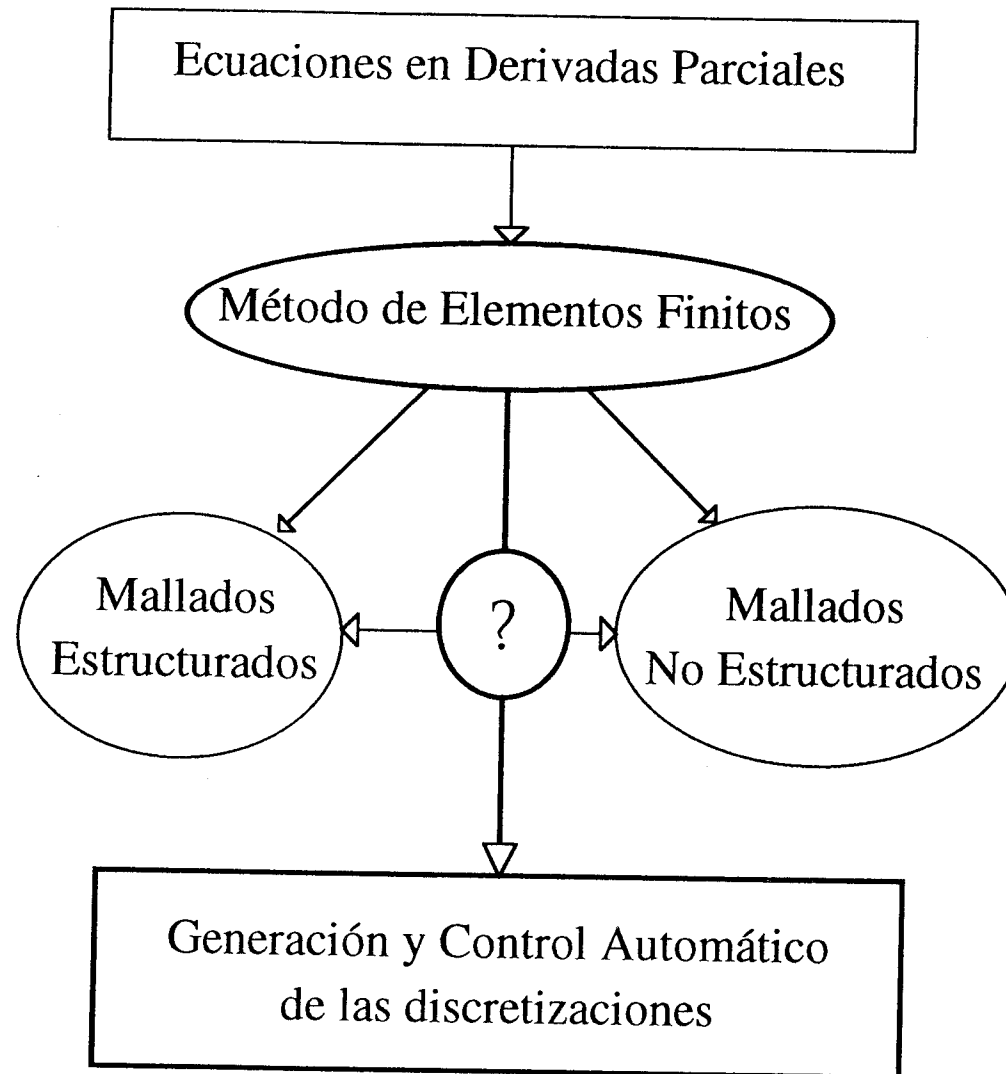
Método de Elementos Finitos

Mallados
Estructurados

?

Mallados
No Estructurados

Generación y Control Automático
de las discretizaciones



Nociones generales

Definición: Malla.

Una malla *conforme* de un poliedro $\Omega \subset E^d$ es un conjunto finito de poliedros $\{\Pi_i, 1 \leq i \leq n\}$ que verifica:

$$1. \quad \Omega = \bigcup_{1 \leq i \leq n} \Pi_i;$$

$$2. \quad \Pi_i \cap \Pi_j = \emptyset \text{ o una } k\text{-cara común a ambos poliedros, } 1 \leq i \neq j \leq n, 0 \leq k \leq d - 1. \quad \square$$

En esta definición la palabra *conforme* se emplea para indicar que no puede haber una intersección entre poliedros que no sea uno de sus vértices, aristas, etc.

Definición: Triangulación.

Una triangulación es una malla en la que todos los poliedros que la forman son símplexes (triángulos en dimensión dos, tetraedros en dimensión tres, etc.).

Características geométricas deseables en una malla

Para la aplicación del M.E.F son deseables las siguientes características geométricas:

- La adaptación a la forma del dominio debe ser lo mejor posible.
- La variación en el tamaño de elementos adyacentes debe ser progresiva.
- La densidad de elementos en determinadas zonas del dominio debe ser mayor (por ejemplo, donde el gradiente de la solución sea elevado, donde la geometría sea más compleja, etc.).
- La forma de los elementos debe ser adecuada. Por ejemplo, en el caso de una triangulación, los símlices deben ser lo más regulares posible; algunas medidas de la regularidad o *calidad* de un 3-símlice (tetraedro), son: $\frac{vol(b_k)}{vol(K)}$, $\frac{vol(b_k)}{vol(B_k)}$, $\frac{rad(b_k)}{diam(K)}$, ..., donde b_k y B_k son las esferas inscrita y circunscrita al tetraedro K . En ningún caso es admisible que existan elementos *degenerados*, es decir, cuya medida de calidad sea nula o *casi* nula.

Clasificación de las mallas

Estructuradas

La conexión entre elementos obedece a un patrón fijo; el número de nodos vecinos a uno dado es siempre el mismo.

Ventajas: Son más simples, más convenientes para su uso en el método de diferencias finitas y requieren menos cantidad de memoria para su almacenamiento en el ordenador.

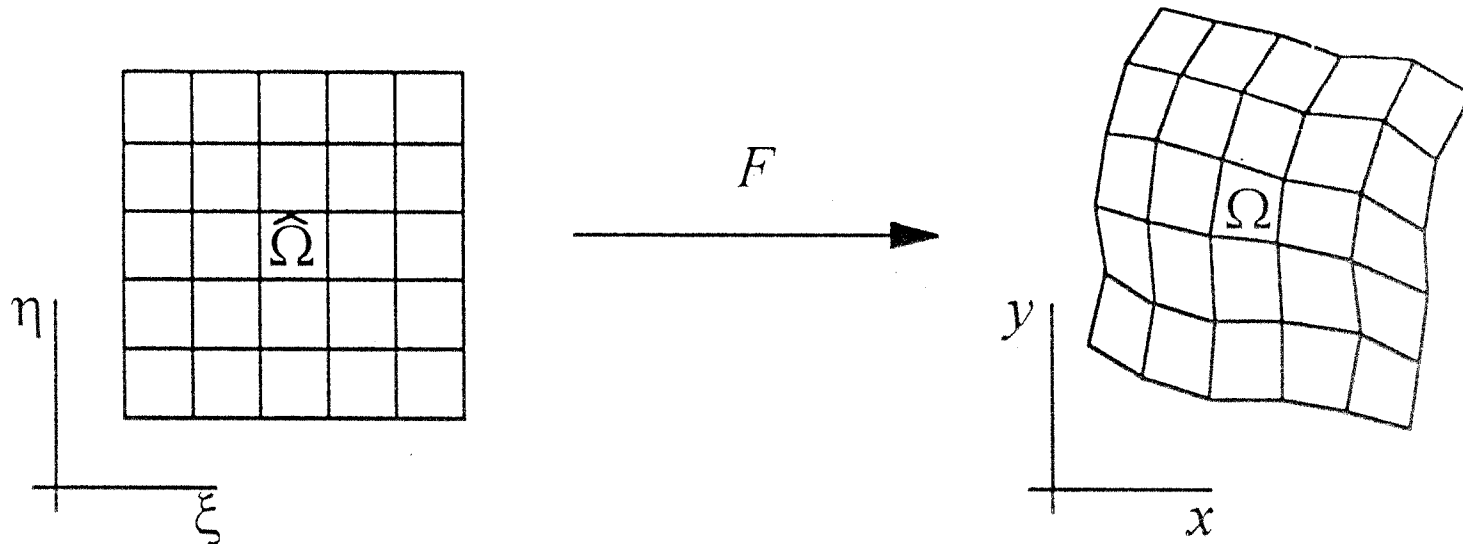
No estructuradas

No hay un patrón fijo de conexión entre elementos; el número de nodos vecinos a uno dado es variable.

Ventajas: Tienen mucha más flexibilidad para adaptarse a dominios con geometrías complicadas, posibilitan el *refinamiento* y *desrefinamiento* local en mayor medida que las estructuradas.

Conceptos generales sobre mallas estructuradas

Los procedimientos de generación de mallas estructuradas están basados en una transformación de coordenadas que establece una aplicación entre una malla canónica de referencia (una malla de un triángulo, un cuadrilátero...., en 2-D, o un tetraedro, un hexaedro..., en 3-D) y la correspondiente malla del dominio real. La transformación $F: \hat{\Omega} \rightarrow \Omega$ debe satisfacer los siguientes requisitos generales:



Algunos procedimientos de generación de mallas estructuradas

□ Métodos algebraicos

La transformación de coordenadas se establece explícitamente *a priori*.

□ Métodos basados en la resolución de un sistema de ecuaciones diferenciales

En este caso la transformación se encuentra al resolver un *sistema generador* de ecuaciones diferenciales en derivadas parciales con unas condiciones de contorno apropiadas. La resolución del sistema permite encontrar las coordenadas de un punto $x(\xi, \eta)$, $y(\xi, \eta)$ del espacio físico en función de las coordenadas ξ , η del espacio lógico; las fronteras del dominio físico corresponden a líneas $\xi = cte$ y

$\eta = cte$. Ejemplo, *sistema generador elíptico*:
$$\begin{cases} \xi_{xx} + \xi_{yy} = P(\xi, \eta) \\ \eta_{xx} + \eta_{yy} = Q(\xi, \eta) \end{cases}$$
 Las funciones de control $P(\xi, \eta)$ y

$Q(\xi, \eta)$ se escogen de manera que nodos de la malla se concentren en las regiones adecuadas.

Algunos procedimientos de generación de mallas no estructuradas

Triangulación de Delaunay.

Triangulación de óptima calidad en 2-D para un conjunto de puntos dado.

Avance Frontal.

Construye la malla a partir de la triangulación de la frontera del dominio.

Mallas Encajadas.

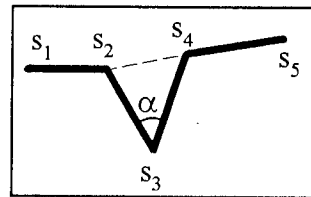
Necesita una triangulación inicial del dominio (malla base).

Muy adecuada para el refinamiento/desrefinamiento local.

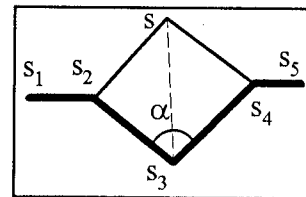
Facilidad en la aplicación del método multimalla.

Generación de mallas no estructuradas. Avance frontal en 2-D

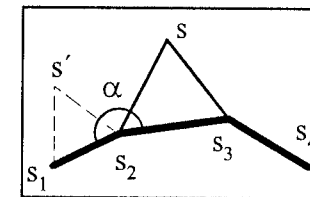
- El frente inicial F se define como el conjunto de segmentos de la frontera del dominio.
- Creación de nuevos elementos a partir del frente actual. Dependiendo del valor del ángulo α , los nuevos triángulos y puntos se crean atendiendo a uno de los tres patrones siguientes:



$$\alpha < \pi/2$$



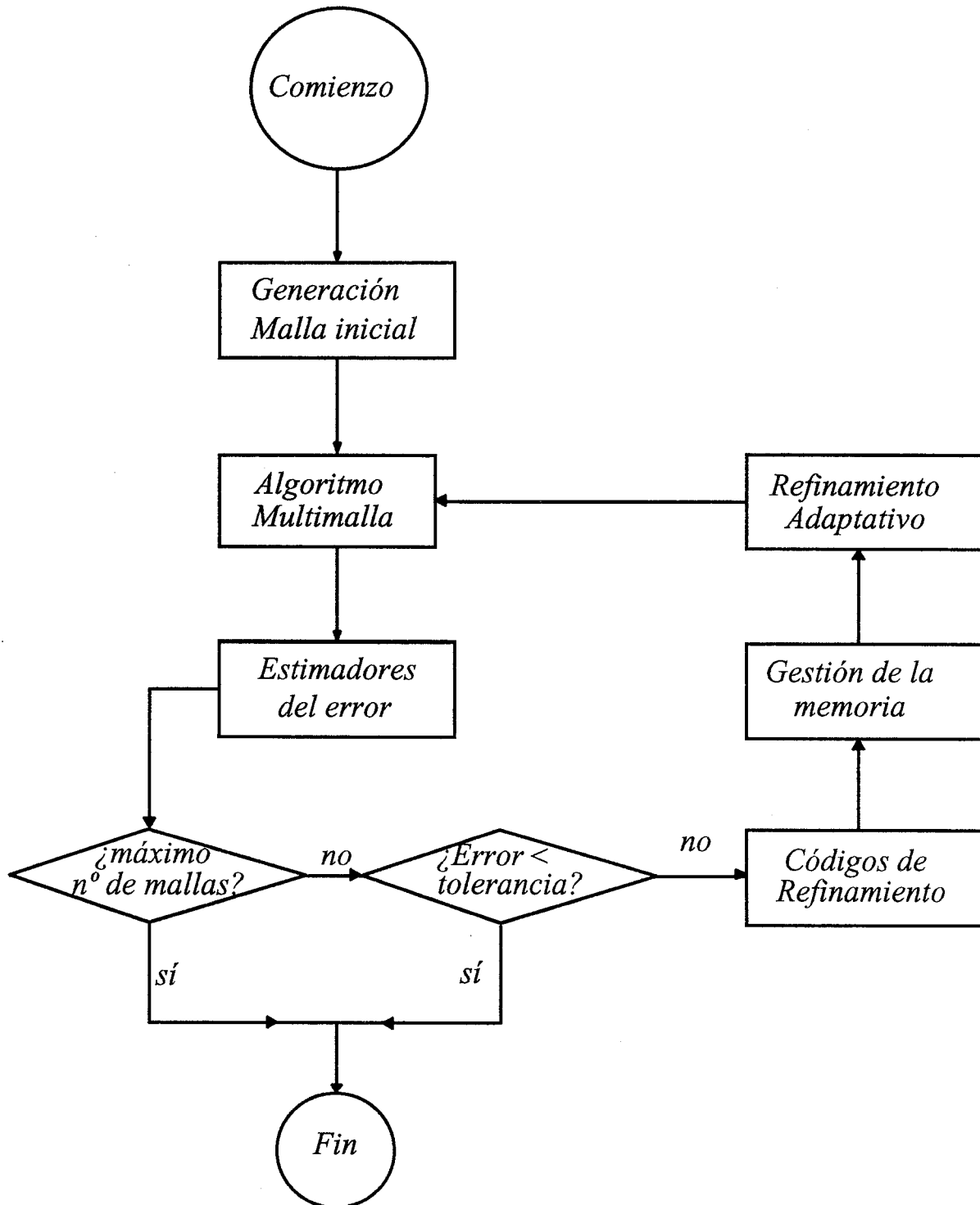
$$\pi/2 \leq \alpha \leq 2\pi/3$$



$$2\pi/3 < \alpha$$

- Actualización del frente. Un nuevo frente se forma suprimiendo los segmentos del presente frente *encerrados* por los nuevos triángulos y añadiendo los *no encerrados* que restan.
- El proceso se repite hasta que el frente se reduce al conjunto vacío.

Código Adaptable



Estrategia Adaptable

- ❑ Indicador de error para el refinamiento local:

$$\varepsilon_i = h_i^q \left| \vec{\nabla} u_h \right| \quad \text{donde consideramos } q = 1 \text{ ó } 2$$

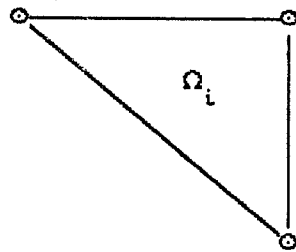
- ❑ Refinamos aquellos elementos en los que:

$$\varepsilon_i \geq \gamma \underset{1 \leq j \leq n}{\text{Max}}(\varepsilon_j) \quad \text{donde } \gamma \in [0,1]$$

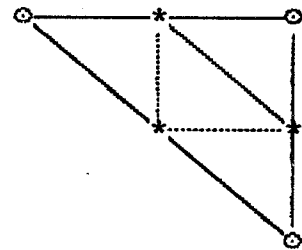
* ESTRATEGIAS DE REFINAMIENTO *

REFINAMIENTO DE UN ELEMENTO

* OPCION 1:

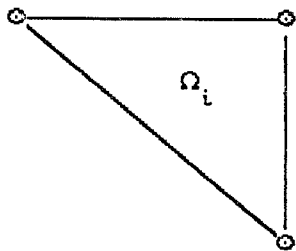


M_n

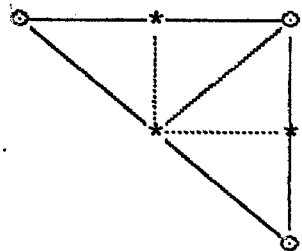


M_{n+1}

* OPCION 2:



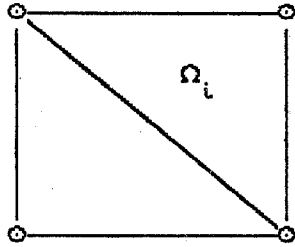
M_n



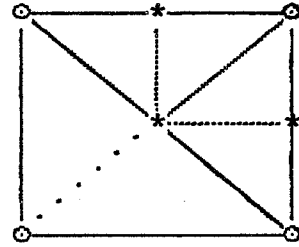
M_{n+1}

TIPOS DE REFINAMIENTOS ADICIONALES

a) Caso de nodo libre en lado mayor:

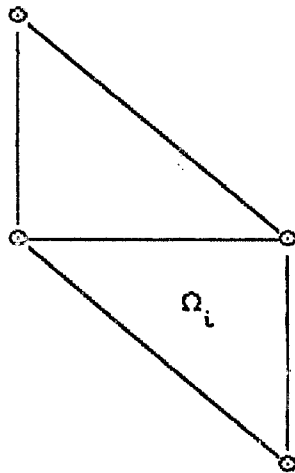


M_n

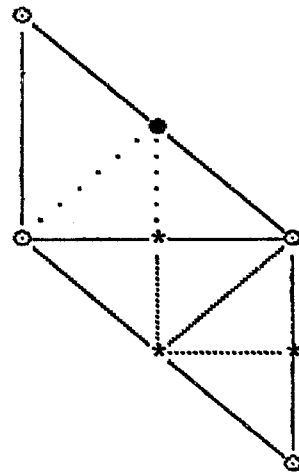


M_{n+1}

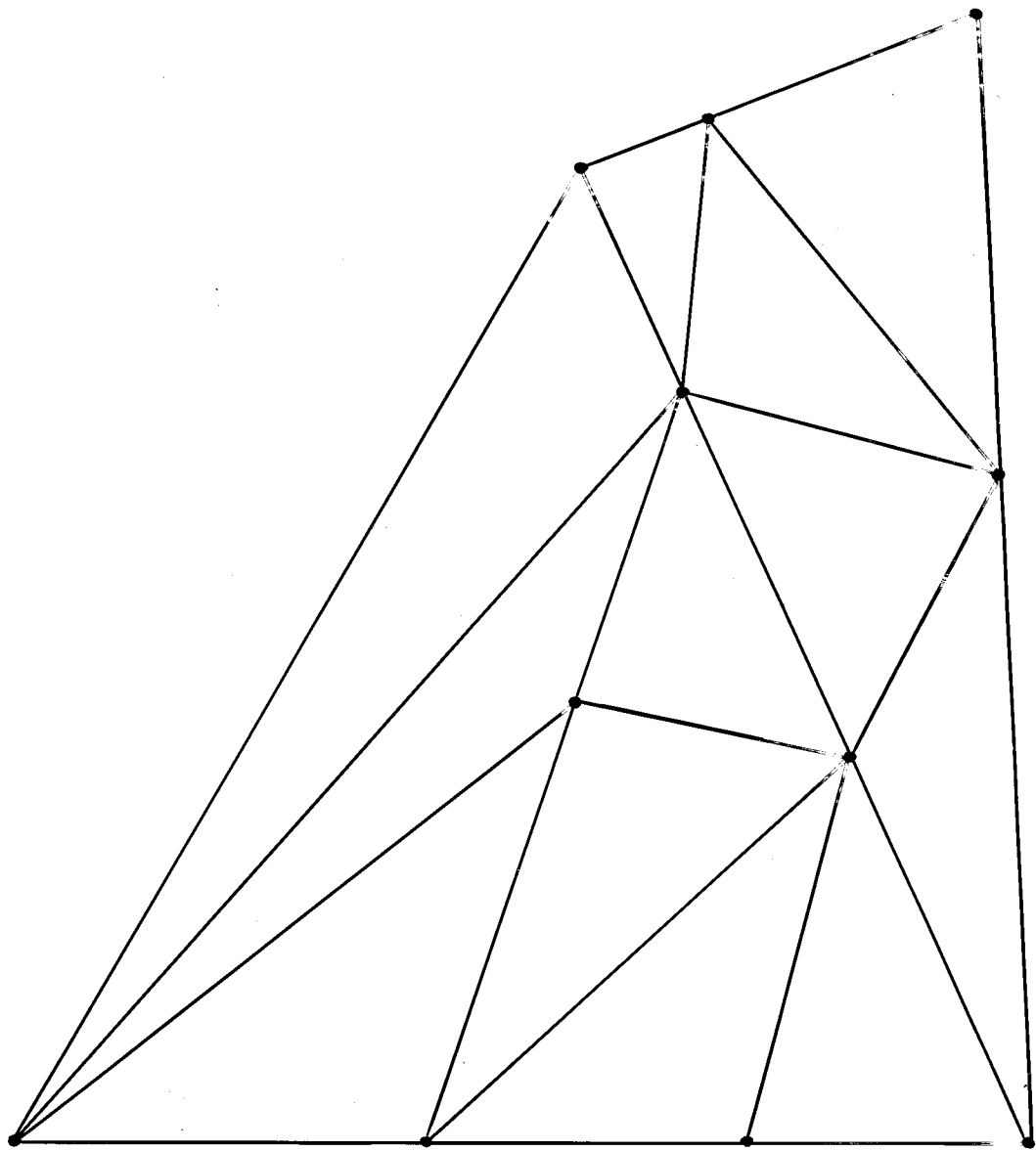
b) Caso de nodo libre en un lado distinto al mayor:



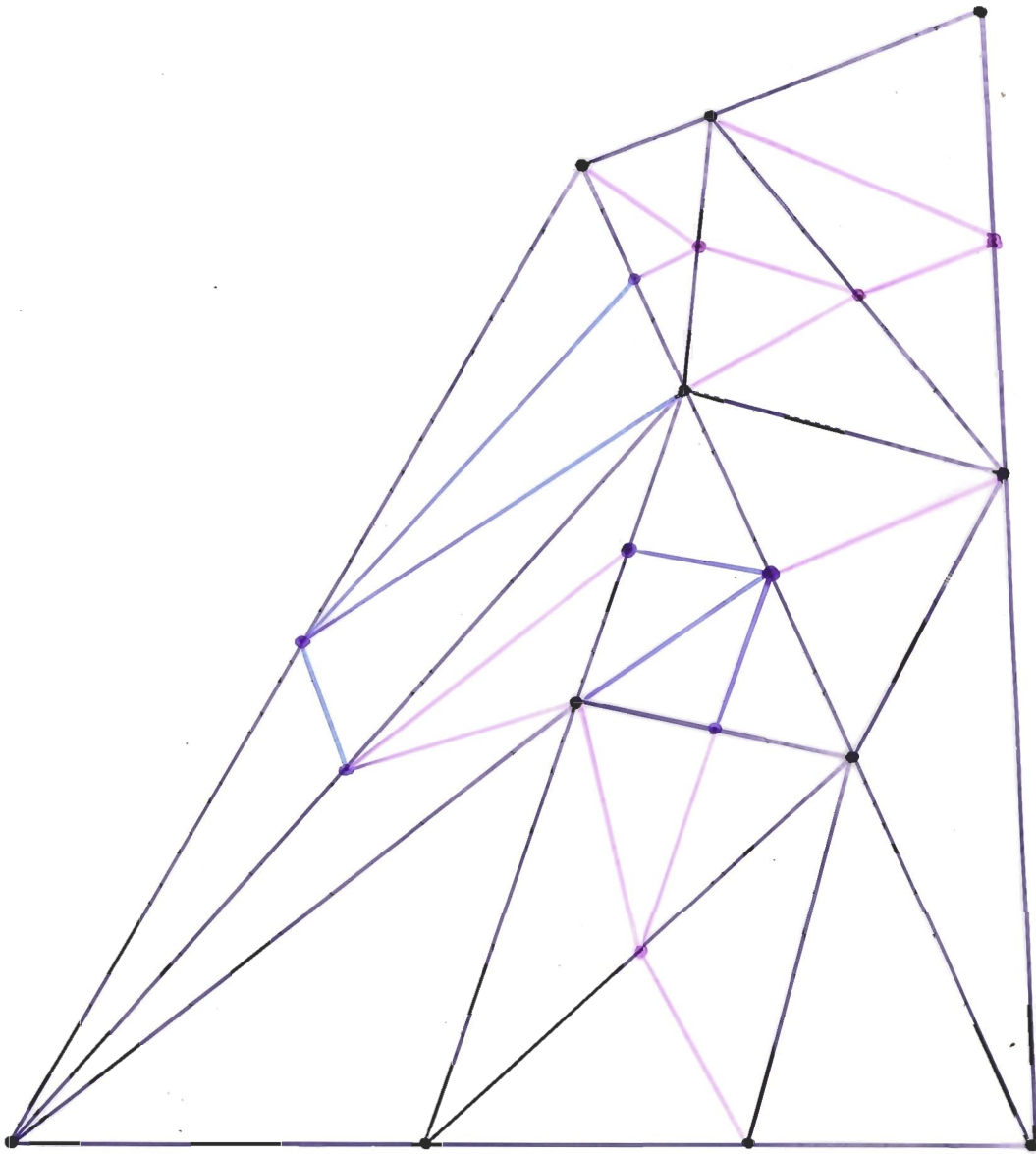
M_n



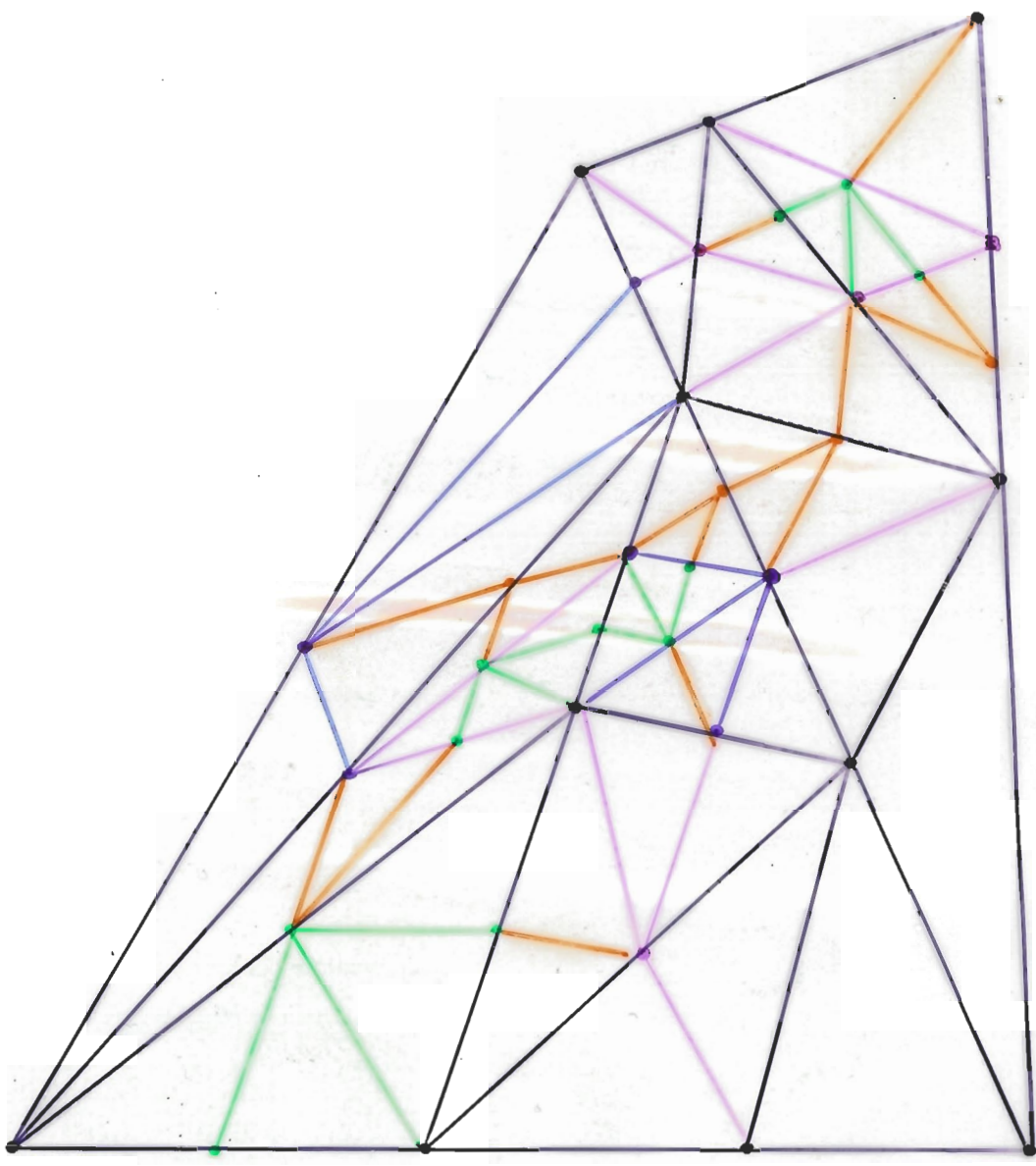
M_{n+1}



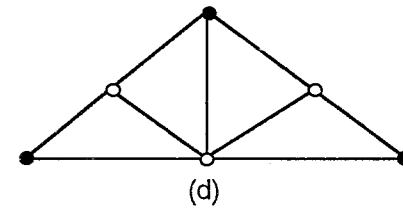
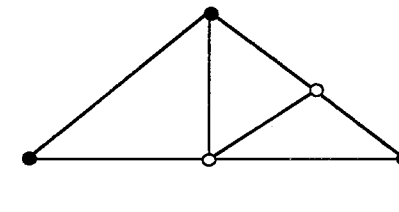
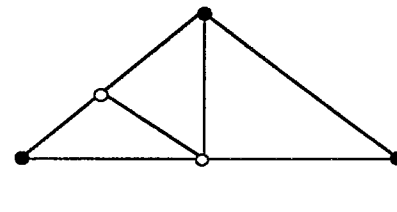
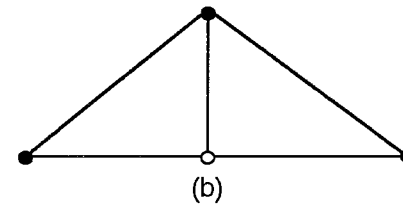
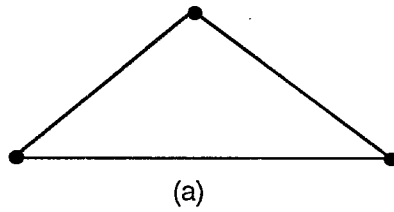
* MALLA 1 *



* МАЛЛА 21 *

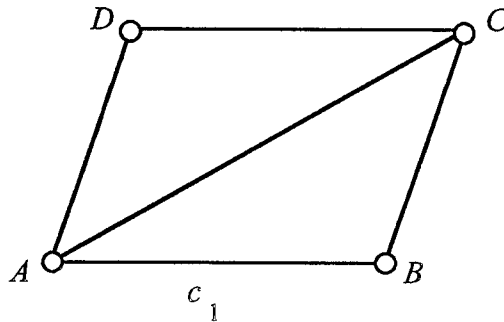


División de un Triángulo

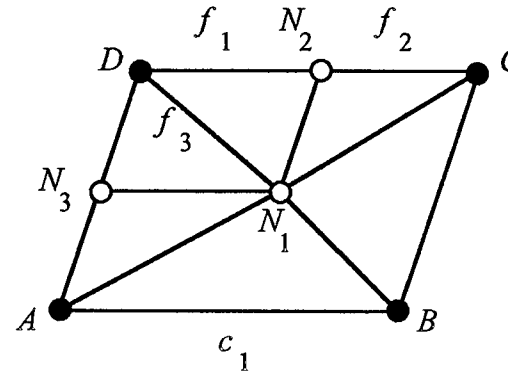


Definiciones y Propiedades

Nivel τ_{j-1}

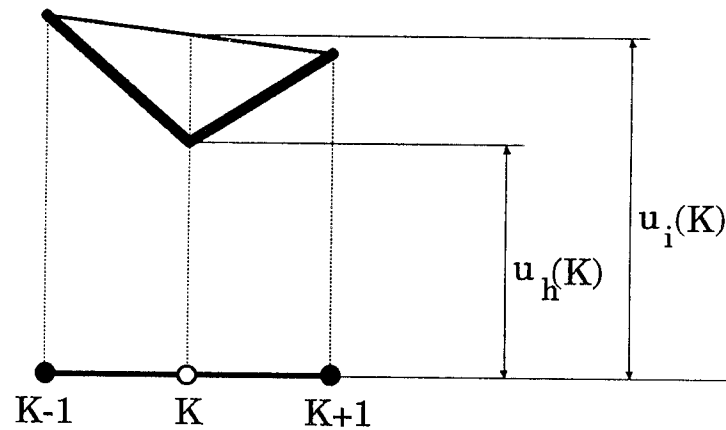
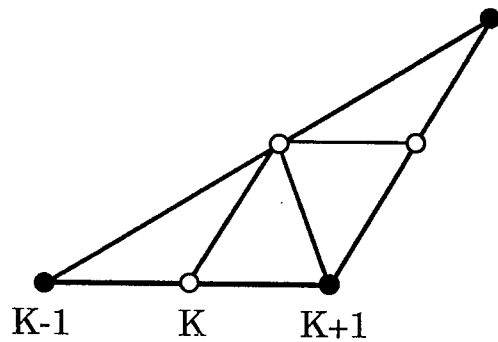


Nivel τ_j



- Nodos, aristas y elementos *propios* y *heredados*.
- Historia o Genealogía de la malla:
 - Aristas y elementos *padres* e *hijos*.
 - Aristas *externas* e *internas*.
- ⇒ Cualquier elemento propio de τ_{j-1} es heredado en τ_j , o tiene sus hijos en τ_j .
- ⇒ Si un elemento no tiene hijos, entonces pertenece a la malla más fina.
- ⇒ Sólo aquellos elementos sin sucesores pueden ser eliminados.

Condición de Desrefinamiento



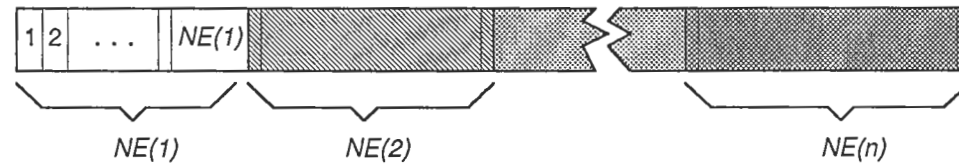
En principio, el nodo propio K puede ser eliminado si:

$$|u_h(K) - u_i(K)| \leq \varepsilon \quad \text{ó} \quad |u_h(K) - u_i(K)| \leq \varepsilon |u_h(K)|$$

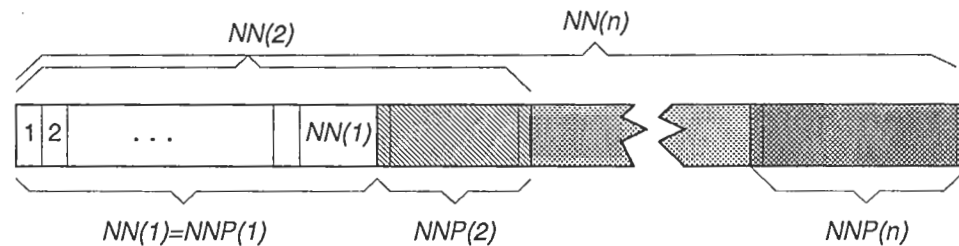
Estructura de Datos

□ VECTORES DE ESTRUCTURA:

(a) Vector para elementos, *IMELEM*; análogamente para aristas, *IMFACE*.



(b) Vector para nodos *IMNODE(1:NUMP)*.



□ VECTORES DE GENEALOGÍA:

(a) Vector para aristas: *IR[1:3,1:NUMF]* (b) Vector para elementos: *IXH[1:6,1:NUMEL]*

□ VECTORES DE NIVEL o INDICADORES DE DESREFINAMIENTO:

NODES(1:NUMN), *NFACES(1:NUMF)* y *NELES(1:NUMEL)*.

□ VECTORES DEL SACO: *NNSAC(*)*, *NFSAC(*)* y *NESAC(*)*

□ ARISTA-ENTORNO: *IEX(1:NUMN)*

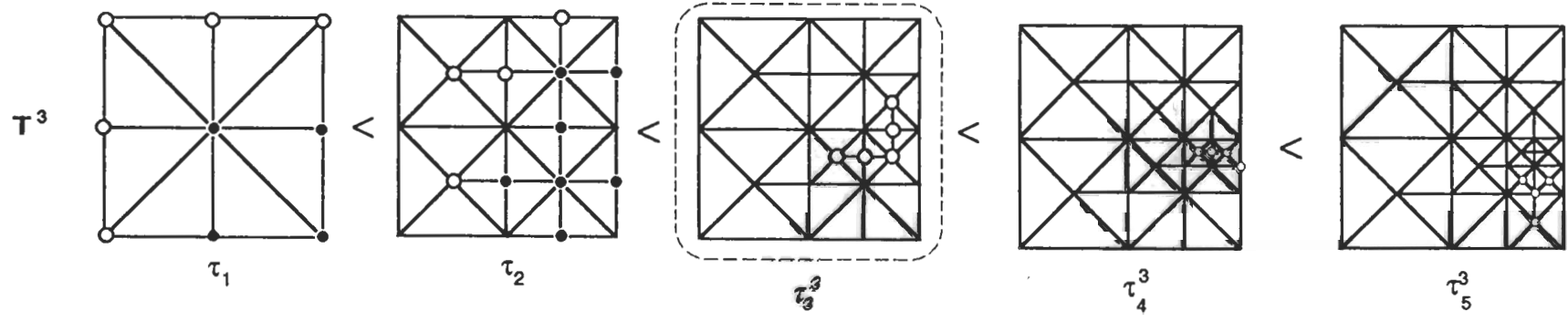
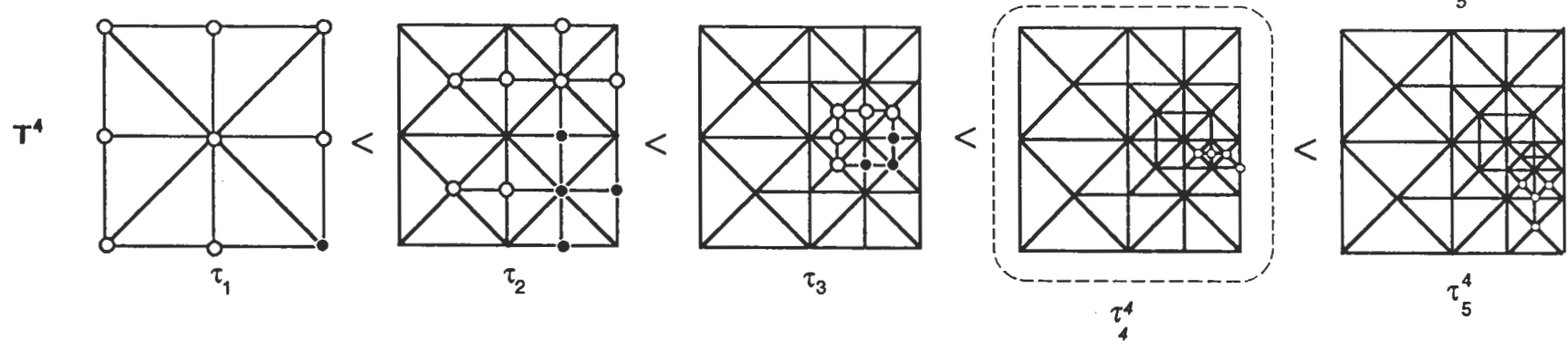
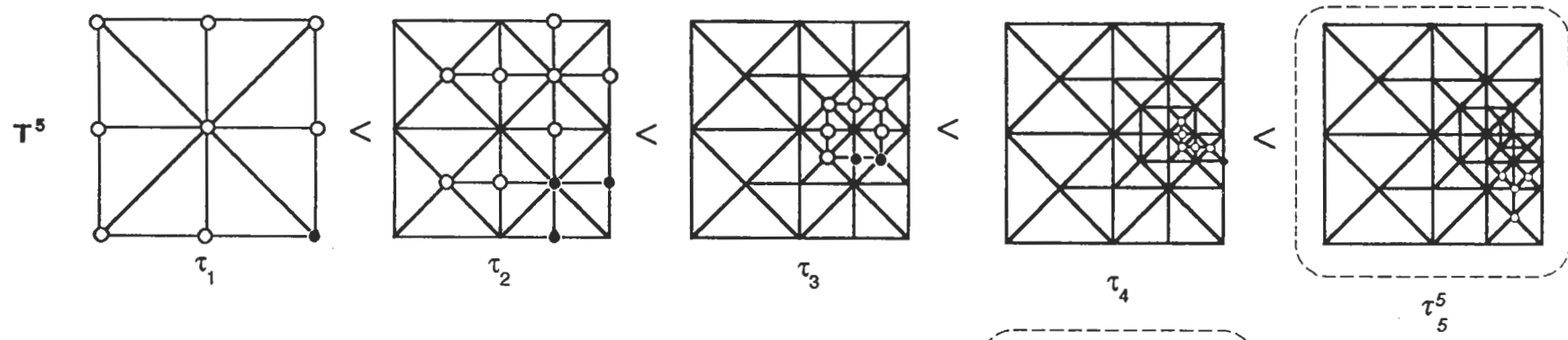
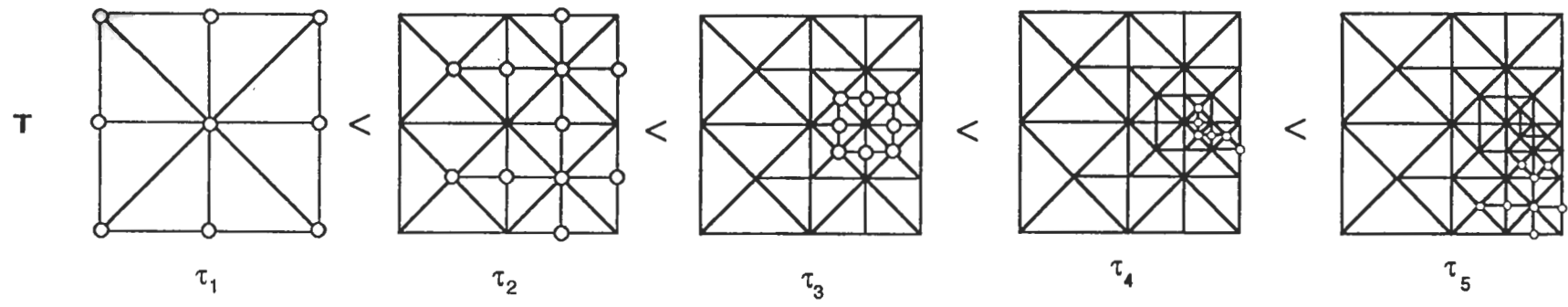
EL ALGORITMO DE DESREFINAMIENTO

ENTRADA: Secuencia $T = \{ \tau_1 < \tau_2 < \dots < \tau_n \}$

Bucle en niveles de T ; Para $j = n$ hasta 2, hace:

1. Se recorren los nodos propios de τ_j y se evalúa la *condición de desrefinamiento*.
2. Se asegura la *conformidad* de la malla que se está creando.
- 3.a. Si algún nodo propio de τ_j debe ser eliminado,
 - 3.a.1. Si algún nodo propio de τ_j debe permanecer,
Se definen *Nuevas conexiones nodales* para el nuevo nivel τ_j^j .
 - 3.a.2. En caso contrario,
Se elimina el *nivel j* de los vectores de estructura.
- 3.b. Si todos los nodos propios de τ_j permanecen, no se modifica el nivel: $\tau_j^j = \tau_j$.
4. Los *cambios se heredan* a los siguientes niveles de malla.
5. Se obtiene una nueva secuencia $T^j = \{ \tau_1 < \tau_2 < \dots < \tau_{j-1} < \tau_j^j < \dots < \tau_n^j \}$.

SALIDA: Secuencia desrefinada $T' = \{ \tau_1 < \tau_2' < \dots < \tau_m' \}$



Conforming procedure

INPUT ($\tau_j, \tau_{j-1}, \text{Nodes}$)

While *Conformity* must be assured:

For each $t \in \tau_{j-1}$:

 If t is 1/2-non-conforming:

 Change the derefinement indicator for the node P of its
 longest side.

 Assure *conformity*.

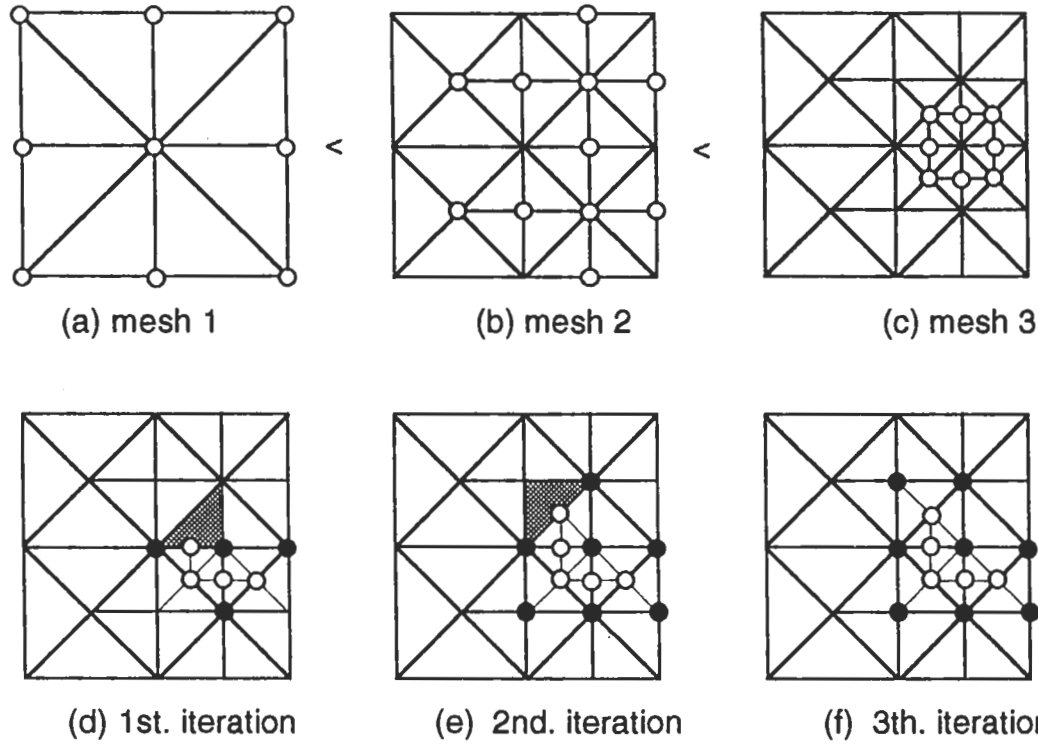
 End if.

End for.

End While.

OUTPUT ($\tau_j, \tau_{j-1}, \text{Nodes}$)

CONFORMITY OF THE MESH



- (a),(b) and (c) sequence of nested meshes. Proper nodes are pointing out.
- (d), (e) and (f) iterations to obtain the conformity (with the derefinement indicators) of the arising mesh τ_3^3

New Conforming procedure

INPUT ($\tau_j, \tau_{j-1}, \text{Nodes}$)

For each proper node $N \in \tau_j$:

Let c be the surrounding edge of N .

For each neighboring element t of c do:

If t is 1/2-non-conforming:

* Change the derefinement vector Nodes for the node P of its longest side.

* Change the derefinement vector Nodes for some nodes of t .

End if.

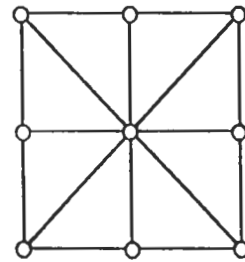
End for.

End for.

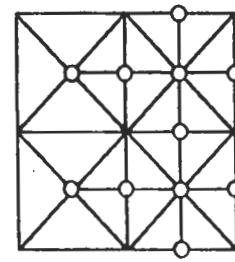
OUTPUT ($\tau_j, \tau_{j-1}, \text{Nodes}$)

CONFORMITY OF THE ARISING NEW MESH

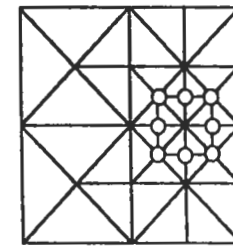
(by means of the new procedure)



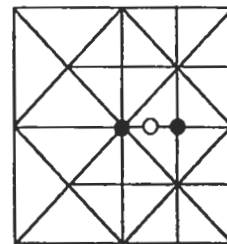
(a) mesh 1



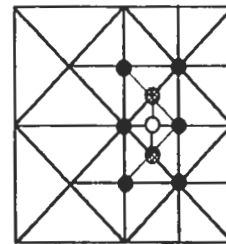
(b) mesh 2



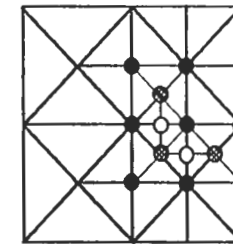
(c) mesh 3



(d) one node remains



(e) Conformity



(f) other node

- (a),(b) and (c) sequence of nested meshes. Proper nodes are pointing out.
- (d), (e) and (f) shaded nodes mean proper nodes that will stay according to the conformity of the mesh.

