

**TECNICAS PRACTICAS DE GENERACION DE REDES
SU INCORPORACION AL GENERADOR DE REDES ENREDO**

Ing. Venere Marcelo Javier
Seccion Mecanica Computacional
Centro Atomico Bariloche

RESUMEN

En este trabajo se describen algunos metodos de generacion de redes que pueden ser incorporados facilmente a generadores existentes. Estos amplian considerablemente las posibilidades de generacion, y permiten obtener redes de mejor calidad.

Entre las mas importantes estan: pegado de redes, generacion adaptiva a una solucion, optimizacion de la forma de los elementos, insercion de elementos, y digitalizacion.

El pegado de redes consiste en pegar dos o mas redes uniendo los nodos con coordenadas iguales. La generacion adaptiva a una solucion permite obtener redes densificadas de acuerdo al problema que se quiere resolver. La optimizacion de la forma de los elementos, re-ubica los nodos internos para obtener elementos de formas adecuadas. La insercion de elementos permite, mediante la division de los mallas, densificar la red localmente. Y la digitalizacion permite modificar las coordenadas de los nodos dando la nueva ubicacion sobre el grafico de la red.

Se describira en forma general al generador de redes ENREDO en su version actual; y en forma mas detallada las nuevas tecnicas incorporadas, y las posibilidades que ofrecen.

INTRODUCCION

Las técnicas de generación desarrolladas en el presente trabajo, han sido implementadas dentro del programa ENREDO. Este es un generador de redes de Elementos Finitos que trabaja en dos dimensiones, con distintos tipos de elementos, y geometría absolutamente general.

Su programación ha sido estructurada en forma modular (ver fig.1), de forma tal que cada modulo cumple una función específica, independiente de los demás. De esta forma la implementación de nuevas técnicas se ve reducida a la incorporación de nuevos modulos.

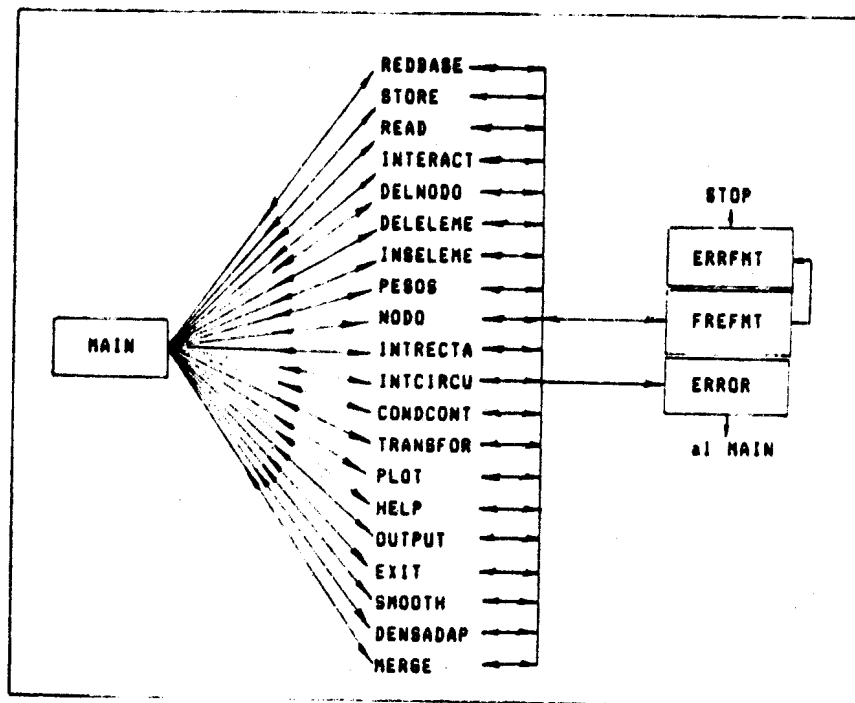


Figura 1: Estructura del programa ENREDO

Actualmente están implementados los seis elementos de la fig. 2. El tercer caso de elementos triangulares, es un caso particular del triángulo cuadrático en el cual los nodos de los lados son forzados a estar alineados con los de los vértices (Para que sea lineal).