EFFICIENCY AND COMPLEXITY

Loop in levels of mesh: n

1.- Derefinement condition: $O(NNP(j))$

2.- Assure conformity: $O(NNP(j))$

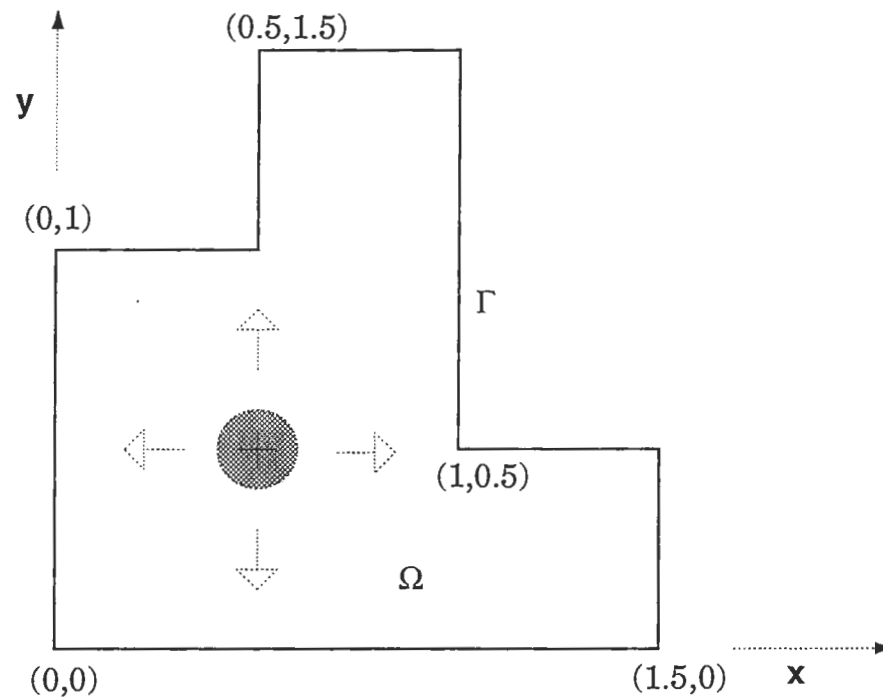
3.- Definition of the new nodal connections: $O(NN)$

4.- Changes are inherited: $O(n \cdot NN)$

Total:

$$O(NN + NN + n \cdot NN + n^2 \cdot NN) = O(NN + n \cdot NN + n^2 \cdot NN) = O(n^2 \cdot NN)$$

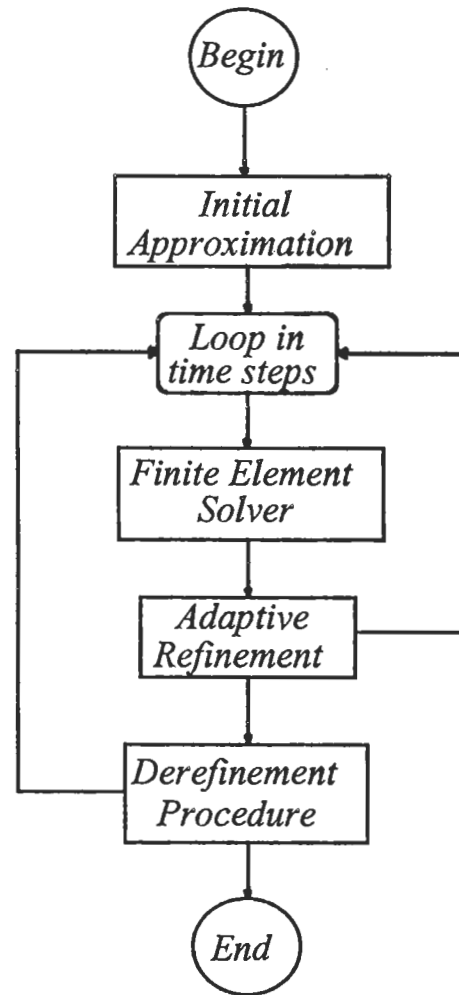
A QUASI-EVOLUTIVE PROBLEM

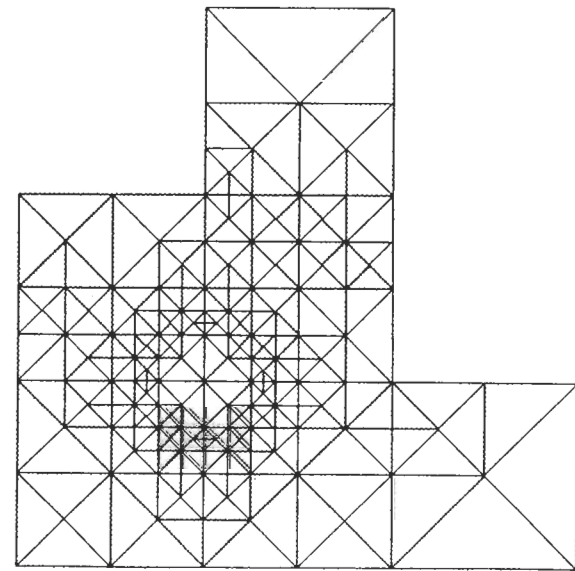
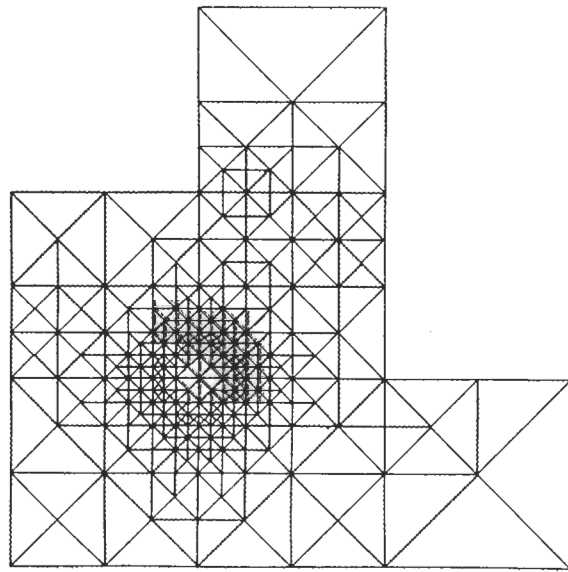
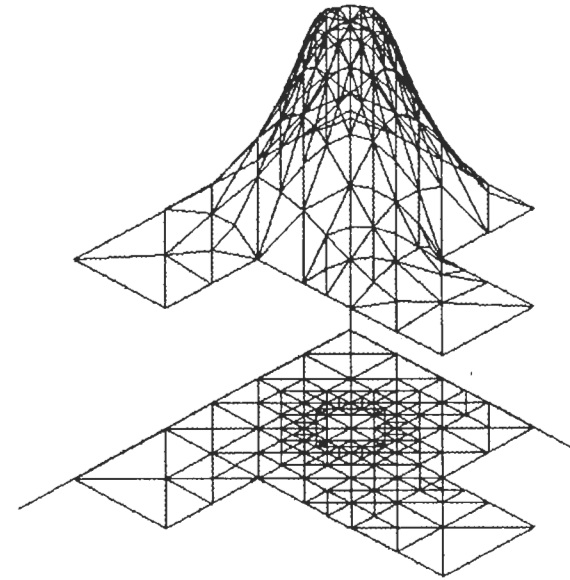
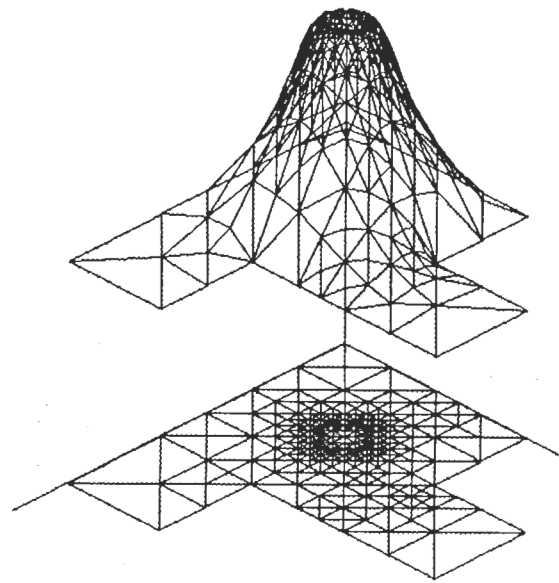


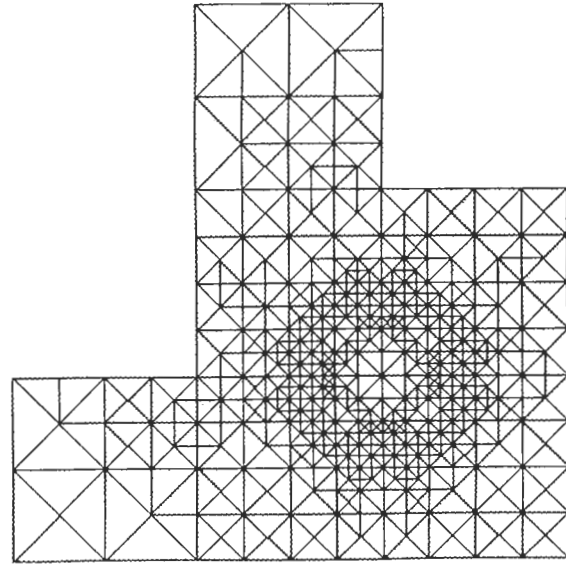
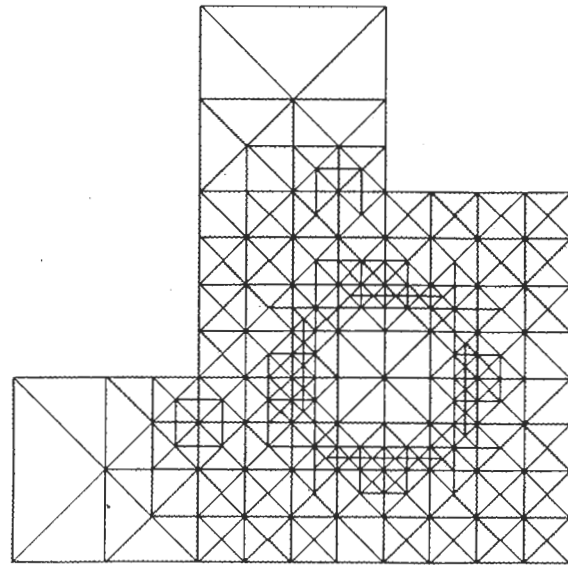
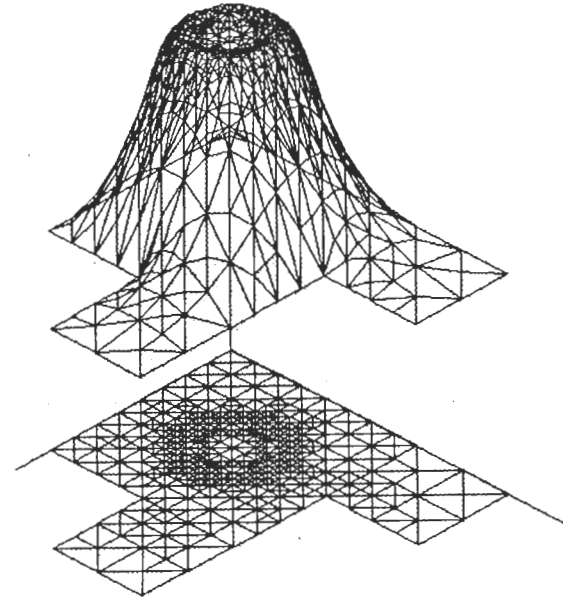
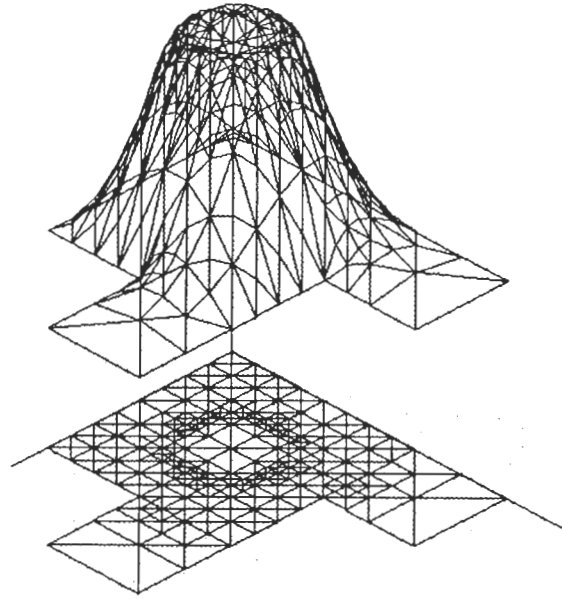
$$\left. \begin{array}{l} \Delta u = f \text{ in } \Omega \\ u = 1 \text{ in } \Gamma = \partial\Omega \end{array} \right\} \text{ where } f(x,y) = \begin{cases} 0 & \text{if } d > r_1(t) \\ 50 & \text{if } r_2(t) < d < r_1(t) \\ -5 & \text{if } d < r_2(t) \end{cases} \text{ and } \begin{cases} d(x,y) = [(x-0.5)^2 + (y-0.5)^2]^{\frac{1}{2}} \\ r_1(t) = 0.1(1+t) \\ r_2(t) = 0.1t \end{cases}$$

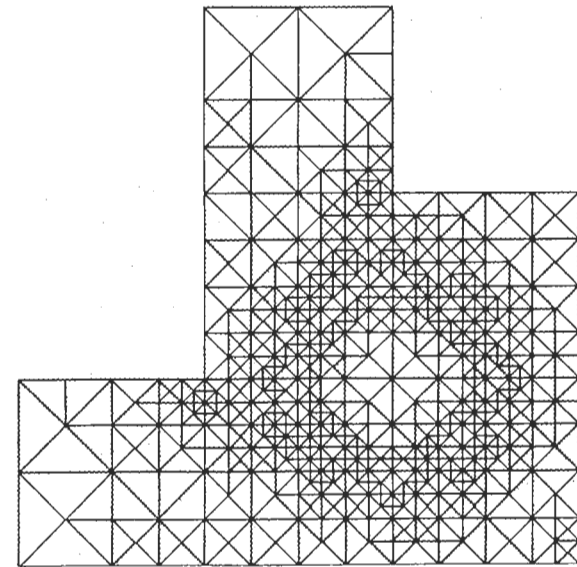
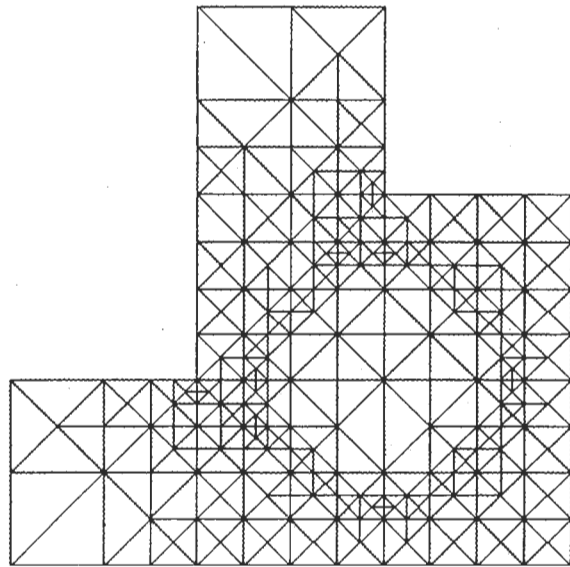
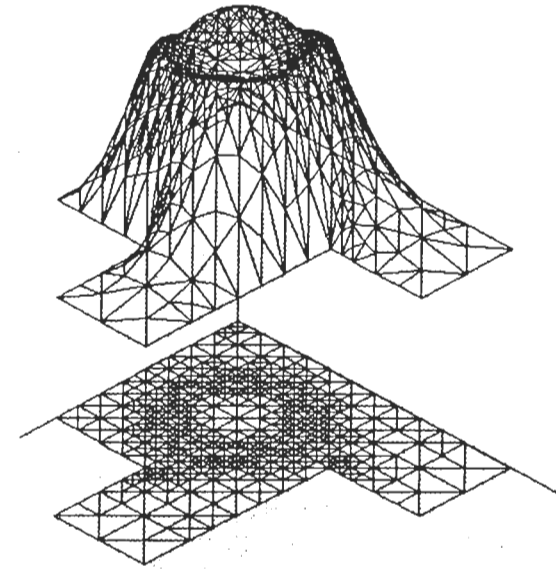
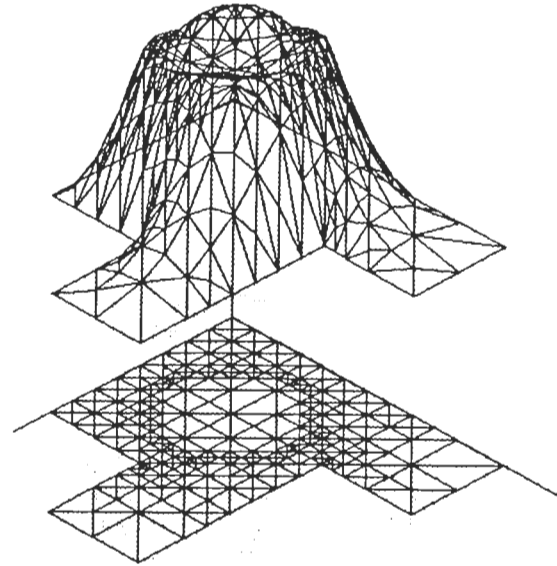
THE READAPTIVE PROCESS

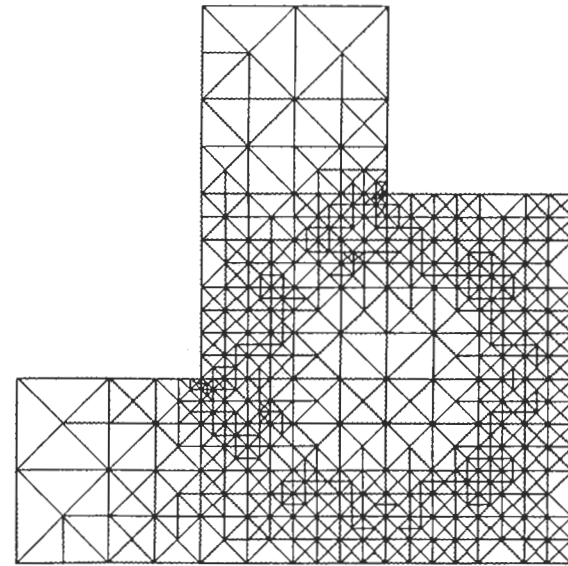
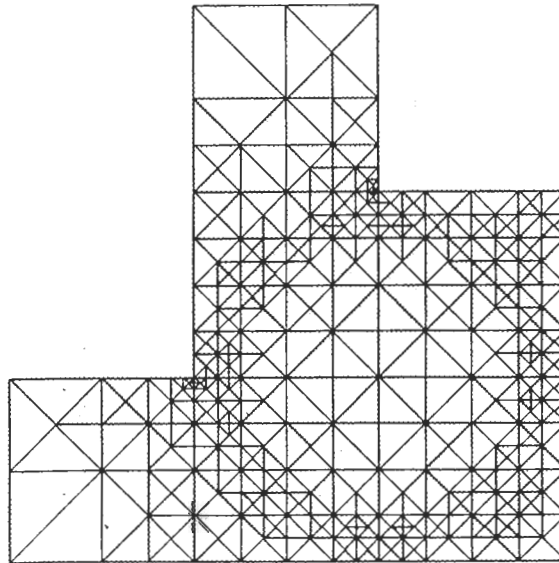
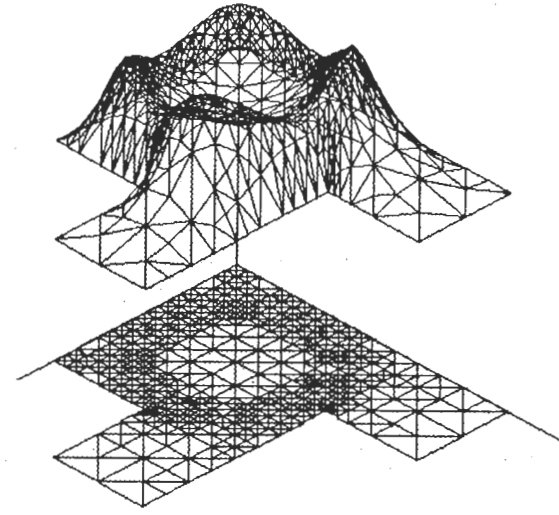
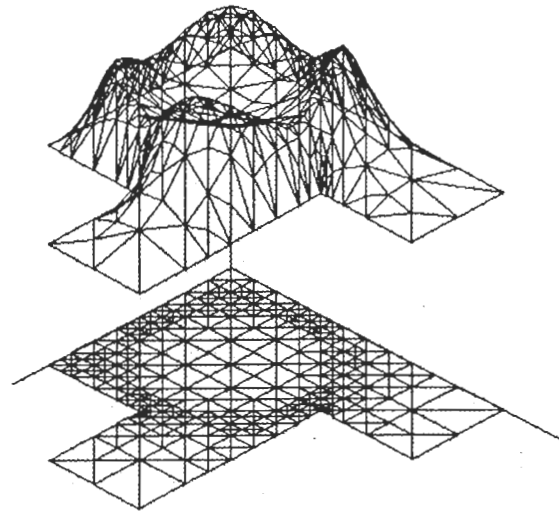
General Aspects

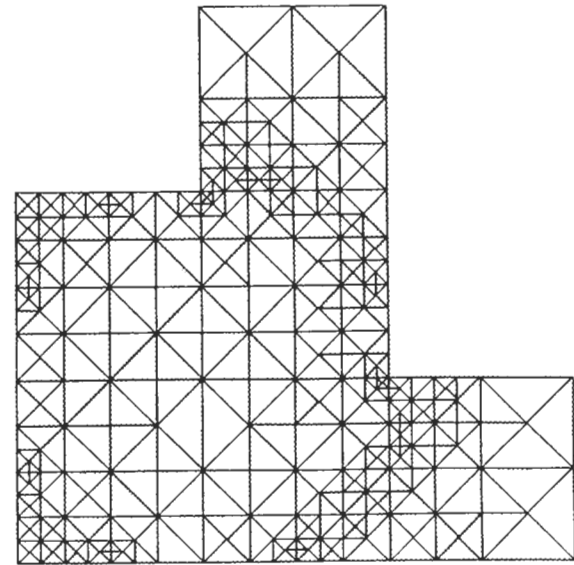
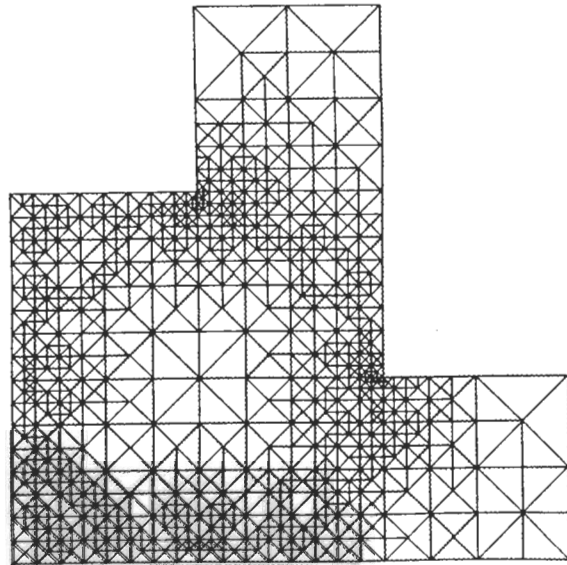
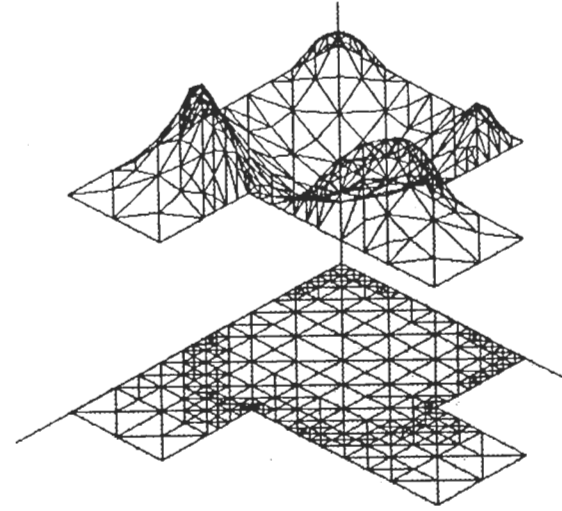
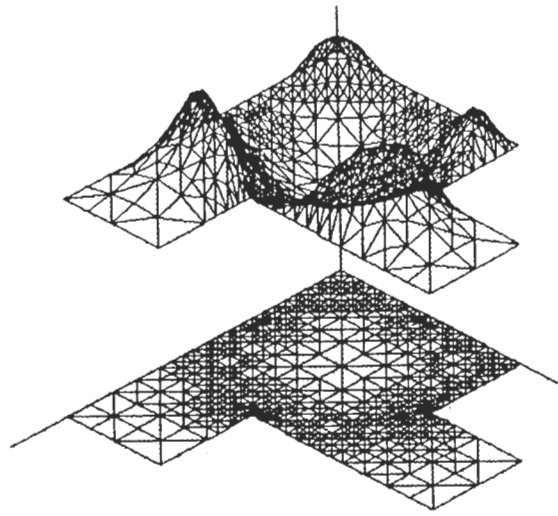


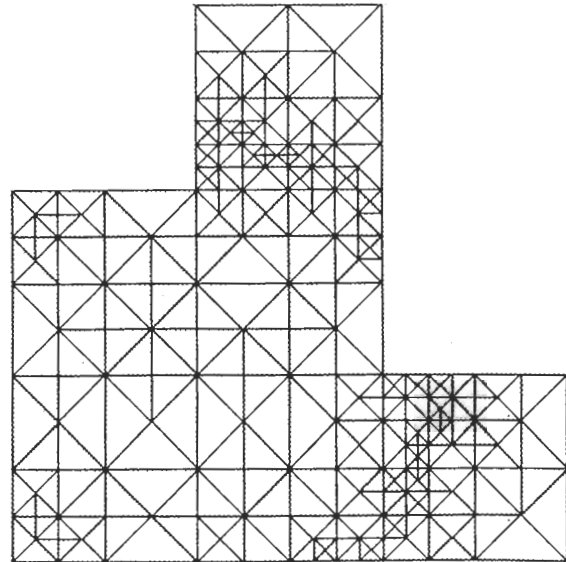
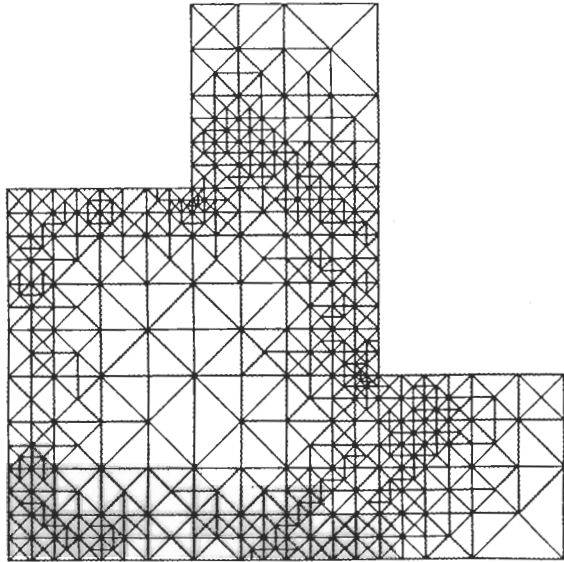
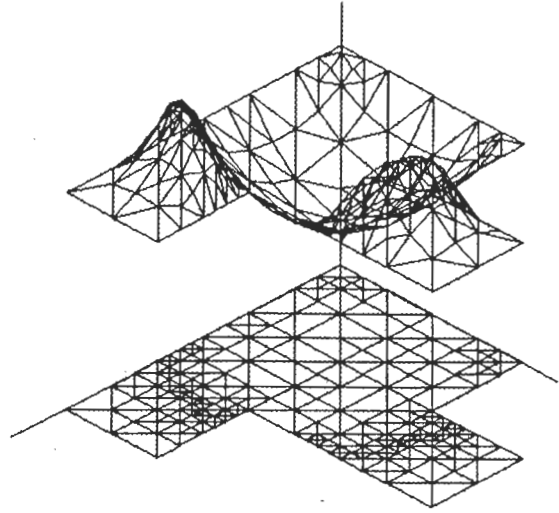
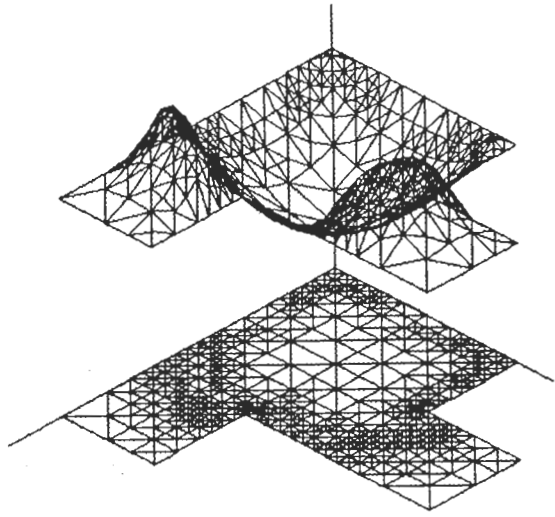


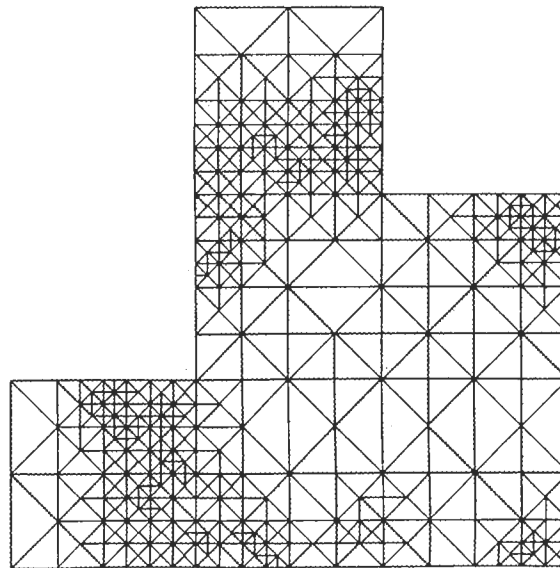
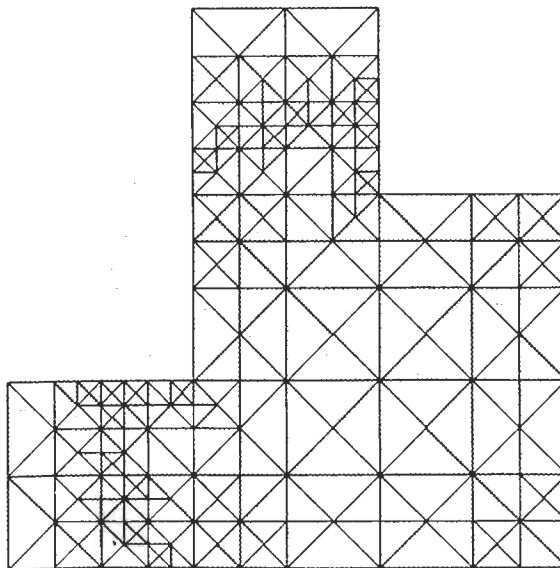
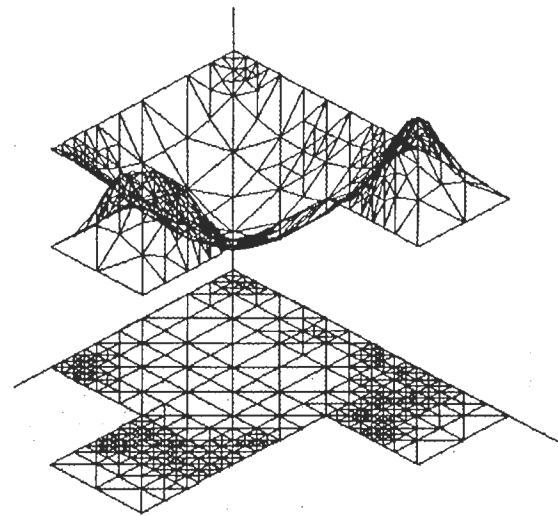
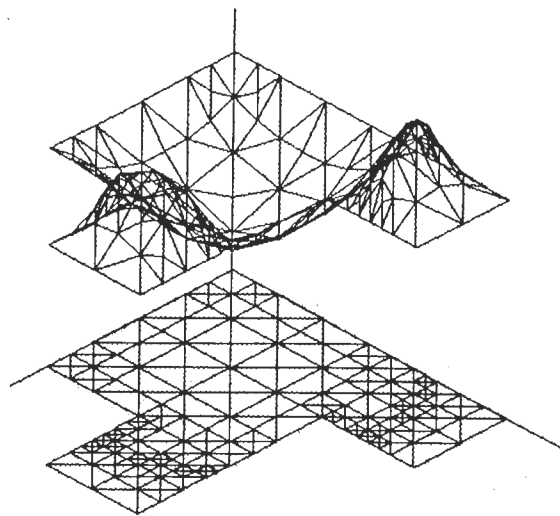


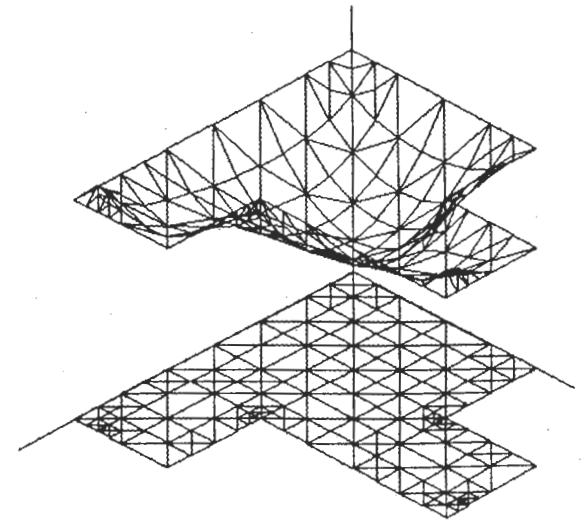
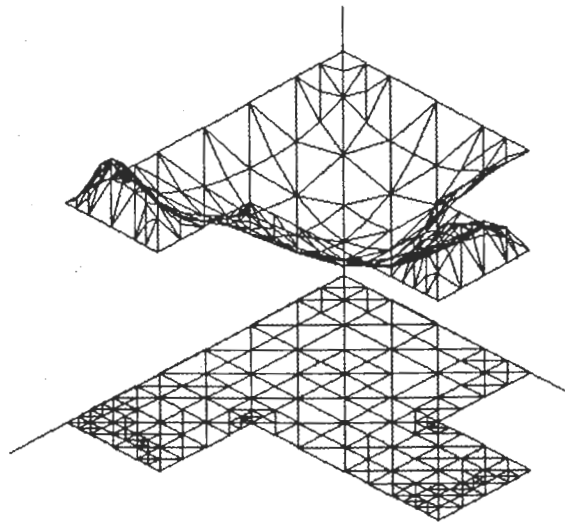
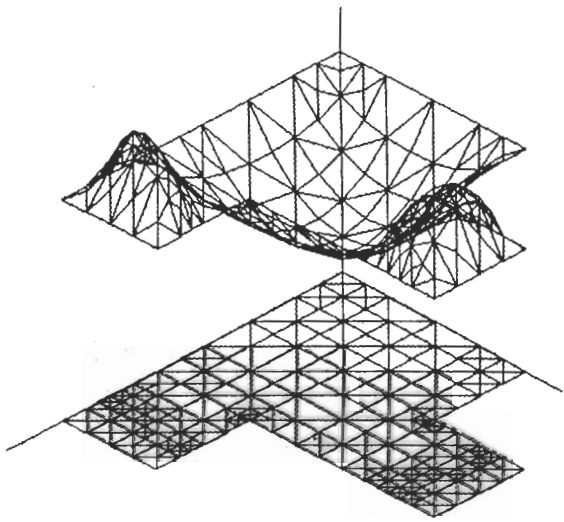
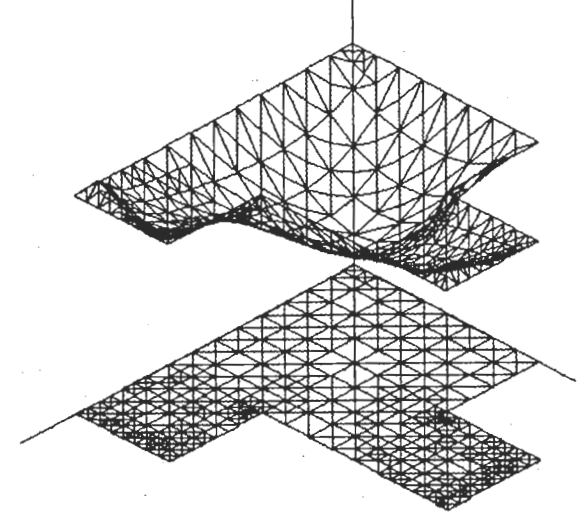
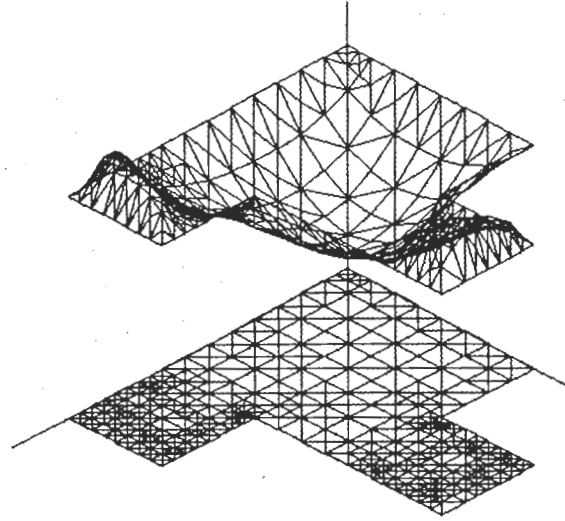
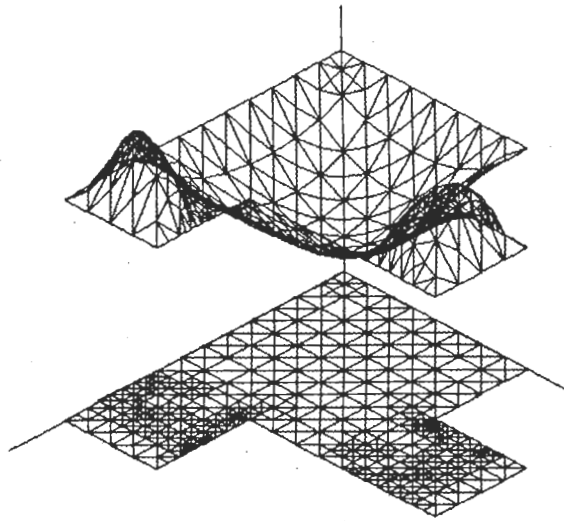


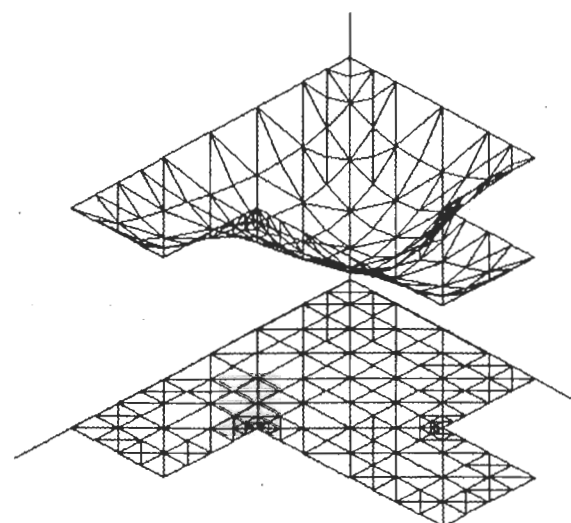
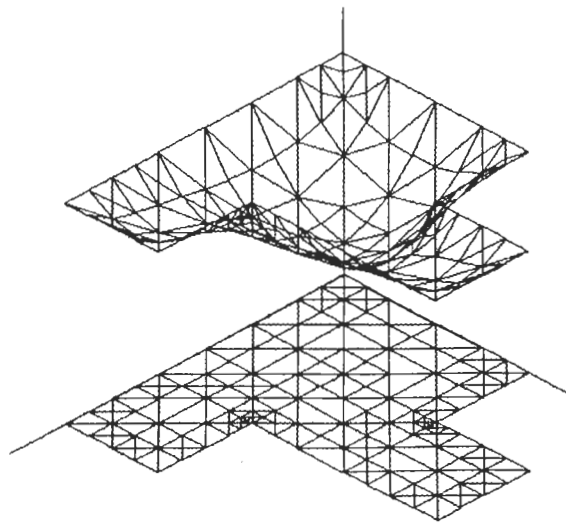
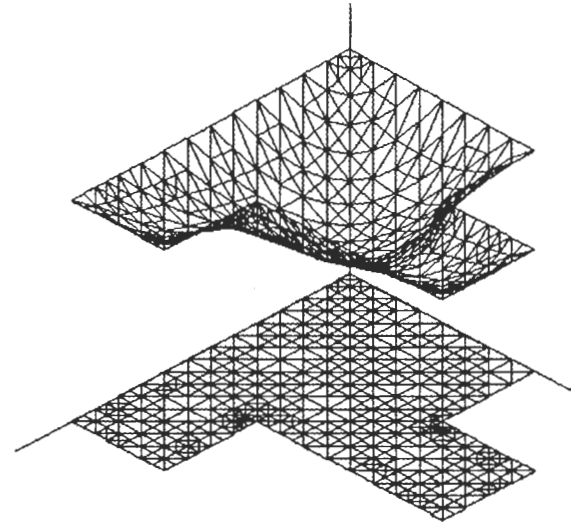
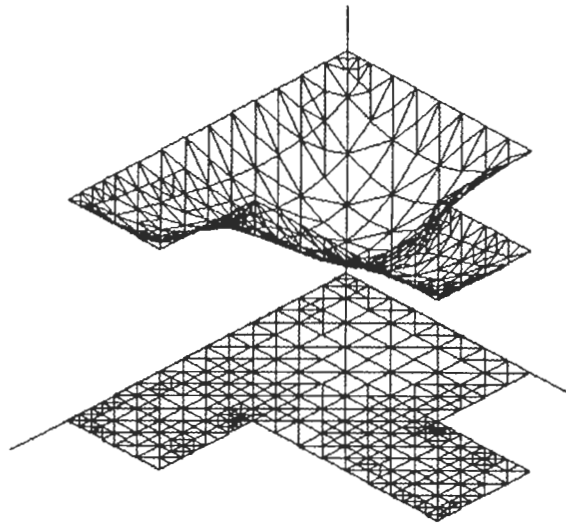






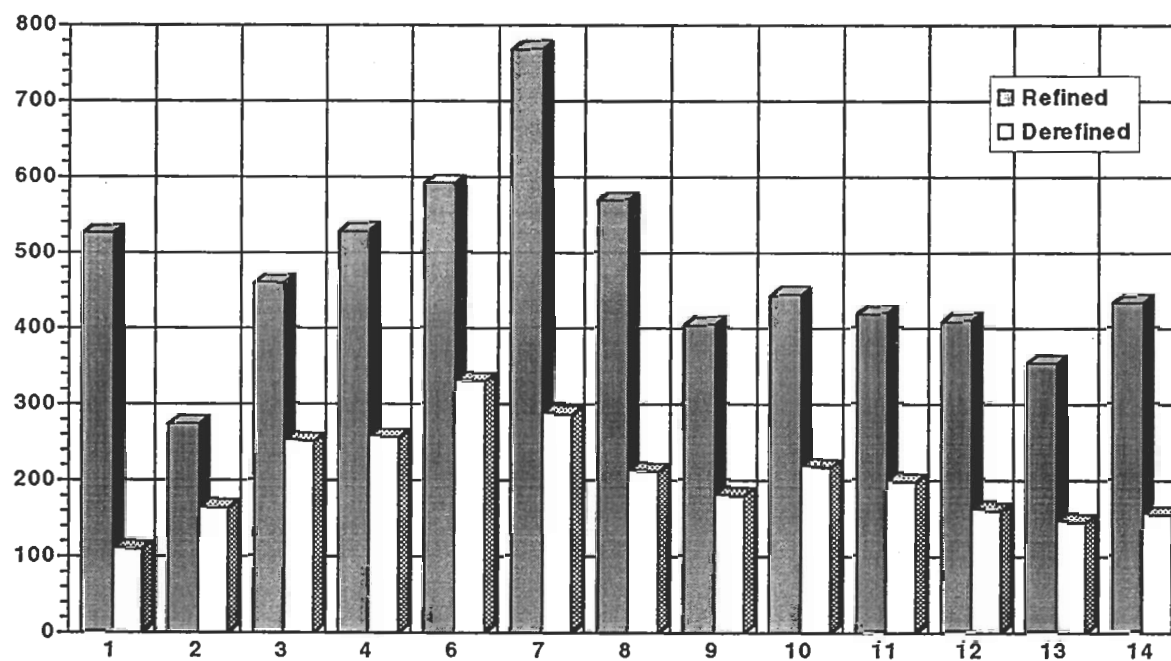






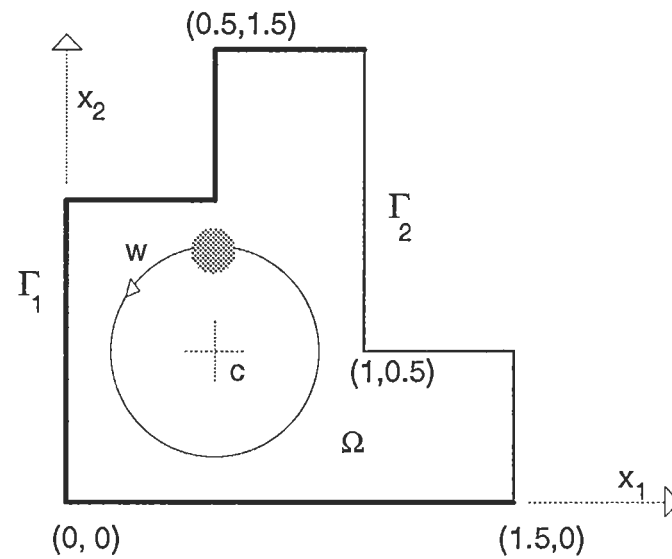
REFINED AND DEREFINED MESHES

NODES



TIME STEP

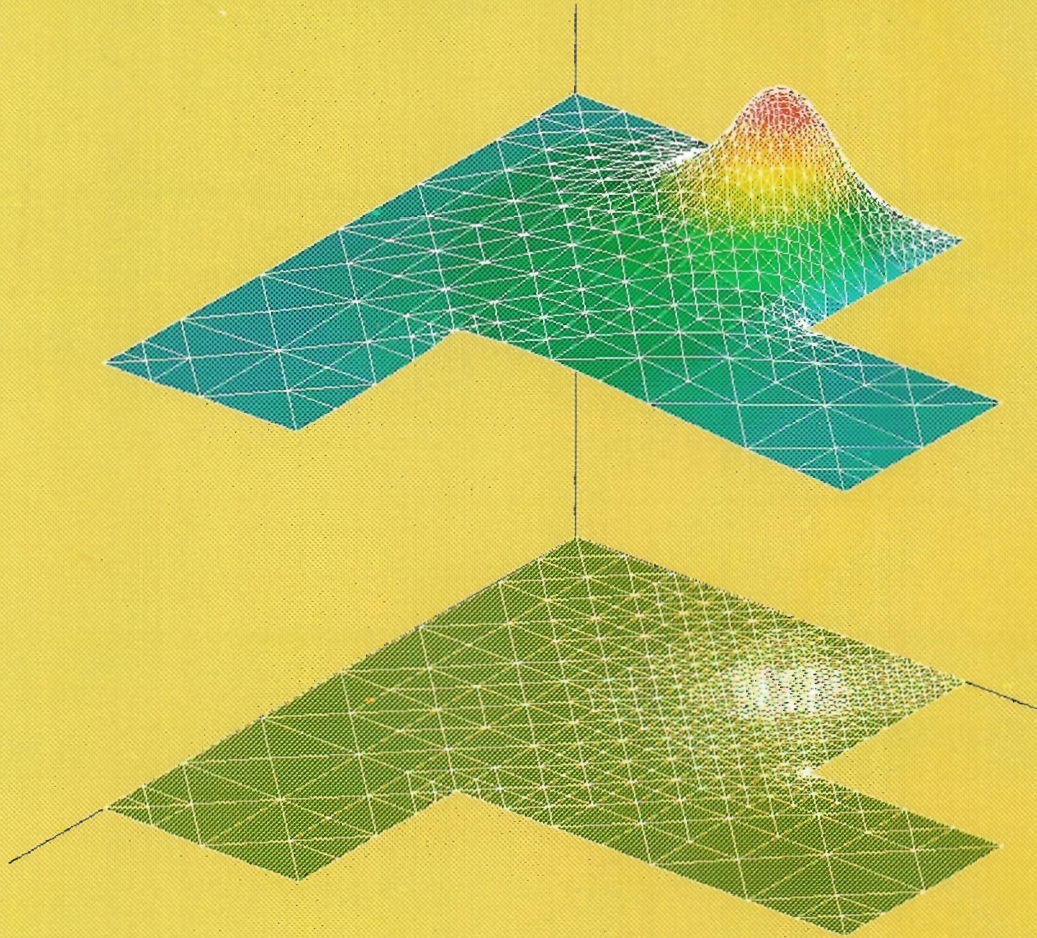
Un Problema de Poisson Quasi-Estacionario



$$f(x_1, x_2) = \begin{cases} 40 & \text{if } d(t) < r \\ 0 & \text{if } d(t) > r \end{cases}$$

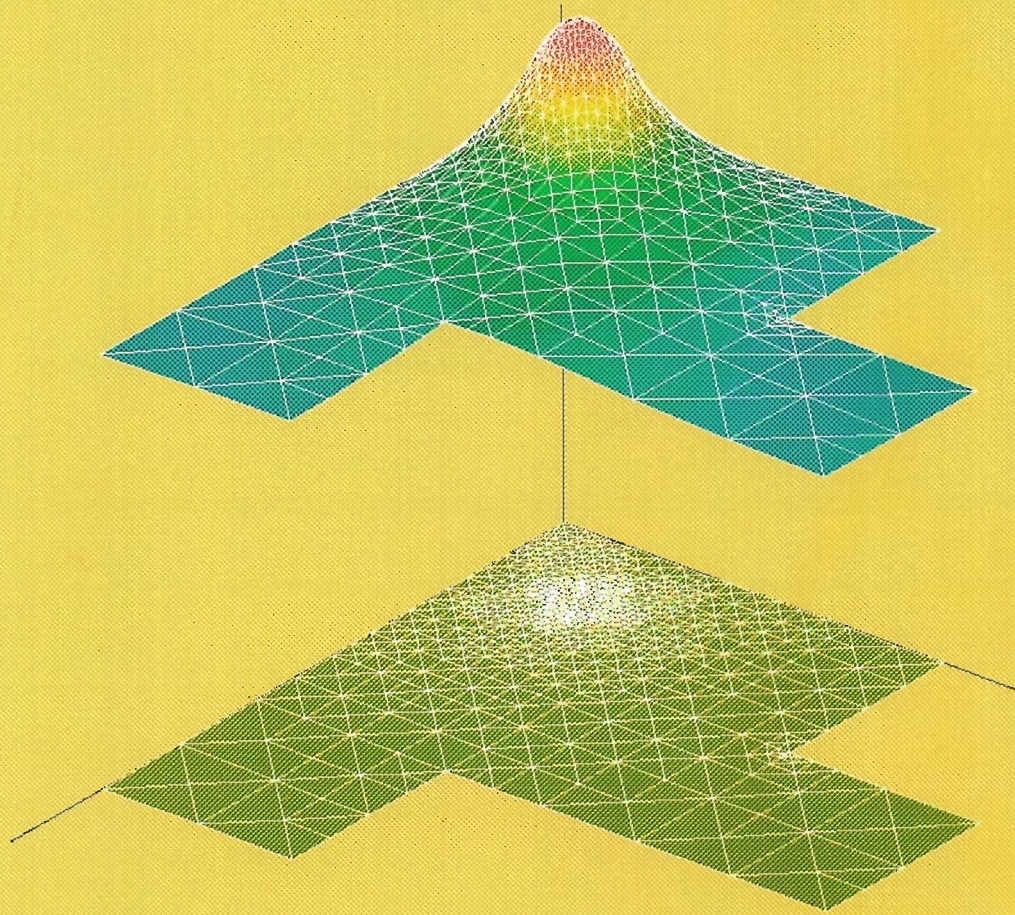
$$\omega = (\pi/4) \text{ rad/seg}$$

PROBLEMA QUASI-EVOLUTIVO



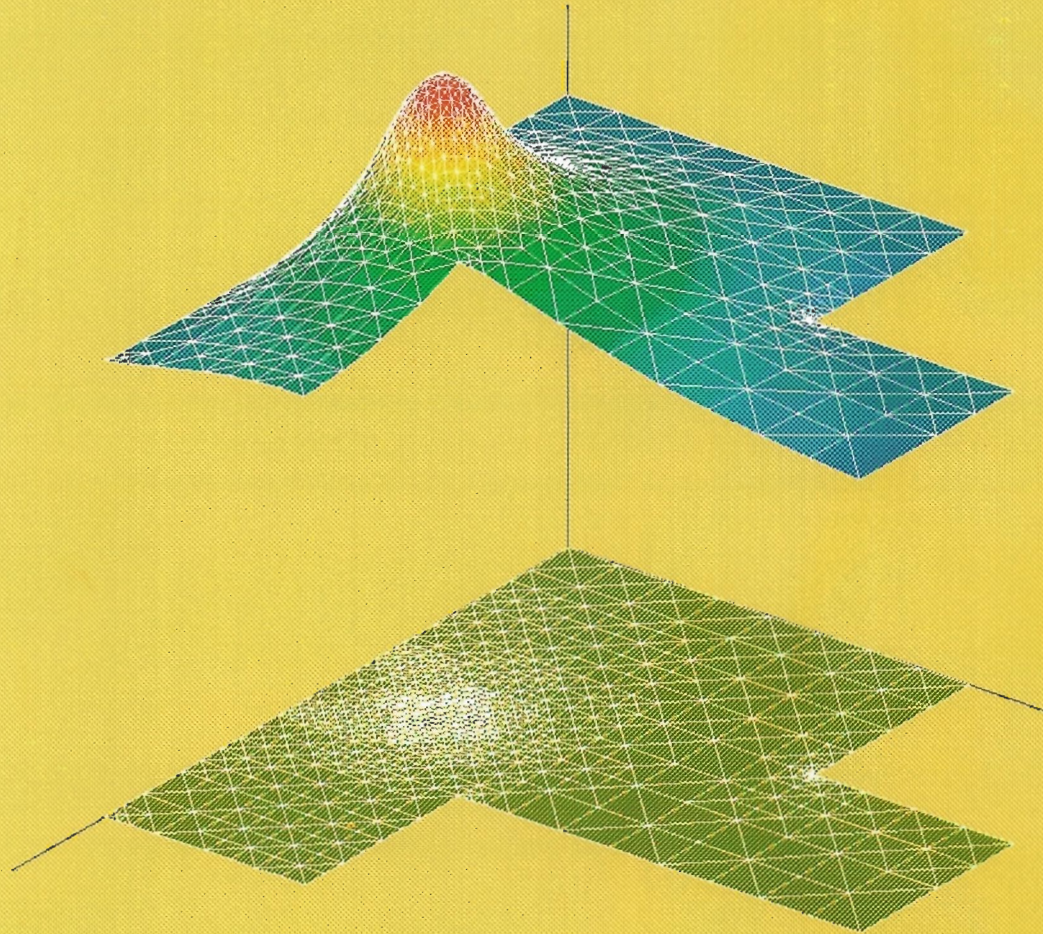
t=1; 838 nodos

PROBLEMA QUASI-EVOLUTIVO



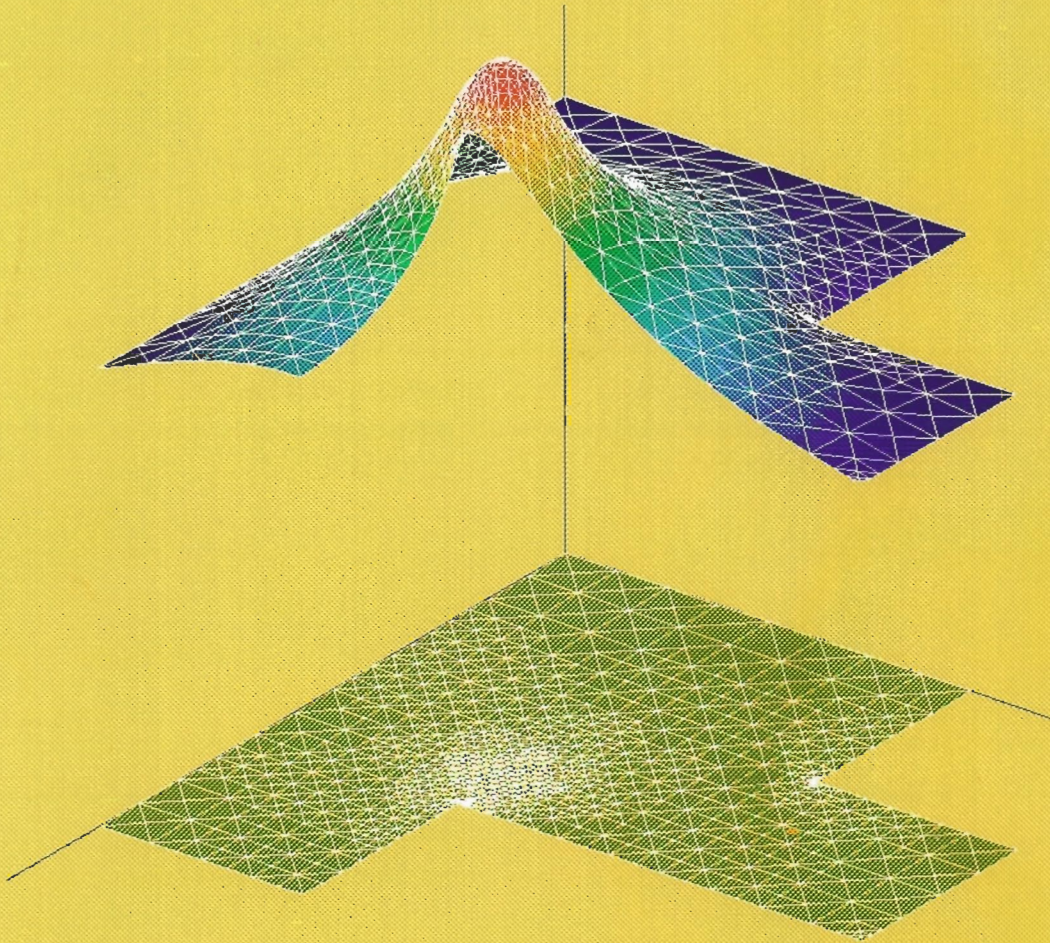
t=3; 812 nodos

PROBLEMA QUASI-EVOLUTIVO



$t=5$; 821 nodos

PROBLEMA QUASI-EVOLUTIVO



t=6; 892 nodos

Problema de Convección-Difusión

$$\frac{\partial u}{\partial t} + \vec{v} \cdot \vec{\nabla} u - \vec{\nabla} \cdot (k \vec{\nabla} u) = f$$

$$\frac{du}{dt} - \vec{\nabla} \cdot (k \vec{\nabla} u) = f$$

$$\frac{du}{dt} \approx \frac{u(\vec{x}, t_{n+1}) - u(\vec{x}, t_n)}{\Delta t} = \frac{u^{n+1}(\vec{x}) - u^n(\vec{x})}{\Delta t}$$

$$u^{n+1}(\vec{x}) - \Delta t \vec{\nabla} \cdot [k(\vec{x}) \vec{\nabla} u^{n+1}(\vec{x})] = \Delta t f^{n+1}(\vec{x}) + u^n(\vec{x})$$

Aproximación de la característica

$$\underline{x}_i = x_i(t_{n+1} - \Delta t) = x_i - v_i(\bar{x})\Delta t + \frac{\Delta t^2}{2} \vec{v}(\bar{x}) \cdot \vec{\nabla} v_i(\bar{x}) + 0(\Delta t^3)$$

$$u^n(\underline{\bar{x}}) = u^n(\bar{x} - \Delta \bar{x}) = u^n(\bar{x}) - \sum_{i=1}^2 \Delta x_i \frac{\partial u^n(\bar{x})}{\partial x_i} + \frac{1}{2} \left[2\Delta x_1 \Delta x_2 \frac{\partial^2 u^n(\bar{x})}{\partial x_1 \partial x_2} + \sum_{i=1}^2 \Delta x_i^2 \frac{\partial^2 u^n(\bar{x})}{\partial x_i^2} \right] + 0(\|\Delta \bar{x}\|^3)$$

$$u^n(\underline{\bar{x}}) = u^n(\bar{x}) - \Delta t \sum_{i=1}^2 v_i(\bar{x}) \frac{\partial u^n(\bar{x})}{\partial x_i} + \frac{\Delta t^2}{2} \sum_{i=1}^2 \left[\vec{v}(\bar{x}) \cdot \vec{\nabla} v_i(\bar{x}) \right] + \frac{\Delta t^2}{2} \sum_{i=1}^2 \sum_{j=1}^2 v_i(\bar{x}) v_j(\bar{x}) \frac{\partial^2 u^n(\bar{x})}{\partial x_i \partial x_j} + 0(\Delta t^3)$$

$$u^{n+1} - \Delta t \vec{\nabla} \cdot [k \vec{\nabla} u^{n+1}] = \Delta t f^{n+1} + u^n - \Delta t \vec{v} \cdot \vec{\nabla} u^n + \frac{\Delta t^2}{2} \sum_i (\vec{v} \cdot \vec{\nabla} v_i) \frac{\partial u^n}{\partial x_i} + \frac{\Delta t^2}{2} \sum_{i,j} v_i v_j \frac{\partial^2 u^n}{\partial x_i \partial x_j}$$

Estabilidad de la Formulación en 1-D

$$C \leq \left[\frac{1}{P_e^2} + \frac{1}{3} \right]^{1/2} + \frac{1}{P_e}$$

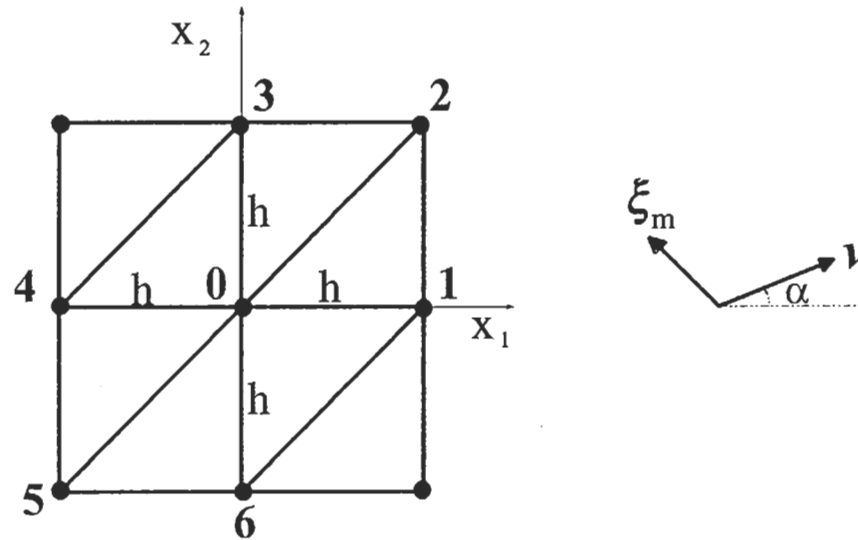
$$C = v \Delta t / h \quad y \quad P_e = v h / k$$

Consistencia de la Formulación en 1-D

$$G(\eta) = G_e(\eta) + O(\eta^3)$$

$$\eta = \xi h$$

Estabilidad de la Formulación en 2-D

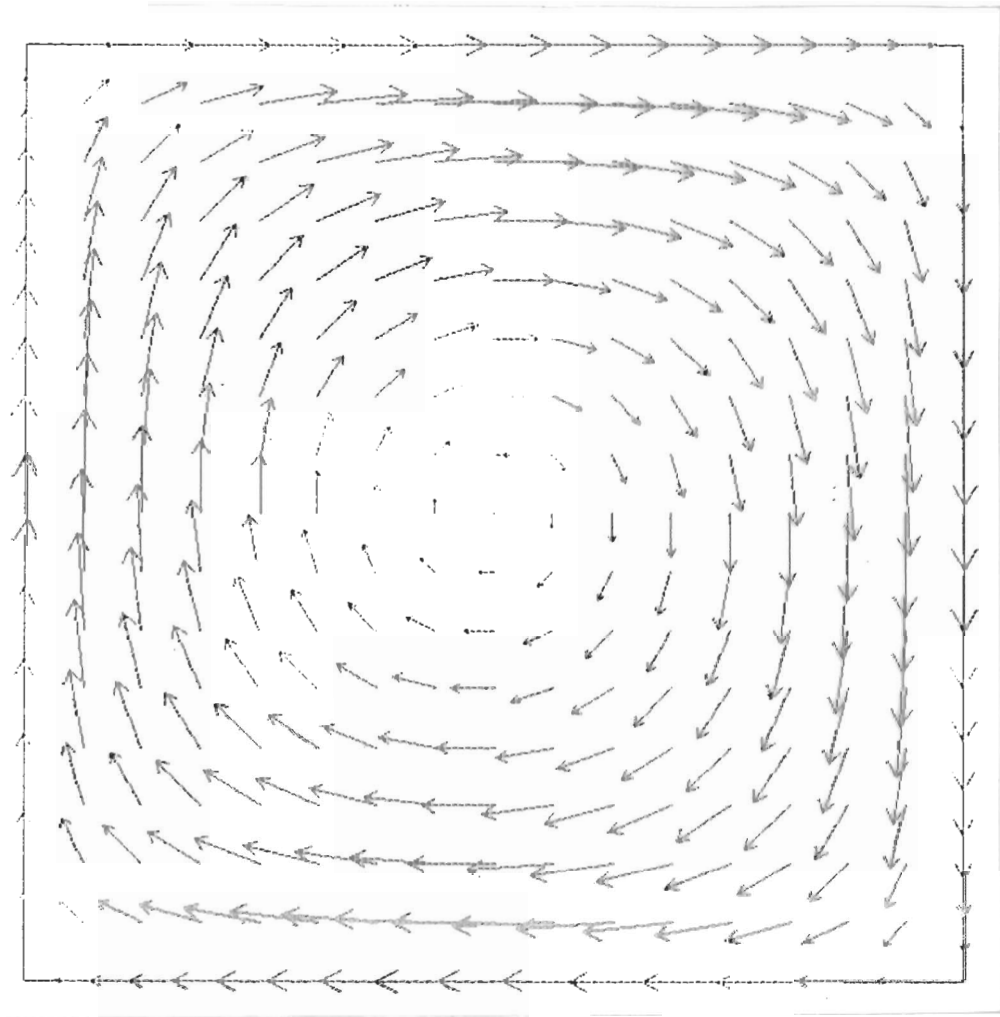


Dirección de propagación: eje x_1 ,

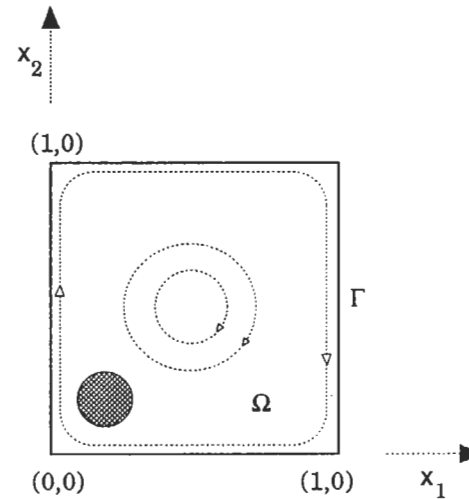
$$C \leq \frac{1}{\cos^2 \alpha} \left[\left(\frac{1}{P_e^2} + \frac{1}{3} \cos^2 \alpha \right)^{1/2} + \frac{1}{P_e} \right]$$

Dirección de propagación: $3\pi/4$,

$$C \leq \frac{1}{1 - \sin 2\alpha} \left[\left(\frac{4}{P_e^2} + \frac{1}{3} (1 - \sin 2\alpha) \right)^{1/2} + \frac{2}{P_e} \right]$$



PROBLEMA DE CONVECCIÓN-DIFUSIÓN

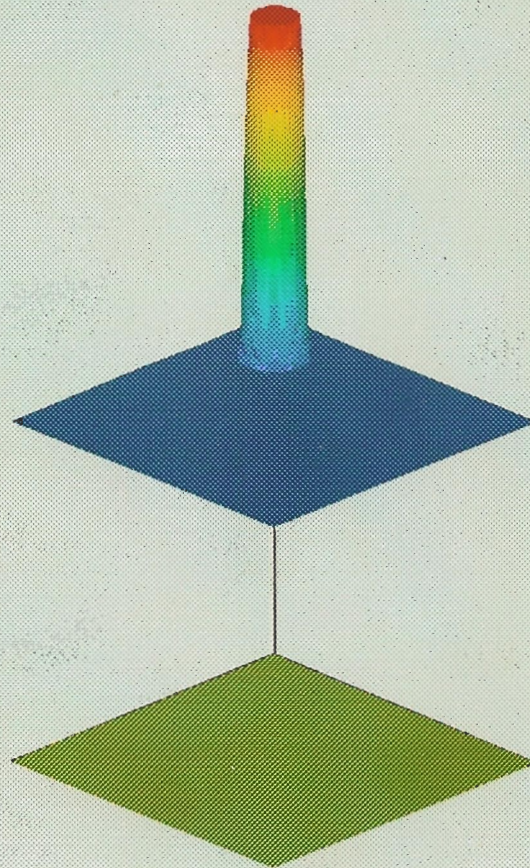


$$\left. \begin{aligned} \frac{\partial u}{\partial t} + \bar{v} \cdot \bar{\nabla} u - \bar{\nabla} \cdot (k \bar{\nabla} u) &= 0 \quad \text{en } \Omega \\ k \frac{\partial u}{\partial n} &= 0 \quad \text{en } \Gamma = \partial \Omega \end{aligned} \right\} \text{ donde } \bar{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \omega x_1 (1-x_1)(x_2-0.5) \\ \omega x_2 (0.5-x_1)(1-x_2) \end{pmatrix}, \quad k = 0.1, \quad \omega = 2.5 \cdot 10^4$$

$$u(\bar{x}, 0) = \begin{cases} 1 & \text{si } d > 0.1 \\ 1 + e^{(0.1^2 - d^2)/(0.07^2 - d^2)} & \text{si } 0.07 < d < 0.1 \\ 2 & \text{si } d < 0.07 \end{cases}$$

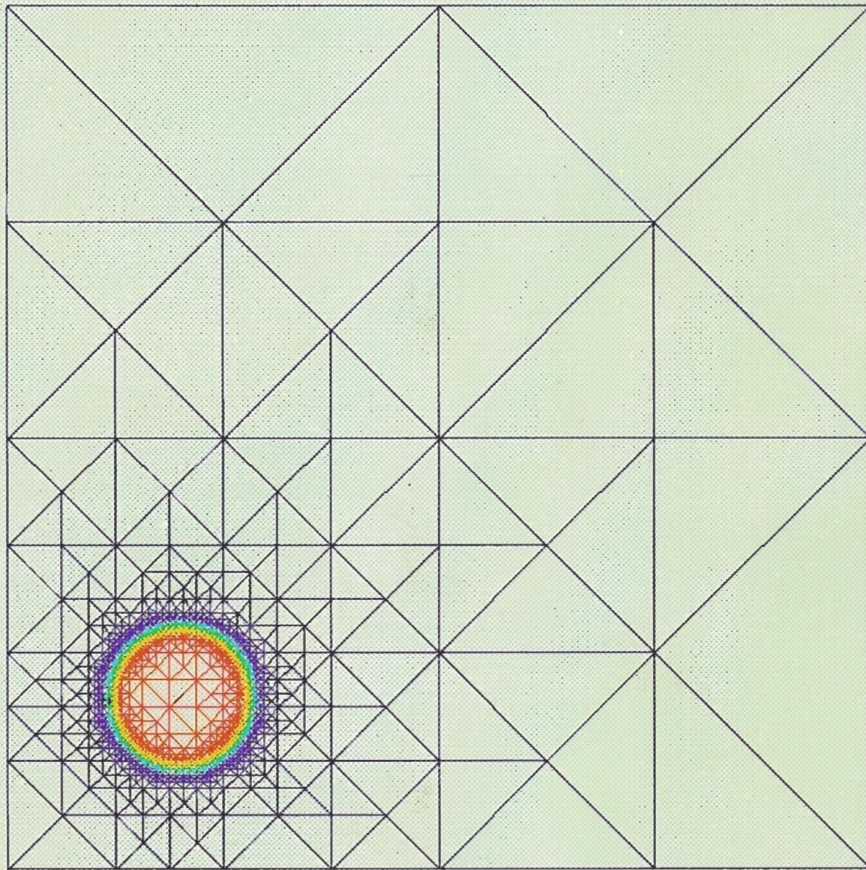
$$P_e = \frac{|\bar{v}|_{\max} L}{k} = 3.125 \cdot 10^4$$

PROBLEMA DE CONVECCION-DIFUSION



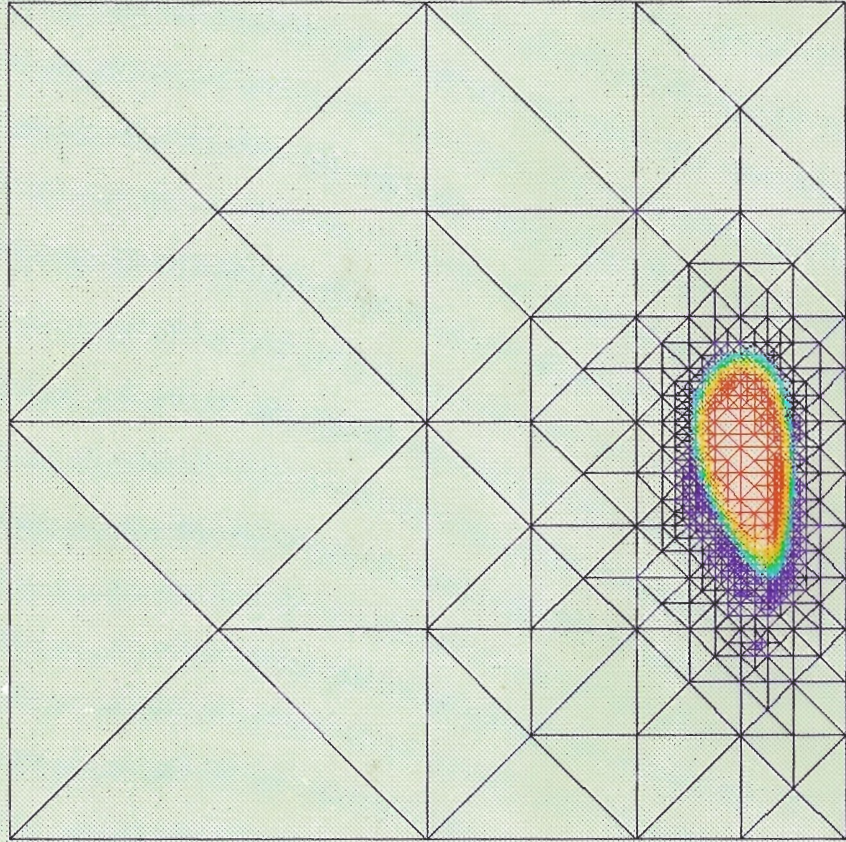
SOLUCION INICIAL

PROBLEMA DE CONVECCION-DIFUSION



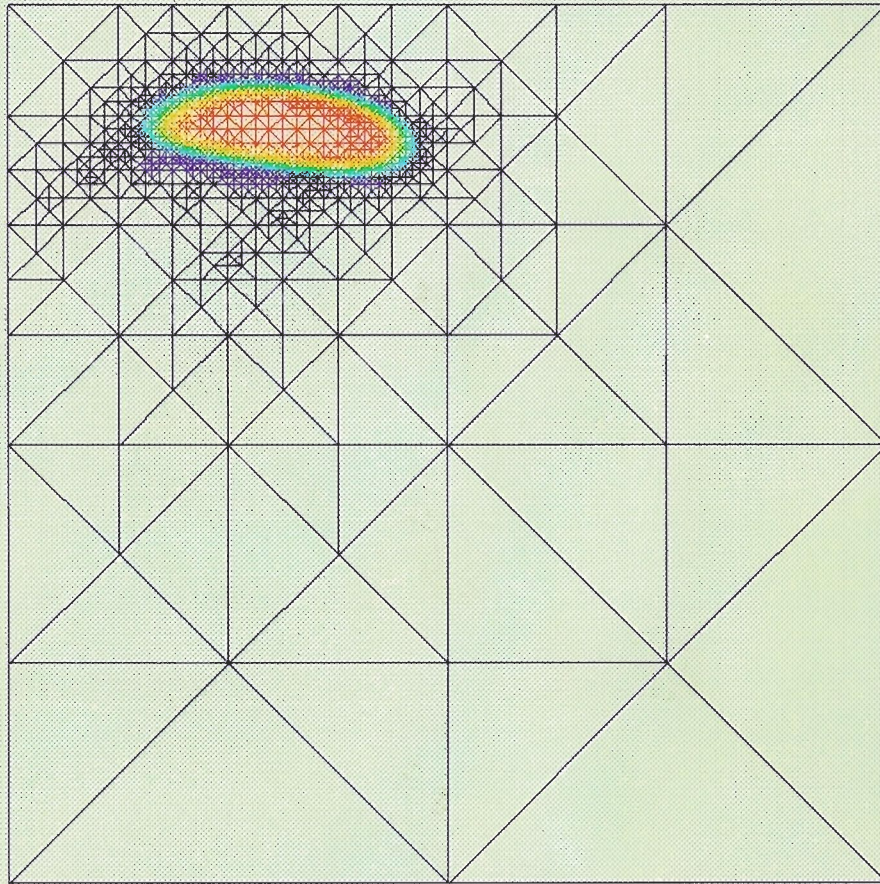
$t=0.0$; 1198 nodos

PROBLEMA DE CONVECCION-DIFUSION



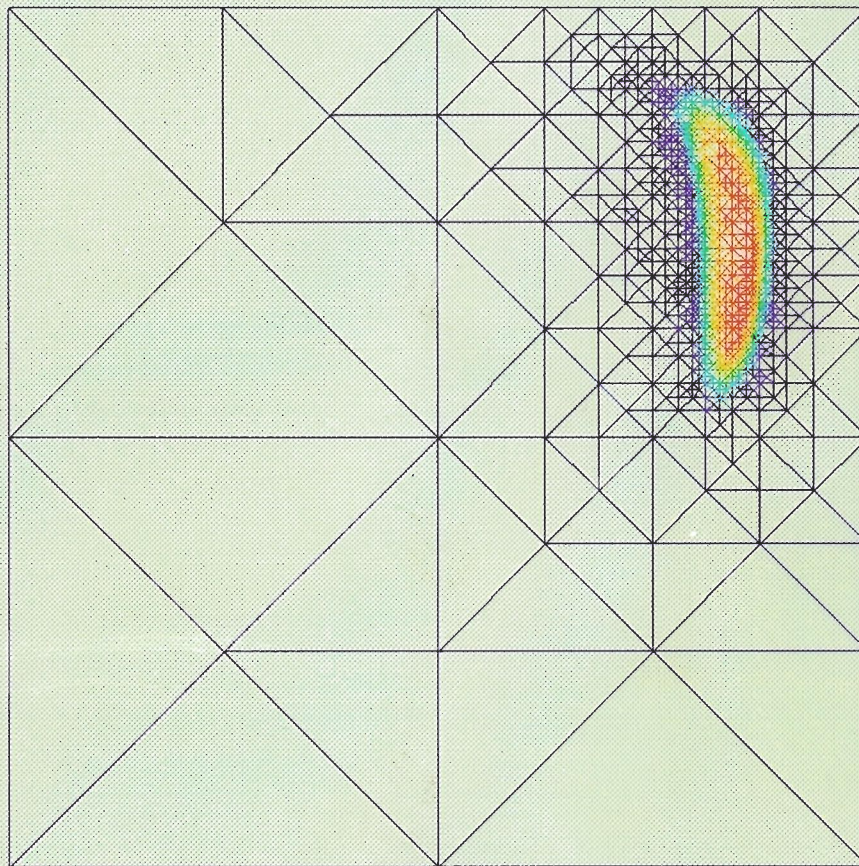
t=0.00014; 982 nodos

PROBLEMA DE CONVECCION-DIFUSION



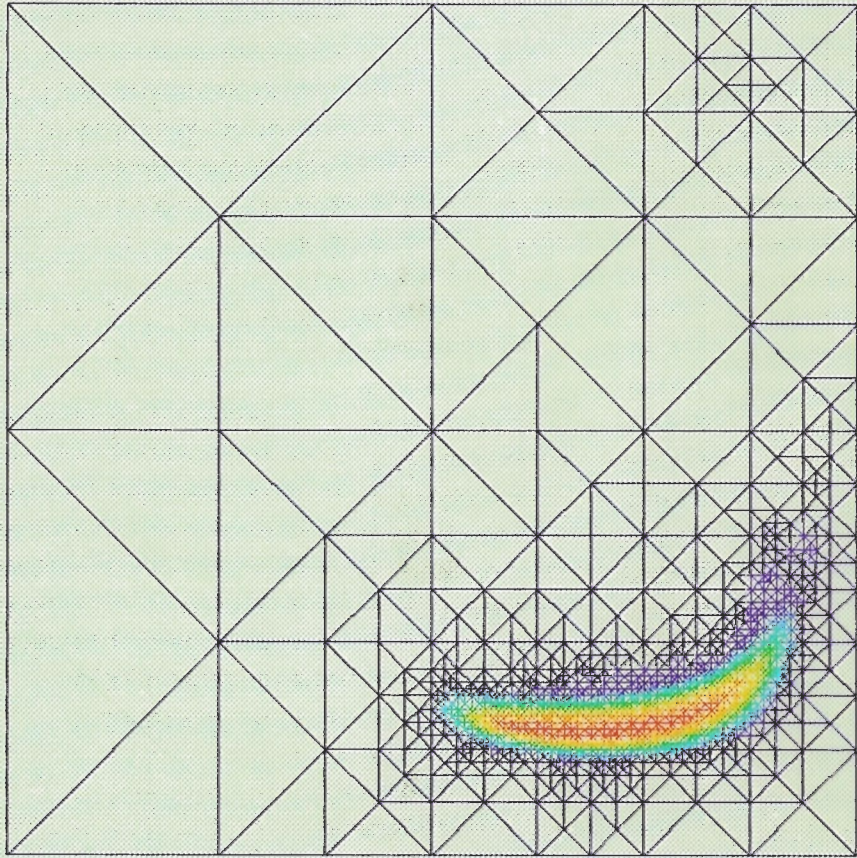
$t=0.00039$; 1183 nodos

PROBLEMA DE CONVECCION-DIFUSION



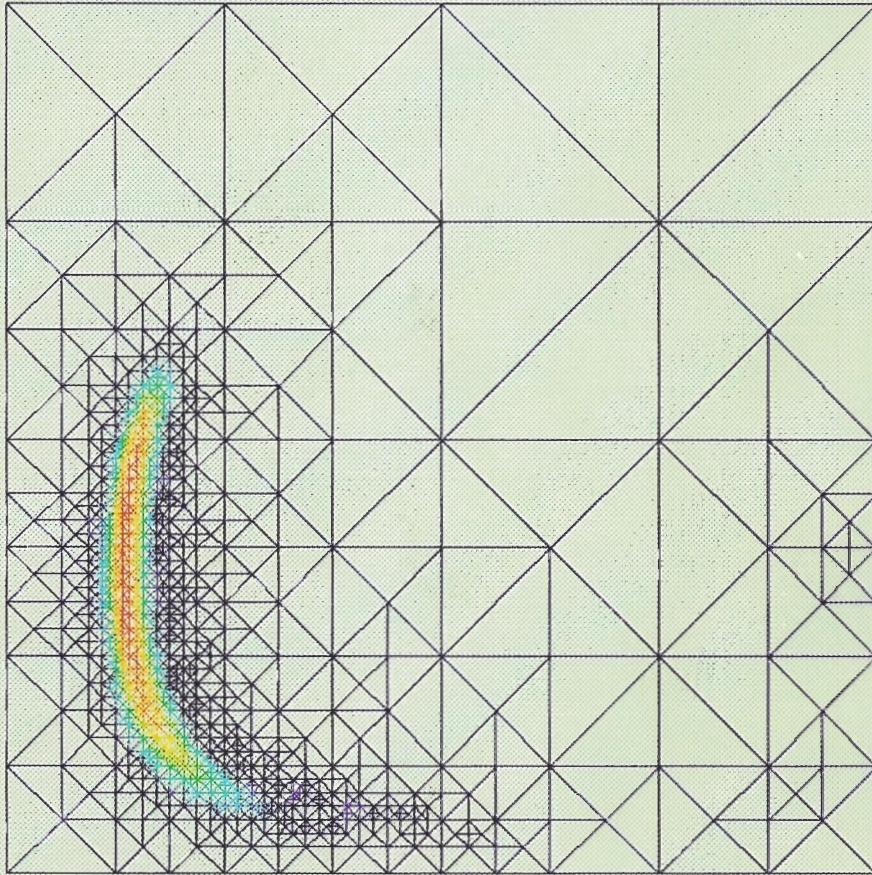
$t=0.00067$; 908 nodos

PROBLEMA DE CONVECCION-DIFUSION

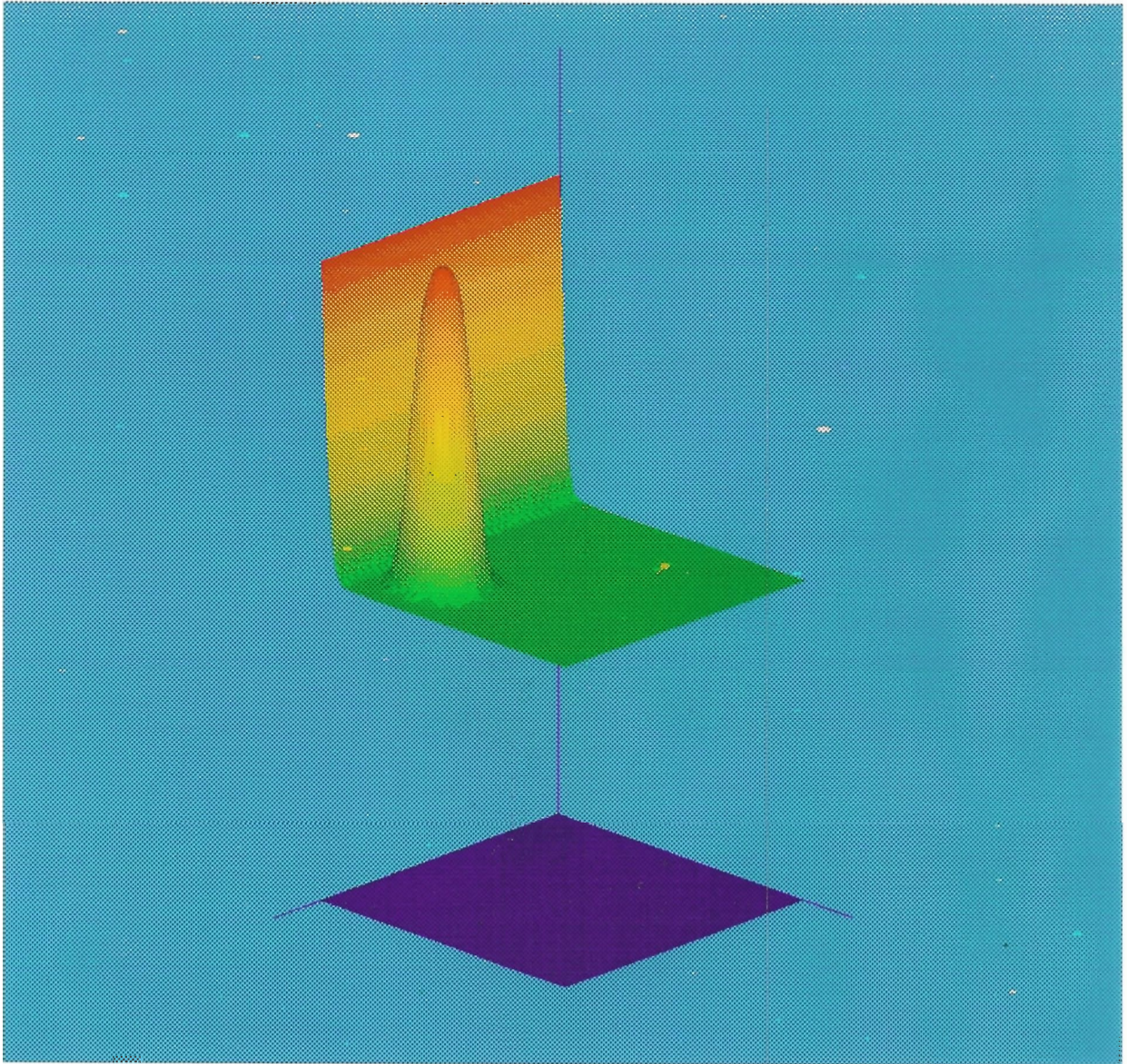


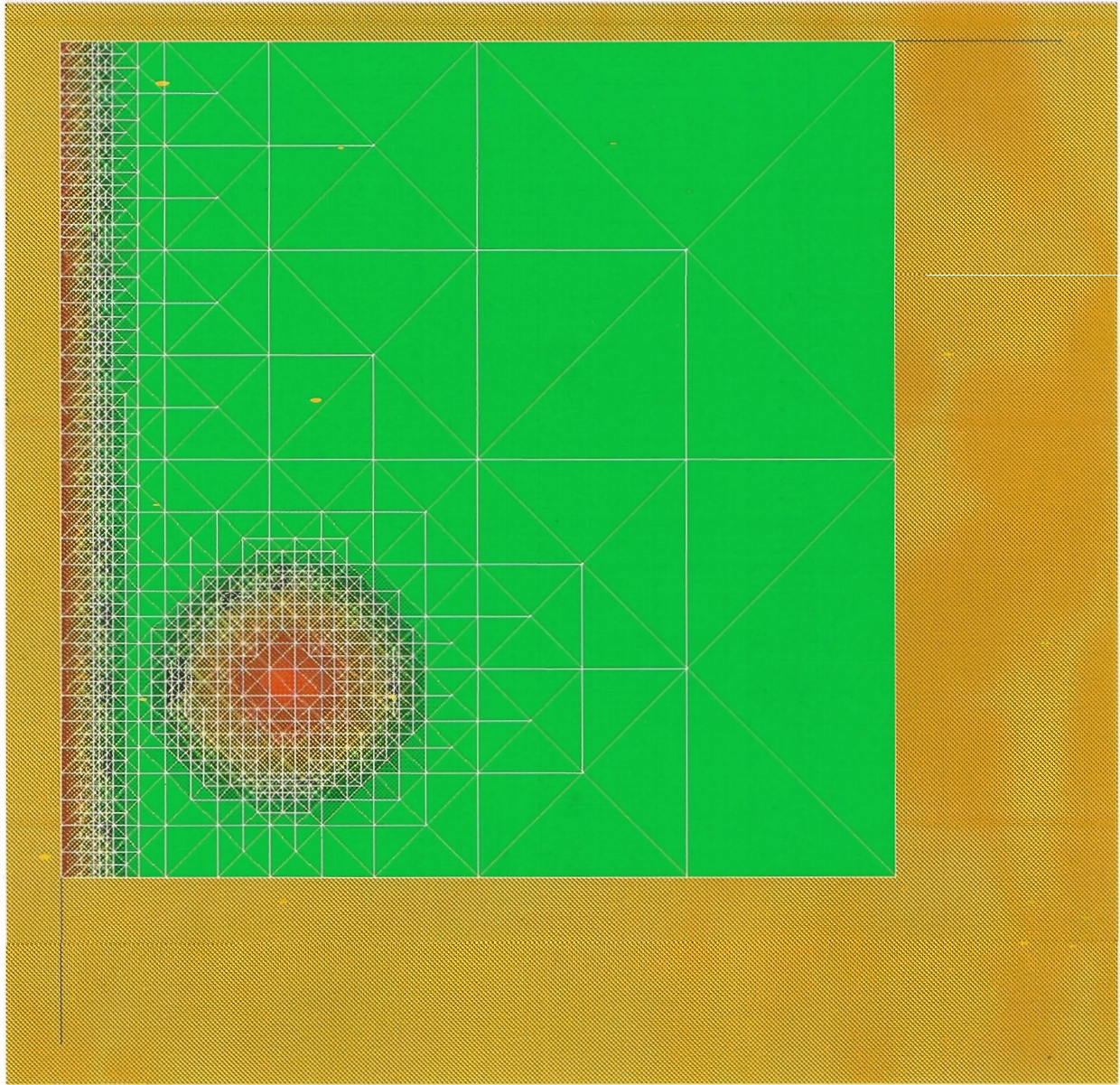
$t=0.00098$; 1105 nodos

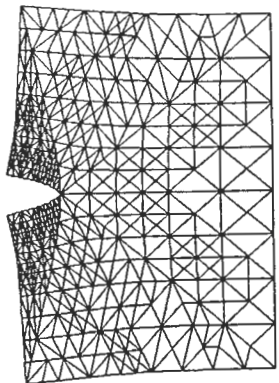
PROBLEMA DE CONVECCION-DIFUSION



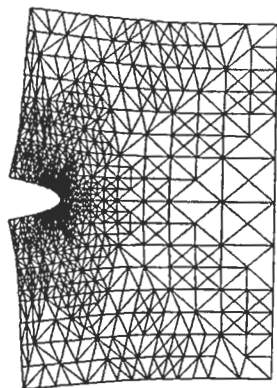
$t=0.0013$; 966 nodos



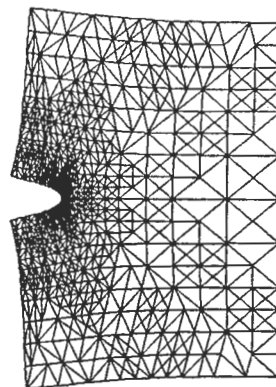




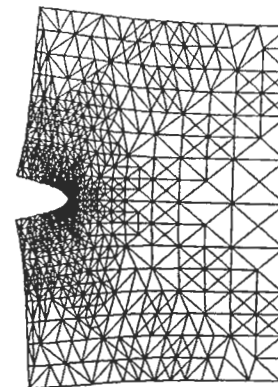
Step 3, 421 nodes.



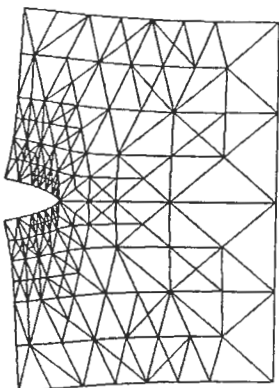
Step 5, 797 nodes



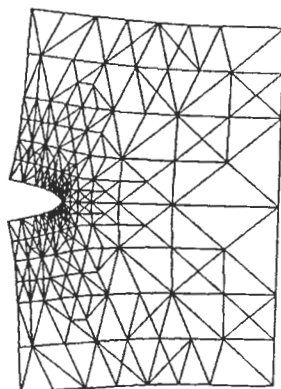
Step 7, 1001 nodes



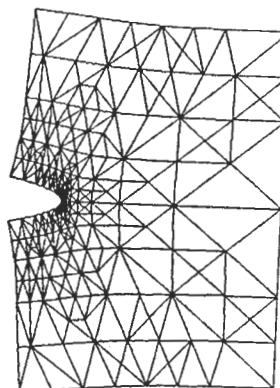
Step 9, 1075 nodes



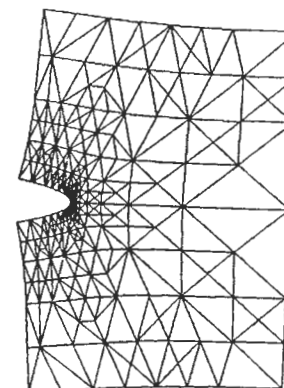
Step 3, 175 nodes.



Step 5, 246 nodes

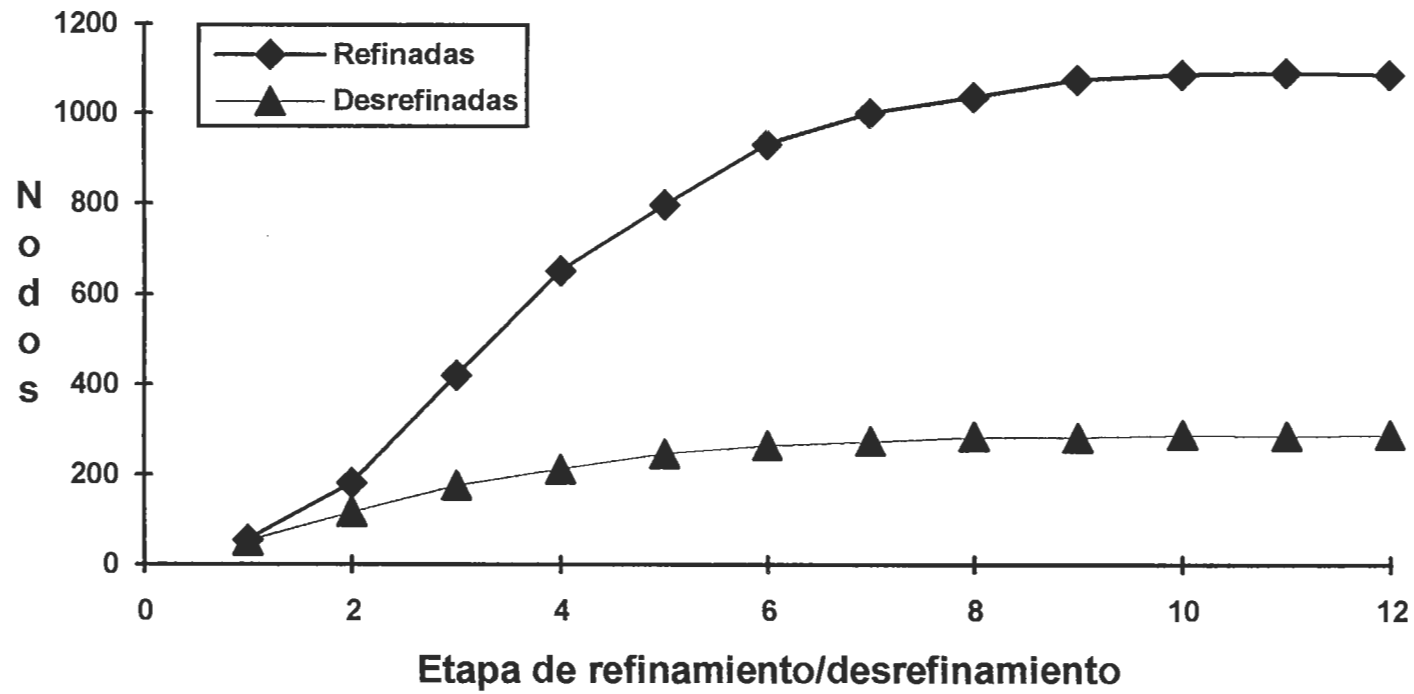


Step 7, 273 nodes



Step 9, 282 nodes

Evolución del Número de Nodos



CONCLUSIONS AND FUTURE RESEARCHES

1.- The refinement/derefinement combination of nested grids give:

- Easy applications of multigrid methods.
- Flexibility in meshes.
- Automatic control of the number of nodes.
- Bounded number of equations in the process.
- Very fast readaptation of the current mesh.
- A good method to approach any initial solution and geometry of the domain.
- A local refinement (derefining after a global refinement).

2.- Linear complexity of the improved derefinement algorithm.

3.- Generalization to three dimensional domains.

**GENERACIÓN DE MALLAS
TRIDIMENSIONALES MEDIANTE LA
TRIANGULACIÓN DE DELAUNAY**

CONTENIDOS

□ ENTRADA DE DATOS

Definición de la geometría del objeto; puntos, líneas, superficies y regiones.

Generación de puntos en el objeto.

□ TRIANGULACIÓN DE DELAUNAY EN 3-D

Algoritmo de Watson.

Problemas asociados a errores de redondeo.

Implementación del algoritmo.

Control sobre los errores de redondeo para conseguir una triangulación aceptable.

□ ELIMINACIÓN DE LA ENVUELTA CONVEXA DE TETRAEDROS

□ EJEMPLOS

□ CONCLUSIONES Y LINEAS DE TRABAJO FUTURAS

Diagrama de Voronoi y Triangulación de Delaunay (I)

- Sea $X \subset E^3$ un conjunto de puntos diferentes no todos en el mismo plano.
- El poliedro de Voronoi, $V(x_i)$, asociado a $x_i \in X$, se define como:

$$V(x_i) = \{x \in E^3 / d(x, x_i) \leq d(x, x_j), \forall x_j \in X, i \neq j\}$$

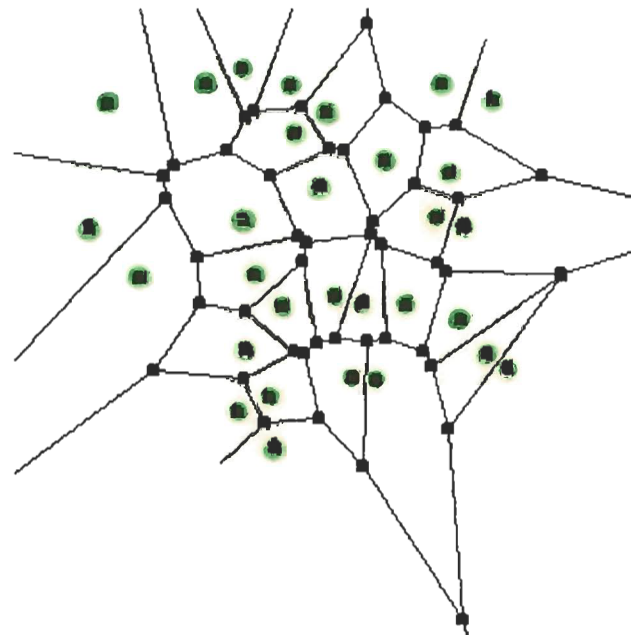
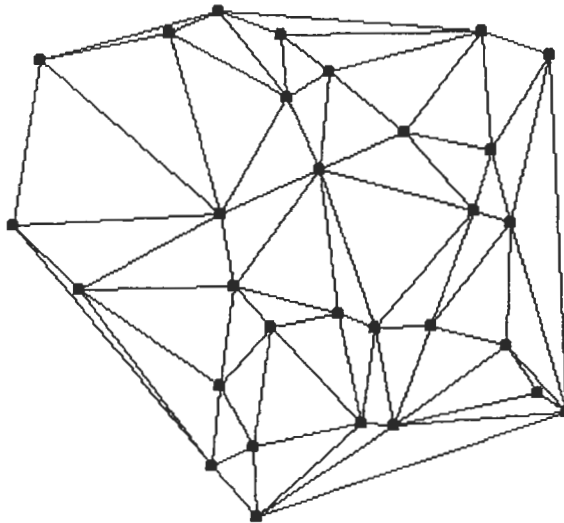


Diagrama de Voronoi y Triangulación de Delaunay (II)

- Sea $R \subset X$ un subconjunto de cuatro o más puntos de X , no todos en el mismo plano, tal que existe un punto, $v_R \in E^3$, equidistante de todos los puntos de R que satisface que

$$d(v_R, x_k) < d(v_R, x_i) \quad x_k \in R \text{ y } \forall x_i \in X - R.$$

- El politopo de Delaunay, $D(R)$, es la envolvente convexa de R .
- La triangulación de Delaunay, $D(X)$, es el conjunto de todos los politopos de Delaunay.



Algoritmo Incremental para la Construcción de la Triangulación de Delaunay

1. Triangulación inicial.

Se construye un prisma que contenga todos los puntos de X y se descompone en cinco tetraedros de manera que formen una triangulación de Delaunay.

2. Adición del punto x_{i+1} a la triangulación.

Hallamos el conjunto de tetraedros, D_1^i , cuyas esferas circunscritas contienen a x_{i+1} .

$$D_1^i = \{T_j \in D(X_i) / x_{i+1} \in B(T_j)\}$$

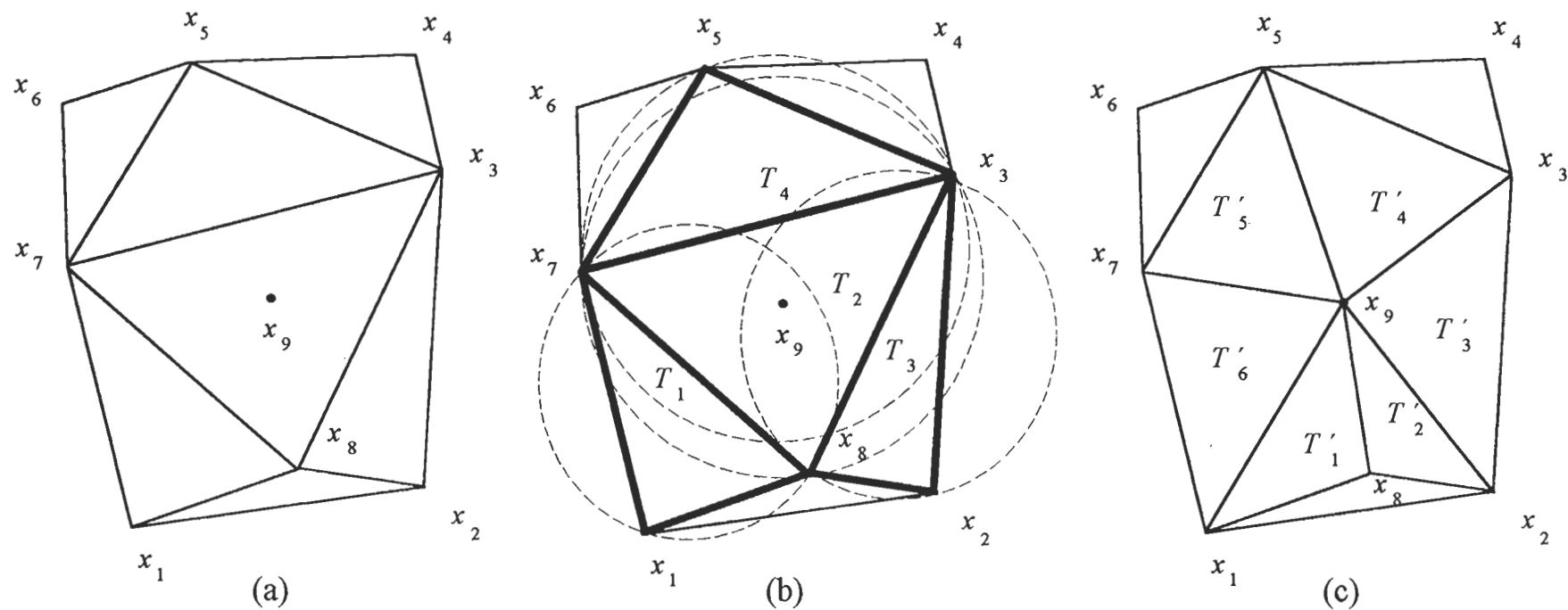
Sea $C_{fD_1}^i$ el conjunto de caras de la frontera de D_1^i , esto es, caras que sólo pertenecen a un tetraedro de D_1^i . Si $D(X_i)$ es la triangulación de los i primeros puntos de X , la triangulación $D(X_{i+1})$ que incluye al punto x_{i+1} se define como:

$$D(X_{i+1}) = (D(X_i) - D_1^i) \cup D_2^i$$

donde $D_2^i = \bigcup_j \{C_j, x_{i+1}\}$ es el conjunto de tetraedros formados con x_{i+1} y las caras C_j de $C_{fD_1}^i$.

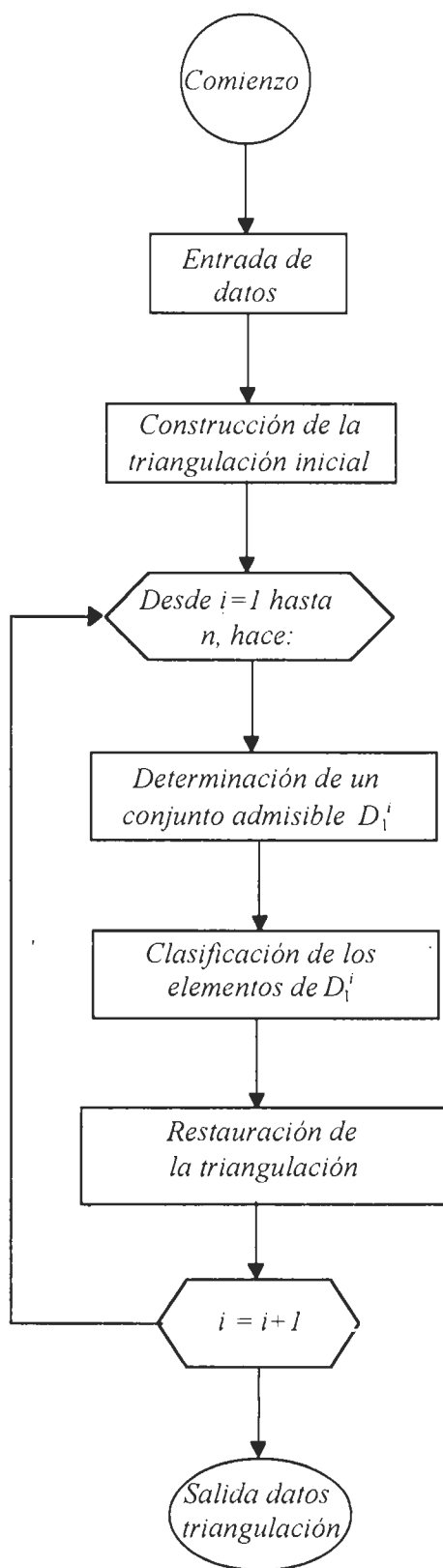
Si las esferas circunscritas a los tetraedros de D_1^i se calculan exactamente, el conjunto D_1^i será en forma de estrella respecto a x_{i+1} y no contendrá ningún otro punto de X aparte de x_{i+1} ; D_2^i estará bien definido.

Ejemplo de Adición de un Punto a la Triangulación



- (a) Se añade el punto x_9 a la triangulación $D(X_8)$ formada con los puntos $X_8 = \{x_1, \dots, x_8\}$.
- (b) Los tetraedros cuyas esferas circunscritas contienen a x_9 son $D_1^{\circ} = \{T_1, \dots, T_4\}$.
- (c) La triangulación que incluye a x_9 es $D(X_9) = (D(X_8) - D_1^{\circ}) \cup D_2^{\circ}$, donde $D_2^{\circ} = \{T'_1, \dots, T'_6\}$.

Diagrama de flujo del programa

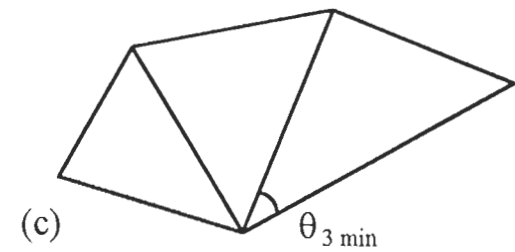
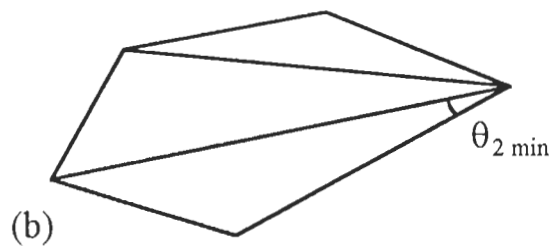
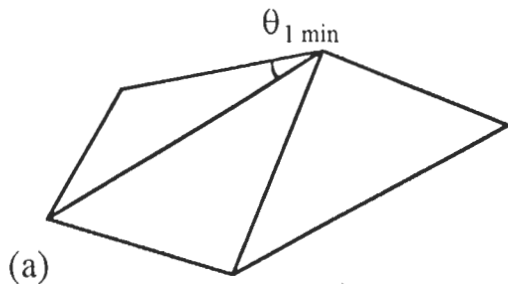


Propiedades de la triangulación de Delaunay (III)

La triangulación de Delaunay presenta algunas propiedades que la hacen muy adecuada para la aplicación del M.E.F. En concreto, para un determinado conjunto de puntos del plano es la triangulación que produce los triángulos más regulares. Sin embargo, no existe ninguna generalización de este resultado en dimensión mayor que dos.

Propiedad

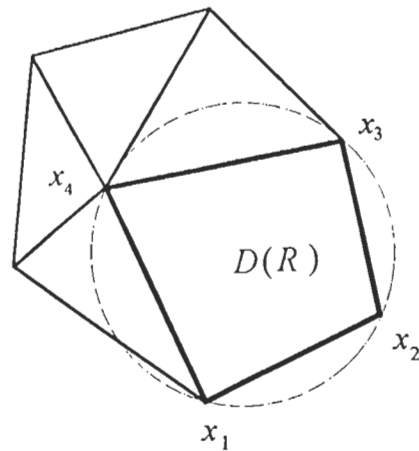
De todas las triangulaciones de un conjunto de puntos $X \subset E^2$ en posición general, la de Delaunay es aquella que hace máximo el ángulo mínimo de cualquier triángulo. \square



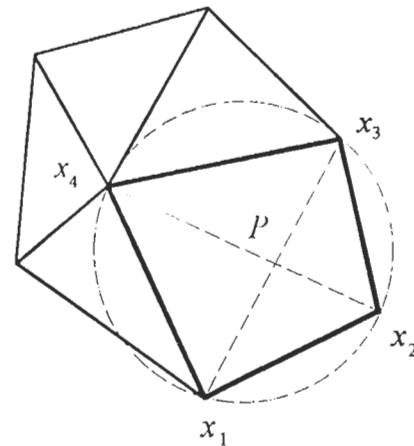
La triangulación de Delaunay (c) es la que hace máximo el ángulo mínimo: $\theta_{3 \min} > \theta_{2 \min} > \theta_{1 \min}$.

Caso degenerado

Si X no está en posición general hay celdas degeneradas (figura (a)). La descomposición en símlices de un politopo correspondiente a una celda degenerada (figura (b)) no es única. Los algoritmos existentes para generar la triangulación de Delaunay tienen dificultades en esta situación. Las dificultades subsisten incluso cuando la disposición de puntos es tal que X no está *claramente* en posición general, es decir, cuando hay más de $k + 1$ puntos de X *próximo*s a la misma k -esfera.

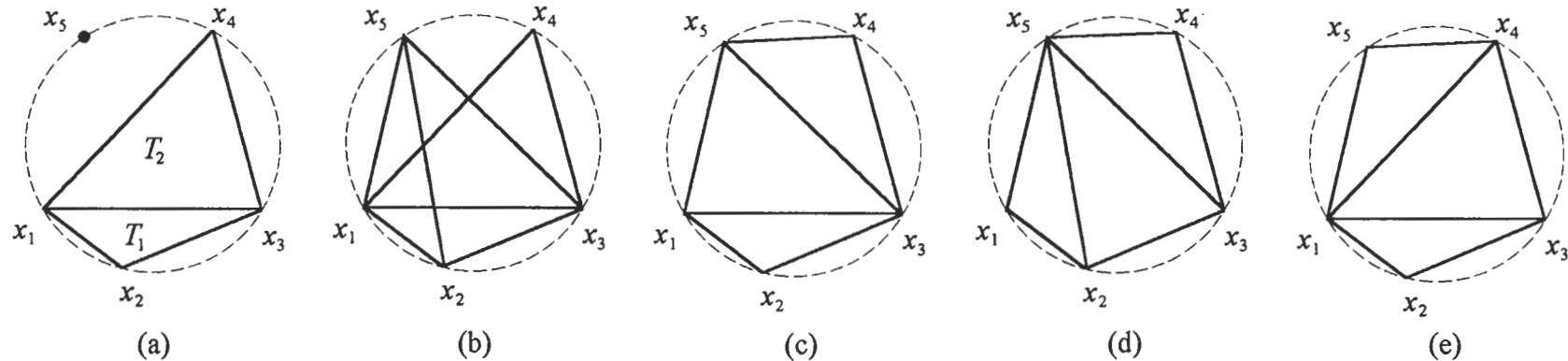


(a)



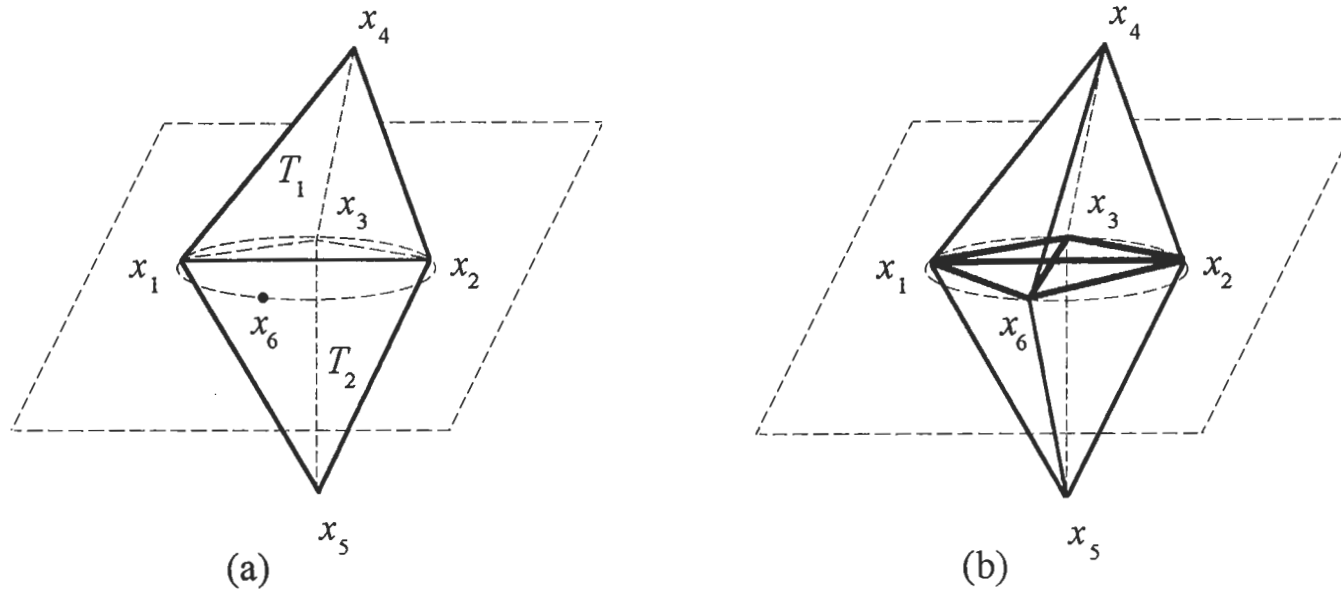
(b)

Problemas Debidos a Errores de Redondeo (I)



- Adición de x_5 a la triangulación de $X_4 = \{x_1, \dots, x_4\}$, figura (a).
- Las triangulaciones posibles, en función de la posición de x_5 calculada por el ordenador, son:
 - (b) Si $x_5 \in B(T_1)$ y $x_5 \notin B(T_2)$, el algoritmo elimina T_1 y conserva T_2 ; triangulación no admisible.
 - (c) Si $x_5 \notin B(T_1)$ y $x_5 \in B(T_2)$, el algoritmo conserva T_1 y elimina T_2 ; triangulación admisible.
 - (d) Si $x_5 \in B(T_1)$ y $x_5 \in B(T_2)$, el algoritmo elimina T_1 y T_2 ; triangulación admisible.
 - (e) Si $x_5 \notin B(T_1)$ y $x_5 \notin B(T_2)$, el algoritmo conserva T_1 y T_2 ; triangulación admisible.

Problemas Debidos a Errores de Redondeo (II)



- (a) Adición del punto x_6 , coplanario con x_1, x_2 y x_3 , a la triangulación formada por los tetraedros T_1 y T_2 .
- (b) Si la posición de x_6 , calculada por el ordenador, es tal que $x_6 \in B(T_1)$ y $x_6 \notin B(T_2)$, se forma el *tetraedro* con volumen nulo, $\{x_1, x_2, x_3, x_6\}$. Esta situación puede ser peor aun si el punto x_6 estuviera por debajo del plano definido por x_1, x_2 y x_3 y el ordenador siguiera evaluando que $x_6 \in B(T_1)$ y $x_6 \notin B(T_2)$. En ese caso los puntos $\{x_1, x_2, x_3, x_6\}$ no formarían un tetraedro.

Alternativas para solucionar las inconsistencias topológicas en la triangulación de Delaunay

□ Aritmética no exacta

Elaborar algún algoritmo capaz de crear una triangulación consistente, aunque ésta no sea exactamente de Delaunay.

⇒ Problemas: Puede llegar a fallar cuando los puntos están muy próximos entre sí.

□ Aritmética exacta

1. Aplicar un factor de escala a los datos de la entrada de manera que los números en coma flotante se conviertan en enteros.
2. Efectuar los cálculos de los determinantes $\Delta(x_{i+1}, T_j)$ y $\det(T_j)$ (que establecen si el punto x_{i+1} está o no dentro de la esfera circunscrita a T_j y la orientación de este tetraedro, respectivamente) utilizando números enteros, exclusivamente.

⇒ Problemas: Limitación en las cantidades que aparecen en los determinantes, y por tanto, en la longitud del dominio. Aumento en el tiempo de CPU en los cálculos. Persistencia de tetraedros *planos*.

Construcción de $D(X_{i+1})$. Consideraciones Preliminares

- El conjunto D_1^i debe satisfacer unas determinadas condiciones para que $D(X_{i+1})$ sea admisible.
- Definiciones:
 1. Diremos que la cara $C \in C_{f D_1}^i$ es ε -visible desde x_{i+1} si y sólo si es visible desde ese punto, y además:

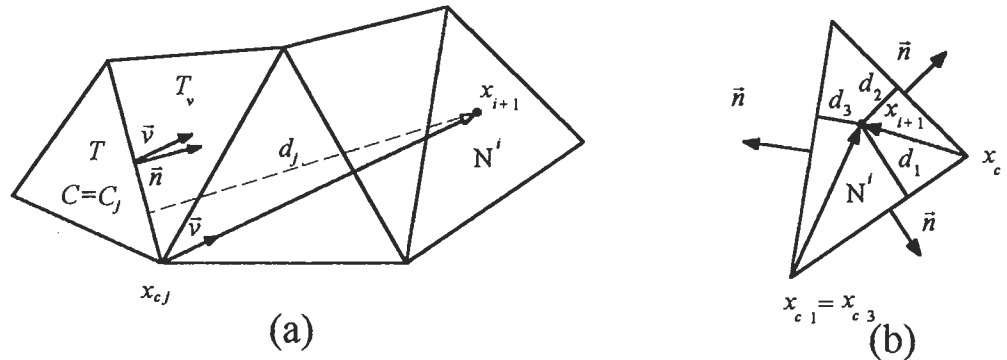
$$\min_{x \in C} \{\cos(\overline{x_{i+1}x}, \vec{n})\} > \varepsilon$$

donde \vec{n} es el vector unitario normal a C y saliente de D_1^i y ε un determinado número positivo.

2. Diremos que un politopo es ε -en forma de estrella desde x_{i+1} si todas sus caras son ε -visibles desde ese punto.
- El conjunto D_1^i debe satisfacer los siguientes requisitos:
 1. Un tetraedro T pertenece a D_1^i sólo si $x_{i+1} \in B(T)$.
 2. El conjunto D_1^i debe ser ε -en forma de estrella desde x_{i+1} .
 3. El conjunto D_1^i no debe contener ningún otro punto de X aparte de x_{i+1} .

Construcción de $D(X_{i+1})$. Búsqueda del Núcleo (I)

- Definimos *núcleo*, N^i , como el tetraedro o conjunto de tetraedros que contienen a x_{i+1} .

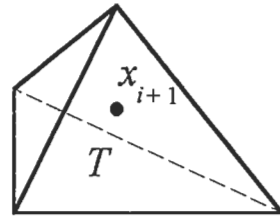


- Algoritmo para encontrar alguno de los tetraedros del núcleo.**

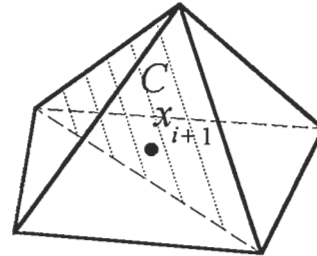
Sea T_u el último de los tetraedros construido en la triangulación.

- $T := T_u$
- Si $(\vec{n} \cdot \vec{v} \leq 0 \quad \forall C \in T)$ entonces $T \in N^i$; fin (figura (b)).
en caso contrario, encontramos la cara C de T que maximiza $\vec{n} \cdot \vec{v}$ (figura (a)).
- Sea T_v el tetraedro que comparte C con T . Hacemos $T := T_v$ y retornamos al paso 2.

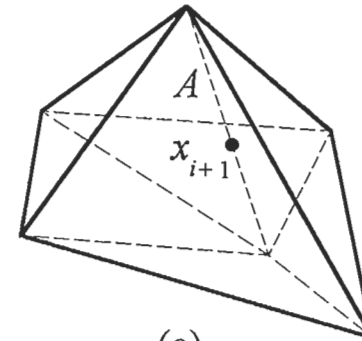
Construcción de $D(X_{i+1})$. Búsqueda del Núcleo (II)



(a)



(b)



(c)

□ Determinación del núcleo:

(a) Si $x_{i+1} \in \overset{\circ}{T}$, el núcleo es T .

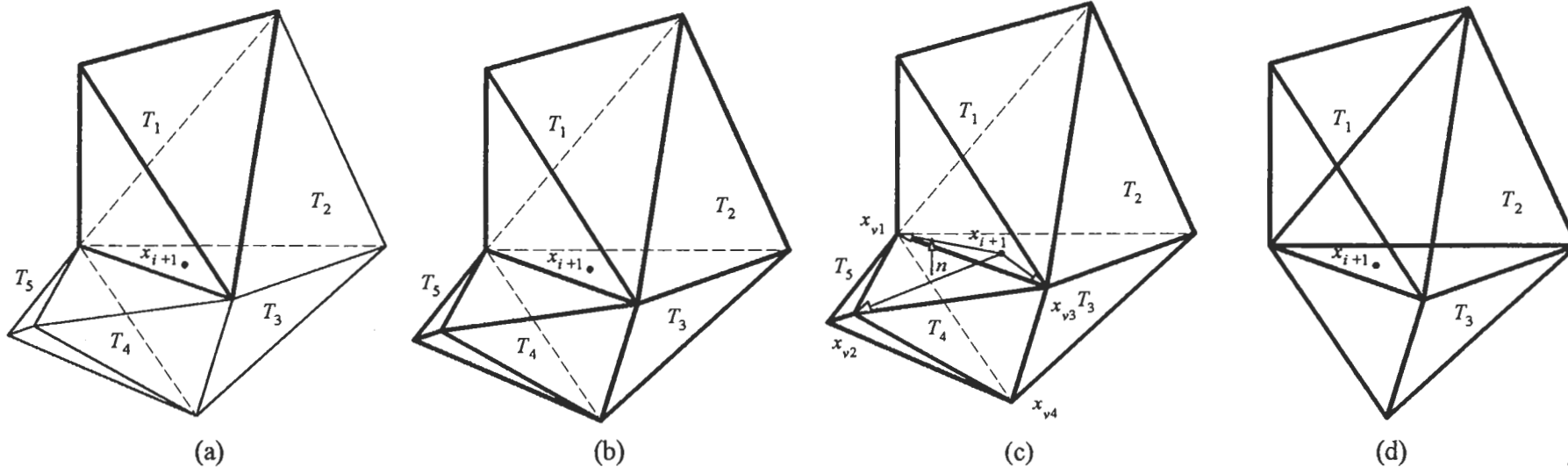
(b) Si $x_{i+1} \in \overset{\circ}{C}$, el núcleo es el par de tetraedros que comparten C .

(c) Si $x_{i+1} \in \overset{\circ}{A}$, el núcleo es el conjunto de tetraedros que comparten A .

□ El núcleo es ε -en forma de estrella respecto del punto x_{i+1} .

□ Encontramos el parámetro ε_N para el que el núcleo es ε -en forma de estrella.

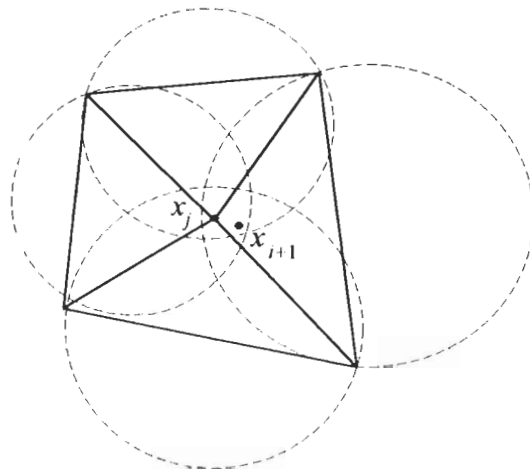
Construcción de $D(X_{i+1})$. Determinación de D_1^i



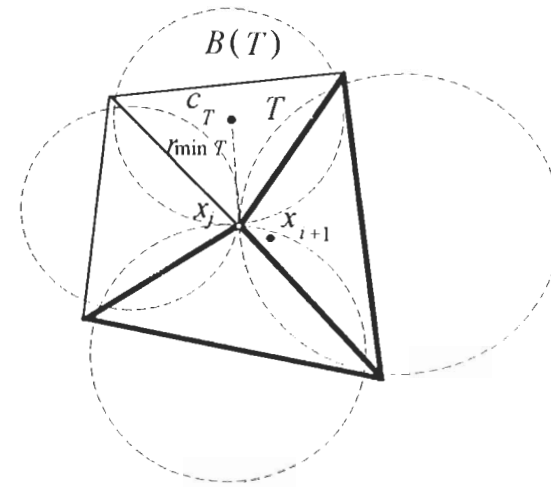
- (a) Núcleo, $N^i = \{T_1\}$, y conjunto de tetraedros, $\{T_1, T_2, T_3, T_4, T_5\}$, cuyas esferas contienen a x_{i+1} .
- (b) Ampliación del núcleo hasta llegar a $D_1^i = \{T_1, T_2, T_3, T_4, T_5\}$.
- (c) El conjunto D_1^i no es ε -en forma de estrella desde x_{i+1} ; por ejemplo, $\cos(\overline{x_{i+1}x_{v2}}, \vec{n}) < 0$.
- (d) El algoritmo comienza de nuevo la determinación de D_1^i . Ahora, el tetraedro T_4 se considera como si su esfera circunscrita, $B(T_4)$, no contuviera a x_{i+1} , y por tanto, el algoritmo se detiene una vez se haya añadido T_3 al conjunto D_1^i .

Construcción de $D(X_{i+1})$. D_1^i sólo debe contener a x_{i+1}

- El algoritmo debe evitar que ningún punto de X , aparte de x_{i+1} , esté contenido en D_1^i .



(a)



(b)

- Los errores de redondeo en el cálculo de las esferas circunscritas pueden hacer que éstas contengan algún punto de X aparte de x_{i+1} (a).
- Para cada tetraedro se calcula el *radio mínimo*, $r_{\min T} = \min\{d(c_T, V_T)\}$, de $B(T)$, donde V_T es cualquiera de los vértices de T y c_T el centro de $B(T)$ (b).
- La determinación de D_1^i se lleva a cabo con estos valores de los radios.

Algoritmo para la determinación de D_1^i

Se expande el núcleo con ciertos controles hasta que se obtiene D_1^i .

Consideremos $B^i = \{B(T) / T \in D(X_i)\}$ y $R_{B(T)}$ el radio de $B(T)$.

procedimiento CONJUNTO D_1^i

entrada: $D(X_i), B^i, N^i, x_{i+1}, \varepsilon, \varepsilon_N$; salida: D_1^i

$D_1^i := N^i$

hallamos $ND_1^i = \{T \text{ adyacentes a } N^i / x_{i+1} \in B(T)\}$.

mientras ($ND_1^i \neq \emptyset$) **hace**

$D_1^i := D_1^i \cup ND_1^i$ (* se actualiza el conjunto D_1^i *)

hallamos $ND_1^i = \{T \text{ adyacentes a } D_1^i / x_{i+1} \in B(T)\}$.

fin mientras

si (D_1^i no es ε -en-forma-de-estrella respecto de x_{i+1}) **entonces**

hallamos $C_m = \{C \in \text{frontera de } D_1^i / C \text{ no es } \varepsilon\text{-visible}\}$.

hallamos $T_m = \{T \in D_1^i / \text{alguna cara de } T \in C_m\}$.

si ($\exists T \in T_m \cap N^i$) **entonces** (* el valor actual de ε es demasiado restrictivo. Se substituye por ε_N , que es menor*)

$\varepsilon := \varepsilon_N$

CONJUNTO D_1^i entrada: $D(X_i), B^i, N^i, x_{i+1}, \varepsilon, \varepsilon_N$; salida: D_1^i

(* el algoritmo comienza de nuevo con un ε menor*)

si no

$R_{B(T)} := 0, \forall T \in T_m$ (* esto evita que el tetraedro T sea incluido de nuevo en D_1^i *)

CONJUNTO D_1^i entrada: $D(X_i), B^i, N^i, x_{i+1}, \varepsilon, \varepsilon_N$; salida: D_1^i

(* el algoritmo comienza de nuevo considerando que las "esferas circunscritas" a T_m tienen radios nulos *)

fin si

fin si

se restauran los valores iniciales de aquellos radios que hayan sufrido variación durante el proceso.

fin procedimiento.

Construcción de D_2^i

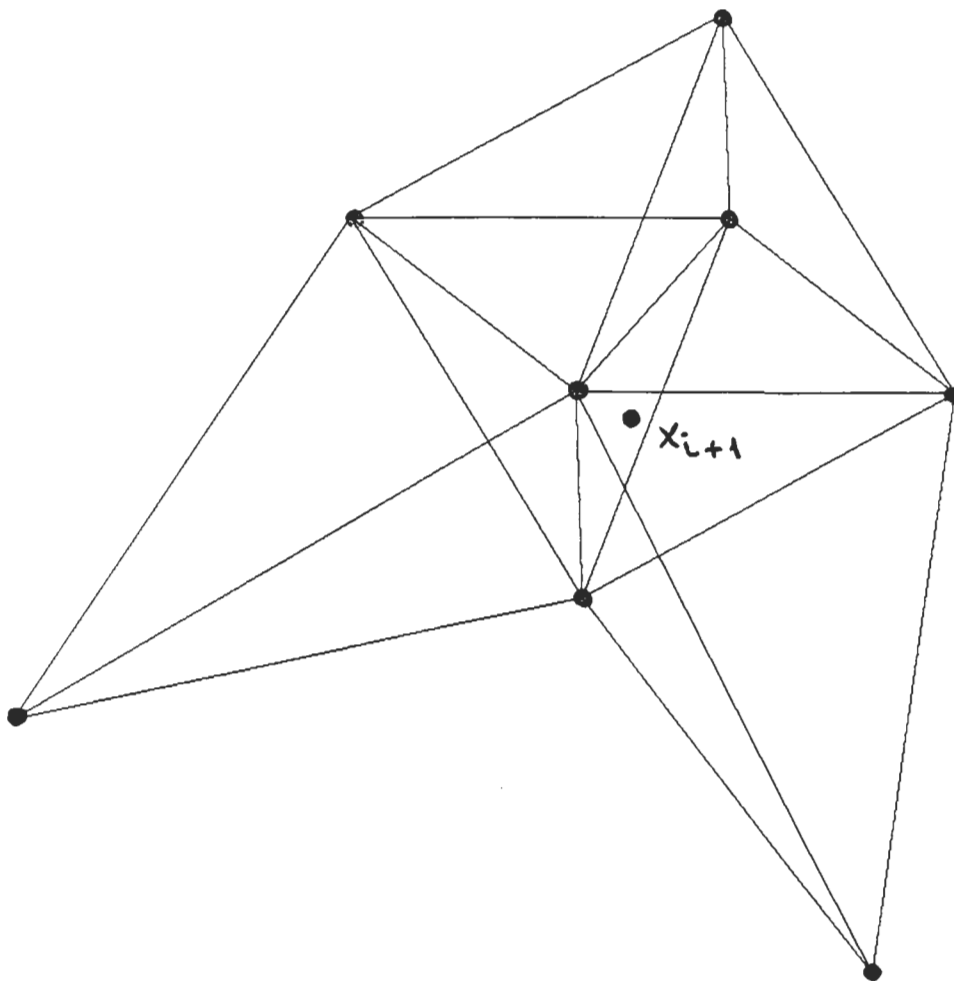
Una vez determinados:

- $D_1^i, C_1^i, A_1^i \in D(X_i)$: conjuntos de tetraedros, caras y aristas que desaparecen al añadir x_{i+1} .

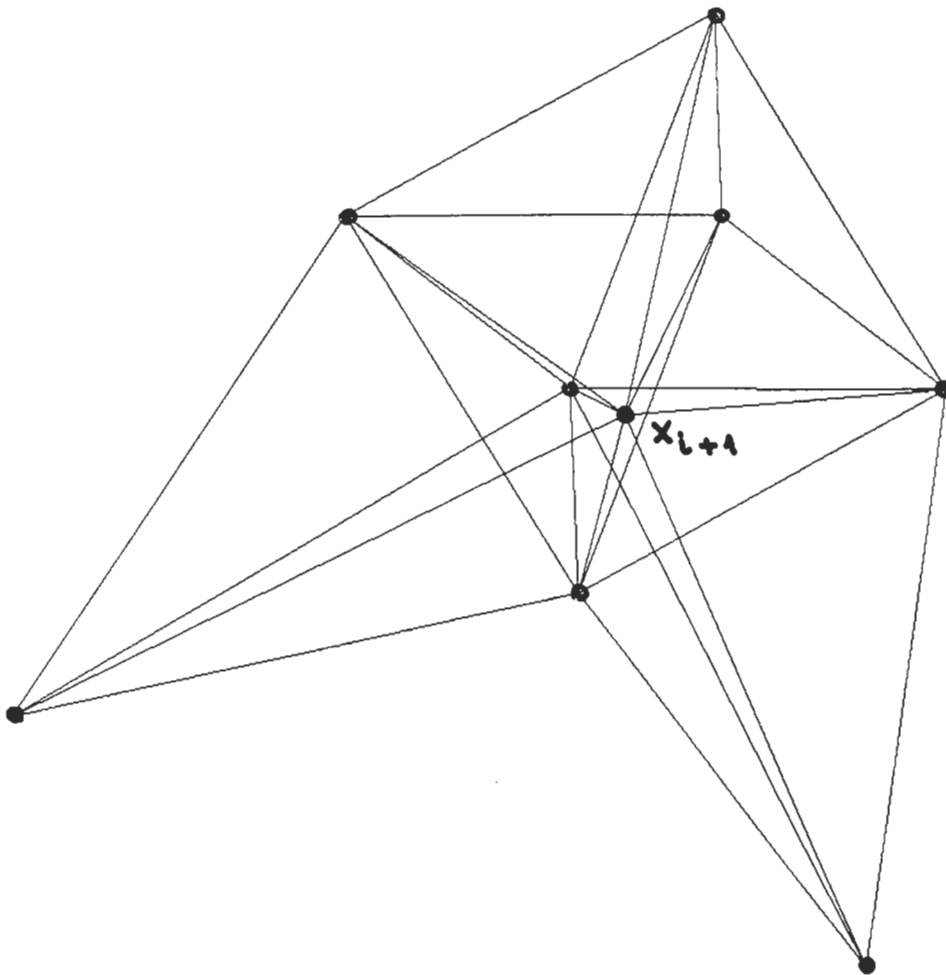
Construimos:

- $D_2^i, C_2^i, A_2^i \in D(X_{i+1})$: conjuntos de tetraedros, caras y aristas formadas a partir de x_{i+1} .
 - El conjunto D_2^i se forma con el punto x_{i+1} y las caras de la frontera de D_1^i .
 - El conjunto C_2^i se forma con el punto x_{i+1} y las aristas de la frontera de D_1^i .
 - El conjunto A_2^i se forma con el punto x_{i+1} y los vértices de la frontera de D_1^i .
- Se restauran las conexiones entre estos nuevos elementos.
- La triangulación de $D(X_{i+1})$ concluye.

Conjunto de tetraedros $T_i \equiv D_A^i$



Conjunto de tetraedros $T2i \equiv D_2^i$



Estructura de datos del programa (I)

Cada punto, arista, cara y tetraedro están representados por un nodo, nv , na , nc y nt , respectivamente.

Coordenadas de los puntos de X : $x(nv)$, $y(nv)$, $z(nv)$.

Coordenadas del centro y radio de la esfera circunscrita a cada tetraedro: $x_e(nt)$, $y_e(nt)$, $z_e(nt)$, $r_e(nt)$.

Descripción de los elementos:

1. Descripción del tetraedro por sus caras: $CT(i,nt) = nc \quad (1 \leq i \leq 4)$.
El vector CT proporciona las cuatro caras del tetraedro nt .
2. Descripción del tetraedro por sus vértices: $VT(i,nt) = nv \quad (1 \leq i \leq 4)$.
El vector VT proporciona los cuatro vértices del tetraedro nt .
3. Descripción de la cara por sus aristas: $AC(i,nc) = na \quad (1 \leq i \leq 3)$.
El vector AC proporciona las tres aristas de la cara nc .
4. Descripción de la cara por sus vértices: $VC(i,nc) = nv \quad (1 \leq i \leq 3)$.
El vector VC proporciona los tres vértices de la cara nc .
5. Descripción de la arista por sus vértices: $VA(i,na) = nv \quad (1 \leq i \leq 2)$.
El vector VA proporciona los dos vértices de la arista na .

⇒ Los vértices de estos elementos son los puntos de la triangulación, es decir, los puntos de X .

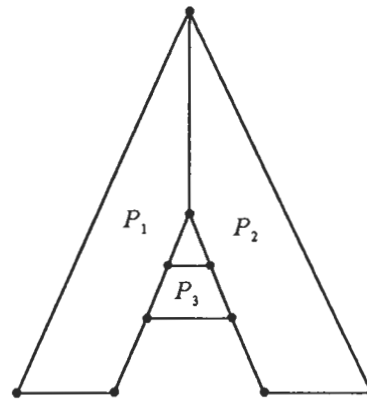
Estructura de datos del programa (II)

Conexiones entre elementos:

1. Tetraedros que comparten una cara: $TCC(i,nc) = nt$ ($1 \leq i \leq 2$).
El vector TCC nos proporciona los tetraedros, nt , que comparten la cara nc . ($TCC(i,nc) = 0$, para $i=1$ ó 2 , si la cara nc está sobre la envolvente exterior del dominio)
 2. Caras que comparten una arista: $CCA(i,na) = nc$ ($1 \leq i \leq tcca(na)$).
El vector CCA nos proporciona las caras, nc , que comparten la arista na . El vector $tcca$ indica el total de caras adyacentes a la arista na .
 3. Aristas que comparten un vértice: $ACV(i,nv) = na$ ($1 \leq i \leq tacv(nv)$).
El vector ACV nos proporciona las aristas, na , que comparten el vértice nv . El vector $tacv$ indica el total de aristas adyacentes al vértice nv .
- ⇒ Ventajas: Es una estructura de datos es muy versátil. Por ejemplo, permite la aplicación del M.E.F con elementos basados en las aristas (edge elements), de gran utilidad en electromagnetismo. Facilita enormemente el ensamblaje de las matrices elementales en forma compacta.
- ⇒ Inconvenientes: Gran gasto de memoria, especialmente si el lenguaje de programación no permite el dimensionamiento dinámico.

Definición del Dominio y Generación Automática de Puntos (I)

- La triangulación, $T(\Omega)$, del dominio queda determinada cuando se generan puntos sobre la superficie e interior de Ω con unas densidades apropiadas.
- La triangulación del dominio se obtiene sustrayendo de la triangulación total, $D(X)$, los tetraedros que no pertenecen a $T(\Omega)$.



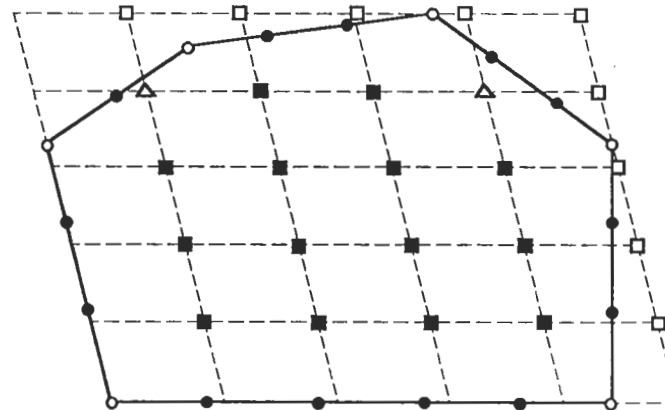
- Para simplificar la generación de puntos, el dominio Ω se descompone en poliedros convexos.
- Los poliedros convexos se definen por sus caras, las caras (convexas) por sus aristas y las aristas por sus vértices.
- El programa genera puntos automáticamente en el interior relativo de cada arista, cara y poliedro.
- La **distancia entre puntos** se especifica previamente en la entrada de datos.

GENERACIÓN DE PUNTOS SOBRE LÍNEAS, SUPERFICIES Y REGIONES

□ LÍNEAS.



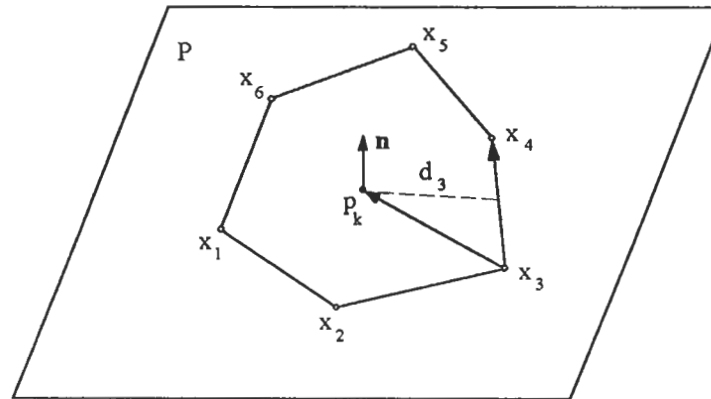
□ SUPERFICIES PLANAS POLIGONALES CONVEXAS.



Origen del sistema
de referencia

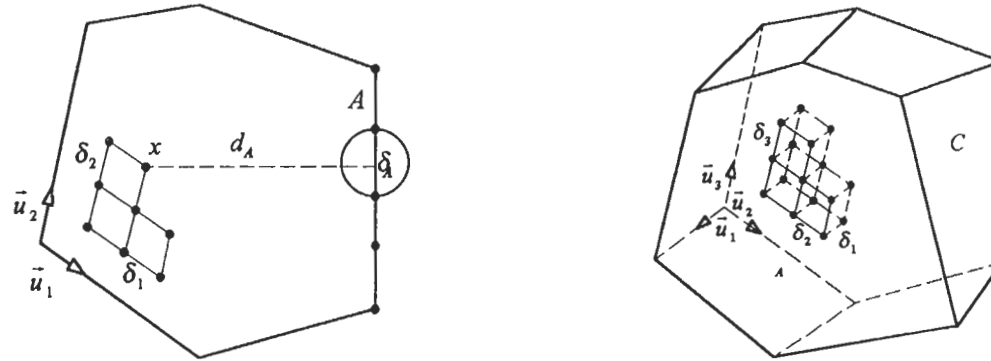
□ REGIONES POLIÉDRICAS CONVEXAS: De forma análoga a las superficies.

PUNTOS INTERIORES A SUPERFICIES O REGIONES. CRITERIO



- Sea $\{x_1, x_2, \dots, x_n\}$ el conjunto de vértices de una poligonal convexa, ordenados cíclicamente.
- Sea n el vector unitario normal a la superficie ($n = x_1x_2 \times x_2x_3 / \|x_1x_2 \times x_2x_3\|$).
- El punto p_k está dentro de la poligonal $\Leftrightarrow \forall i=1,2, \dots, n$ resulta $d_i = [(\mathbf{n} \times x_i x_{i+1}) \cdot x_i p_k / \|x_i x_{i+1}\|] > 0$ considerando $x_{n+1} = x_1$; la cantidad d_i representa la distancia a cada lado de la poligonal.
- ☞ Una idea **similar** se utiliza para generar puntos en el interior de las regiones poliédricas convexas en que se descompone el objeto.

Definición del Dominio y Generación Automática de Puntos (II)



□ **Generación de puntos en las aristas.**

La distancia δ_A entre puntos interiores de una arista A se especifica en la entrada de datos.

□ **Generación de puntos en las caras.**

Los puntos interiores se generan en las direcciones de \vec{u}_1 y \vec{u}_2 con unas determinadas distancias entre ellos, δ_1 y δ_2 , respectivamente.

Si $d_A > 1/2\delta_A$, la arista A no es eliminada por efecto de los puntos interiores de la cara.

□ **Generación de puntos interiores en los poliedros.**

Los puntos interiores se generan en las direcciones de \vec{u}_1 , \vec{u}_2 y \vec{u}_3 con unas determinadas distancias entre

ellos. Si la distancia de los puntos interiores a la cara C es mayor que $d_C = \frac{1}{2}\sqrt{\delta_1^2 + \delta_2^2} + \frac{1}{2}\max\{\delta_A\}$,

donde δ_A es la distancia entre puntos en cada una de las aristas de C , entonces esta cara no es eliminada por efecto de los puntos interiores de poliedros adyacentes.

CONSERVACIÓN DE LA GEOMETRÍA DEL OBJETO

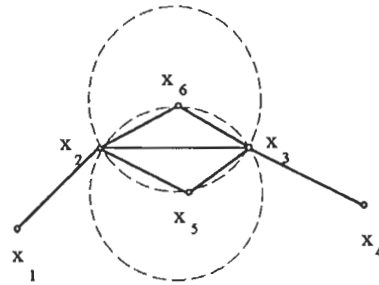


Fig.1

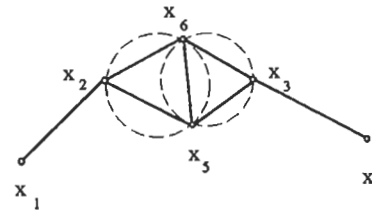


Fig.2

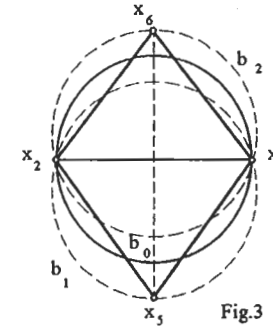


Fig.3

- La triangulación de la fig.1 no es una triangulación de Delaunay ya que no respeta la propiedad de las esferas vacías (según la cual, las esferas circunscritas a los tetraedros no contienen ningún punto de la triangulación). En cambio, la de la fig. 2 sí lo es.
- Supongamos que los segmentos (x_1, x_2) , (x_2, x_3) y (x_3, x_4) pertenecen a la geometría del objeto y que se deben insertar los puntos x_5 y x_6 a ambos lados del segmento (x_2, x_3) .
- Cuando se inserte el punto x_6 en la triangulación el segmento (x_2, x_3) desaparecerá dando lugar al segmento (x_5, x_6) . No se ha respetado la geometría del objeto.
- Una condición suficiente para conservar el segmento (x_2, x_3) es que las distancias de los puntos x_5 y x_6 a dicho segmento sea mayor que la mitad de la distancia entre x_2 y x_3 .
- Si los puntos x_5 y x_6 **cumplen** la condición anterior se cumple:
 $x_5 \text{ y } x_6 \notin b_0 \Rightarrow x_5 \notin b_2 \text{ y } x_6 \notin b_1$ y por tanto la triangulación de Delaunay de los puntos x_2, x_3, x_5 y x_6 conserva el segmento (x_2, x_3) .

Clasificación de los elementos de la triangulación (I)

OBJETIVO: Una vez realizada la triangulación $D(X)$, a cada punto, arista, cara y tetraedro pertenecientes al dominio se le debe asignar un número de referencia que permita imponer las condiciones de contorno oportunas o asociarle un determinado tipo de material.

REALIZACIÓN: Supongamos que V_i , C_i , y A_i son, respectivamente, los subvolúmenes convexos, superficies convexas y segmentos en los que se descompone el dominio. A cada uno de estos elementos se le ha asignado, en la entrada de datos, un número de referencia que posteriormente puede ser interpretado como una condición de contorno, un tipo de material, etc.

Clasificación de los tetraedros de la triangulación:

$$\forall t \in D(X) \begin{cases} t \notin \text{ningún } V_i \Rightarrow t \notin \text{dominio} ; \text{ } rnt(t) := 0 \\ t \in V_i ; \begin{cases} \text{asignamos a } t \text{ el número de referencia de } V_i \\ \text{almacenamos los } n^{\circ} \text{ globales de estos tetraedros: } n := n + 1 ; rnt(t) := n \end{cases} \end{cases}$$

Clasificación de las caras de la triangulación:

$$\forall c \in D(X) \begin{cases} c \notin \text{ningún } C_i \Rightarrow c \notin \text{frontera del dominio ni a ninguna interfase entre materiales} ; rnc(c) := 0 \\ c \in C_i ; \begin{cases} \text{asignamos a } c \text{ el número de referencia de } C_i \\ \text{almacenamos el } n^{\circ} \text{ global de estas caras: } n := n + 1 ; rnc(c) := n \end{cases} \end{cases}$$

Clasificación de los elementos de la triangulación (II)

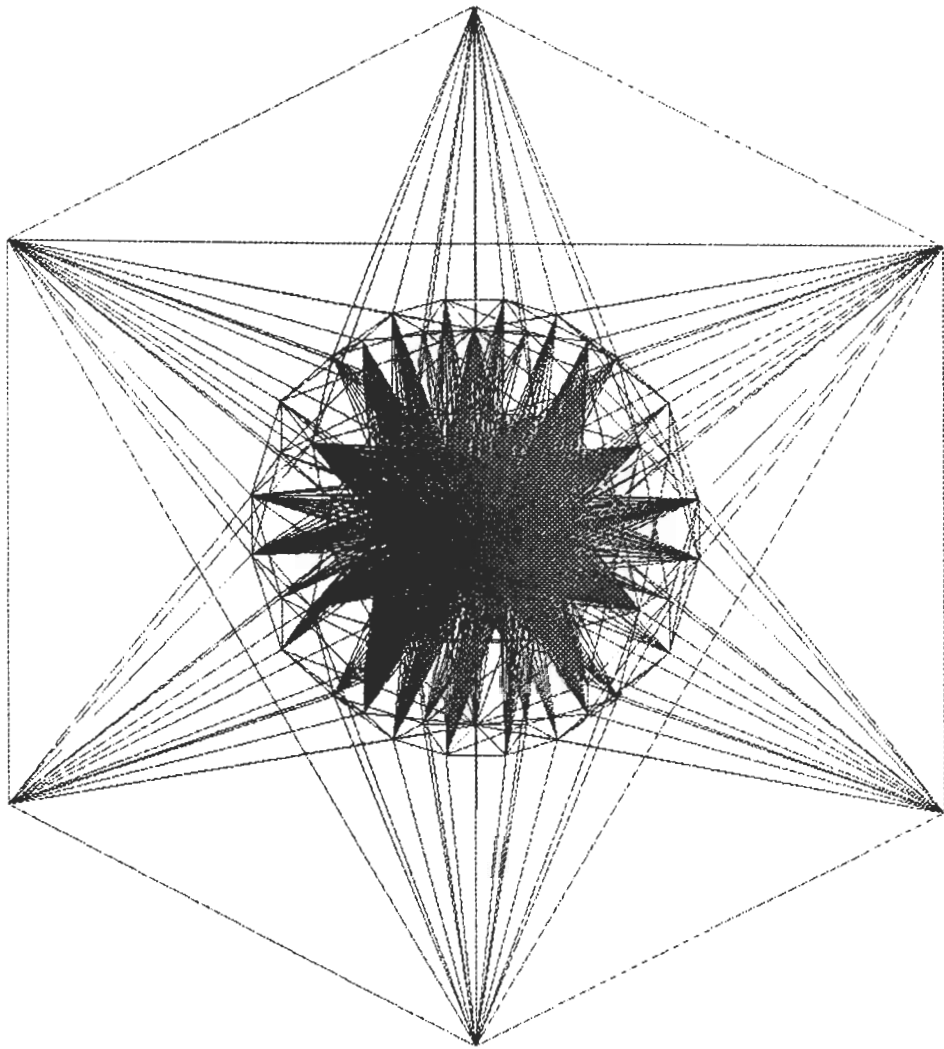
Clasificación de las aristas de la triangulación:

$$\forall a \in D(X) \left\{ \begin{array}{l} a \notin \text{ningún } A_i, C_i, V_i \Rightarrow a \notin \text{al dominio, } rna(a) := 0 \\ a \in A_i; \left[\begin{array}{l} \text{asignamos } a \text{ a el número de referencia de } A_i \\ \text{almacenamos el } n^o \text{ global de estas aristas: } n := n + 1; rna(a) := n \end{array} \right. \\ a \in C_i; \left[\begin{array}{l} \text{asignamos } a \text{ a el número de referencia de } C_i \\ \text{almacenamos el } n^o \text{ global de estas aristas: } n := n + 1; rna(a) := n \end{array} \right. \\ a \in V_i; \left[\begin{array}{l} \text{asignamos } a \text{ a el número de referencia de } V_i \\ \text{almacenamos el } n^o \text{ global de estas aristas: } n := n + 1; rna(a) := n \end{array} \right. \end{array} \right.$$

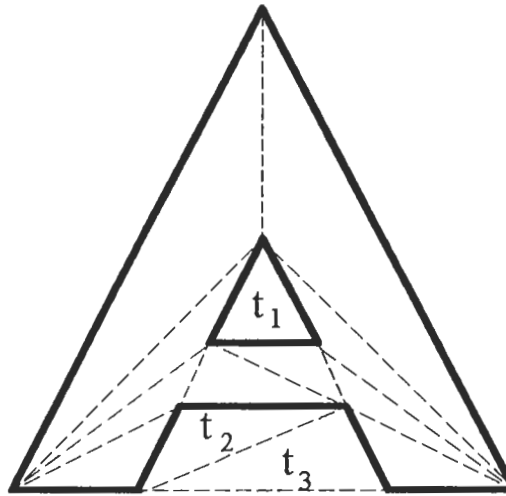
Clasificación de los puntos de la triangulación:

$$\forall v \in D(X) \left\{ \begin{array}{l} v \in \text{a alguno de los vértices de la envolvente del dominio} \Rightarrow v \notin \text{dominio} \\ v \notin \text{a ninguno de los vértices de la envolvente del dominio} \Rightarrow v \in \text{dominio} \end{array} \right.$$

Nota: Para el cálculo de las eventuales integrales de línea que pudieran aparecer en el M.E.F., sólo se necesitaría conocer las aristas pertenecientes a alguno de los segmentos A_i . No obstante, cuando se utilizan elementos basados en las aristas (edge elements) éstas hacen el papel de nodos, y por tanto se hace necesaria su total clasificación.



ELIMINACIÓN DE LOS TETRAEDROS NO PERTENECIENTES AL OBJETO



- El algoritmo de Watson crea una triangulación de la envuelta convexa de la nube de puntos. Es por tanto necesario eliminar los tetraedros exteriores al objeto.
- Para suprimir estos tetraedros hallamos el centro geométrico de cada tetraedro y mediante un procedimiento similar al empleado para situar puntos dentro del objeto, localizamos aquellos que estén fuera del mismo.
- En la figura superior los triángulos t_1 , t_2 y t_3 tienen sus centros fuera del objeto, por tanto serán eliminados de la triangulación final.

Programas principales

PROGRAMA "ENTRADA"

- *Lee los datos que definen la geometría del dominio.*
- *Genera automáticamente los puntos de la triangulación*



PROGRAMA "MALLA"

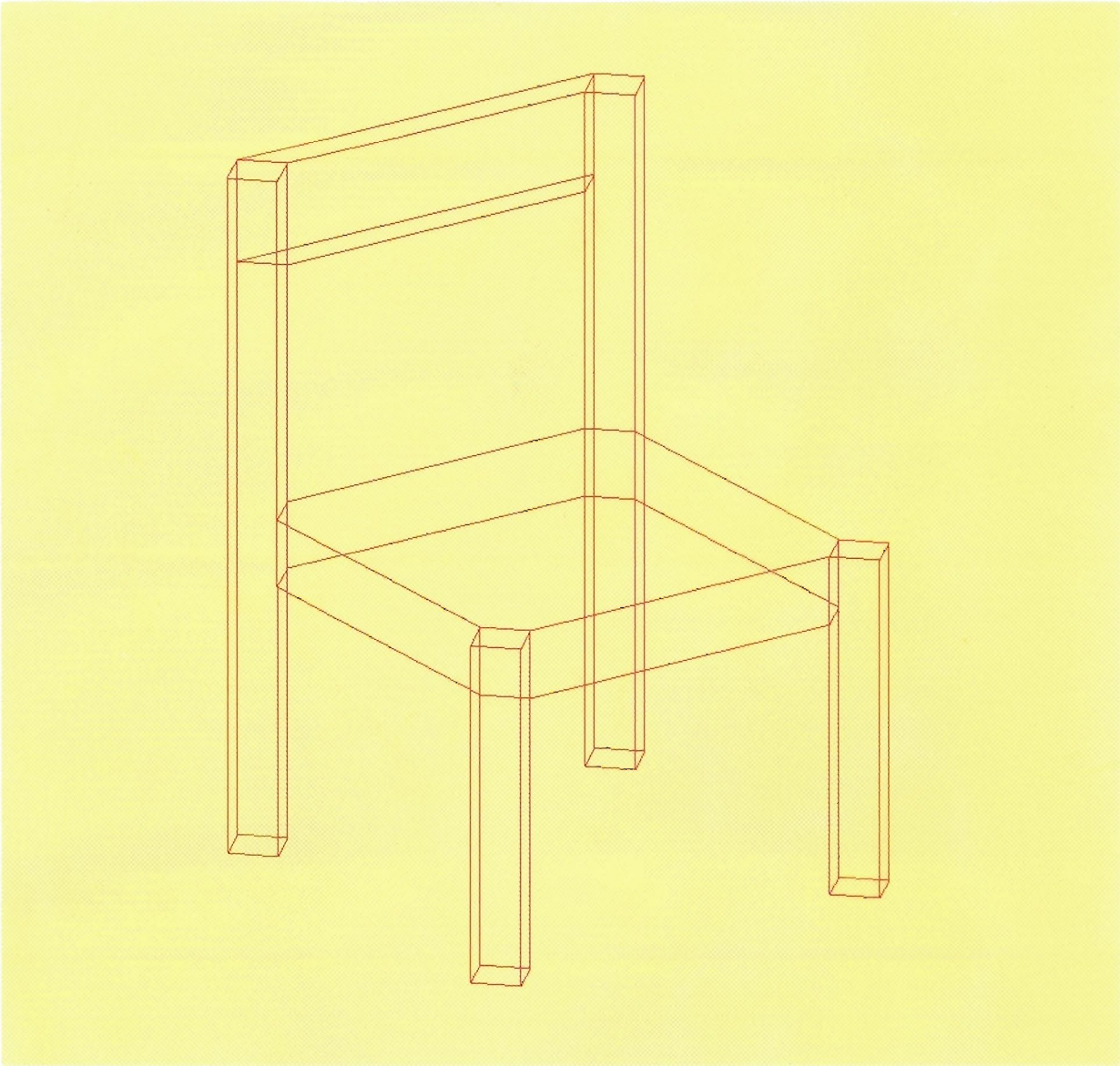
- *Genera la triangulación de los puntos creados por "ENTRADA"*

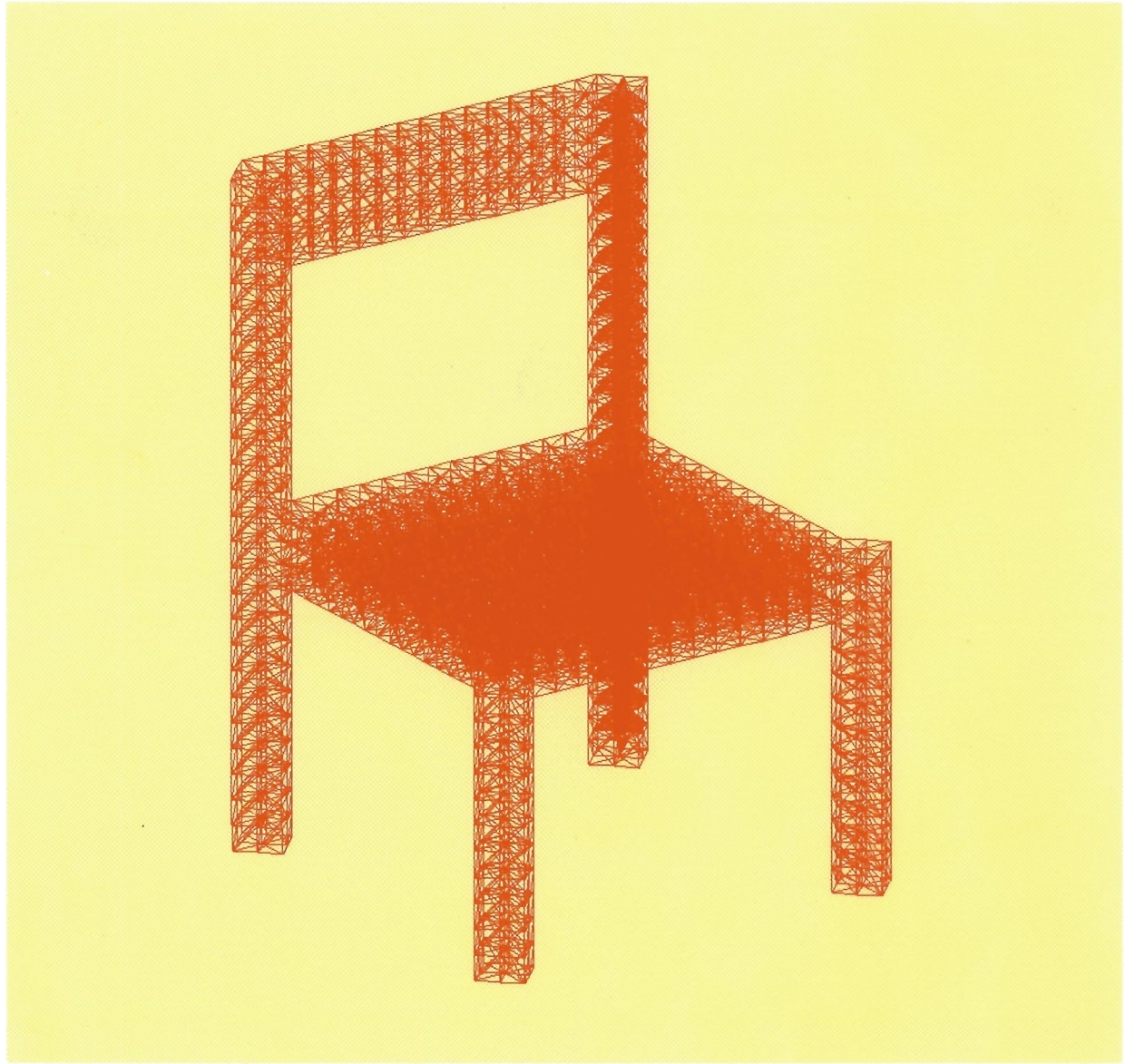


PROGRAMA "CLASIFICA"

- *Establece cuales son los tetraedros, caras y aristas de la triangulación que pertenecen al dominio.*
- *Asigna numeros dereferencia a tetraedros, caras y aristas para imponer condiciones de contorno*
- *Genera una salida de datos apta para la aplicación del M.E.F. y la representación gráfica de la triangulación*





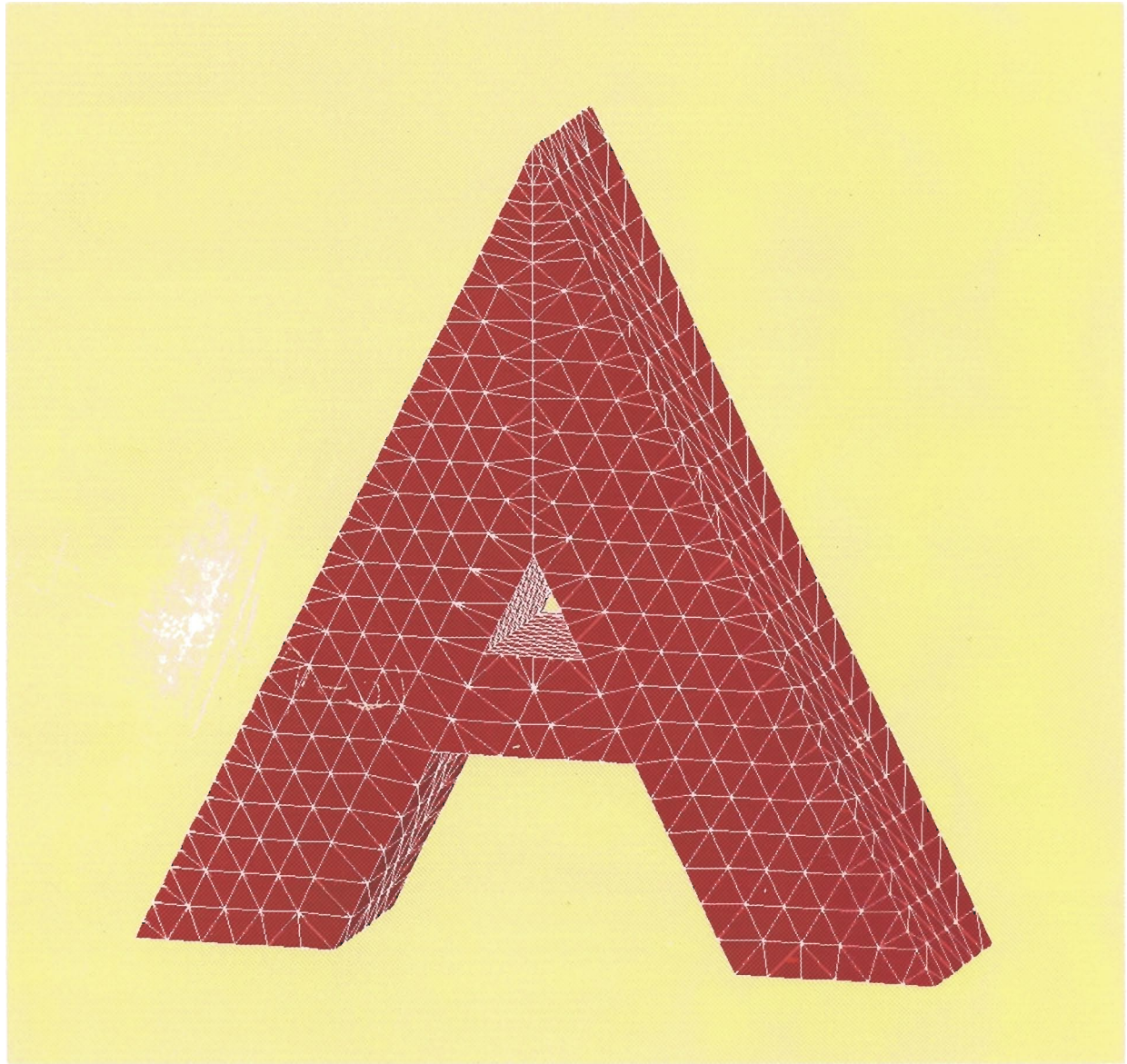


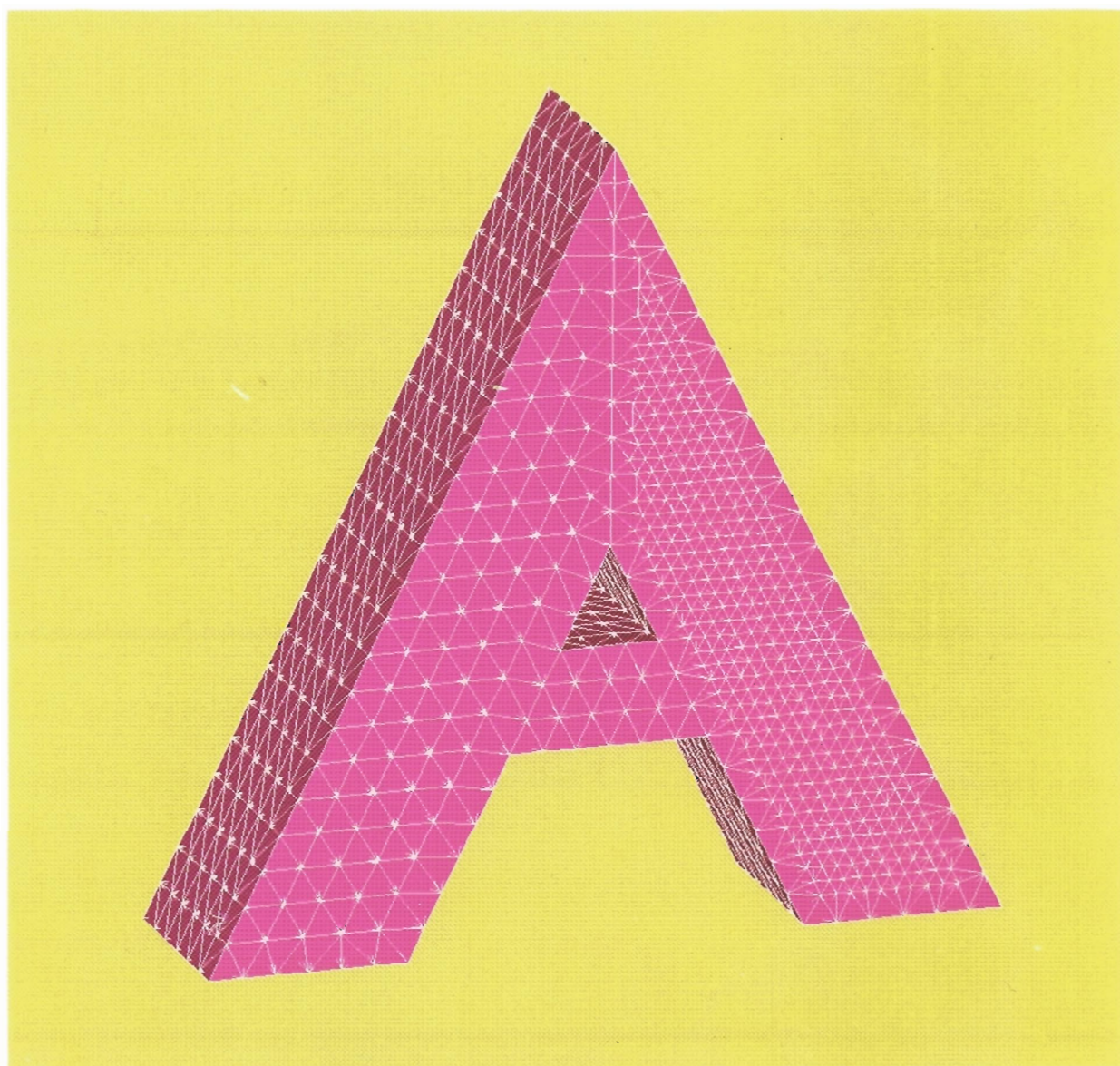




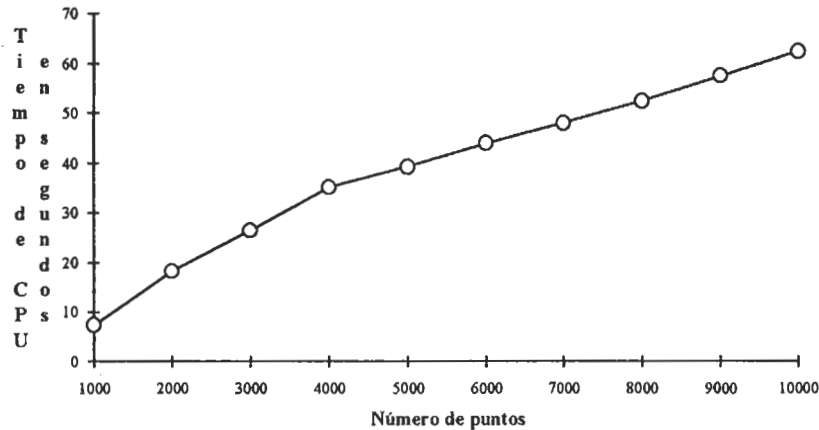
Problema de Triangulación No conforme con
la frontera del dominio



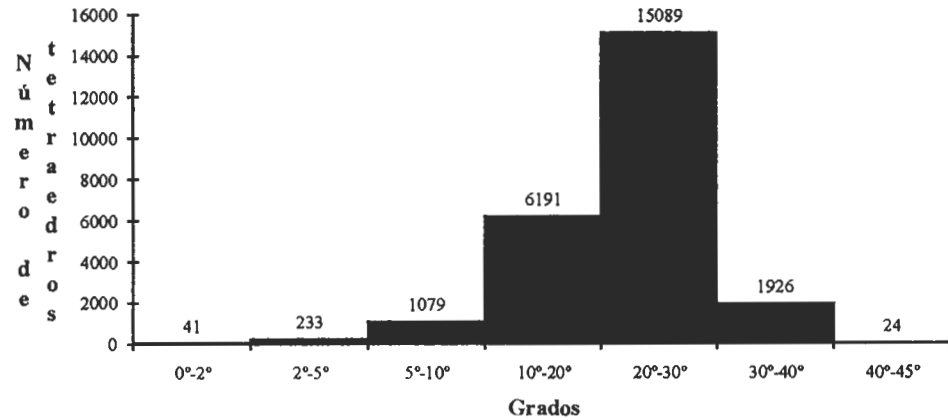




Tiempo de CPU y Medida de Calidad de la Triangulación



(a)



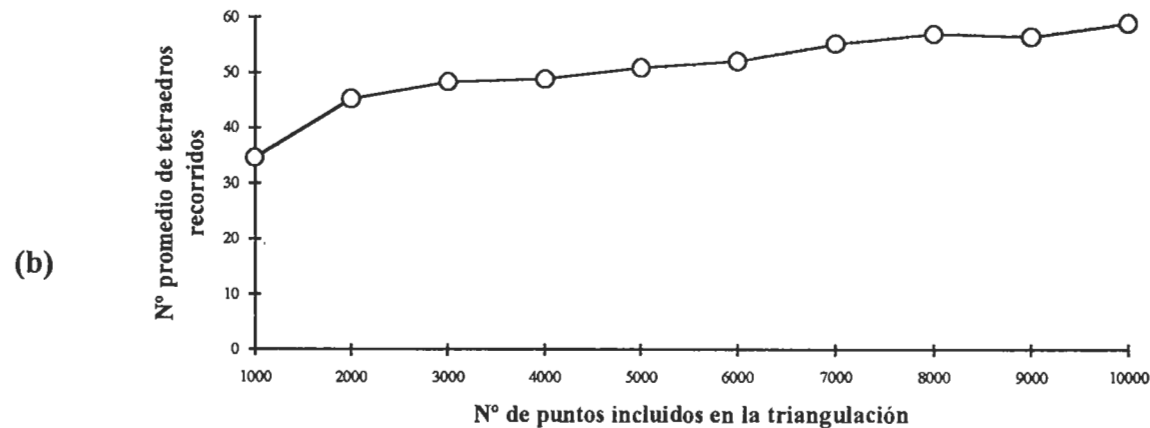
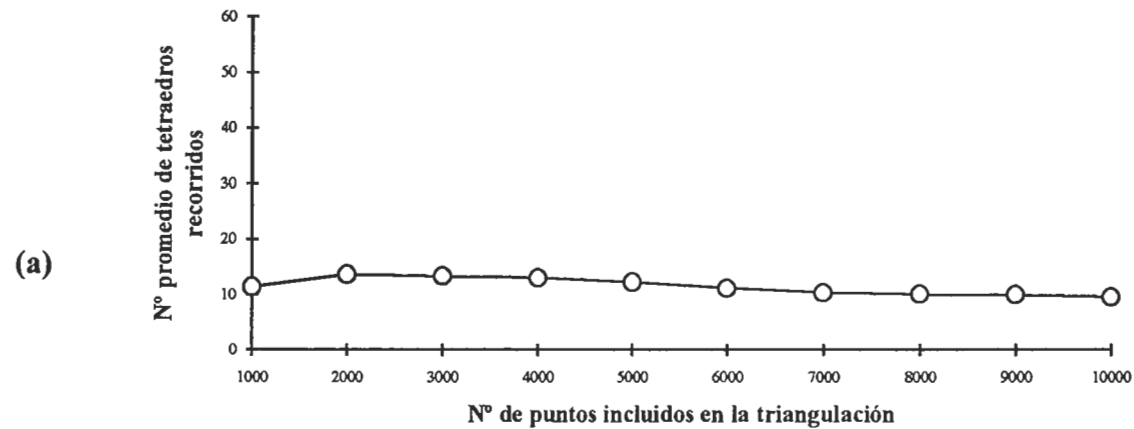
(b)

- Figura (a): Tiempo de CPU en segundos en función del número de puntos en una estación de trabajo HP-730. Aproximadamente se obtiene una complejidad lineal en función del número de puntos.
- Figura (b): Número de tetraedros frente al ángulo ϕ_p en grados. Se puede dar una medida de la calidad de un tetraedro T mediante la expresión $\min_{1 \leq p \leq 4} \{\phi_p\}$, donde:

$$\phi_p = \text{sen}^{-1}(1 - \cos^2 \alpha_p - \cos^2 \beta_p - \cos^2 \gamma_p + 2 \cos \alpha_p \cos \beta_p \cos \gamma_p)^{1/2}$$

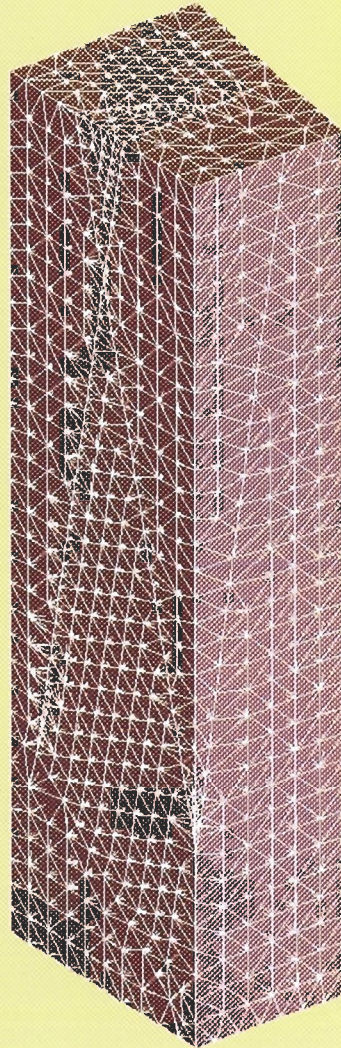
y α_p , β_p , y γ_p son los ángulos planos correspondientes al vértice p . El valor de ϕ_p es 45° para un tetraedro equilátero y 0° para uno plano.

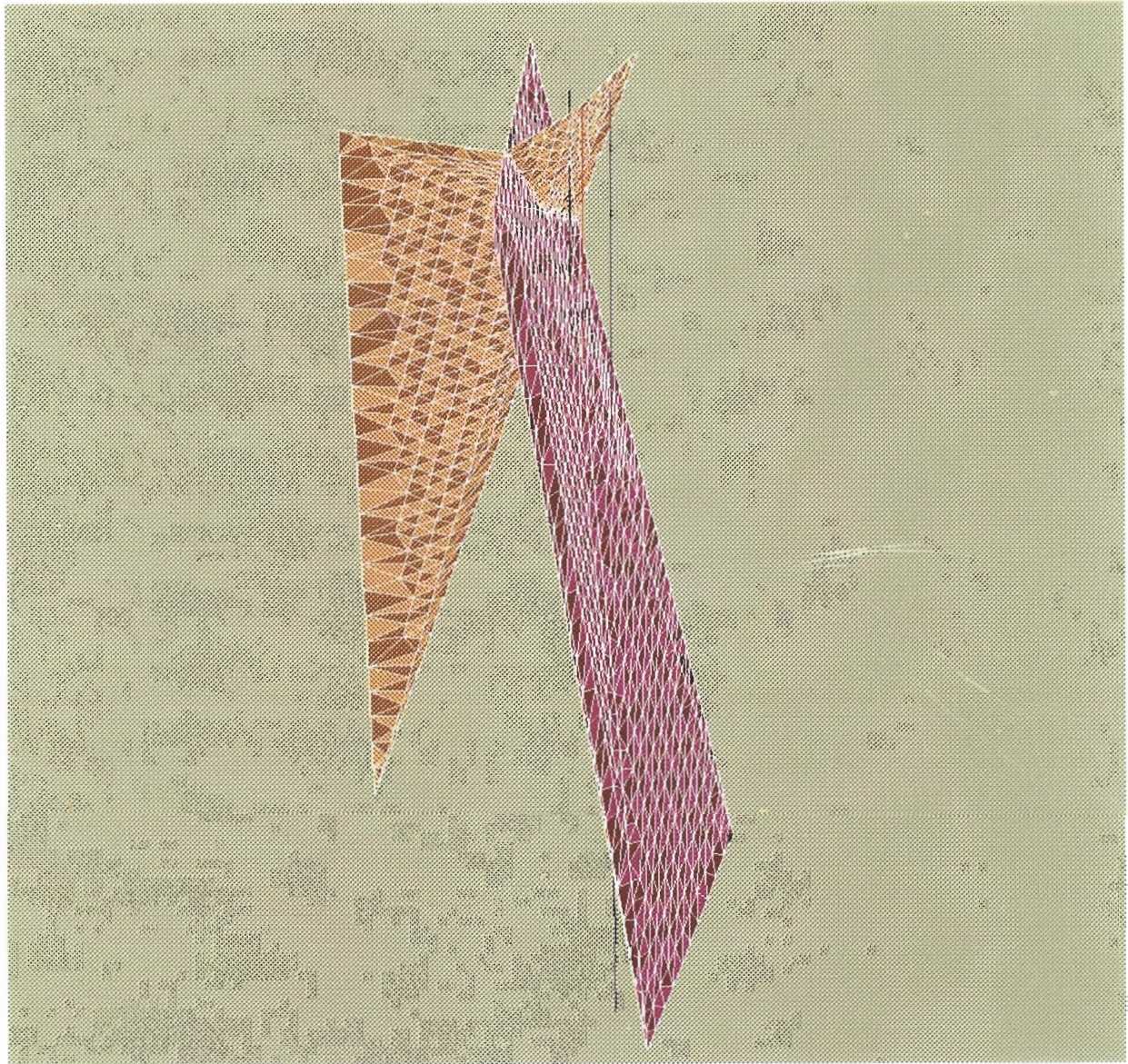
Características del Algoritmo de Búsqueda del Núcleo

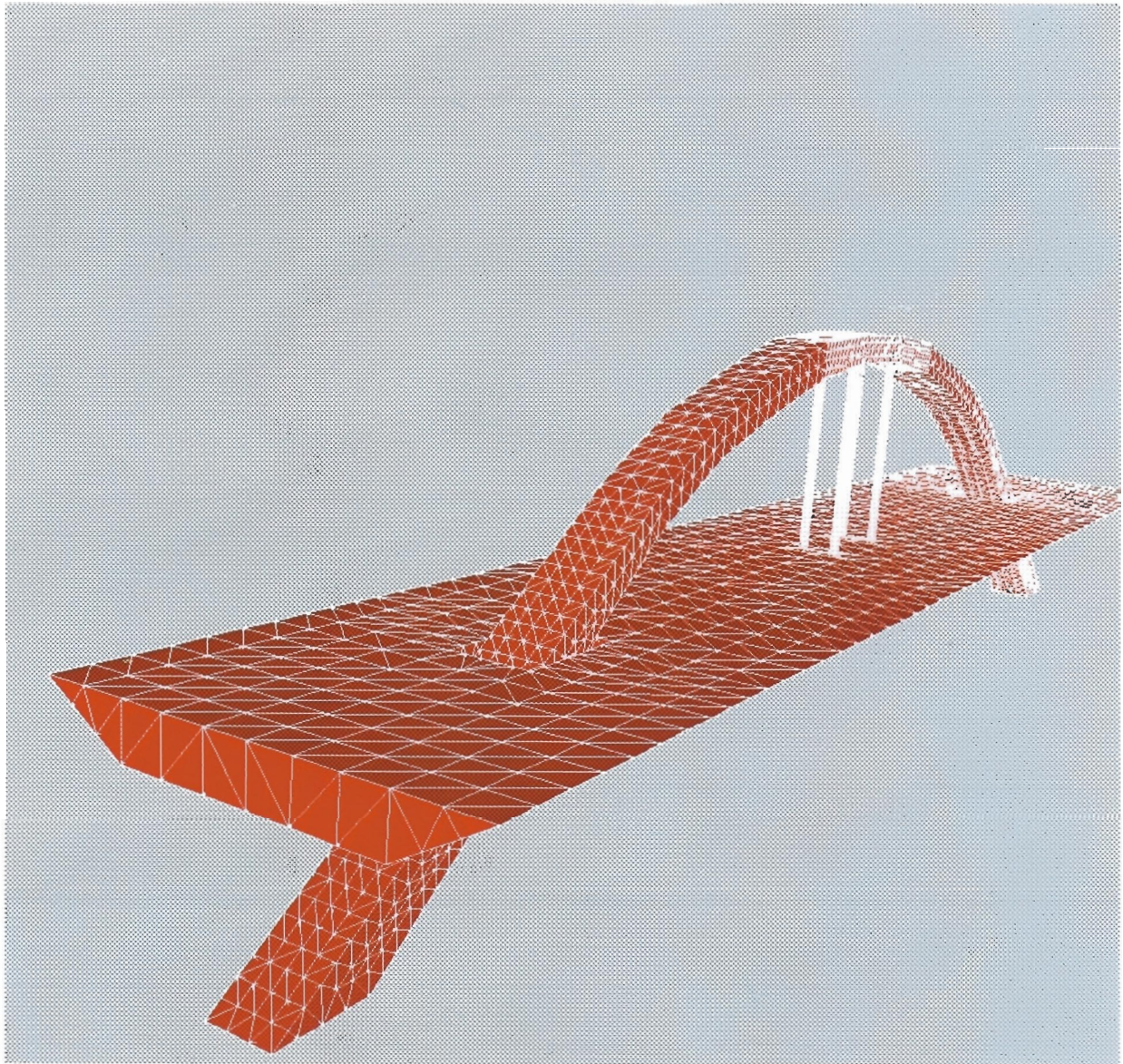


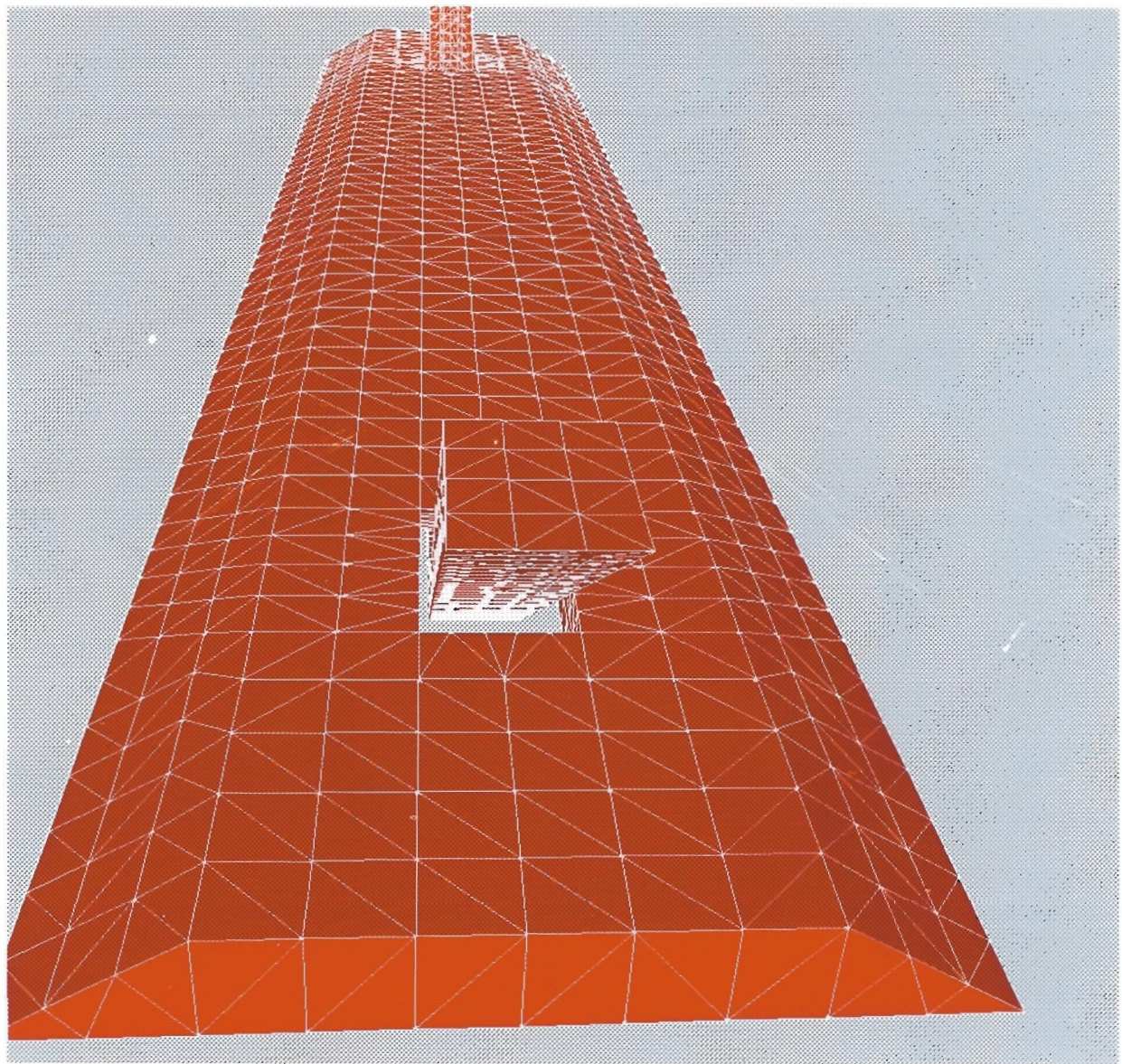
- Nº promedio de tetraedros recorridos para encontrar el núcleo, en función del nº de puntos. Empezando por el último tetraedro incluido en la triangulación (a) y empezando por el nº 1 (b).

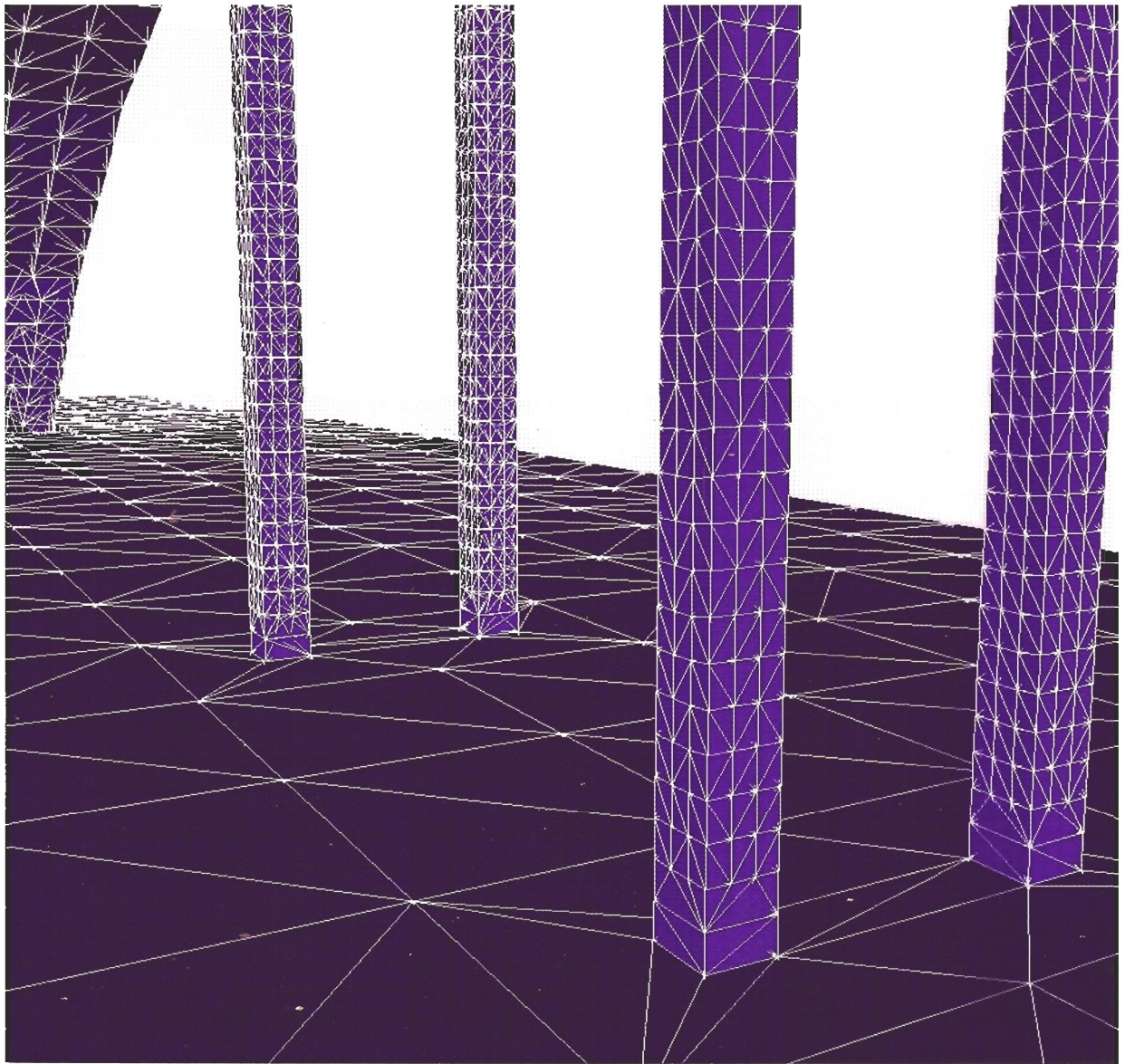
DOMINIO CON FRACTURAS Y SONDEOS











Almacenamiento compacto de la matriz

Ejemplo de almacenamiento compacto de la matriz del sistema

Matriz del sistema		Matriz compacta		Matriz de posiciones
$\begin{pmatrix} k_{11} & k_{12} & 0 & k_{14} & 0 & 0 \\ k_{21} & k_{22} & 0 & 0 & 0 & 0 \\ 0 & k_{32} & k_{33} & k_{34} & 0 & 0 \\ k_{41} & 0 & k_{43} & k_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{55} & k_{56} \\ 0 & 0 & 0 & 0 & k_{65} & k_{66} \end{pmatrix}$	\longleftrightarrow	$\begin{pmatrix} k_{11} & k_{12} & k_{14} \\ k_{22} & k_{21} & \\ k_{33} & k_{32} & k_{34} \\ k_{44} & k_{41} & k_{43} \\ k_{55} & k_{56} & \\ k_{66} & k_{65} & \end{pmatrix}$	\cup	$\begin{pmatrix} 3 & 2 & 4 \\ 2 & 1 & \\ 3 & 2 & 4 \\ 3 & 1 & 3 \\ 2 & 6 & \\ 2 & 5 & \end{pmatrix}$

⇒ El generador de malla suministra la información necesaria para construir la matriz de posiciones **sin hacer búsquedas en toda la malla.**

CONCLUSIONES Y PERSPECTIVAS FUTURAS

CONCLUSIONES.

1. El algoritmo de refinamiento/desrefinamiento de mallas encajadas proporciona:

- Facilidad para la aplicación del método multimalla.
- Mallas flexibles con control automático del número de nodos (incógnitas).
- Adaptación muy rápida de la malla (complejidad lineal).
- Un buen método para aproximar cualquier solución inicial o geometría.
- Un refinamiento local (desrefinando después de un refinamiento global).

2. El algoritmo de triangulación tridimensional proporciona:

- Solución de los problemas debidos a errores de redondeo.
- Posibilidad de fijar diferentes densidades de puntos en aristas, caras y poliedros.
- Generación de mallas de buena calidad.
- Pruebas experimentales con una complejidad casi lineal.

PERSPECTIVAS FUTURAS.

- Desarrollo del algoritmo de refinamiento/desrefinamiento en 3-D.
- Mejora de la entrada de datos y conformidad con la frontera en 3-D.
- Aplicación de las técnicas de mallado en problemas de Ingeniería en 3-D.

HISTORIA

Problema de elasticidad con

sólo refinamiento

Classical formulation:

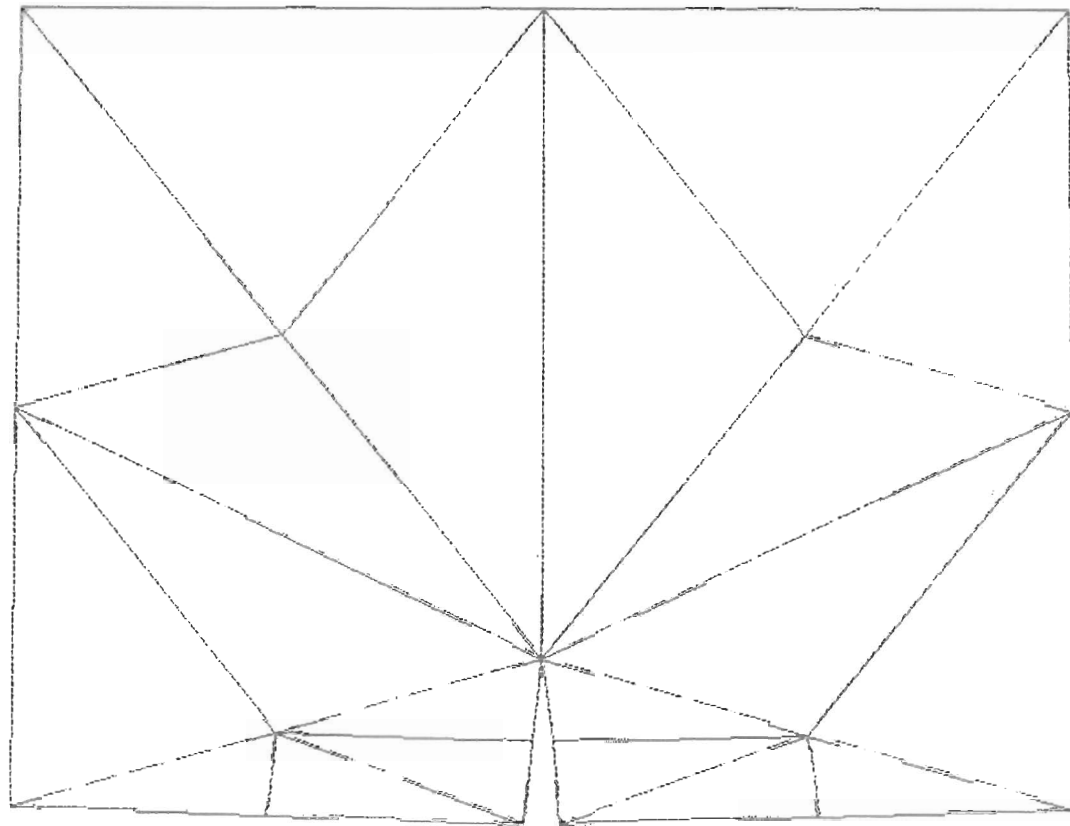
$$\sum_{j=1}^2 \frac{\partial}{\partial x_j} \sigma_{ij}(\bar{u}) = 0 \quad \text{in } \Omega, \quad \text{where } i = 1, 2$$

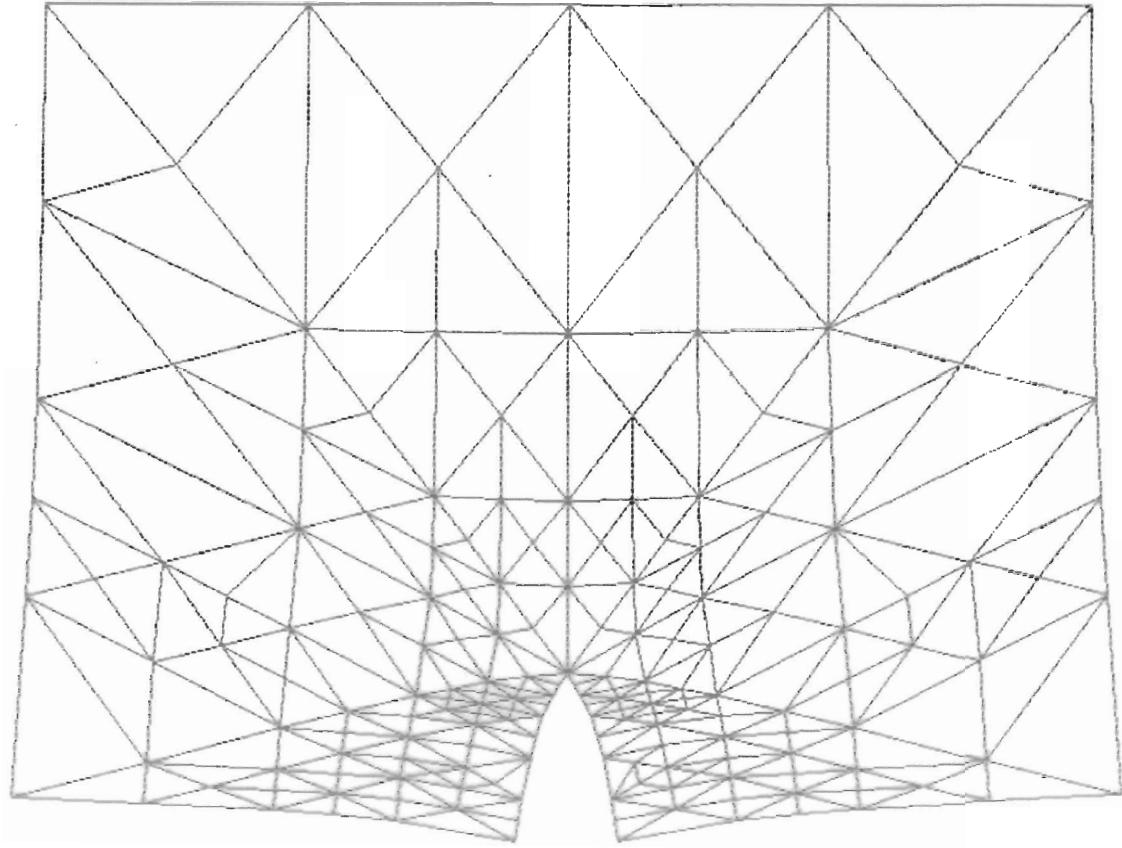
$$\bar{u}_i = \bar{0} \quad \text{on } \Gamma_0, \quad u_2(A) = 0$$

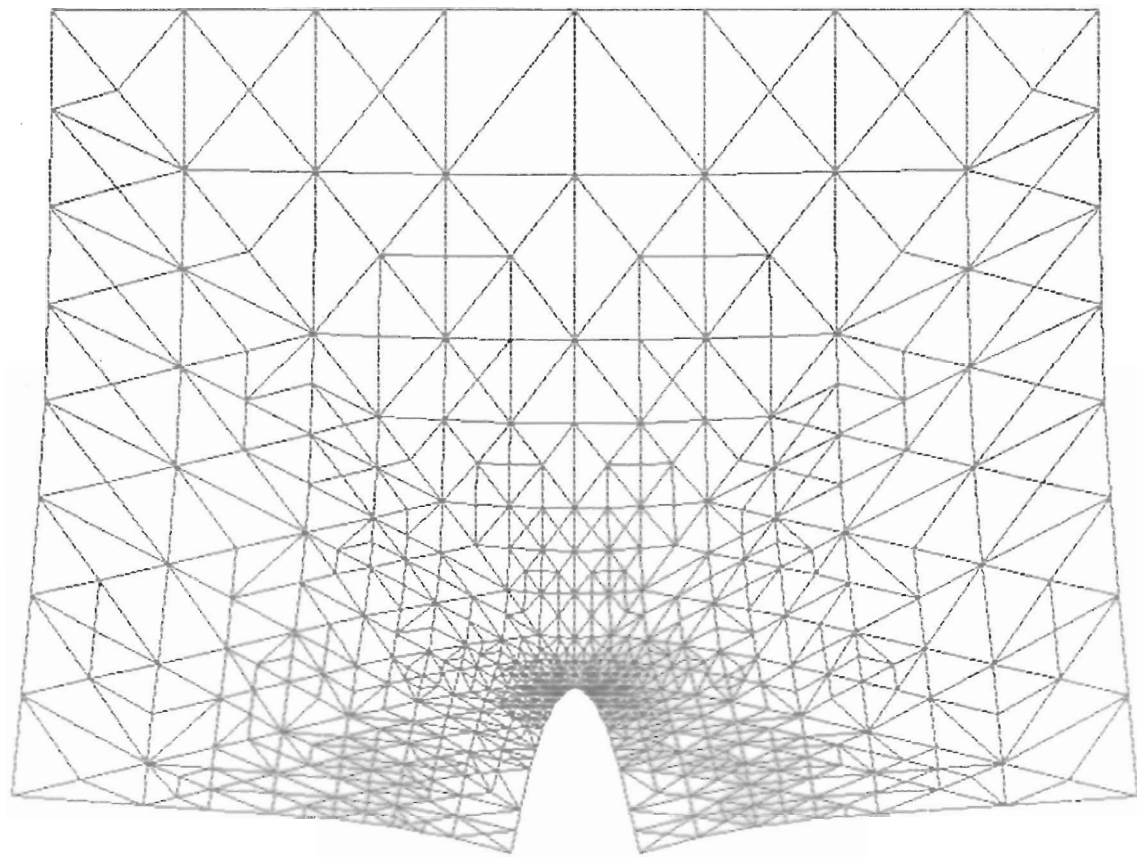
$$\sum_{j=1}^2 \sigma_{1j}(\bar{u}) n_j = 0 \quad \text{on } \Gamma_{1,1}, \quad \sum_{j=1}^2 \sigma_{2j}(\bar{u}) n_j = 1 \quad \text{on } \Gamma_{1,1}$$

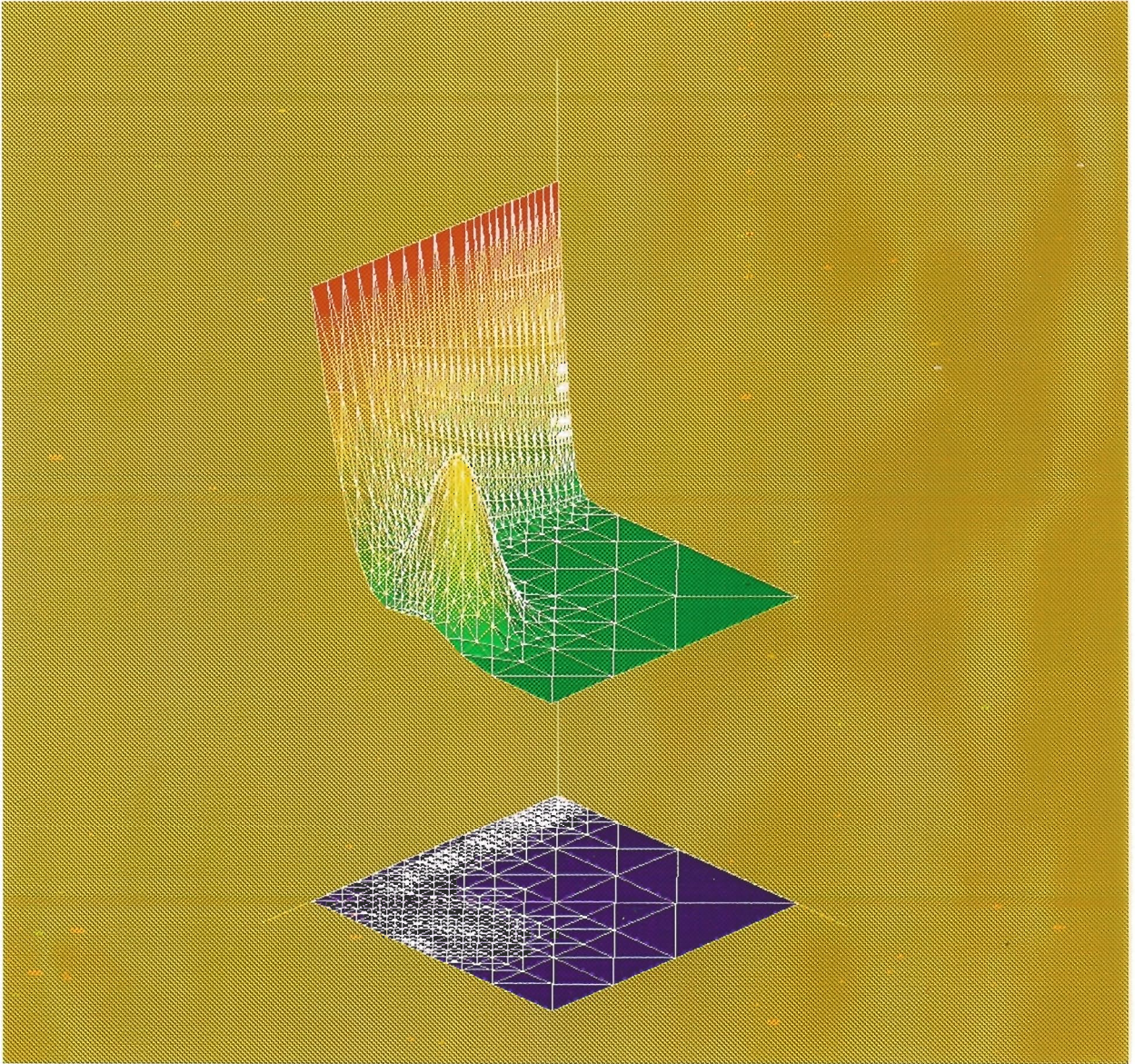
$$\sum_{j=1}^2 \sigma_{1j}(\bar{u}) n_j = 0 \quad \text{on } \Gamma_{1,2}, \quad \sum_{j=1}^2 \sigma_{2j}(\bar{u}) n_j = -1 \quad \text{on } \Gamma_{1,2}$$

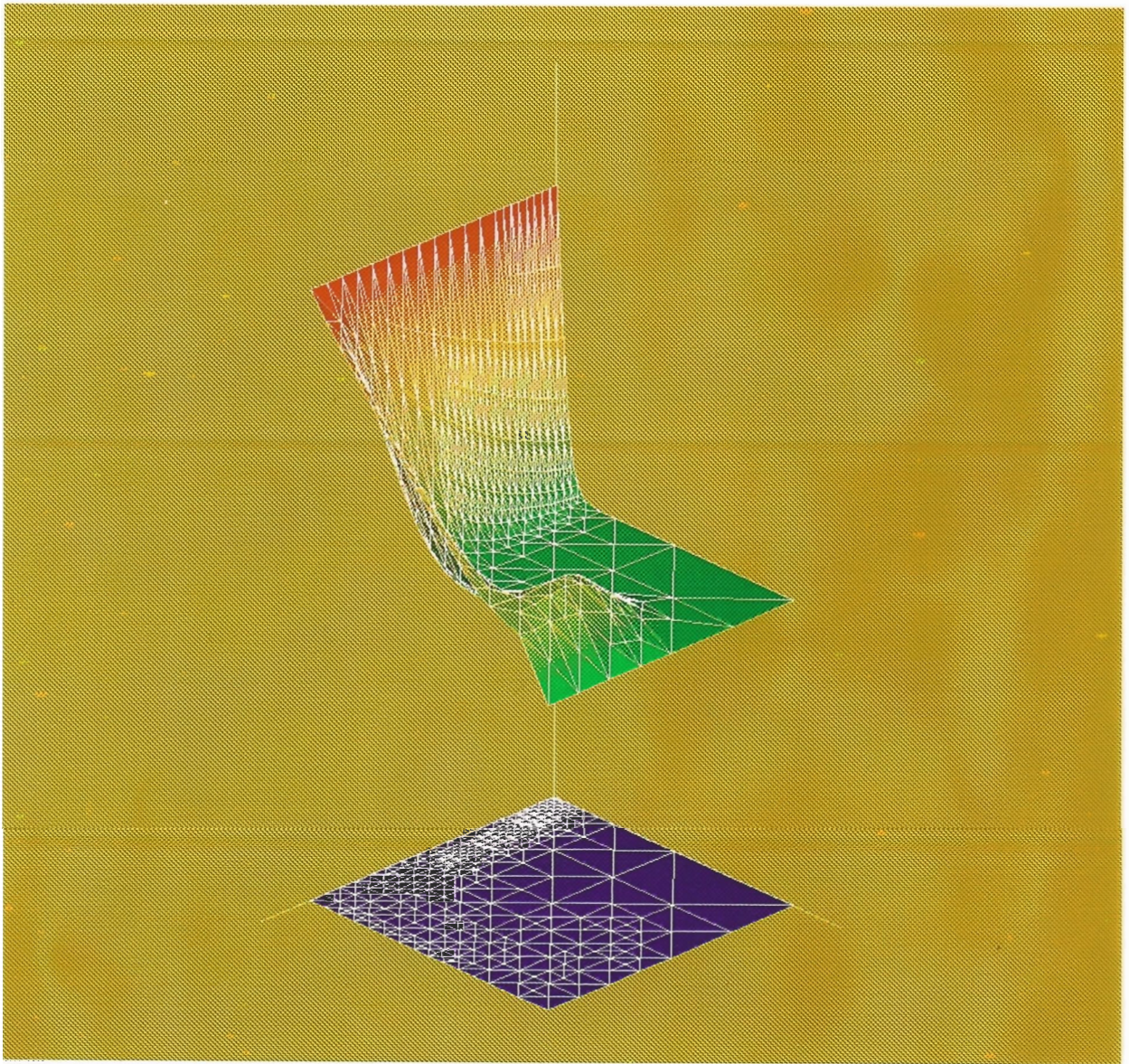
$$\sum_{j=1}^2 \sigma_{ij}(\bar{u}) n_j = 0 \quad \text{where } i = 1, 2 \quad \text{on } \Gamma_{1,3}$$

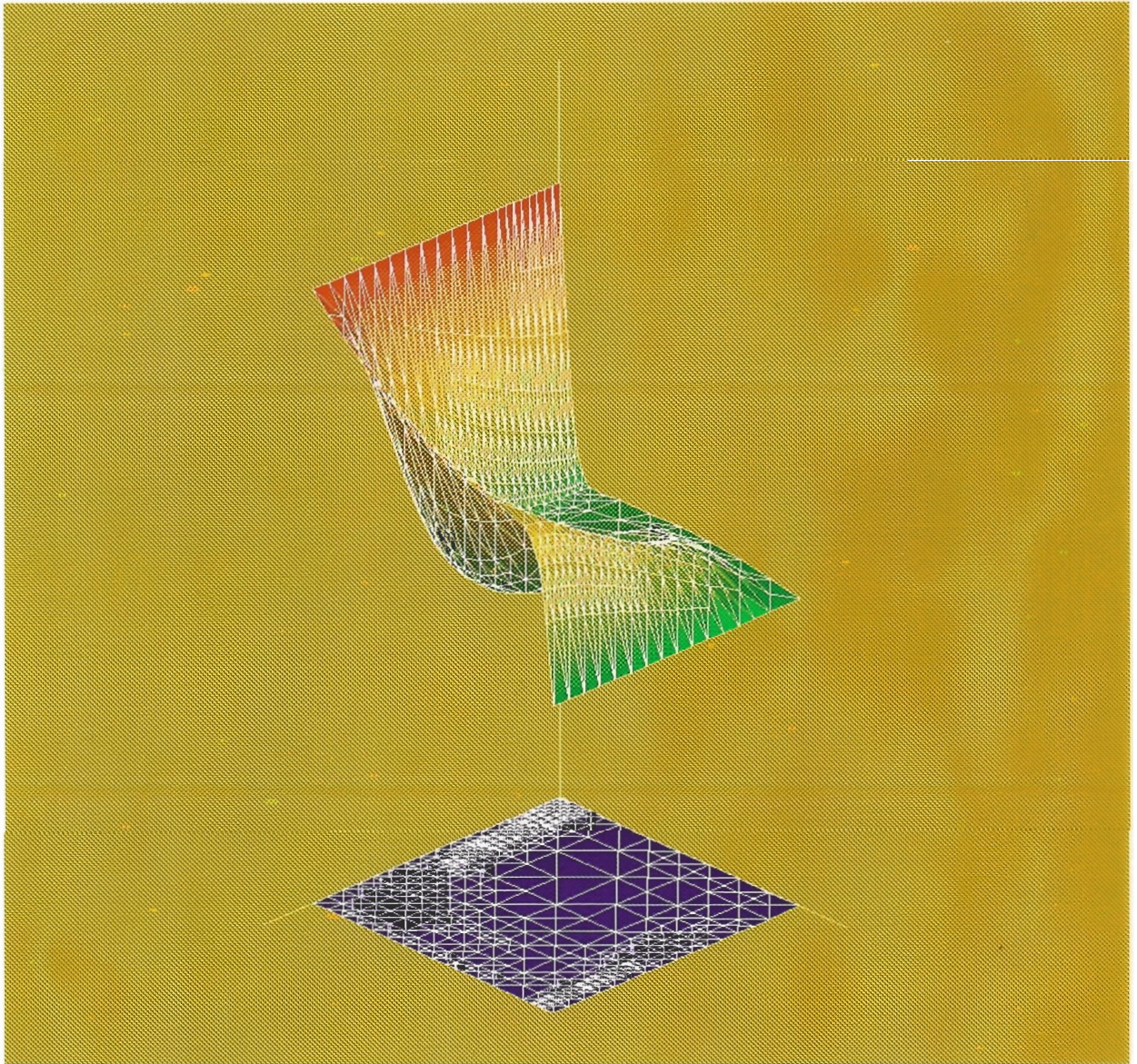


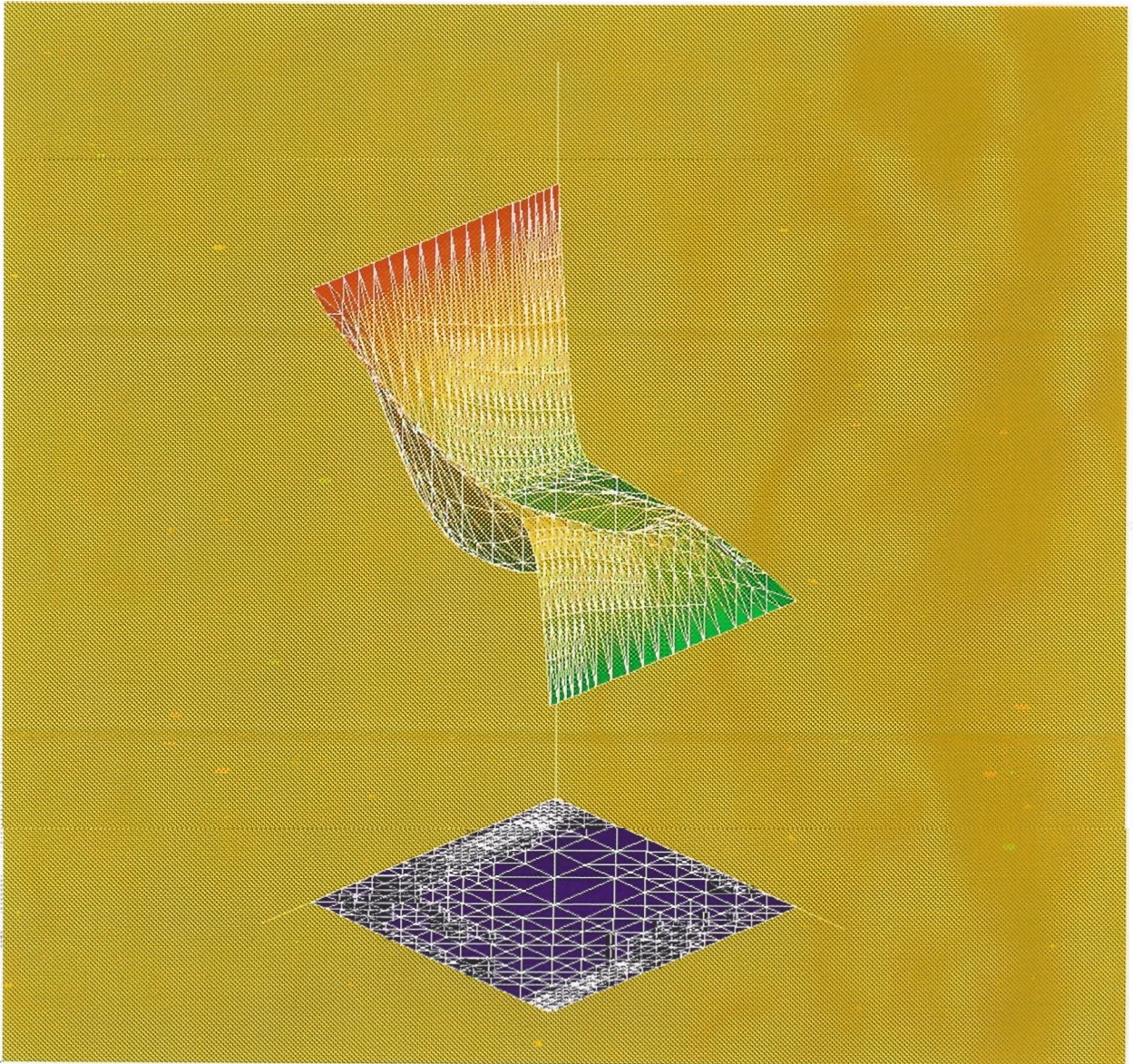


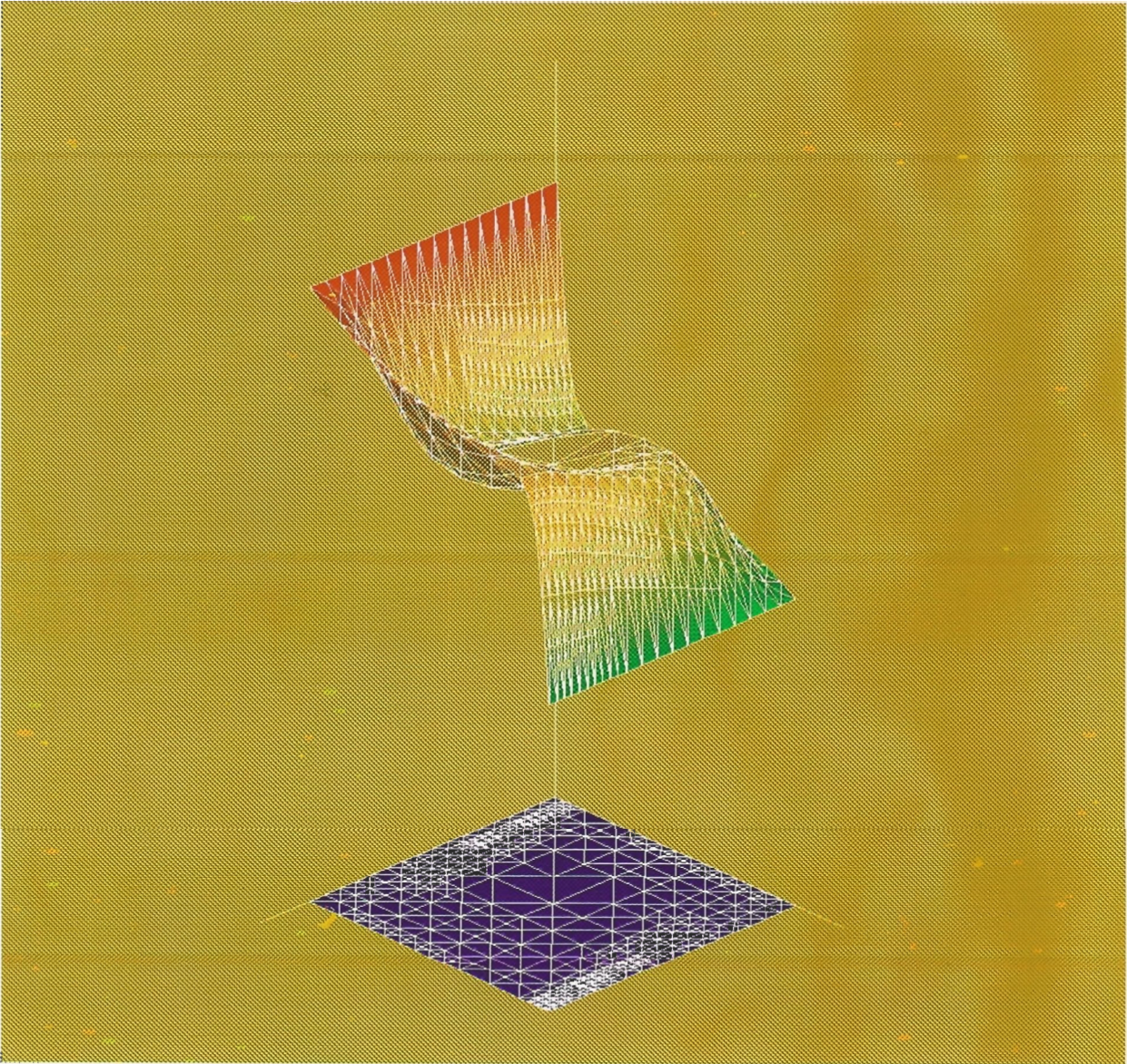




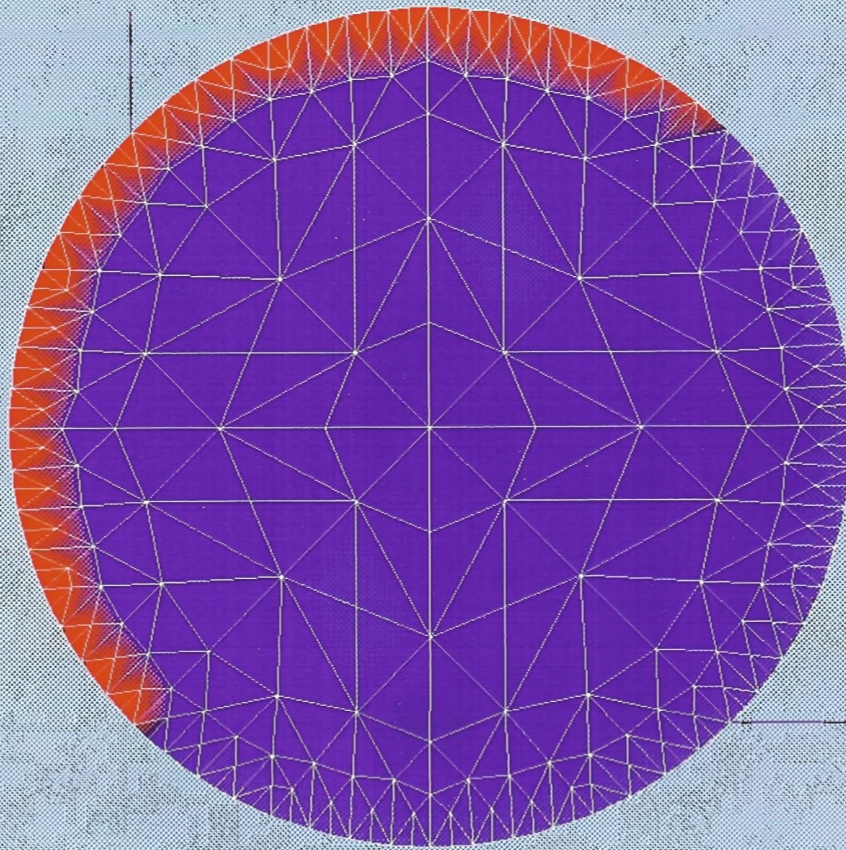








STEADY CONVECTION-DIFFUSION PROBLEM



Initial mesh, 297 nodes

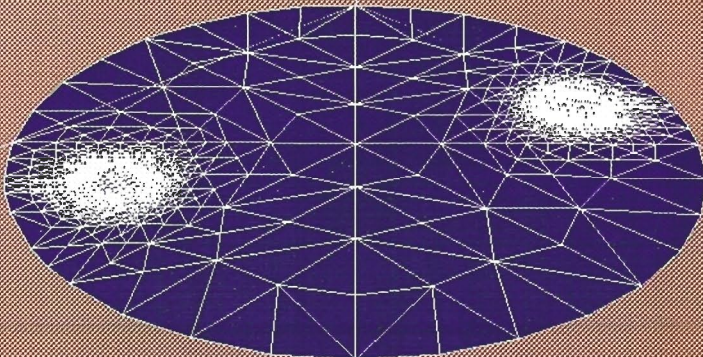
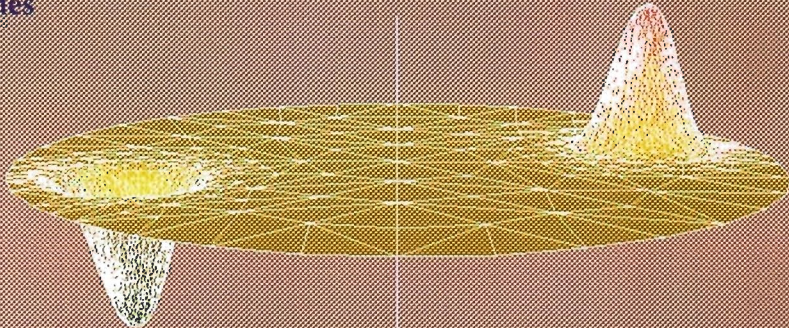
STEADY CONVECTION-DIFFUSION PROBLEM



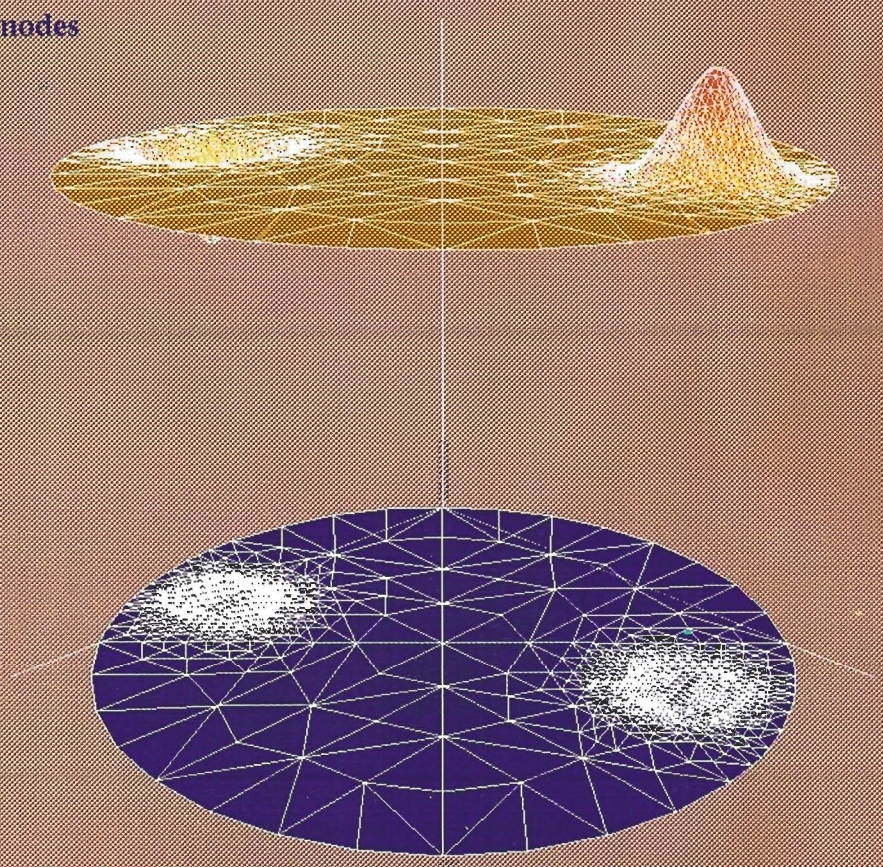
1421 nodes

240 time steps

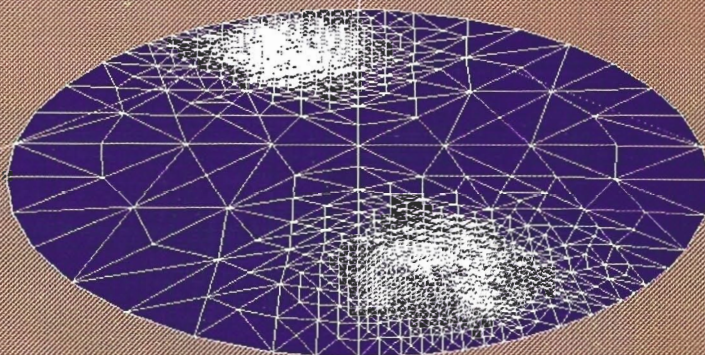
2115 nodes



480 time steps
1923 nodes

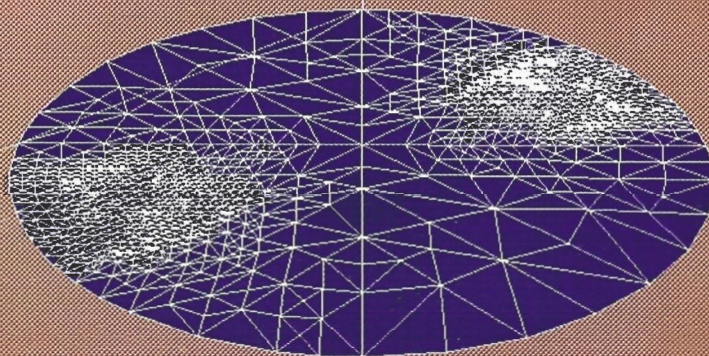


720 time steps
1873 nodes



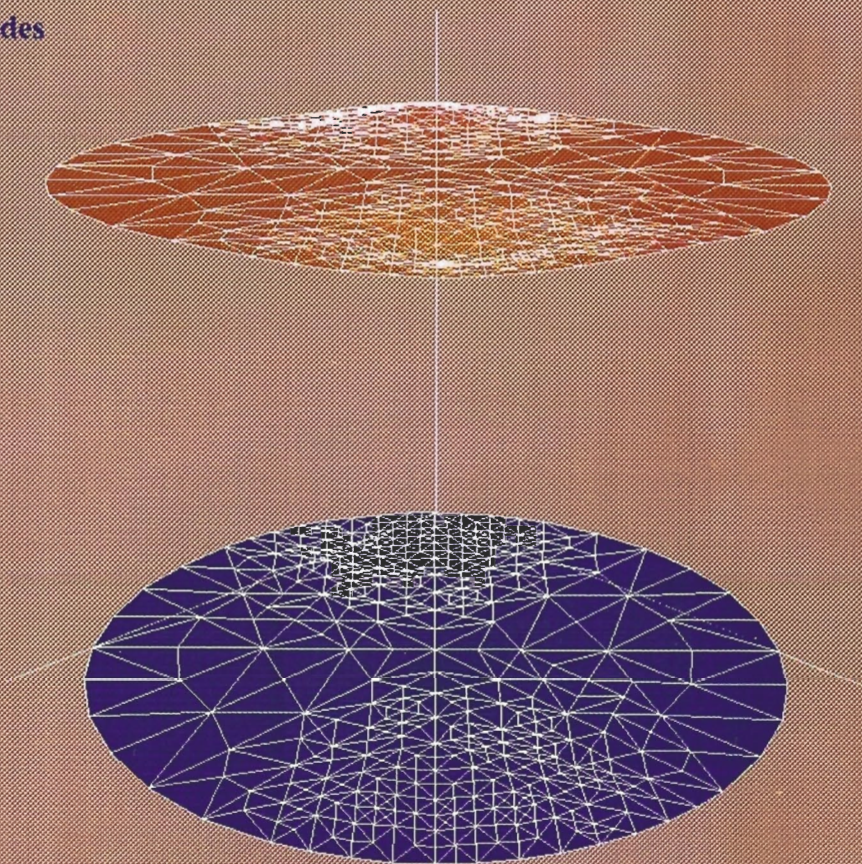
960 time steps

1383 nodes



1200 time steps

449 nodes

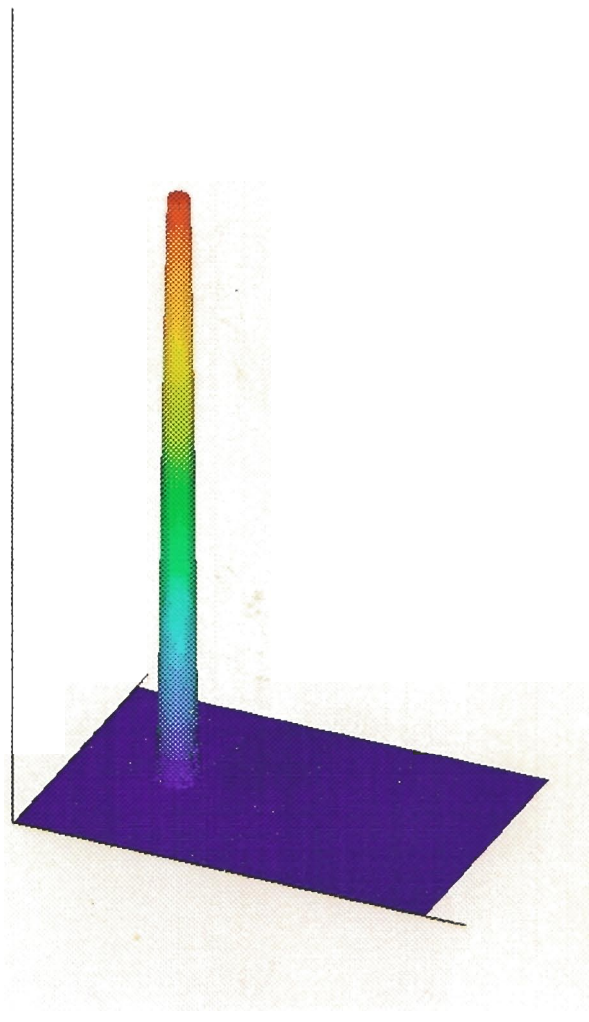


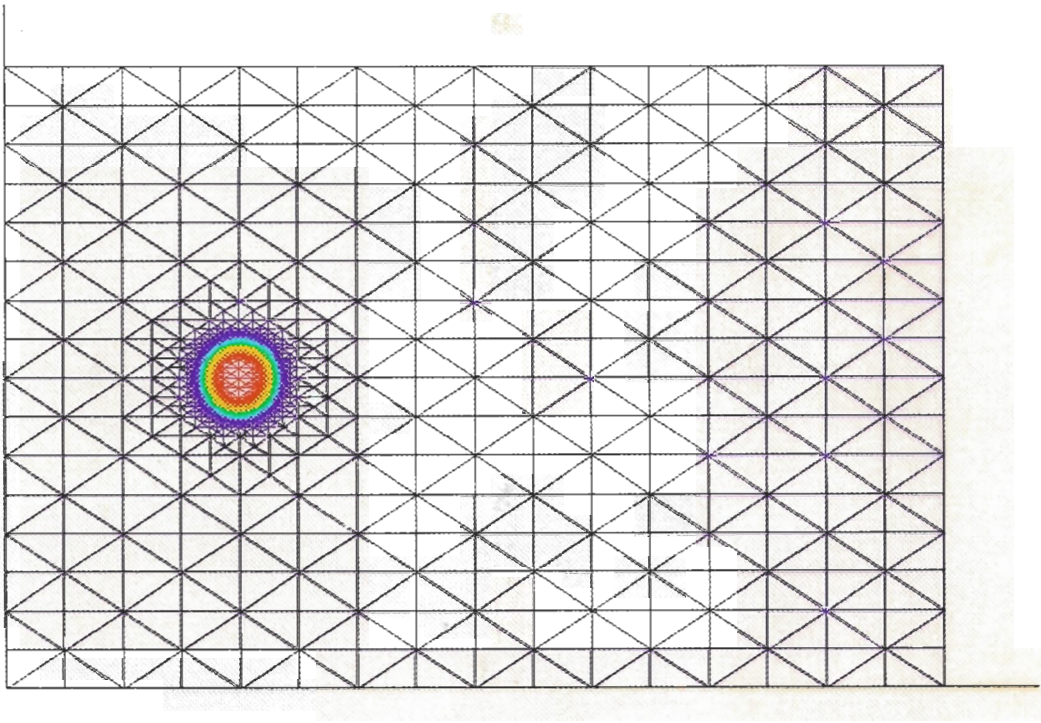
SIMULACIÓN DE PROPAGACIÓN DE FUEGO

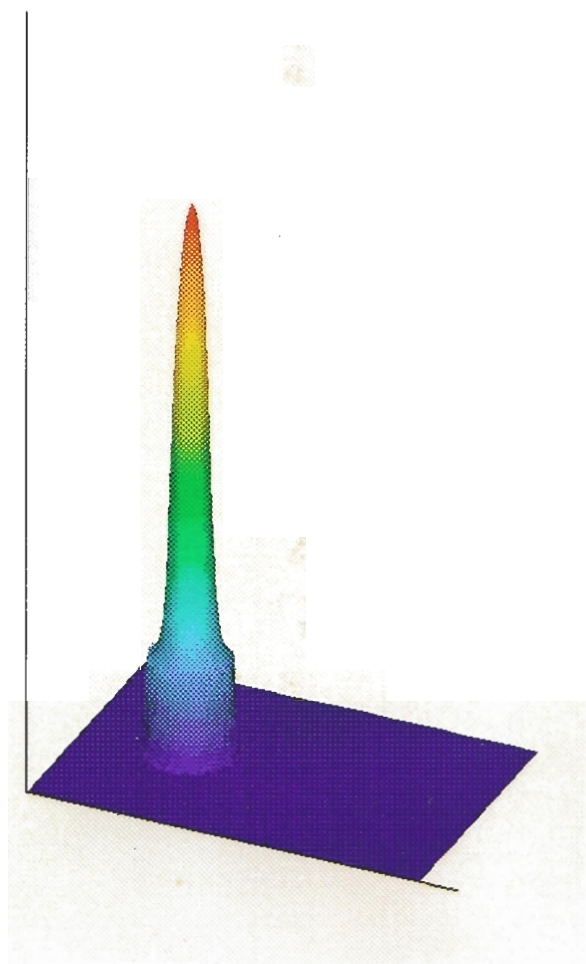
$$\frac{\partial T}{\partial t} - \text{div}(\mathbf{K} \vec{\nabla} T) + \vec{v} \cdot \vec{\nabla} T + h(T)(T - T_{\infty}) = f(t, T)$$

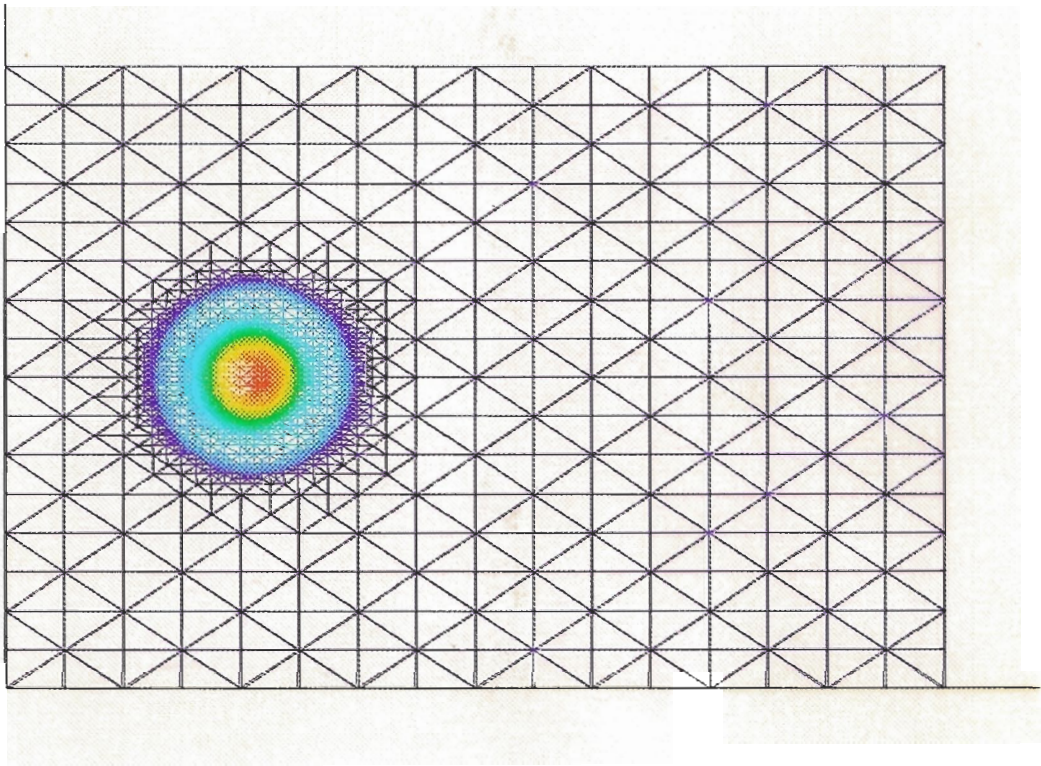
$$\frac{dC_A}{dt} = -\lambda_A C_A ; \lambda_A = A e^{-\frac{E_A}{RT}}$$

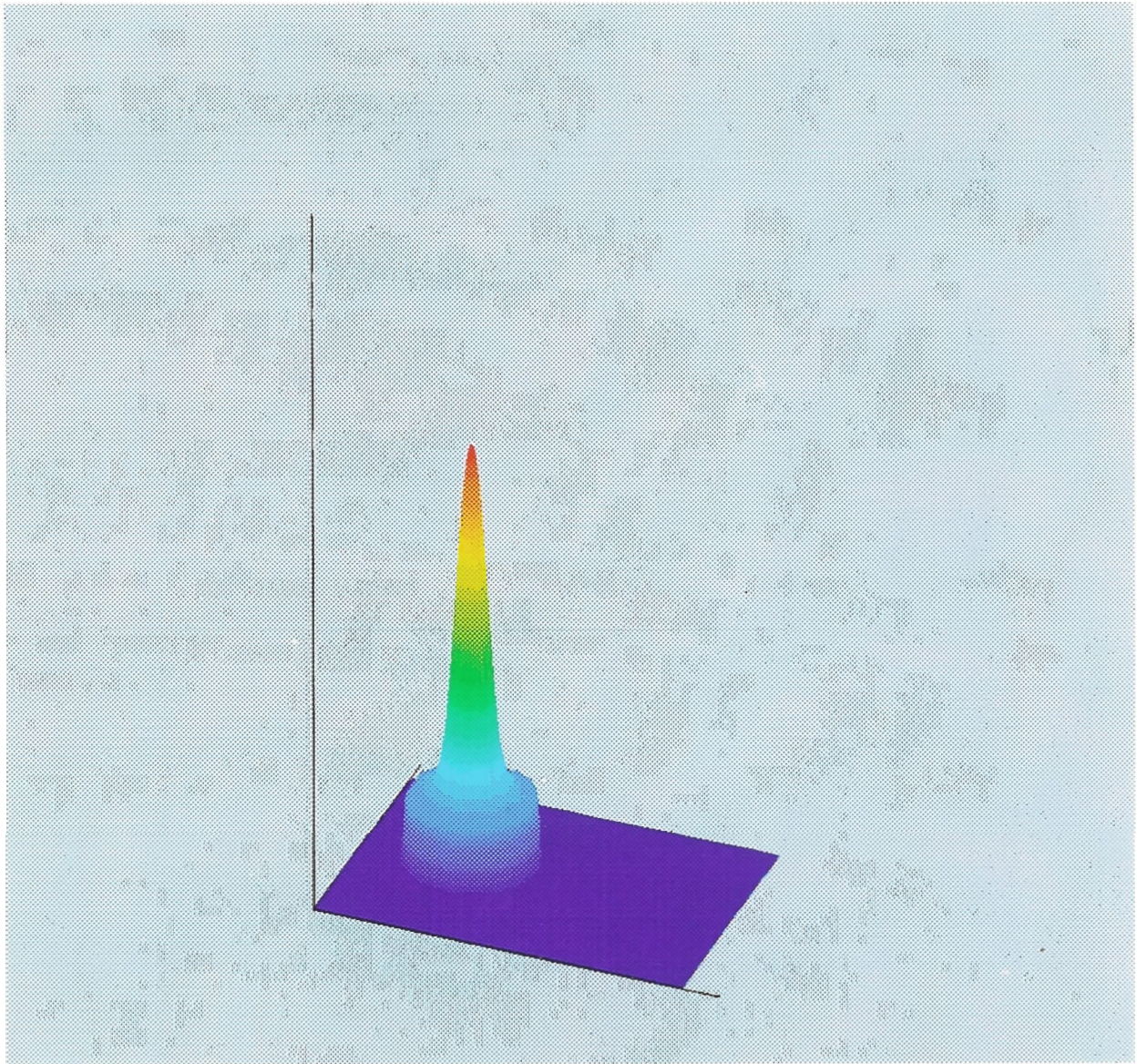
- Análisis mediante diferencias finitas y elementos finitos.
- Estudio de estabilidad del esquema en diferencias finitas (paso de tiempo).











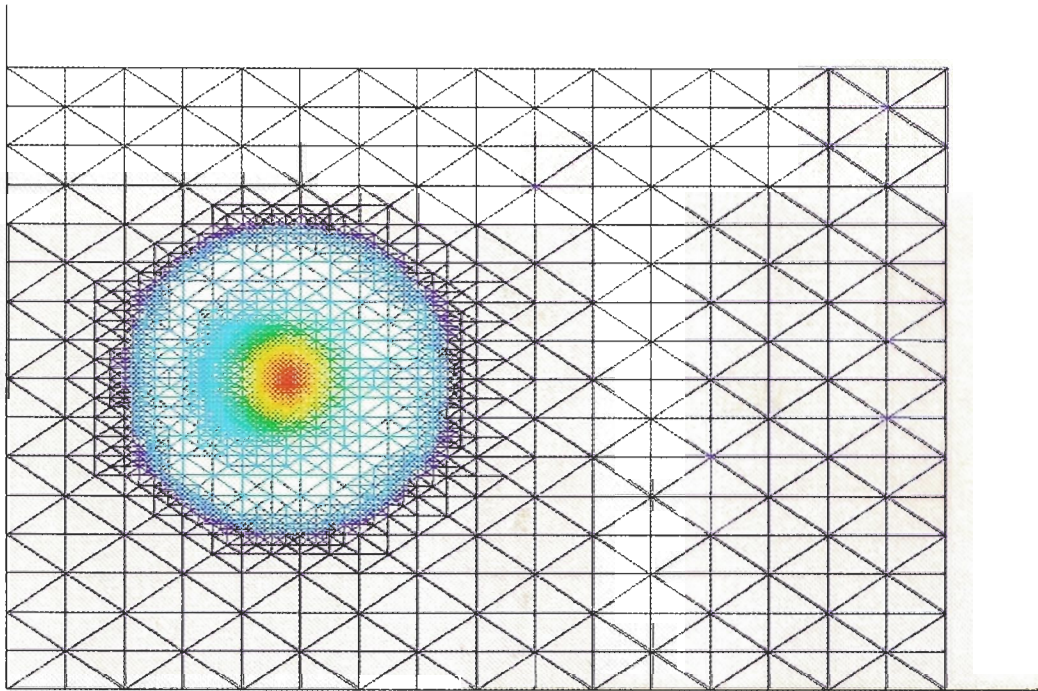
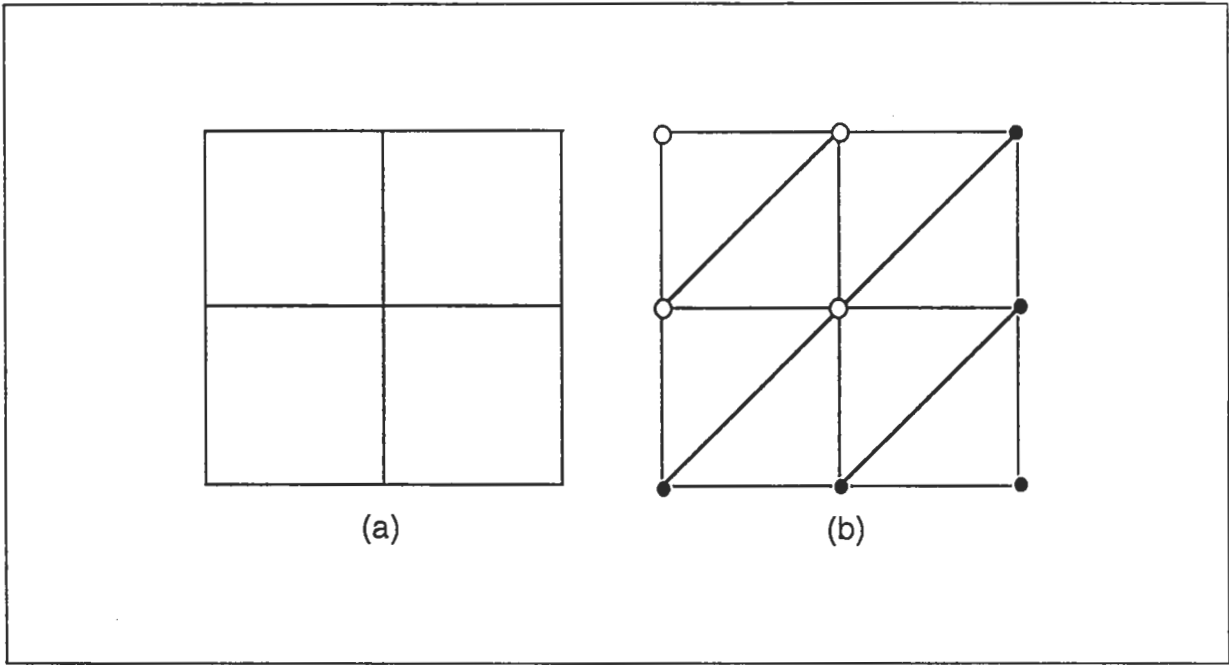
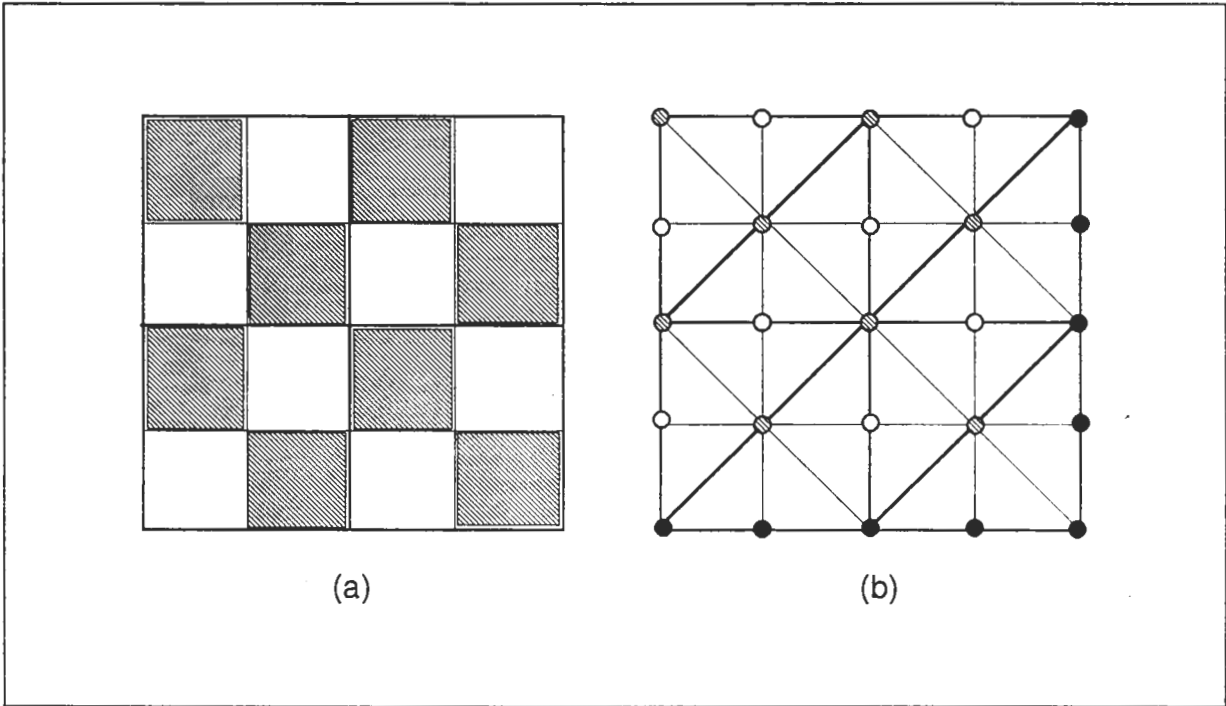




Imagen clásica 256x256.



(a) Imagen 2x2 pixels, (b) malla triangular asociada



(a) Imagen 4x4 pixels, (b) malla triangular

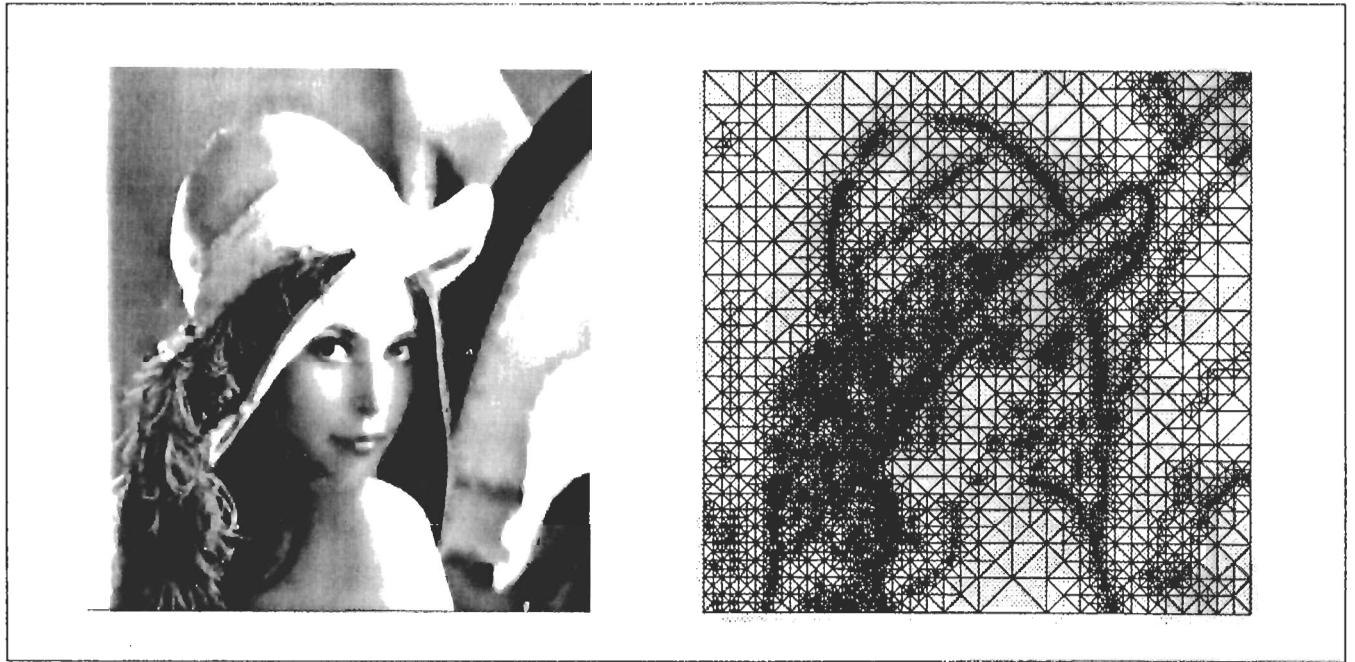


Imagen y malla para $\varepsilon = 30$: 7.736 nodos, SNR(dB) = 22.58.

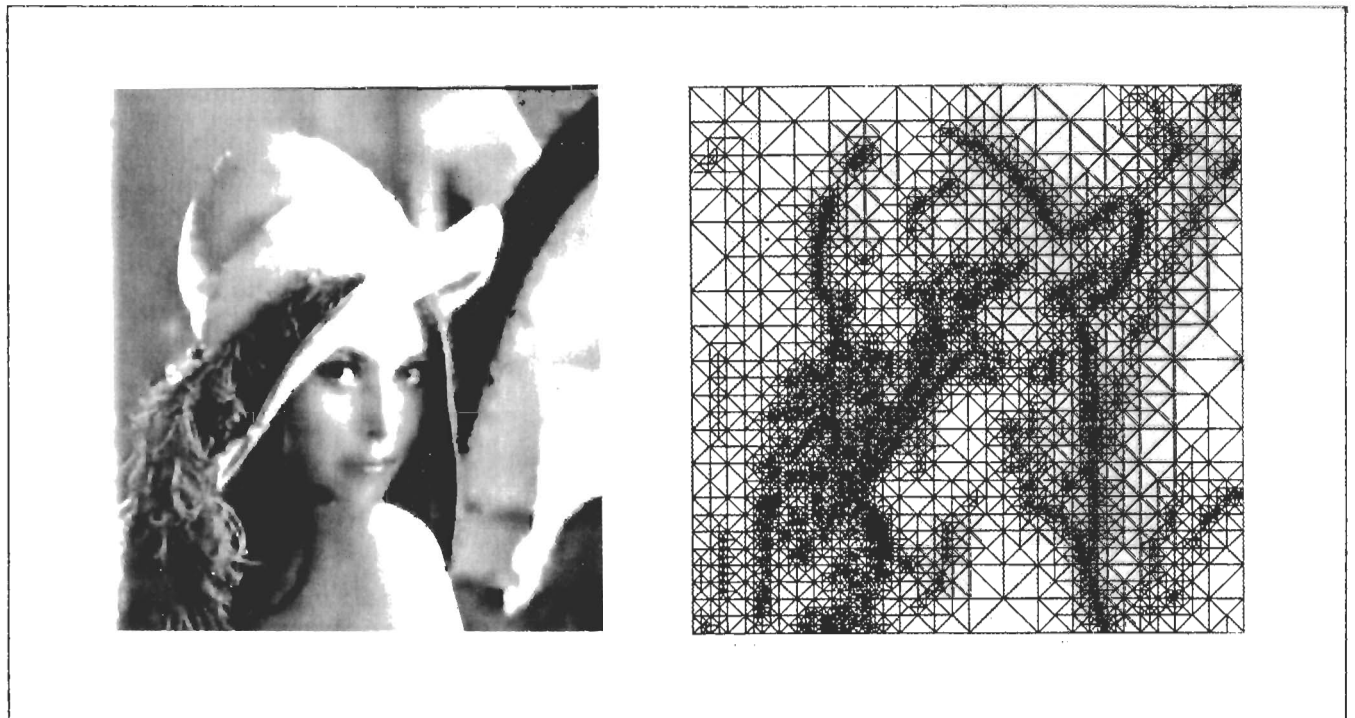
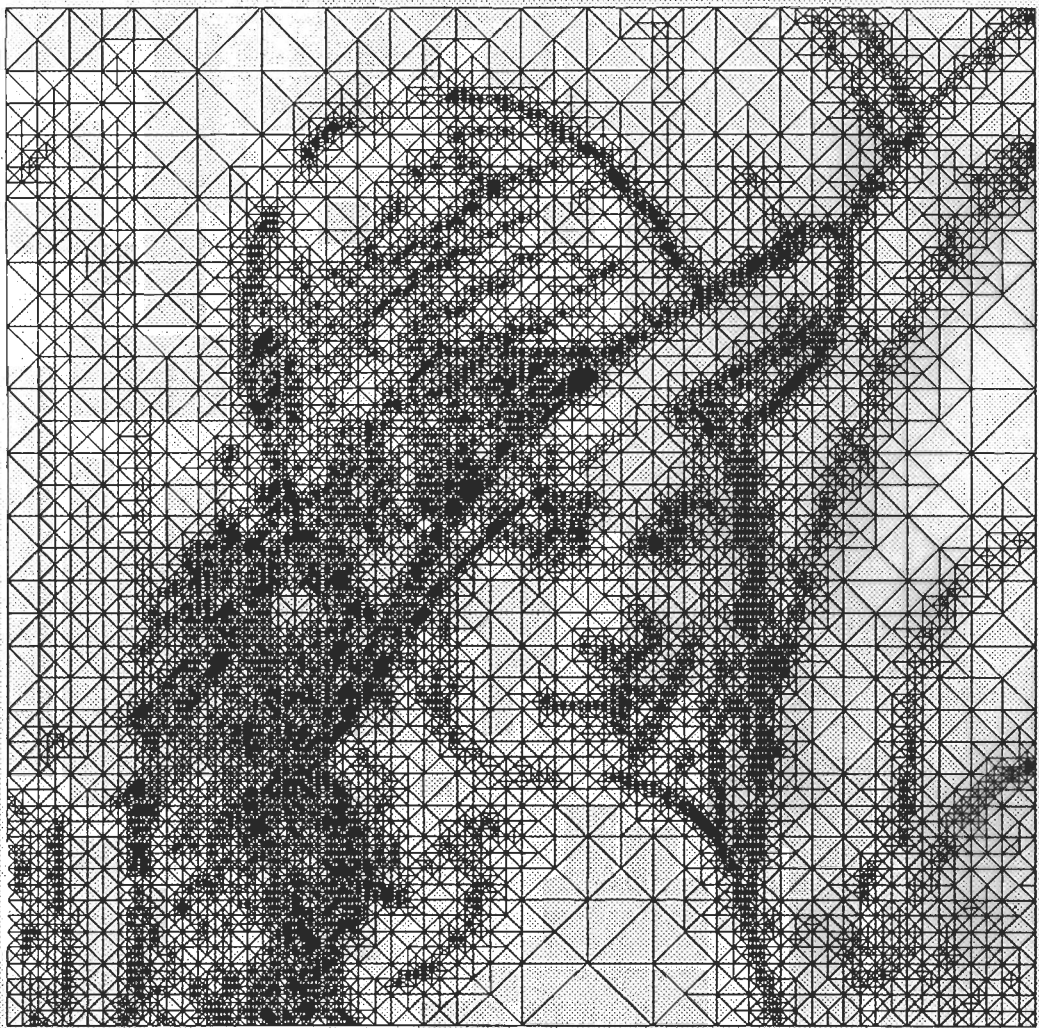


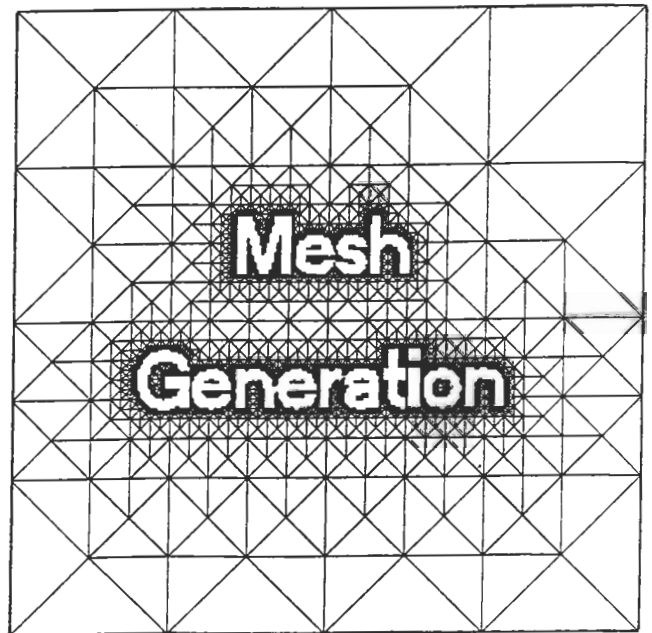
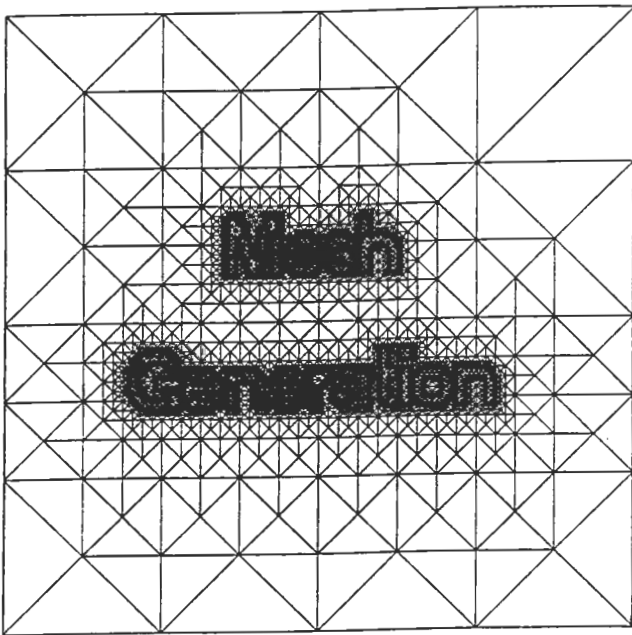
Imagen y malla para $\varepsilon = 40$: 5.555 nodos, SNR(dB) = 19.95.



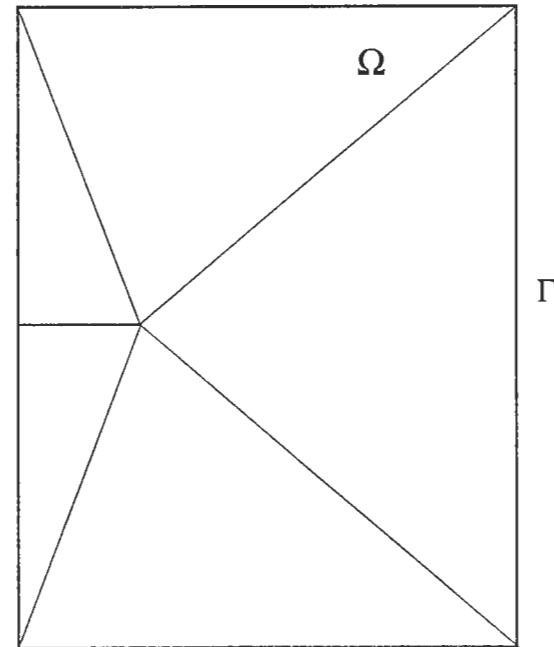
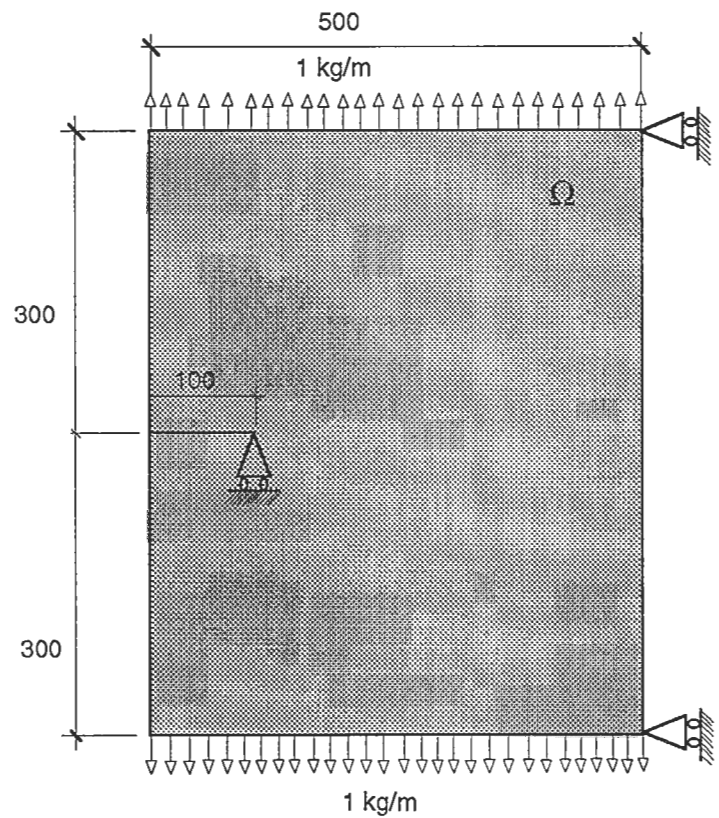




Mesh Generation



Un Problema de Elasticidad Lineal

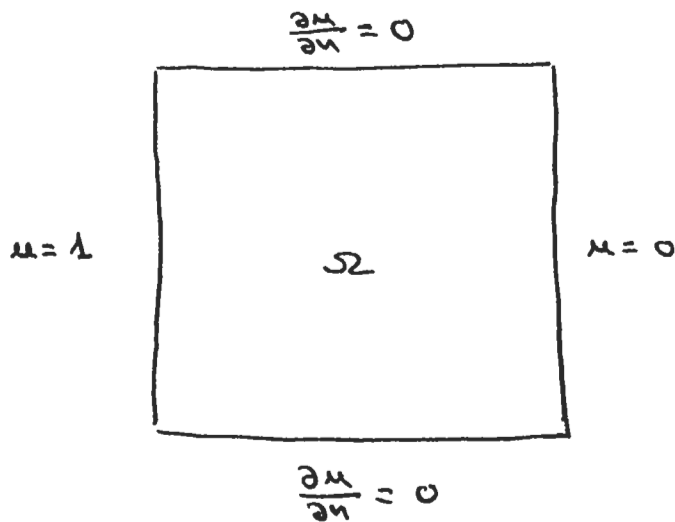


HISTORIA PASADA

(TESIS - 1989)

Problema Evolutivo cuando solo se disponía del algoritmo de refinamiento

$$\frac{\partial \mu}{\partial t} + \vec{v} \cdot \nabla \mu - \nabla \cdot \left(\underset{\substack{1 \\ \downarrow}}{K} \nabla \mu \right) = 0 \quad \text{en } \Omega$$



condición inicial: $\mu(\vec{x}, 0) = 0$

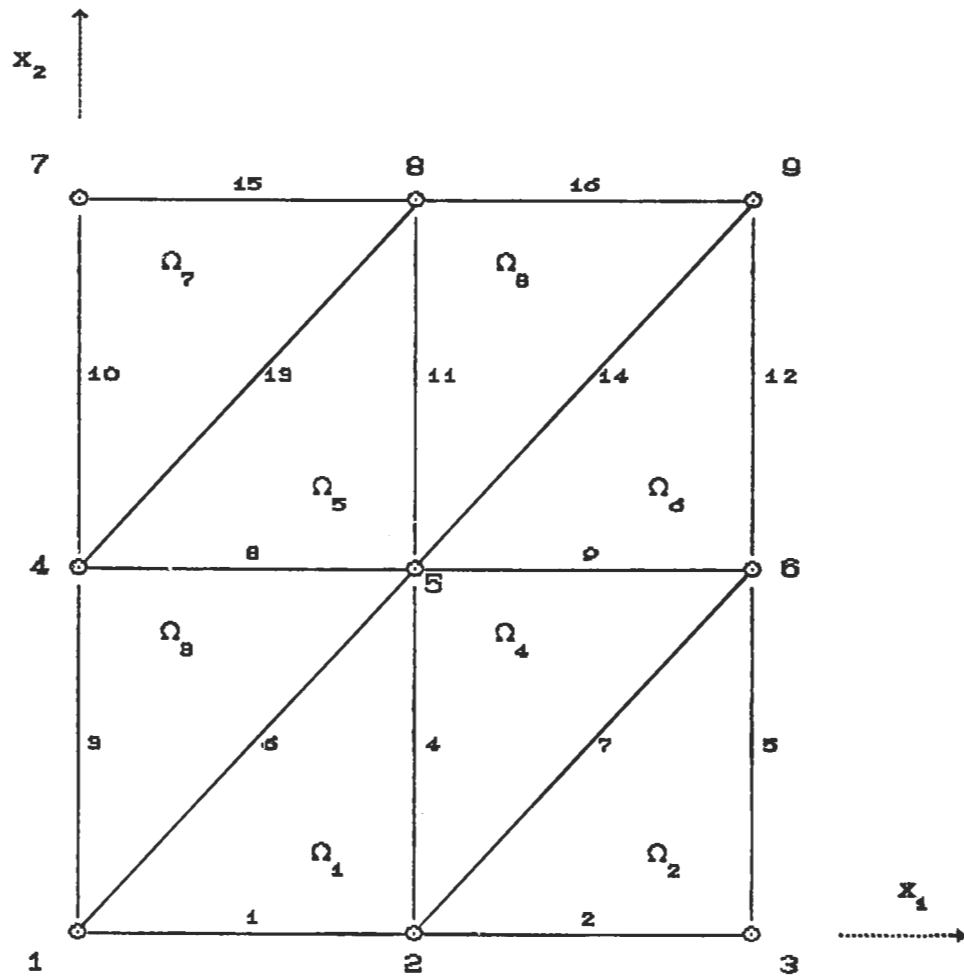
HISTORIA NEPTUNO

Generación de malla inicial

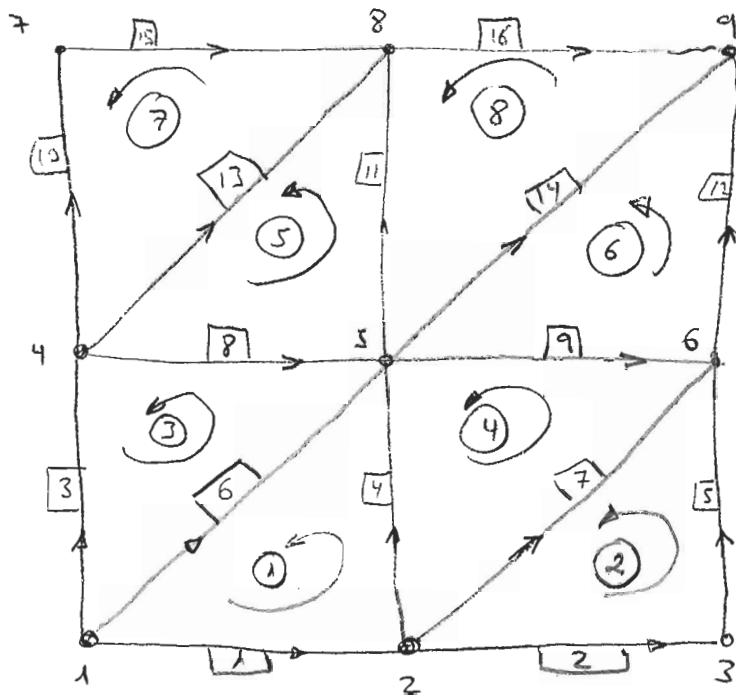
y
entrada de datos



MALLADO DE "PARTIDA" ; Nodos, caras y elementos.



NR = 1



NR = 2

NEPT *** CONVECCION-DIFUSION, CAMPO CIRCULAR, (A) ***

9,16,8,1,2,0,0,5,2,1,2,3,2,3,0,1

COORD

1,1,0.,0.

3,0,1.,0.

4,1,0.,0.5

6,0,1.,0.5

7,1,0.,1.

9,0,1.,1.

FACE

1,0,1,2,1

3,1,1,4,0

4,0,2,5,0

5,2,3,6,0

6,0,1,5,1

8,0,4,5,1

10,1,4,7,0

11,0,5,8,0

12,2,6,9,0

13,0,4,8,1

15,0,7,8,1

ELEM

1,1,1,4,-6,1

3,1,6,-8,-3,1

5,1,8,11,-13,1

7,1,13,-15,-10,1

RNOD

1,3,1

7,0,1

3,3,2

9,0,2

MATE

1 4
3,0,0,0,4,5
1,0,0.

CCON

1,1

2,2

FUNZ

C1-DIRICH-NR=1

1,1.

C2-DIRICH-NR=2

1,0.

C3-CONDUCT.-K

1,1.

C4-VELOC.X-V1

6,1000.

C5-VELOC.Y-V2

7,1000.

NOPR

END

MACR

TOL
LOOP
NNIV
REFN
NEXT
LOOP
NNIV
MATR
PREC
SEGM
GRCJ
IMPR
INER
REFN
NEXT
END
PLOT
POSP
6,0.,1.
STOP

1.E-6

4.

3.

3.

1.

3.

0.4

SUBROUTINE ACONTR

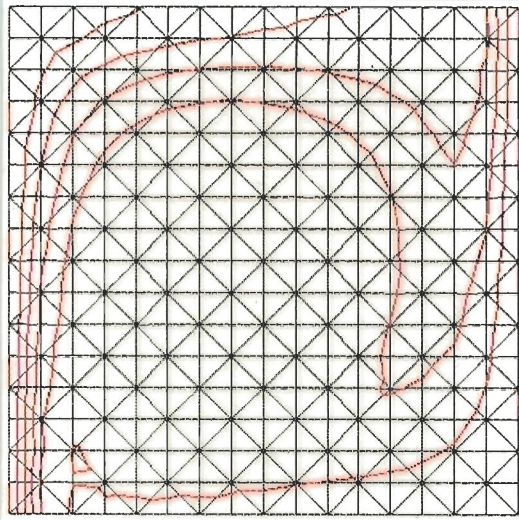
C-----Subprograma de control de todo el proceso
 C-----Lee el titulo,datos de control iniciales,inicializa la matriz
 C-----de indicadores,llama al modulo de entrada de datos,al modulo de ejecucion,
 C-----y a los modulos de preproceso y posproceso

C
 C O :Controla el salto de pagina
 C HEAD :Titulo
 C NUMNP :No.de nodos (totales)
 C NUMNF :No.de caras (totales)
 C NUMEL :No.de elementos
 C NUMAT :No.de materiales
 C NUREF :No.de numeros de referencia
 C NUFNO :No.de funciones dadas en forma implicita (Fronteras curvas)
 C NUFN1 :No.de funciones de un parametro
 C NUFN2 :No.de funciones de dos parametros
 C NDM :Dimension del dominio de definicion del problema
 C NDF :No.de grados de libertad por nodo
 C NDF1 :No.de grados de libertad por nodo variable acoplada
 C NEN :No.de caras por elemento
 C NEC :No.de nodos por cara
 C NEL :No.de nodos por elemento
 C NIN :No.de nodos internos por elemento
 C NEQ :No.de ecuaciones
 C NSIM :si = 0 -> m.simetrica ;si #0 -> m.no simetrica

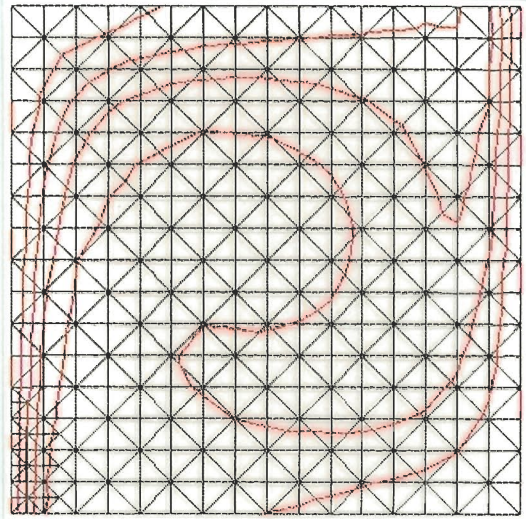
C-----

```
COMMON M(1)
COMMON/DIREC/IM(4,1)
COMMON/MEMOR/IMAX,IMAXI,NBLOQ,IM1BL,IM2BL,IMEM,IPR
COMMON /CDATA/ HO,HEAD(20),NUMNP,NUMNF,NUMEL,NUMAT,NUREF,
1 NUFNO,NUFN1,NUFN2,NEN,NEC,NEL,NIN,NEQ
COMMON /NDIM/ NDM,NDF,NDF1,NEN1,NEC1,NSIG
COMMON /NULOG/ NL,NE
COMMON /MSIM/ NSIML
LOGICAL PCOMP,NSIML
DIMENSION TITL(20),WD(8)
DATA WD/4HNEPT,4HMACR,4HPLOT,4HPOSP,4HVELC,4HFLEC,4HDEFR,4HSTOP/
```

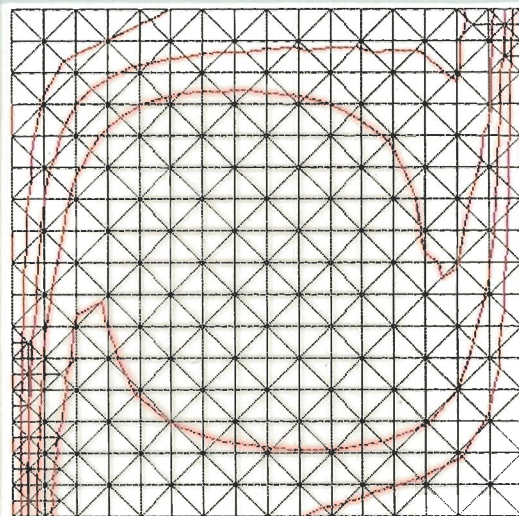
C
 C--- Lee una tarjeta y se dirige al proceso correspondiente
 C



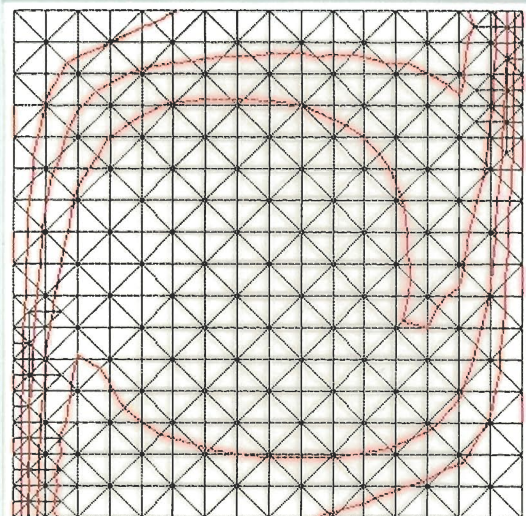
* MALLA 1 (Tiempo=0.05511) *



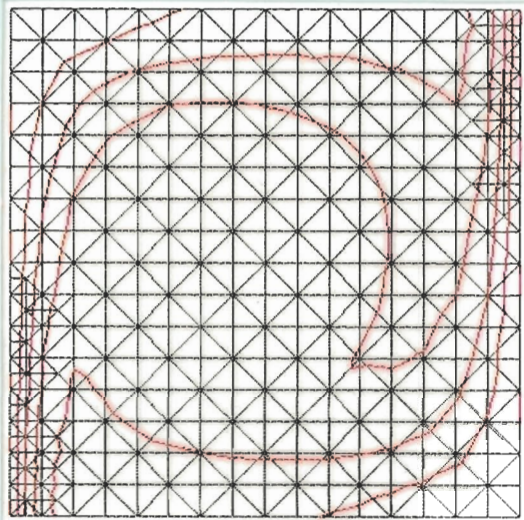
* MALLA 2 (Tiempo=0.08267) *



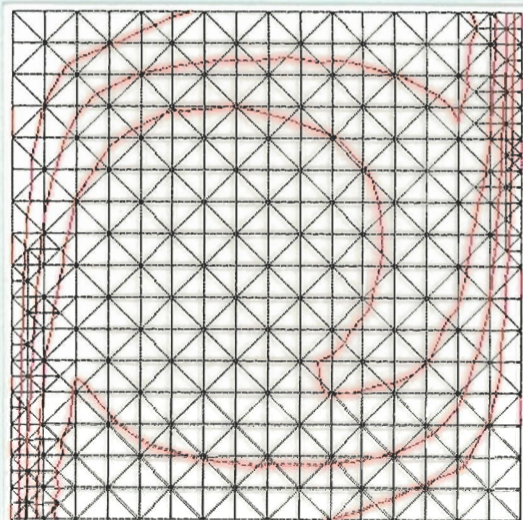
* MALLA 3 (Tiempo=0.11023) *



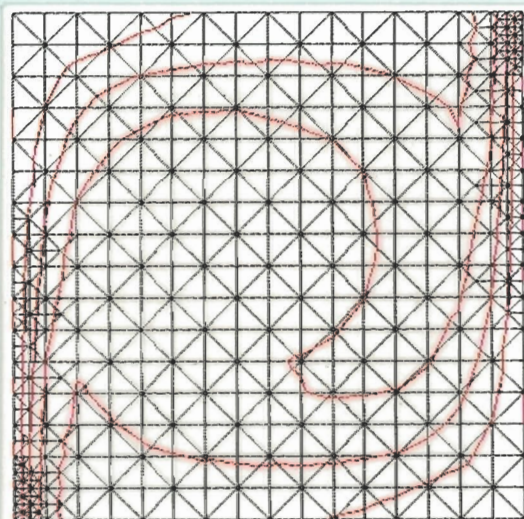
* MALLA 4 (Tiempo=0.13778) *



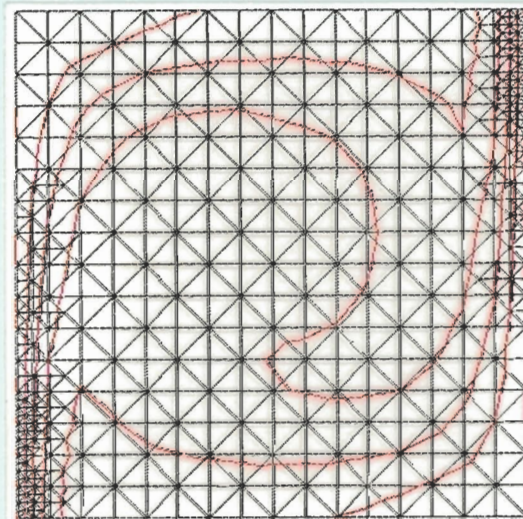
* MALLA 5 (Tiempo=0.16534) *



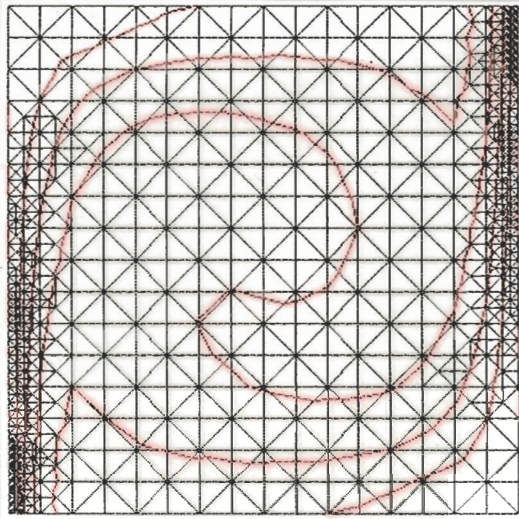
* MALLA 6 (Tiempo=0.18483) *



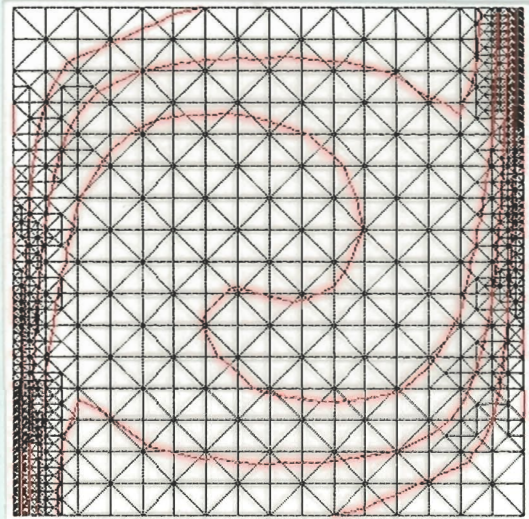
* MALLA 7 (Tiempo=0.19860) *



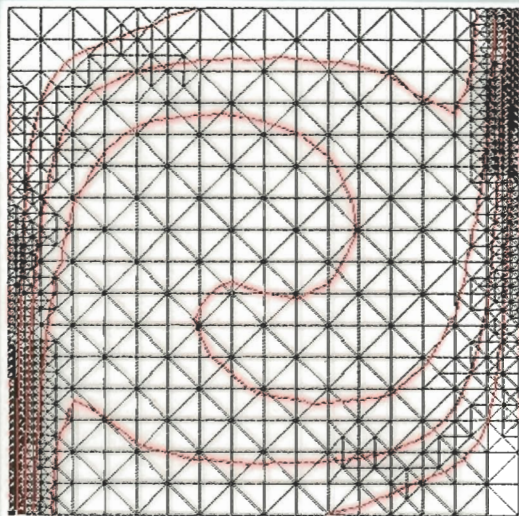
* MALLA 8 (Tiempo=0.21238) *



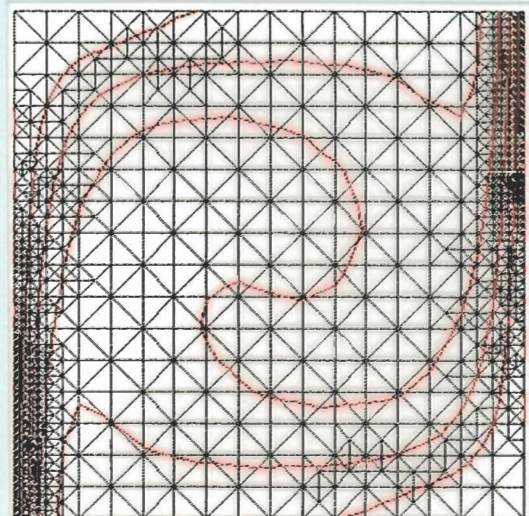
* MALLA 13 (Tiempo=0.26631) *



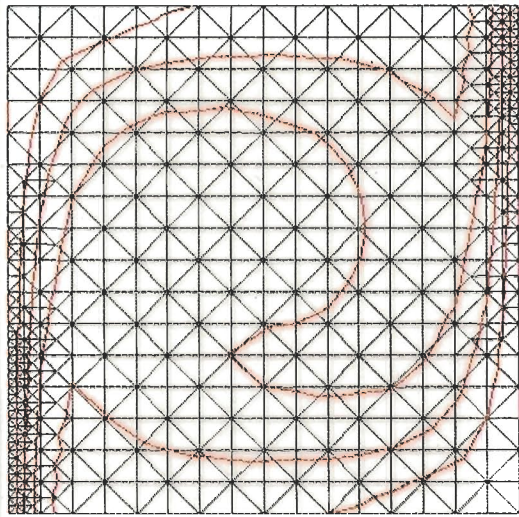
* MALLA 14 (Tiempo=0.27320) *



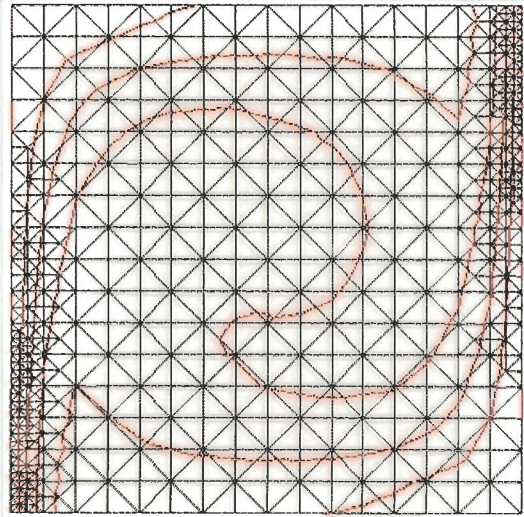
* MALLA 15 (Tiempo=0.28009) *



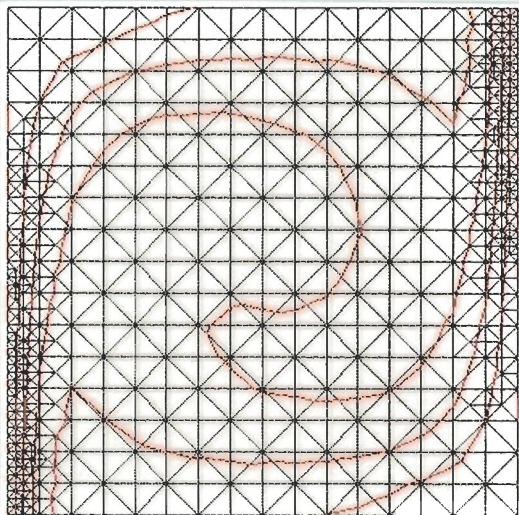
* MALLA 16 (Tiempo=0.28698) *



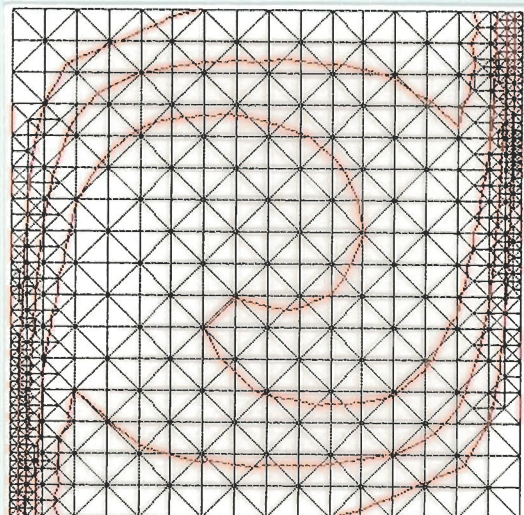
* MALLA 9 (Tiempo=0.22616) *



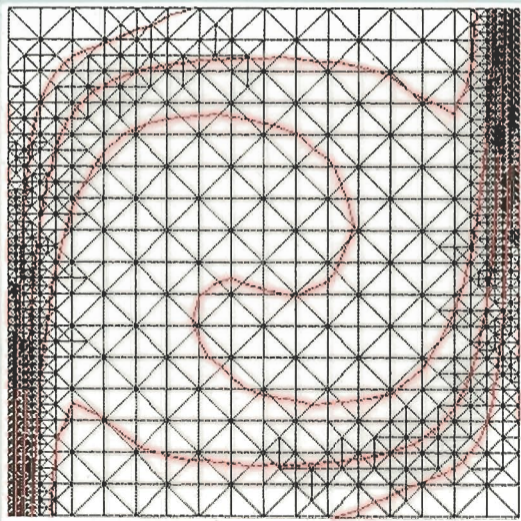
* MALLA 10 (Tiempo=0.23994) *



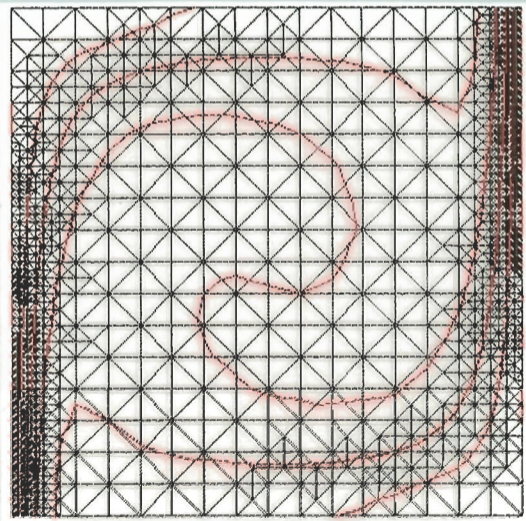
* MALLA 11 (Tiempo=0.24968) *



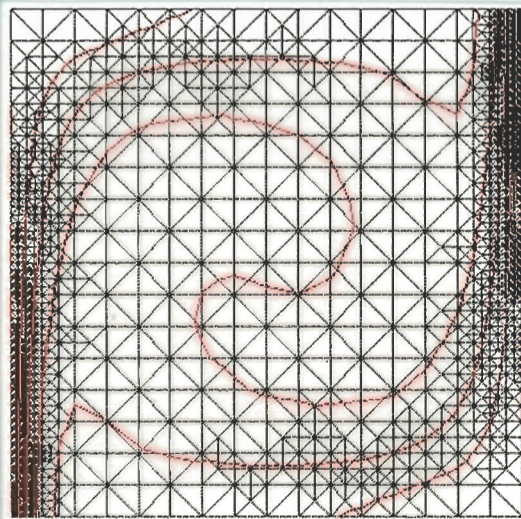
* MALLA 12 (Tiempo=0.25943) *



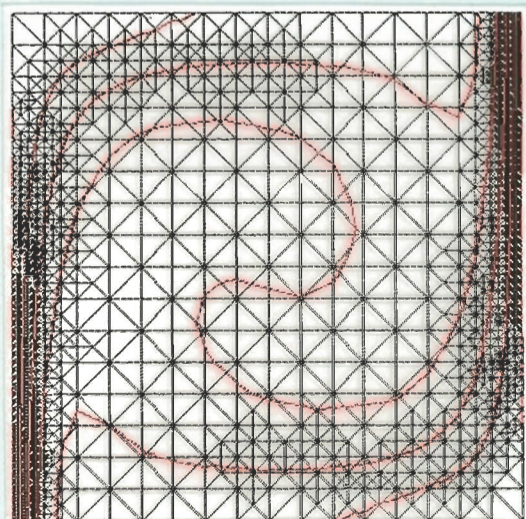
* MALLA 17 (Tiempo=0.29185) *



* MALLA 18 (Tiempo=0.29672) *

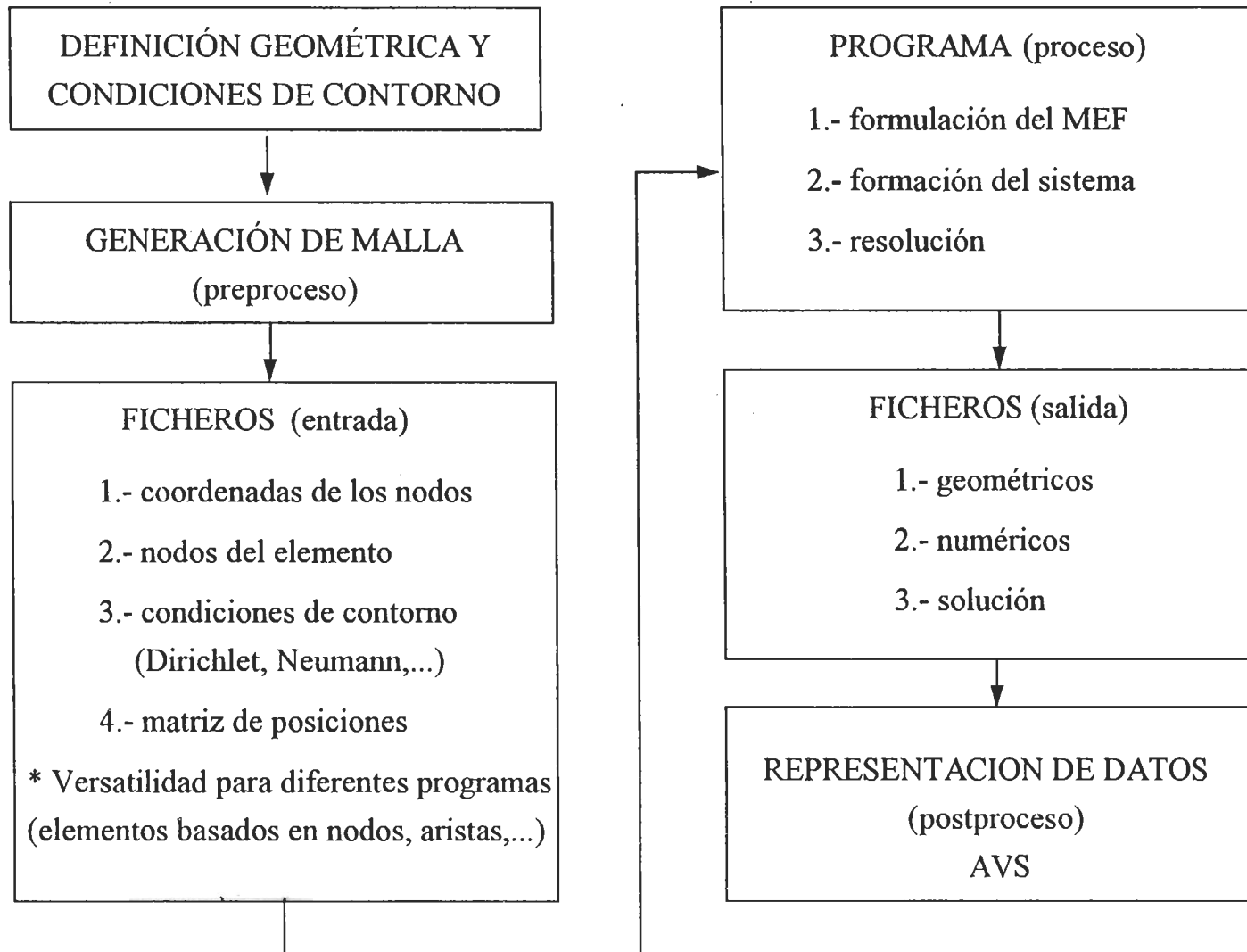


* MALLA 19 (Tiempo=0.30017) *



* MALLA 20 (Tiempo=0.30361) *

Principales etapas del código 3-D



Ecuaciones de Maxwell

$$\vec{\nabla} \cdot \vec{D} = \rho$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\vec{\nabla} \cdot \vec{B} = 0$$

$$\vec{\nabla} \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J}_{cond} = \vec{J}_{tot}$$

$$\vec{D} = \epsilon \vec{E}$$

$$\vec{B} = \mu \vec{H}$$

\vec{E} : campo eléctrico, \vec{D} : vector inducción eléctrica, \vec{B} : campo magnético, \vec{H} : vector de inducción magnética,
 ϵ : permitividad del medio dieléctrico, μ : permeabilidad del medio, ρ : es la densidad total de carga eléctrica,

\vec{J}_{cond} : **densidad de corriente por** conducción, \vec{J}_{tot} : densidad de corriente total, tal que $\vec{\nabla} \cdot \vec{J}_{tot} = 0$.

Simulación numérica en electromagnetismo

Problemas electrostáticos en 3-D

□ Formulación diferencial

Resolver la ecuación de Poisson

$$\nabla \cdot (\varepsilon(\vec{r}) \nabla \Phi(\vec{r})) = -\rho(\vec{r}), \text{ en } \Omega$$

sujeta a unas condiciones de contorno Dirichlet

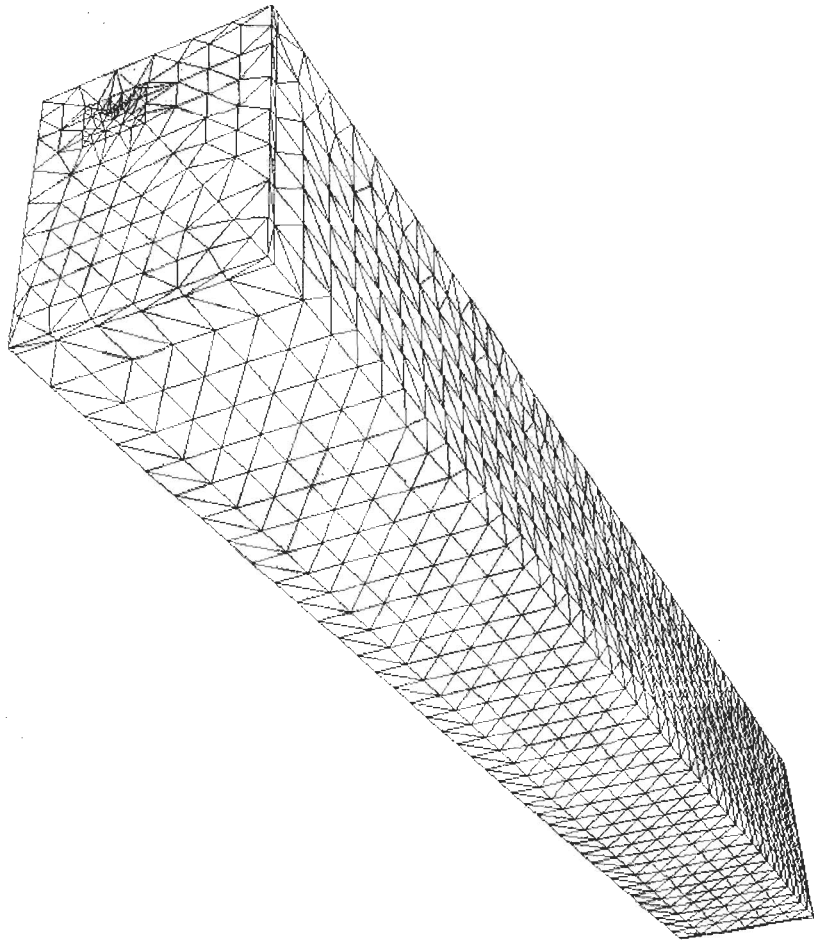
$$\Phi = p \text{ en } \partial\Omega$$

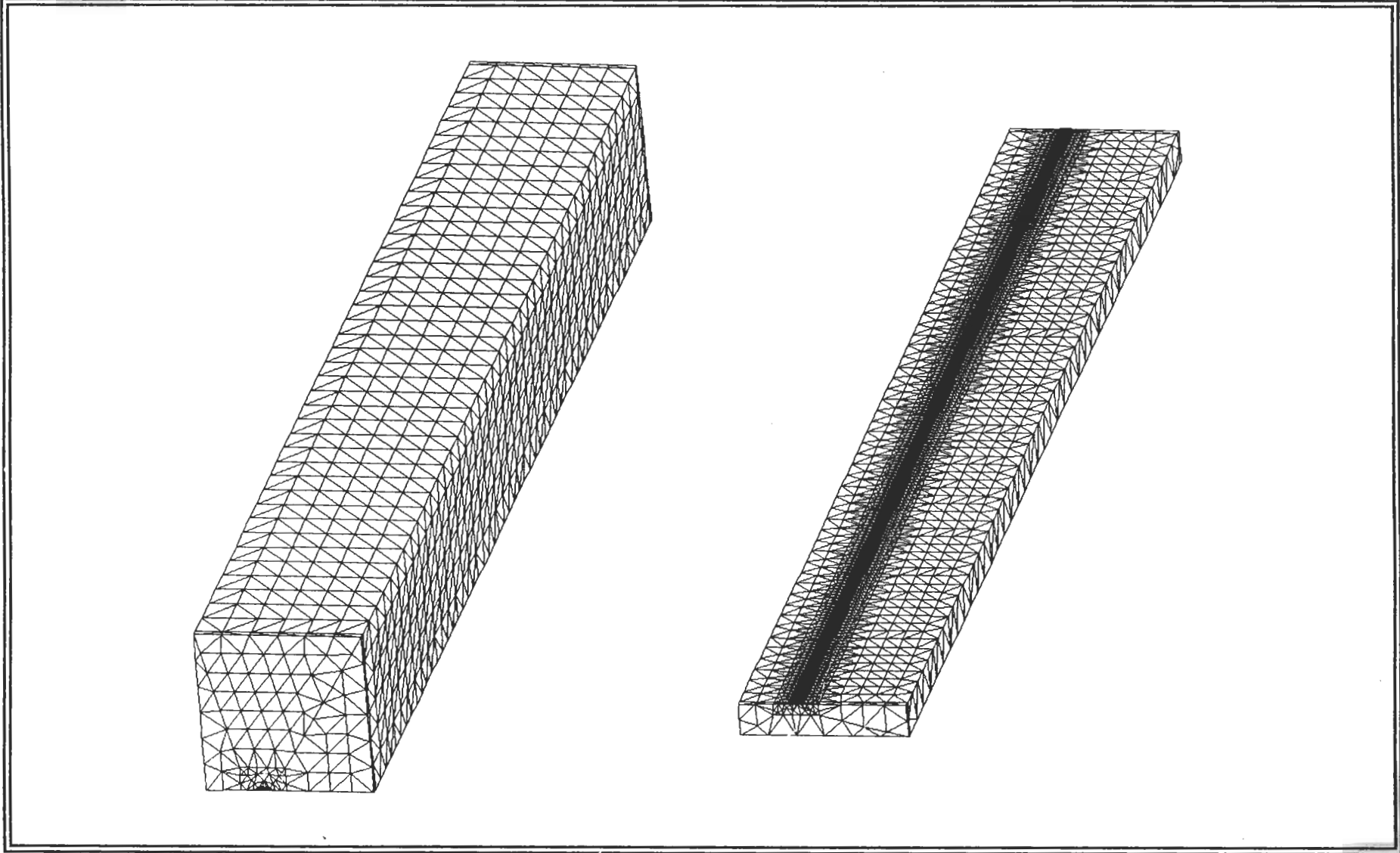
□ Formulación débil del problema

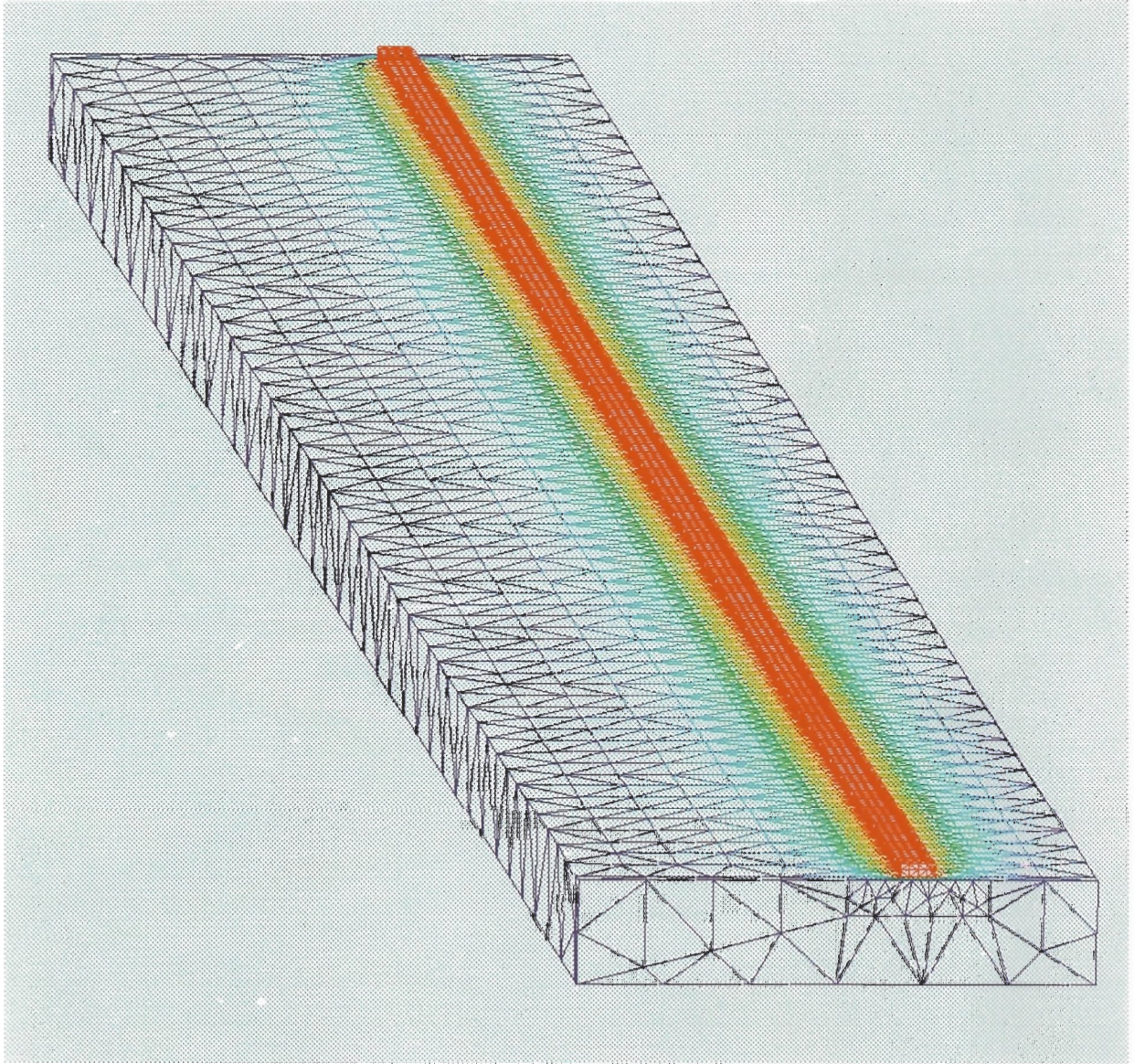
Encontrar una función Φ tal que $\Phi = p$ sobre $\partial\Omega$ y se satisfaga la ecuación

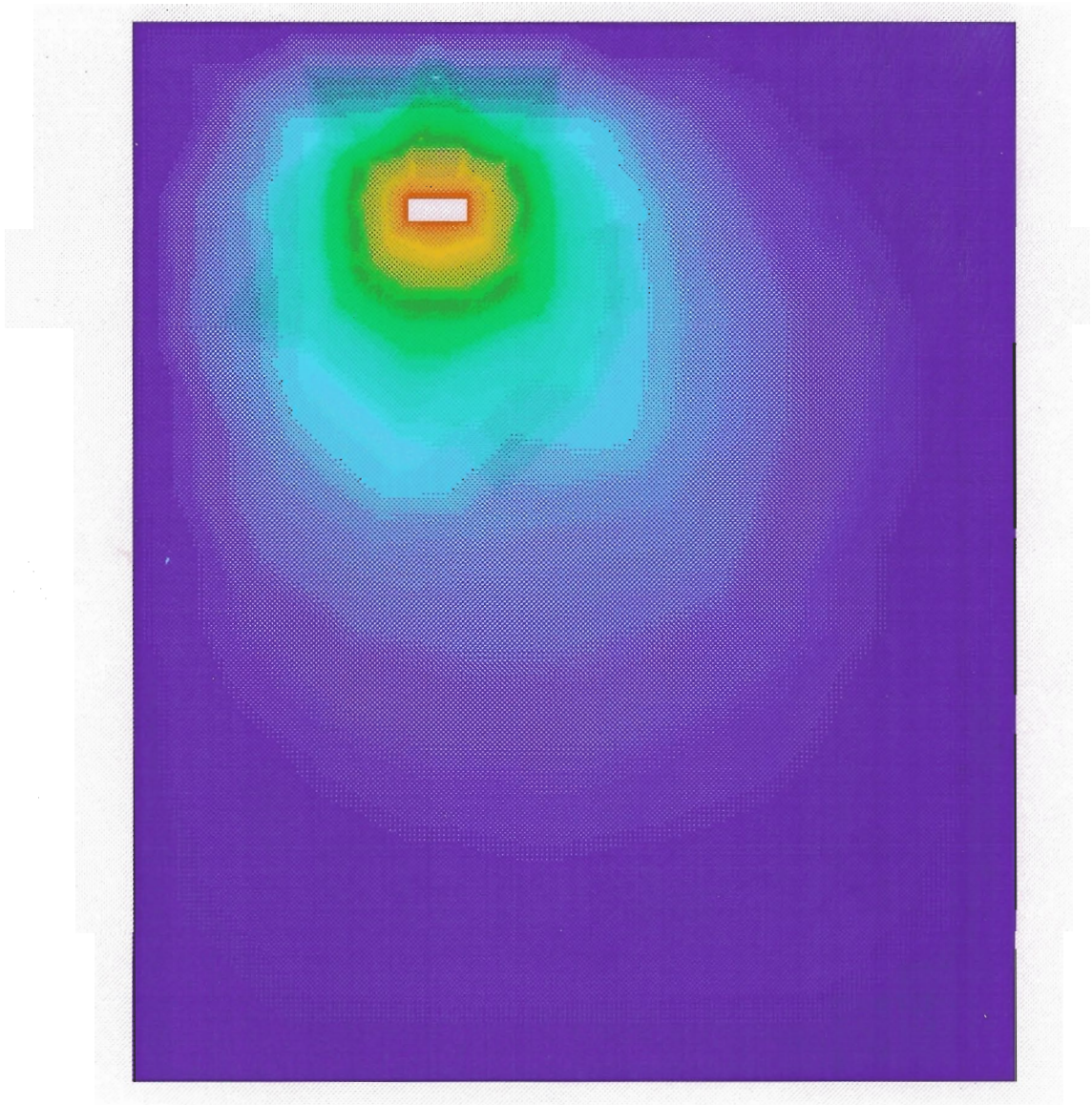
$$\int_{\Omega} \varepsilon(\nabla\Phi) \cdot (\nabla\Psi) d\Omega = \int_{\Omega} \rho\Psi d\Omega$$

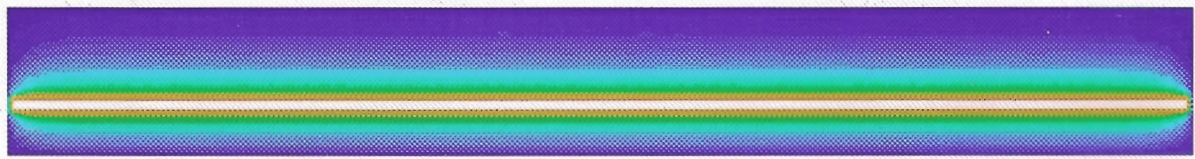
para toda función prueba admisible Ψ .











Ondas electromagnéticas guiadas

Circuitos de microondas de N puertas

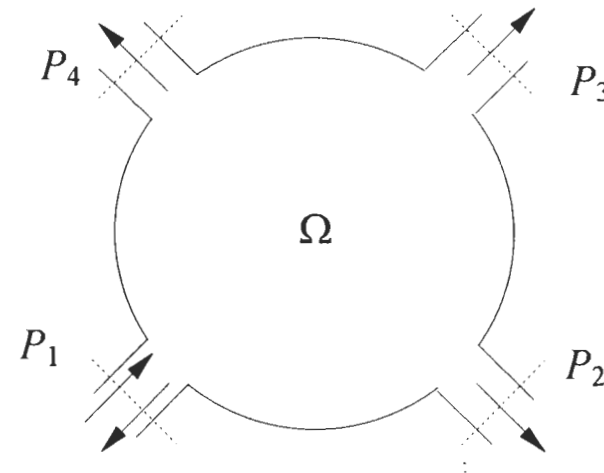
□ Formulación diferencial

Hallar el campo eléctrico \vec{E} que verifican la ecuación (en este caso $k_0 = \omega\sqrt{\epsilon_0\mu_0}$ está prefijado)

$$\nabla \times (\nabla \times \vec{E}) - \epsilon_r \mu_r k_0^2 \vec{E} = \vec{0} \quad \text{en } \Omega$$

sujeta a las condiciones de contorno:

- Conductor: $\vec{n} \times \vec{E} = \vec{0}$
- Puertas: $\vec{n} \times (\nabla \times \vec{E}) + \gamma_i \vec{n} \times (\vec{n} \times \vec{E}) = \vec{U}$,
donde $\vec{U} = \vec{0}$ en P_i $i \neq 1$ y $\vec{U} = -2\gamma_1 \vec{e}_{t_1}$ en P_1 .



□ Formulación variacional

Obtener el mínimo del siguiente funcional

$$F(\vec{E}) = \frac{1}{2} \int_{\Omega} [(\nabla \times \vec{E}) \cdot (\nabla \times \vec{E}) - k^2 \vec{E} \cdot \vec{E}] dV + \frac{1}{2} \int_{\cup P_i} \gamma_i (\vec{n} \times \vec{E}) \cdot (\vec{n} \times \vec{E}) dS - 2 \int_{P_1} \gamma_1 \vec{e}_{t_1} \cdot \vec{E} dS \quad \text{en } \Omega$$

- Las condiciones de contorno en las puertas están incluidas en la formulación.

Simulación numérica en electromagnetismo

□ Discretización

Aproximando \vec{E} mediante *edge elements*, ensamblando las matrices y vectores elementales en $[K]$ y $\{B\}$, respectivamente, e imponiendo la condición de mínimo al funcional, llegamos al sistema de ecuaciones

$$[K]\{E\} = \{B\}$$

Se imponen las condiciones Dirichlet y se resuelve. Hemos empleado el método del **Gradiente Conjugado** con diferentes **precondicionadores** para acelerar la convergencia.

Shape Functions

Shape function on a tetrahedral element Ω_e :

$$\vec{N}_i^e = \pm l_i^e (\zeta_{i_1}^e \nabla \zeta_{i_2}^e - \zeta_{i_2}^e \nabla \zeta_{i_1}^e) \quad (\text{edge element})$$

where $\zeta_{i_k}^e$ is the Lagrange shape function whose value is 1 in the k -th ($k = 1, 2$) node of i -th edge of Ω_e and l_i^e is the length of this edge.

The sign \pm assures an only direction of the edge in the mesh.

Auxiliary functions over elemental surfaces Σ_s :

$$\vec{S}_1^s = \vec{n}_s \times \vec{N}_{i_1}^e \Big|_{\Sigma_s} \quad \vec{S}_2^s = \vec{n}_s \times \vec{N}_{i_2}^e \Big|_{\Sigma_s} \quad \vec{S}_3^s = \vec{n}_s \times \vec{N}_{i_3}^e \Big|_{\Sigma_s}$$

where i_1, i_2 **and** i_3 are the edges of Σ_s and \vec{n}_s is the unitary and normal to Σ_s vector.

Propiedades de los *edge elements*

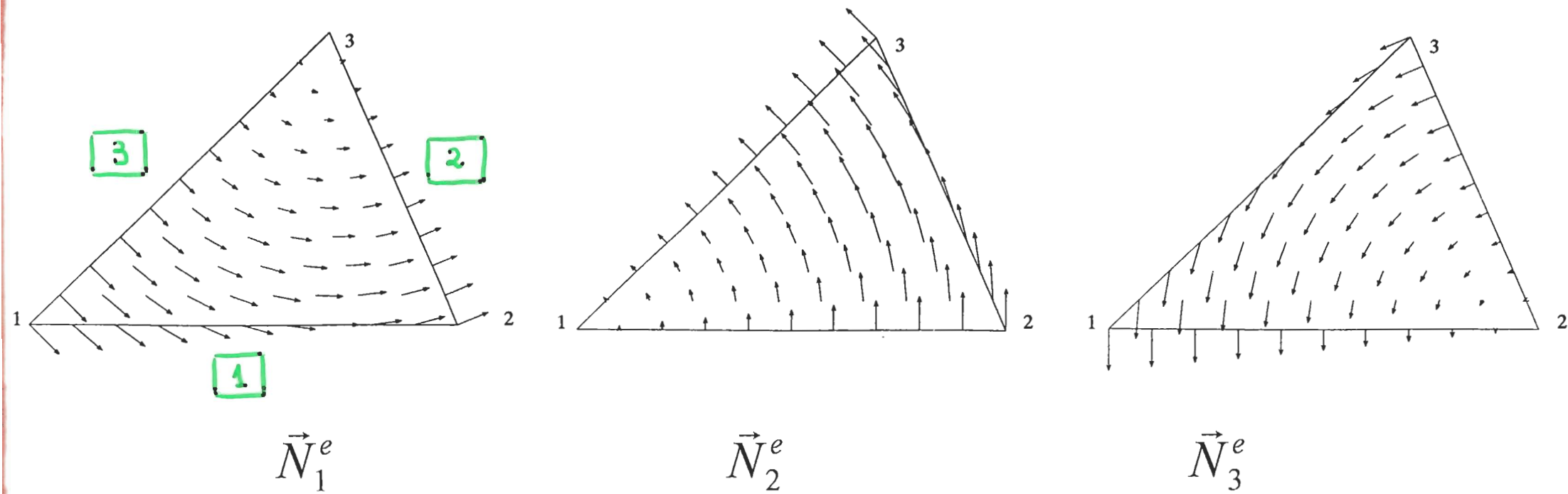
Grados de libertad:

componentes tangenciales del campo en las aristas

Propiedades más relevantes:

- Divergencia nula: $\nabla \cdot \vec{N}_i^e = 0 \Rightarrow \nabla \cdot \vec{E} = 0$ (elimina soluciones *espúreas*)
- Rotacional constante en cada elemento: $\nabla \times \vec{N}_i^e = cte$
- Componente tangencial del campo interpolado continua en las caras, aun siendo discontinua la componente normal \Rightarrow adecuados para imponer condiciones de contorno en conductores y aproximar el campo en la entrecara de materiales con diferentes constantes dieléctricas

Edge elements en 2-D



Interpolación del campo eléctrico

Campo eléctrico interpolado en el elemento Ω_e

$$\vec{E}_h^e = \sum_{i=1}^6 E_i^e \vec{N}_i^e = \{E^e\}^T \{\vec{N}^e\} = \{\vec{N}^e\}^T \{E^e\}$$

donde E_i^e son las componentes tangenciales de \vec{E}_h^e sobre las aristas del tetraedro,

$$\{E^e\}^T = (E_1^e, E_2^e, \dots, E_6^e) \text{ y } \{\vec{N}^e\}^T = (\vec{N}_1^e, \vec{N}_2^e, \dots, \vec{N}_6^e)$$

Componente tangencial del campo interpolado sobre Σ_s

$$\vec{n}_s \times \vec{E}_h^s = \sum_{i=1}^3 E_i^s \vec{S}_i^s = \{E^s\}^T \{\vec{S}^s\} = \{\vec{S}^s\}^T \{E^s\}$$

Cálculo de los parámetros S

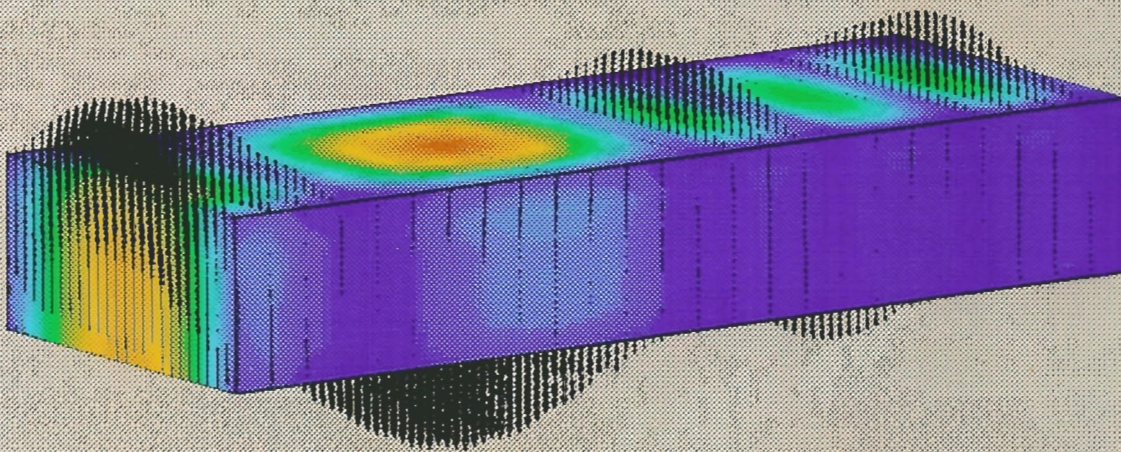
Cálculo de los parámetros S_{i1} a partir del campo interpolado mediante elementos finitos, \vec{E}_i , en cada puerta

$$S_{11} = \frac{1}{Z_{w1}^*} \iint_{P_1} \vec{E}_1 \cdot \vec{e}_{t_1}^* dS - 1$$

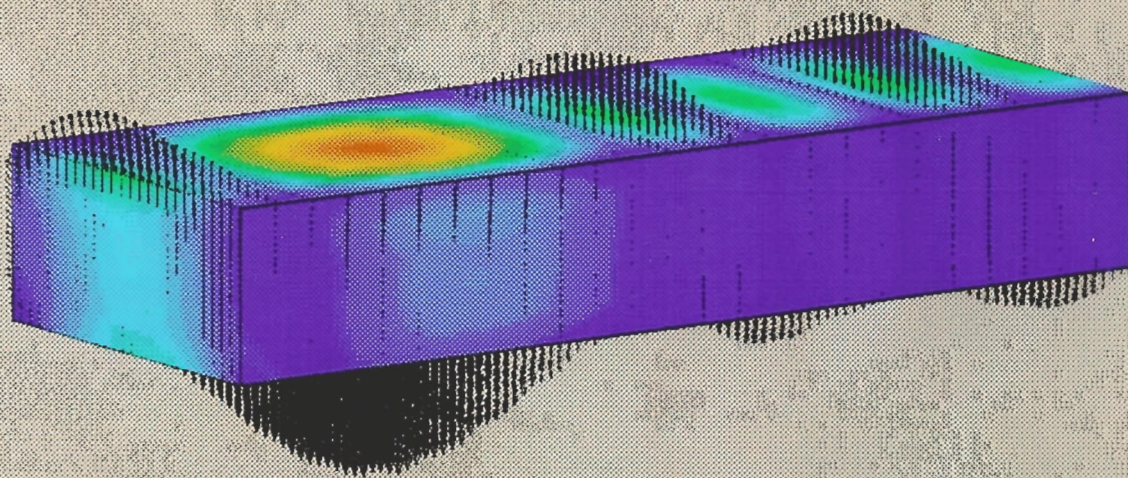
$$S_{i1} = \frac{1}{Z_{w_i}^*} \iint_{P_i} \vec{E}_i \cdot \vec{e}_{t_i}^* dS \quad i \neq 1$$

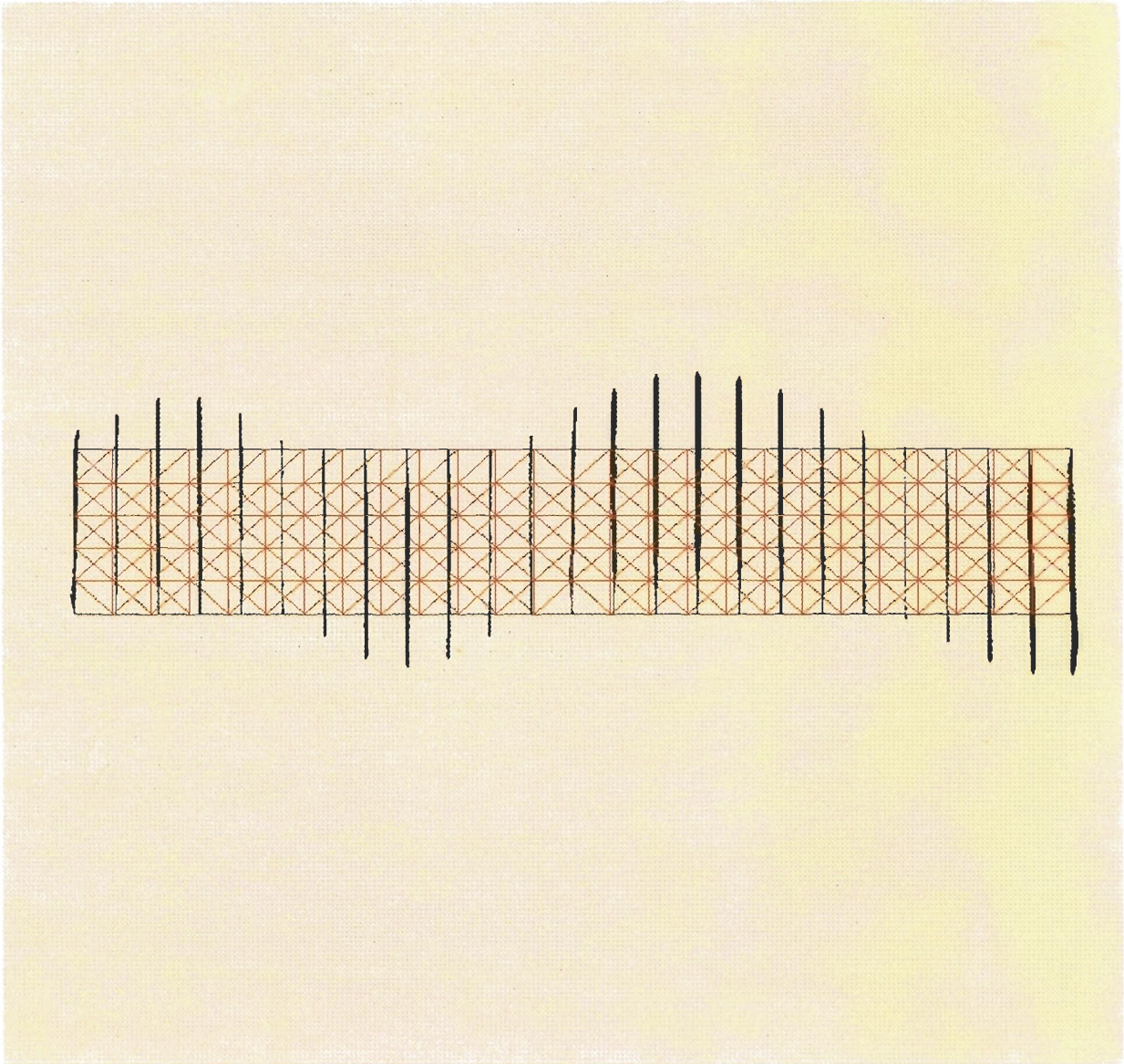
- ⇒ Los parámetros S se obtienen como un **promedio del campo sobre toda la puerta.**
(Resultados más exactos que los obtenidos a través de expresiones en las que sólo se consideren valores puntuales)

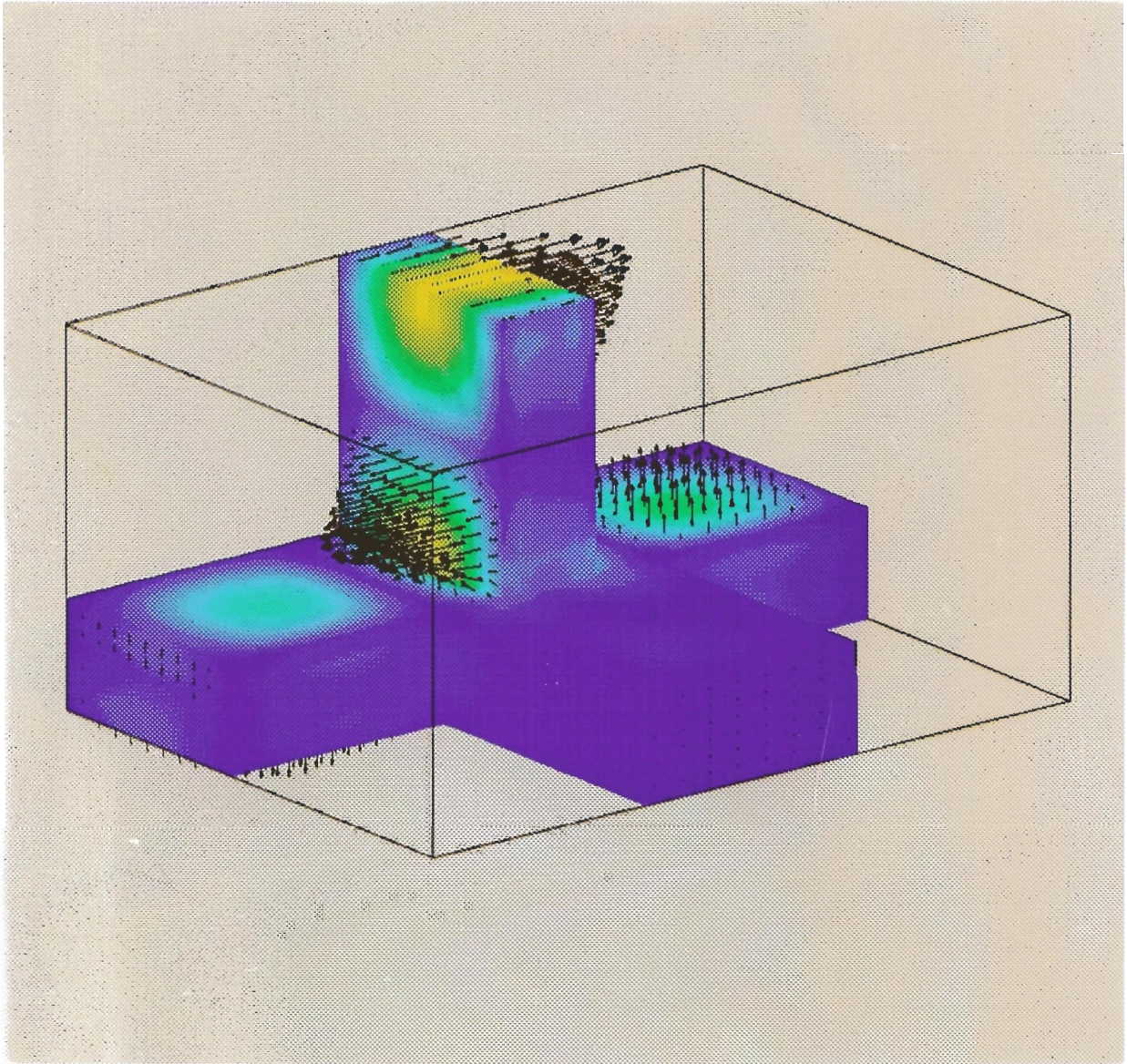
Cambio de dielectrico en una guia: $\text{Re}(E)$

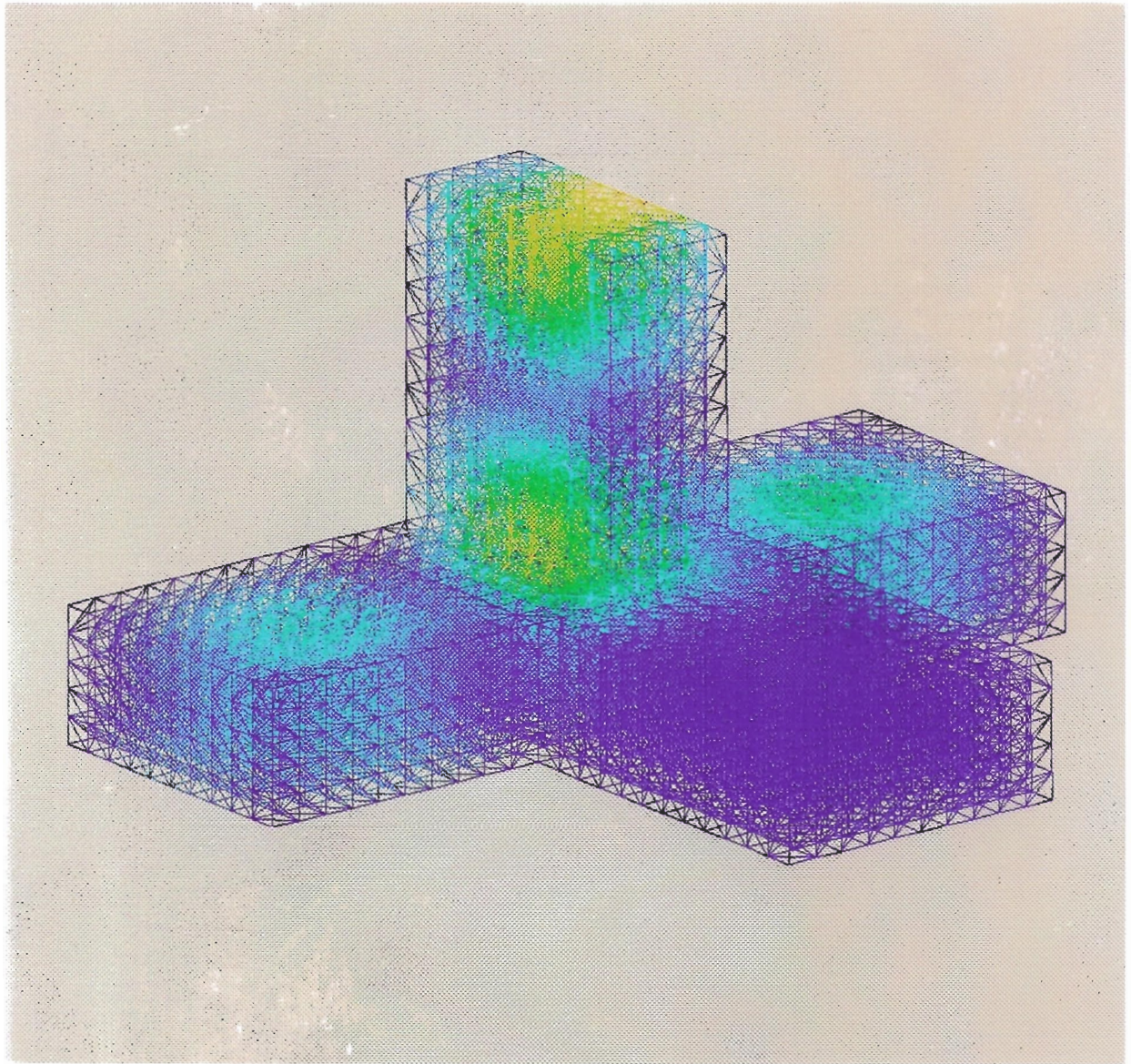


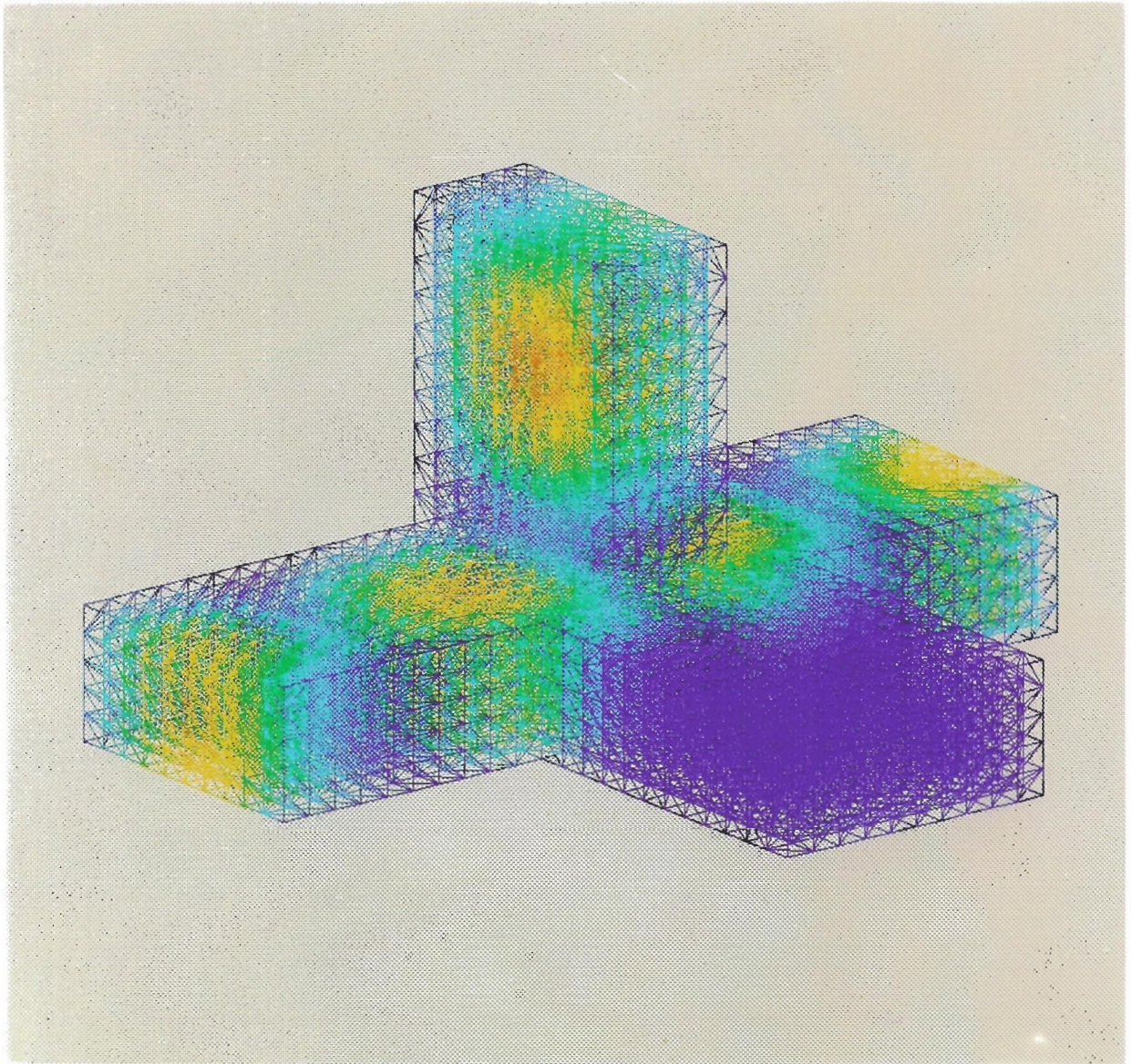
Cambio de dielectrico en una guia: $\text{Im}(E)$





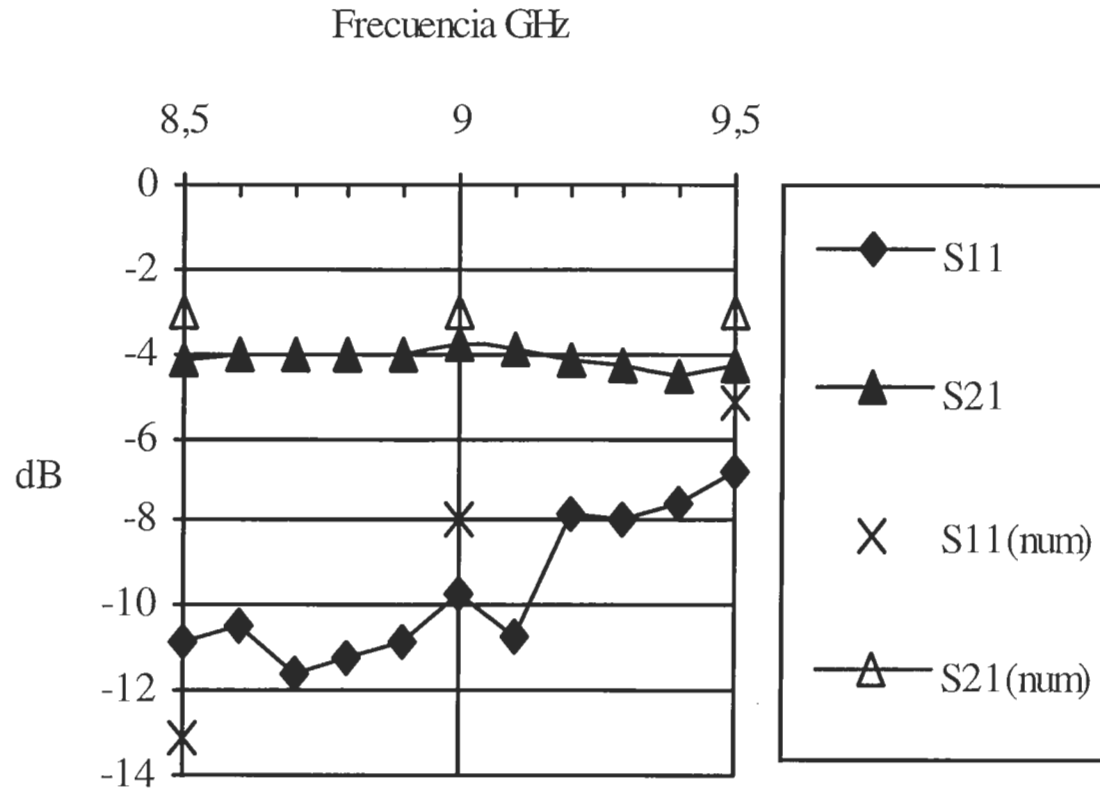




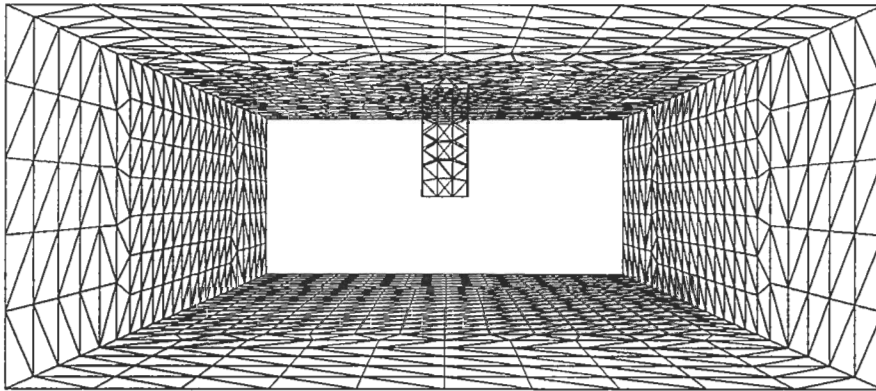


Parámetros S en una T-híbrida

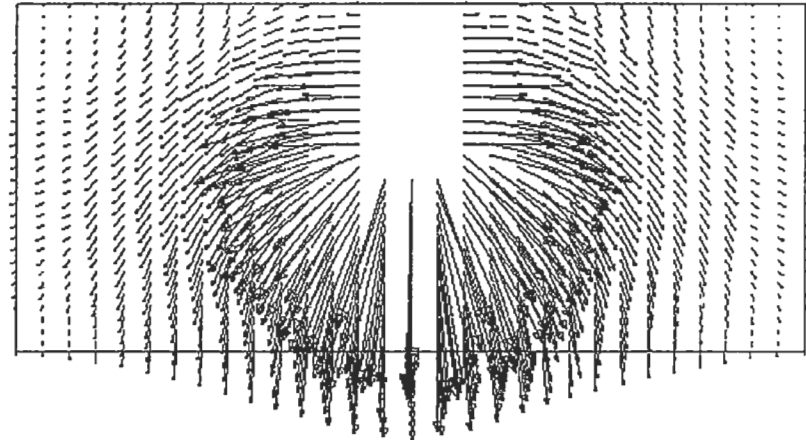
Comparación entre resultados numéricos y experimentales



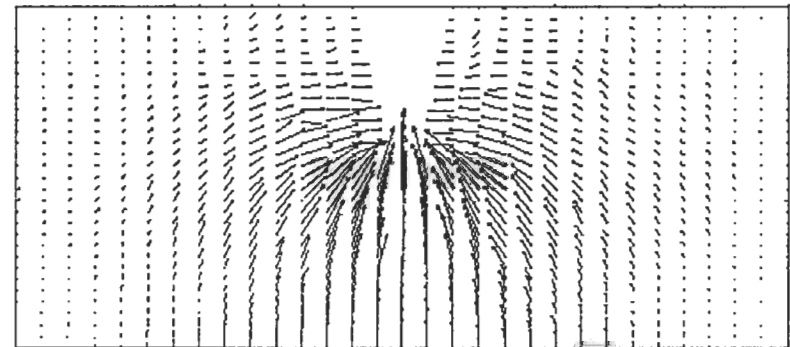
Wave Guide with a Conductive Post



Inner view of surface triangulation



$\text{Re}(\mathbf{E})$



$\text{Im}(\mathbf{E})$

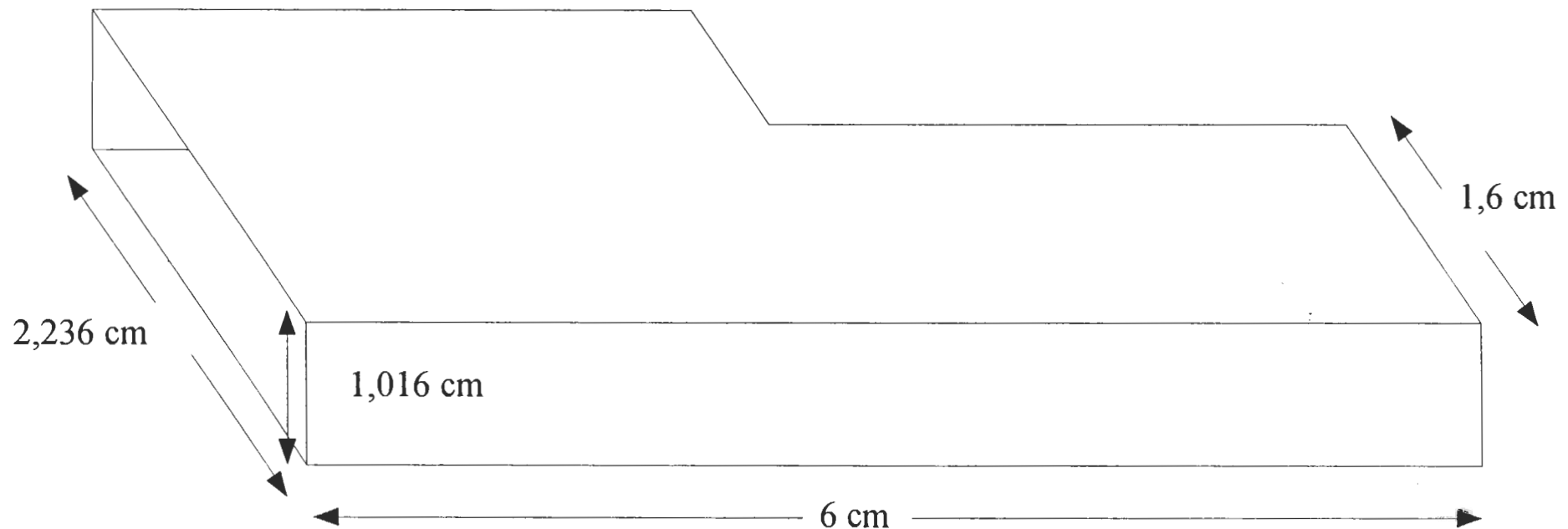
Estrechamiento en una guía rectangular

Datos:

Frecuencia de trabajo: 10 GHz.

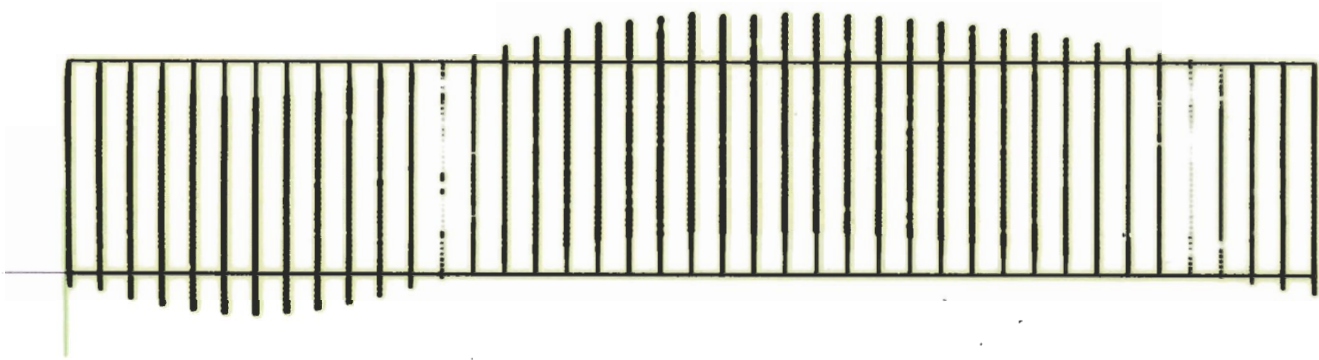
Módulo del coeficiente de reflexión (analítico): 0,43

Módulo del coeficiente de reflexión (numérico): 0,47

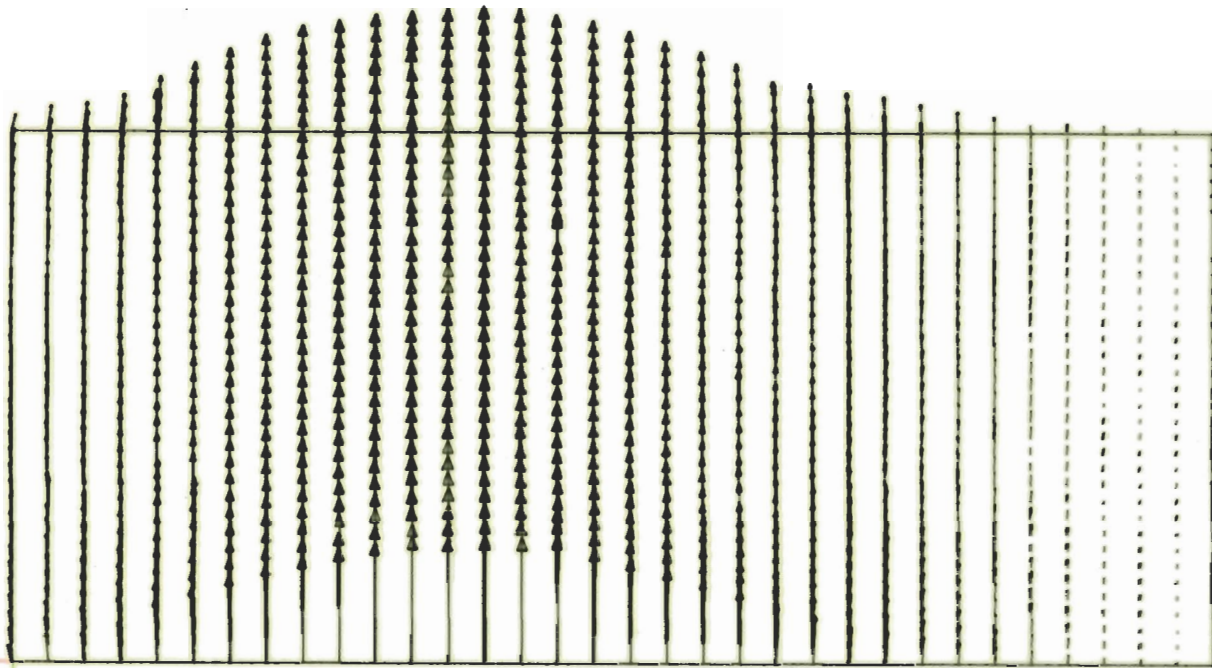




Corte longitudinal: Parte Real del campo

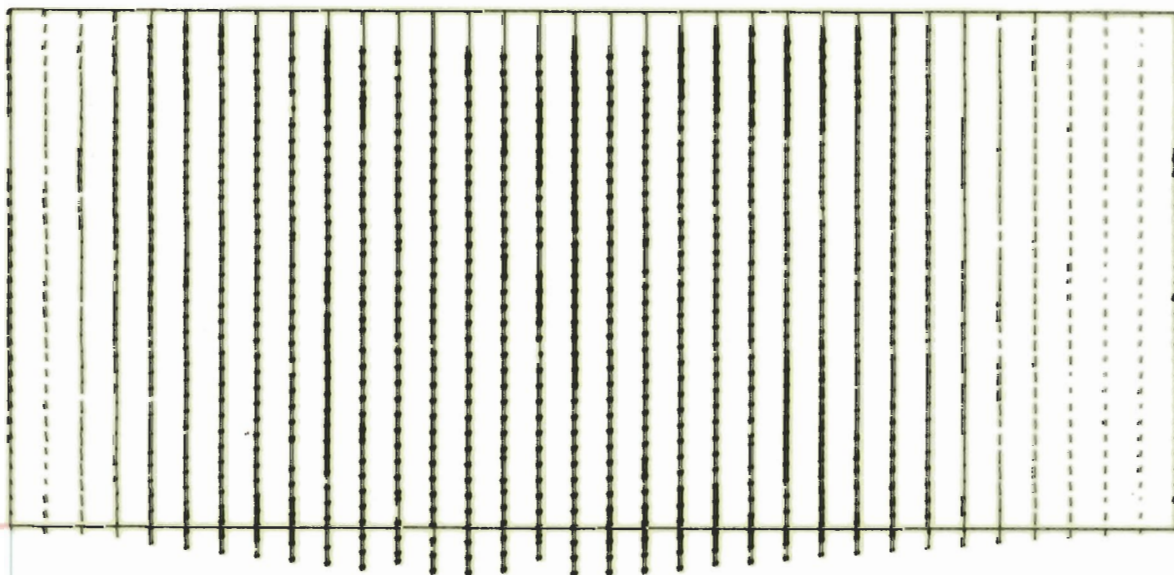


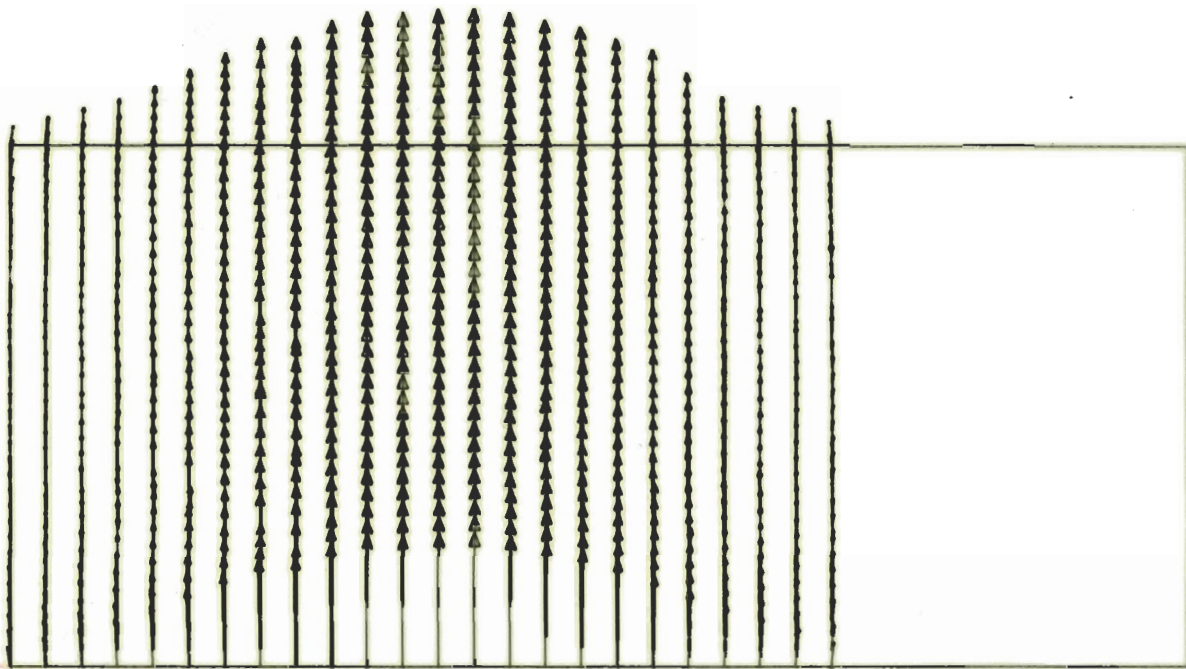
Corte longitudinal: Parte Imaginaria del campo



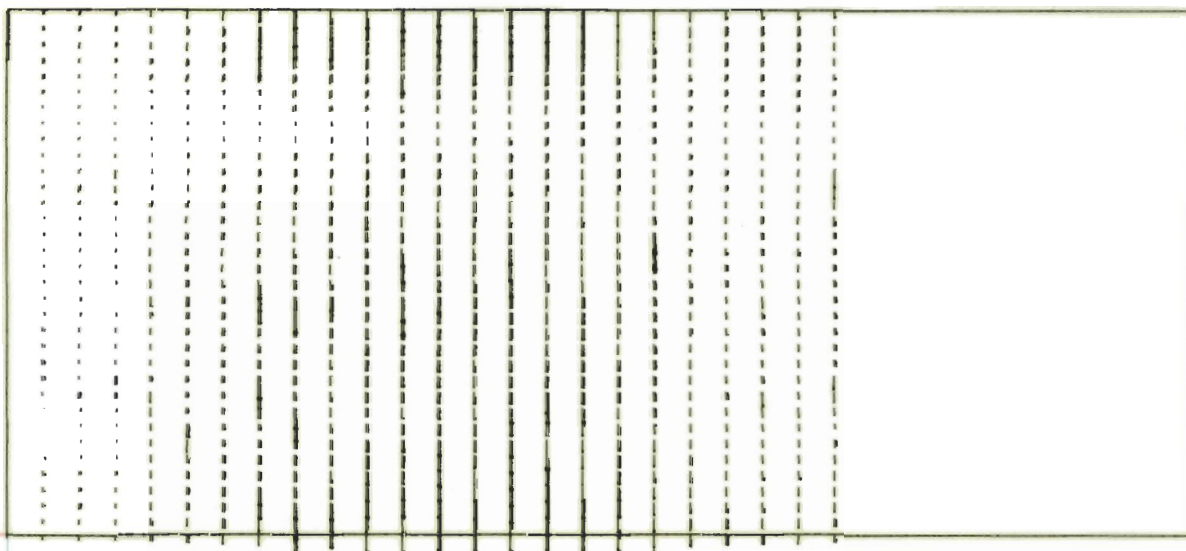
Parte Imaginaria del campo en la guía ancha

Parte Real del campo en la guía ancha

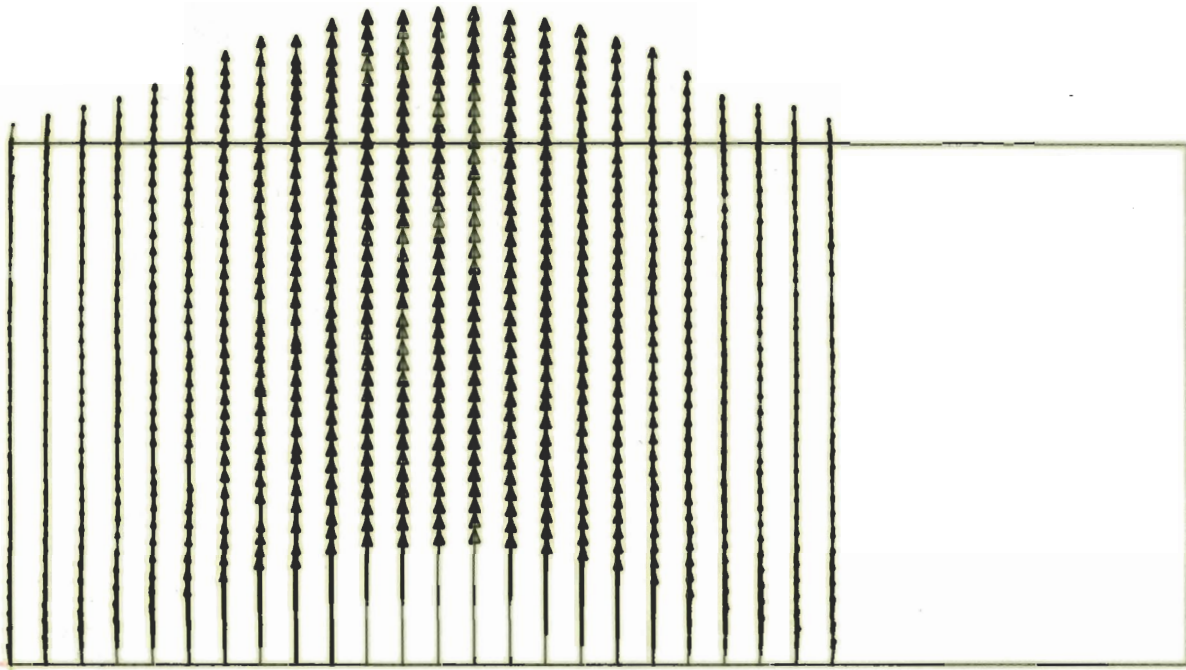




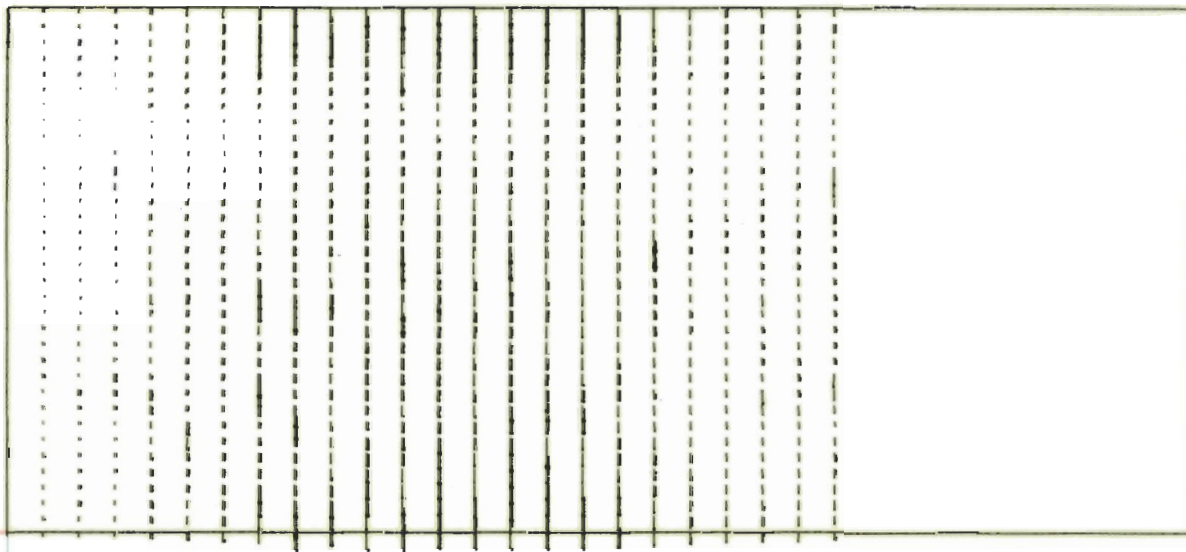
Parte Imaginaria del campo en la guía estrecha



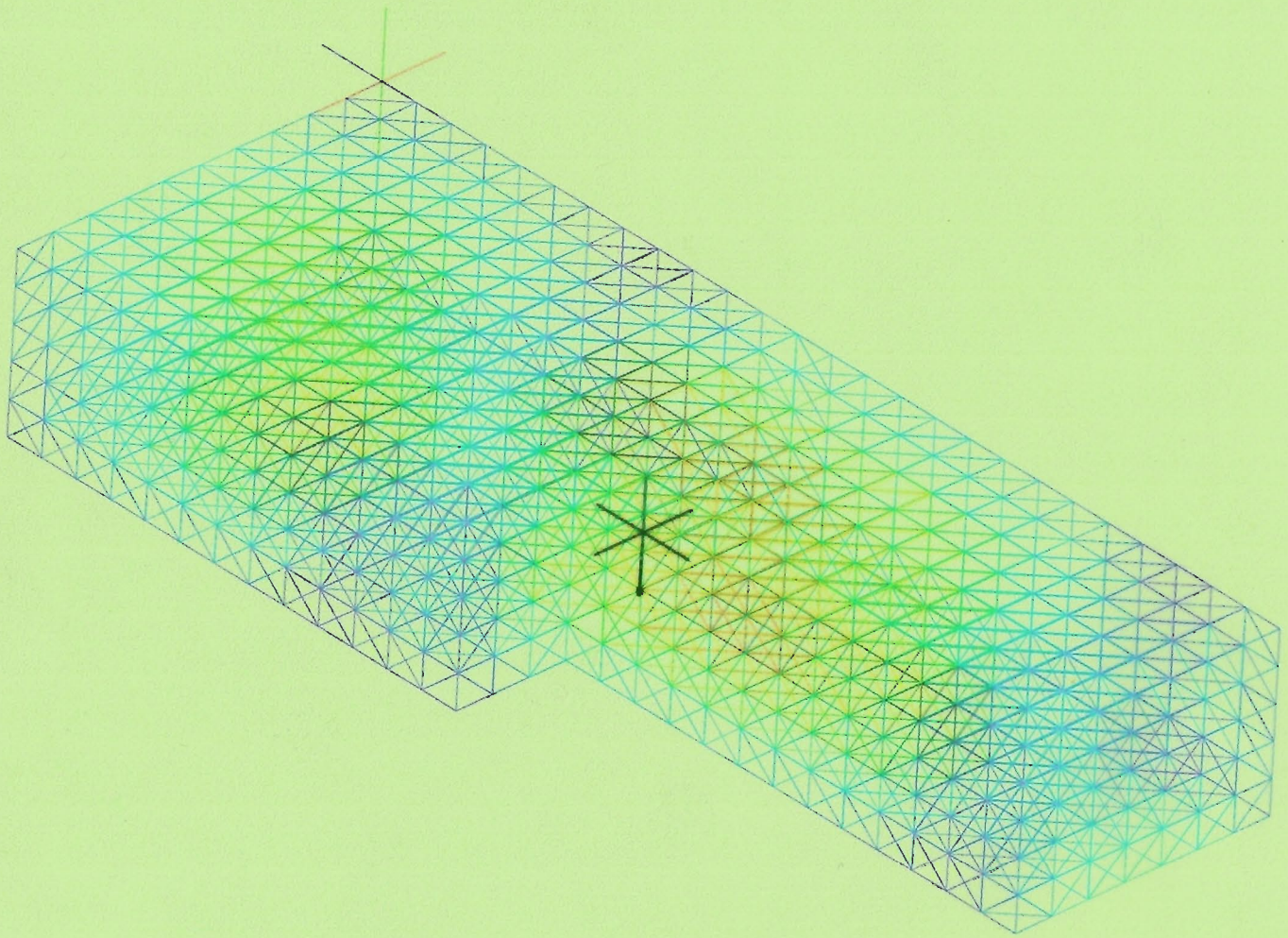
Parte Real del campo en la guía estrecha



Parte Imaginaria del campo en la guía estrecha



Parte Real del campo en la guía estrecha



INTRODUCCIÓN

OBJETIVO:

– Cavity 3D:

- Paredes perfectamente conductoras
- Medio isótropo y parcialmente homogéneo
- Medio no conductor
- No existen fuentes



Frecuencias de resonancia

Modos de resonancia



MEF

PLANTEAMIENTO

Ecuación diferencial

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \mathbf{E} \right) - k_0^2 \epsilon_r \mathbf{E} = 0$$

k_0^2 : autovalor
 \mathbf{E} : autofunción

Condición de contorno sobre la pared conductora

$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{cuando} \quad \mathbf{r} \in \partial \Omega$$

Problema variacional correspondiente

$$\int_{\Omega} \frac{1}{\mu_r} (\nabla \times \mathbf{E}) \cdot (\nabla \times \mathbf{F}) d\Omega = k_0^2 \int_{\Omega} \epsilon_r \mathbf{E} \cdot \mathbf{F} d\Omega$$

$\mathbf{n} \times \mathbf{E} = 0$ sobre $\partial \Omega$, para toda función prueba \mathbf{F} suficientemente regular tal que $\mathbf{n} \times \mathbf{F} = 0$ sobre $\partial \Omega$

Discretización

$$\mathbf{E}^e = \sum_{i=1}^6 E_i^e \mathbf{N}_i^e = \{E^e\}^T \{N^e\}$$

PLANTEAMIENTO

$$\sum_{e=1}^E \{E^e\} [k^e] \{F^e\} = k_0^2 \sum_{e=1}^E \{E^e\} [m^e] \{F^e\}$$

Matrices elementales

$$[k^e] = \int_{\Omega} \frac{1}{\mu_r} (\nabla \times \{N^e\}) \cdot (\nabla \times \{N^e\}^T) d\Omega$$

$$[m^e] = \int_{\Omega} \epsilon_r \{N^e\} \cdot \{N^e\}^T d\Omega$$

Ensamblaje

Matrices elementales



Matrices globales

$$[k^e]$$



$$[K]$$

$$[m^e]$$

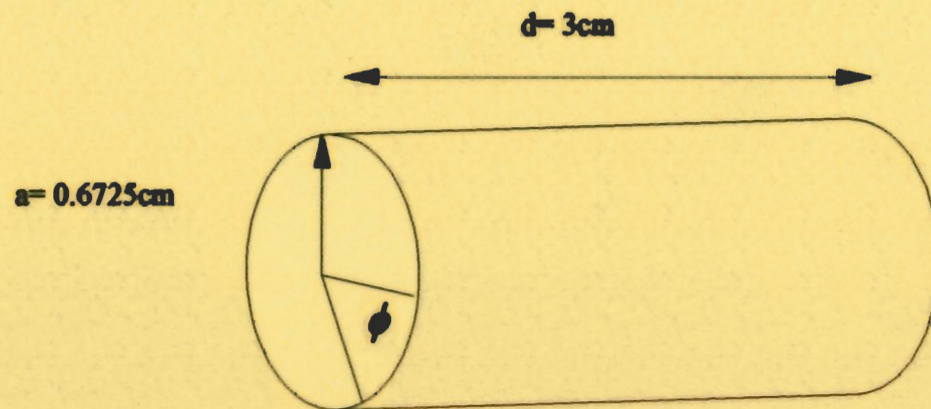


$$[M]$$

Problema de autovalores generalizado

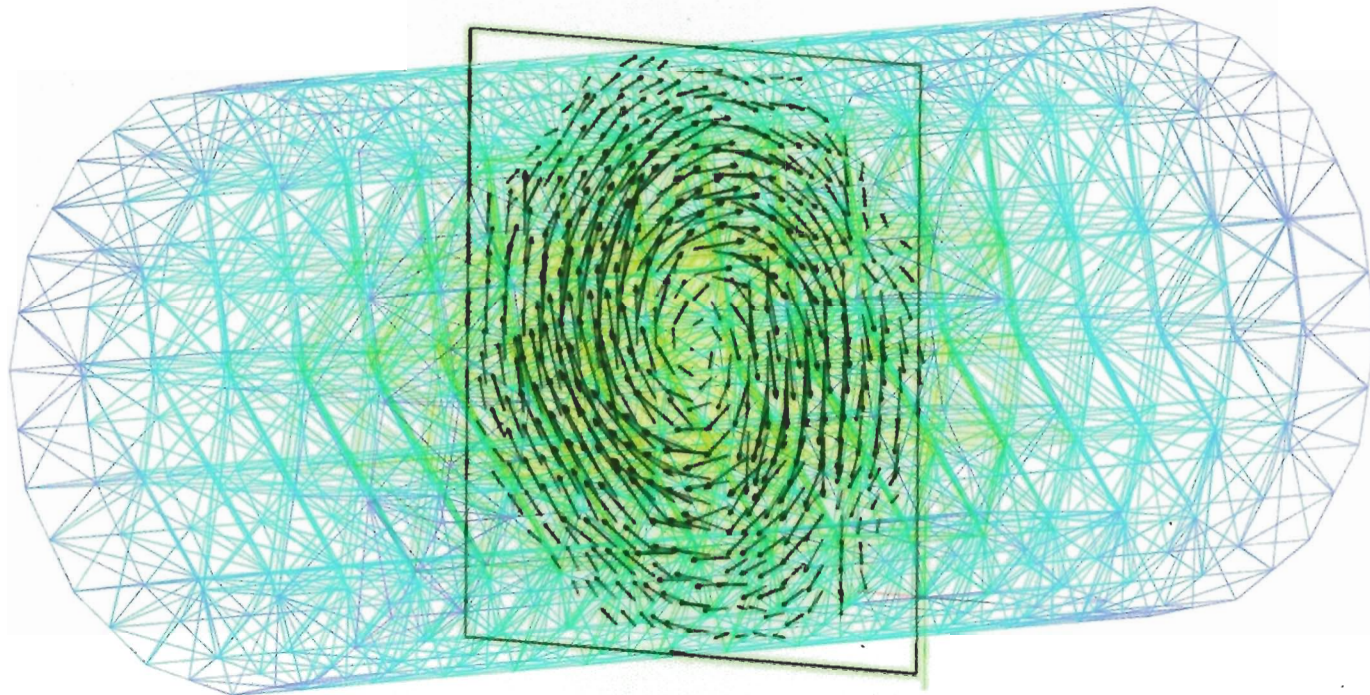
$$[K] \{E\} = k_0^2 [M] \{E\}$$

Cavidad cilíndrica



Geometría de la cavidad

CAVIDAD CILINDRICA TE011



CAVIDAD CILINDRICA TE011



Cavidad cilíndrica

modos	fr. Cal.	fr. Obt.
TE_{111}	2.9312	2.9415
TE_{112}	3.447	3.48804
TM_{010}	3.576	3.56165
TM_{011}	3.726	*
TM_{012}	4.1442	4.17400
TE_{113}	4.1671	4.17500
TE_{211}	4.6606	4.73567
TM_{013}	4.76	4.74327
TE_{212}	5.0012	*
TE_{114}	5.004	5.00108
TM_{014}	5.5076	5.48615
TE_{213}	5.522	*
TE_{011}	5.7932	5.79968
TE_{115}	5.9085	*
TE_{012}	6.0705	*

SIMULACIÓN DE CAMPOS DE VIENTOS

Sea $\Omega \subset \mathbb{R}^d$ un dominio acotado de frontera $\Gamma = \Gamma_1 \cup \Gamma_2$,

\vec{u}_0 : campo de viento obtenido mediante interpolación de datos experimentales.

□ Objetivo: encontrar un campo de velocidades \vec{u} que se ajuste a \vec{u}_0 , verificando,

□ Condición de incompresibilidad: $\operatorname{div} \vec{u} = 0$ en Ω

□ Condición de impermeabilidad: $\vec{u} \cdot \vec{n} = 0$ en Γ_1

□ Entonces, el campo \vec{u} es la solución del problema: "Hallar $\vec{u} \in K$ que verifique,

$$\left\{ \begin{array}{l} J(\vec{u}) = \min_{\vec{v} \in K} J(\vec{v}) \\ K = \{ \vec{v}; \operatorname{div} \vec{v} = 0, \vec{v} \cdot \vec{n}|_{\Gamma_1} = 0 \} \end{array} \right. \quad \text{donde} \quad J(\vec{v}) = \frac{1}{2} \int_{\Omega} (\vec{v} - \vec{u}_0)^t P (\vec{v} - \vec{u}_0)$$

□ Este problema puede ser formulado introduciendo un multiplicador de Lagrange.

$$L(\vec{v}, q) = J(\vec{v}) + \int_{\Omega} q \operatorname{div} \vec{v}$$

$$H_{0,\Gamma_2}^1(\Omega) = \left\{ \varphi \in H^1(\Omega); \varphi|_{\Gamma_2} = 0 \right\} \quad H(\operatorname{div}, \Omega) = \left\{ \vec{v} \in (L^2(\Omega))^d; \operatorname{div} \vec{v} \in L^2(\Omega) \right\}$$

$$H_{0,\Gamma_1}(\operatorname{div}, \Omega) = \left\{ \vec{v} \in H(\operatorname{div}, \Omega); \int_{\Gamma} \vec{v} \cdot \vec{n} \varphi = 0, \forall \varphi \in H_{0,\Gamma_2}^1(\Omega) \right\}$$

□ Buscamos el punto silla del lagrangiano: (\vec{u}, λ) en $H_{0,\Gamma_1}(\operatorname{div}, \Omega) \times L^2(\Omega)$ tal que,

$$L(\vec{u}, q) \leq L(\vec{u}, \lambda) \leq L(\vec{v}, \lambda) \quad \forall q \in L^2(\Omega) \text{ y } \forall \vec{v} \in H_{0,\Gamma_1}(\operatorname{div}, \Omega)$$

$$\frac{\partial}{\partial \vec{v}} L(\vec{u}, \lambda) = 0 \quad \text{y} \quad \frac{\partial}{\partial q} L(\vec{u}, \lambda) = 0$$

$$\int_{\Omega} \vec{v}^t P \vec{u} + \int_{\Omega} \lambda \operatorname{div} \vec{v} = \int_{\Omega} \vec{v}^t P \vec{u}_0 \quad \forall \vec{v} \in H_{0,\Gamma_1}(\operatorname{div}, \Omega)$$

$$\int_{\Omega} q \operatorname{div} \vec{u} = 0 \quad \forall q \in L^2(\Omega)$$

□ Si además consideramos que $\lambda \in H^1(\Omega)$, se obtiene la siguiente formulación:

$$-\vec{\nabla} \cdot (P^{-1} \vec{\nabla} \lambda) = \vec{\nabla} \cdot \vec{u}_0 \quad \text{en } \Omega$$

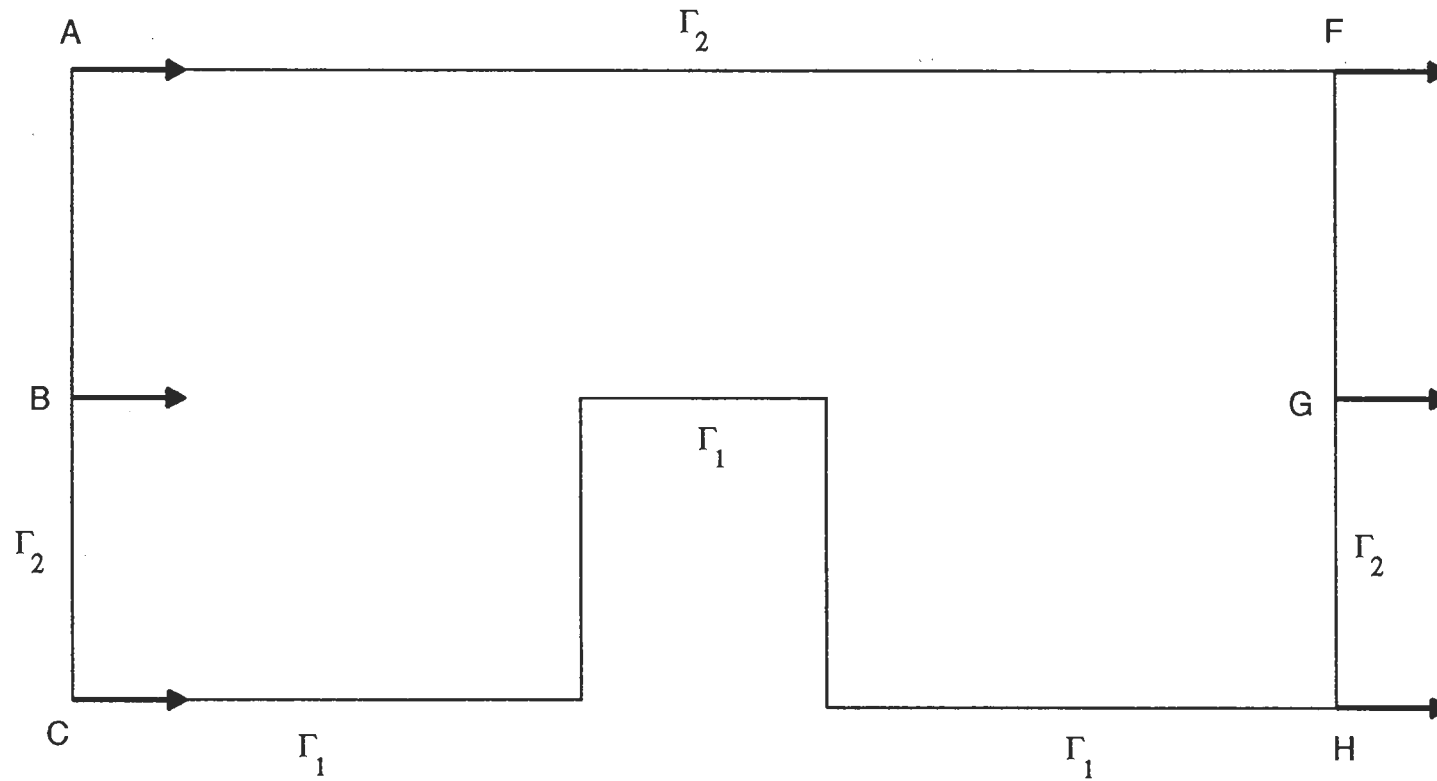
$$-P^{-1} \frac{\partial \lambda}{\partial \vec{n}} = \vec{n} \cdot \vec{u}_0 \quad \text{en } \Gamma_1$$

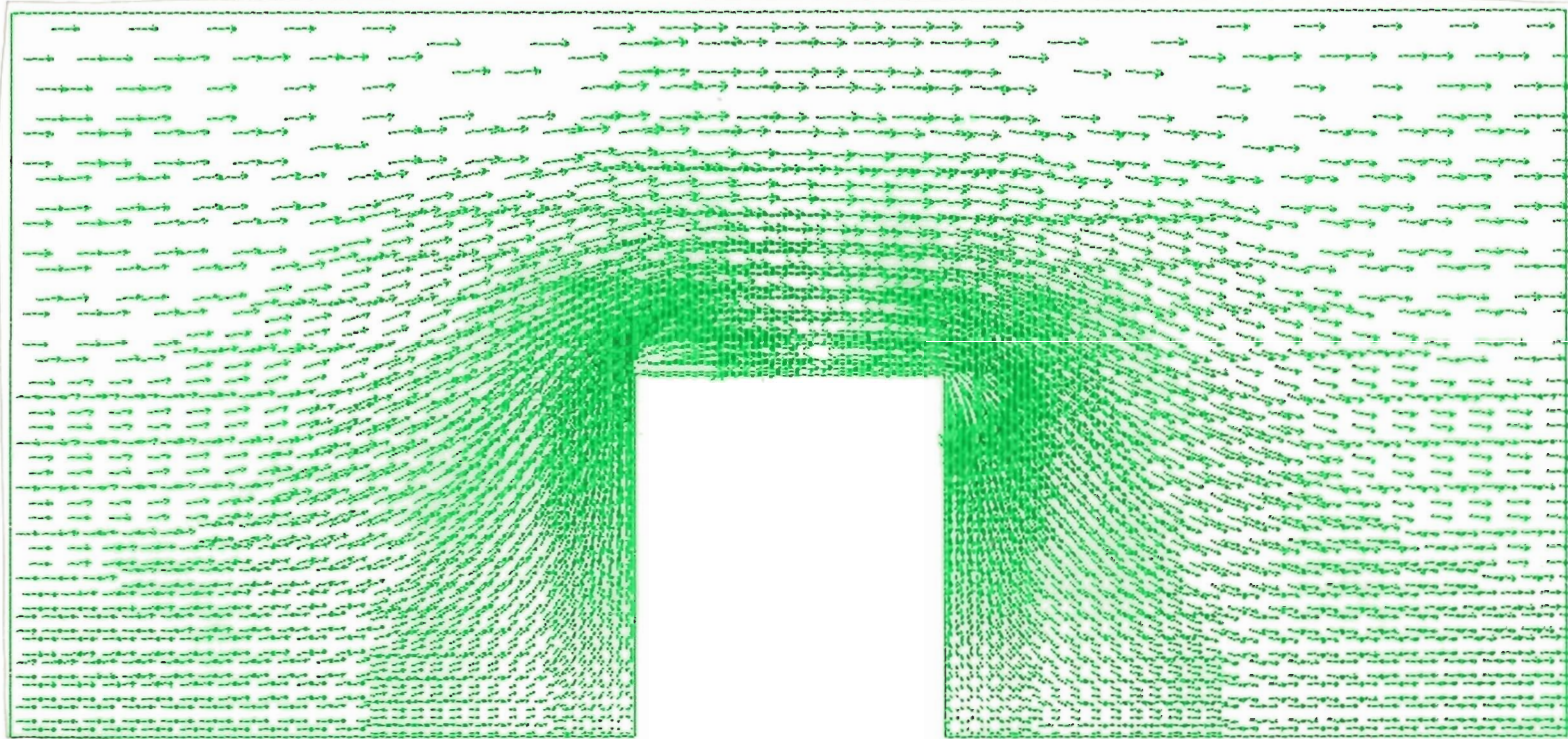
$$\lambda = 0 \quad \text{en } \Gamma_2$$

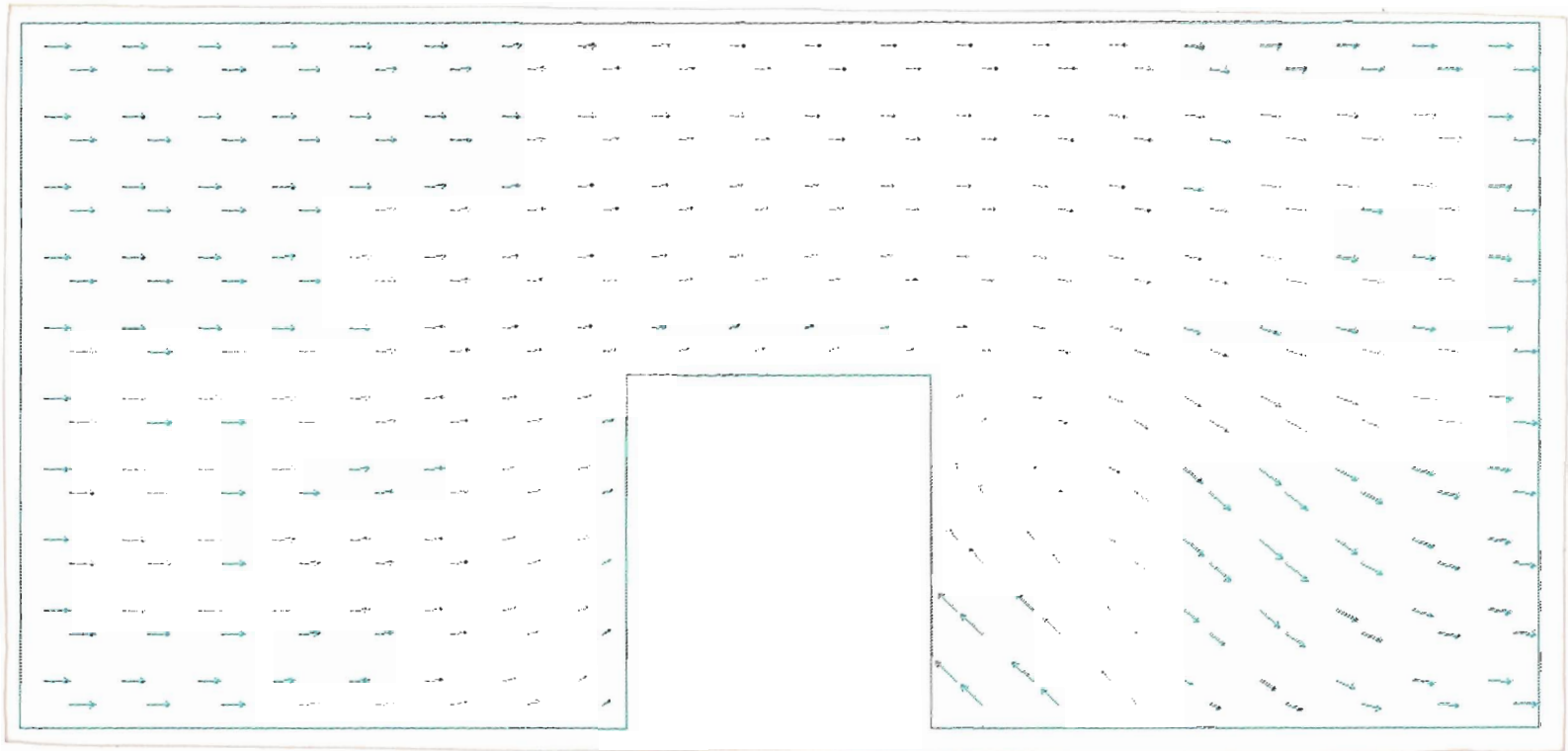
donde

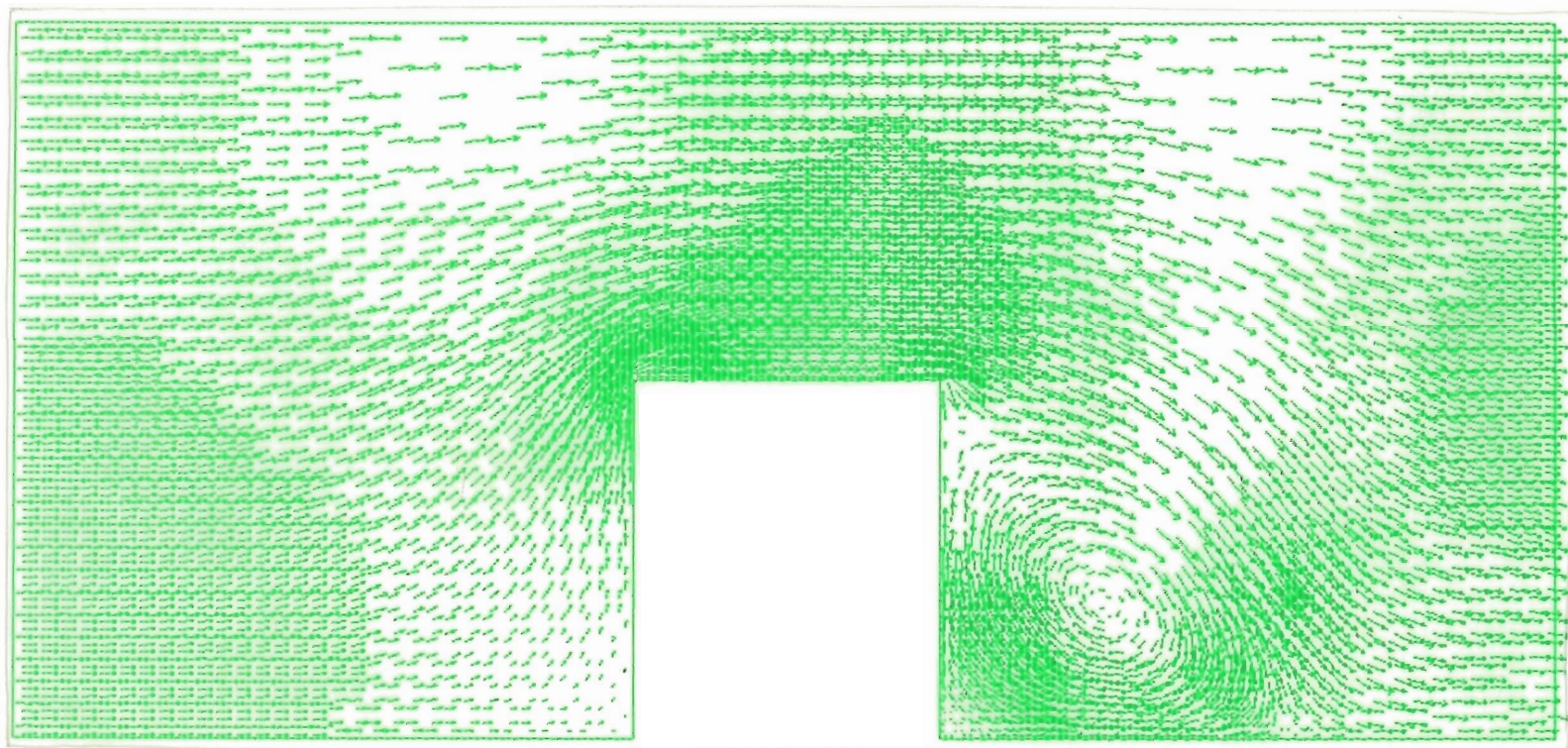
$$\vec{u} = \vec{u}_0 + P^{-1} \vec{\nabla} \lambda \quad \text{en } \Omega$$

Un Problema Modelo en 2-D









Un Problema Modelo en 2-D