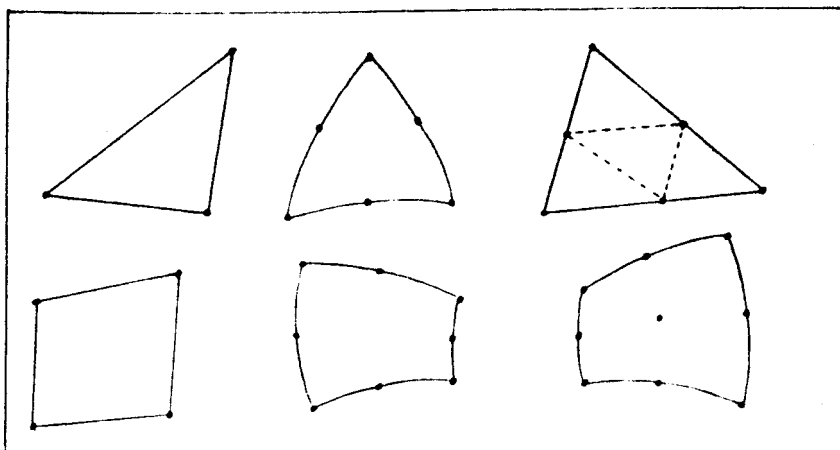


Figura 2: Elementos implementados

Dado que las redes de elementos triangulares pueden introducir errores en el calculo a causa de una orientacion preferencial, se han implementado tres tipos de orientaciones distintas (ver fig. 3).

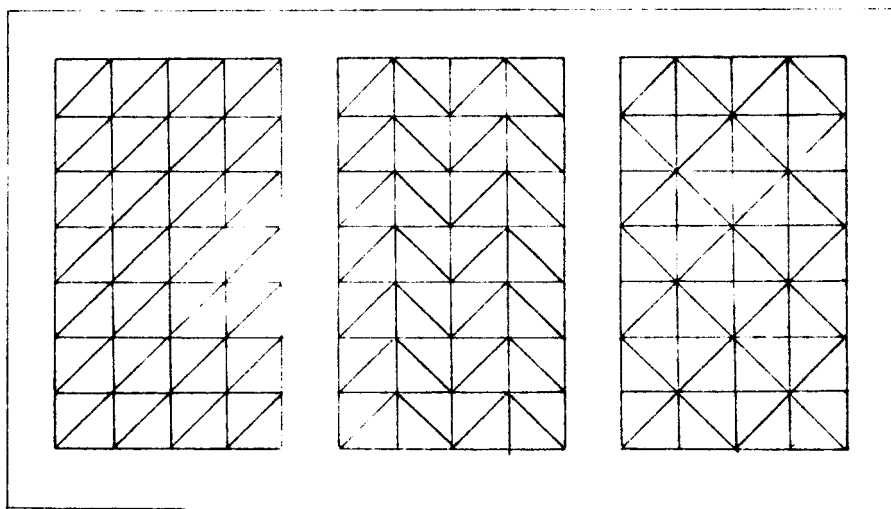


Fig. 3: Orientaciones posibles en redes de elementos triangulares

A fin de dar una idea del sistema sobre el que se trabajo, describre someramente las características del programa ENREDO.

ENREDO trabaja en general en forma interactiva, en dos modalidades distintas: en modo comando, o en modo grafico (PLOT).

Desde el modo grafico se puede obtener informacion de la red (numeracion de nodos y elementos, distribucion de los misasos, etc) y modificar la red ejecutando comandos desde este modo o digitalizando nodos.

La otra modalidad de trabajo es a travez de comandos. Estos se componen de dos partes; una palabra clave (Keyword), que identifica al modulo que se desea activar; y a continuacion los datos que requiere el modulo en cuestion:

REDB TRI6 14 10 0.7
└───┬──────────┘
Keyword Datos

Existe tambien la posibilidad de trabajar leyendo los comandos desde un archivo, e incluso alternar todas las modalidades de trabajo durante una misma generacion.

Los comandos implementados y sus funciones son los siguientes:

- REDBase : Genera una red base o primaria de forma simple.
- STORE : Habilita el grabado de los comandos en un archivo.
- READ : Desactiva el interactivo y lee de un archivo.
- INTERact : Vuelve al modo interactivo.
- DELNode : Borra nodos, y los elementos formados por estos.
- DELEleme : Borra elementos.
- INSEleme : Densifica localmente la red dividiendo elementos.
- PEsos : Genera nuevas coordenadas, en una direccion.
- Node : Adjudica nuevas coordenadas a nodos dados.
- INTRecta : Genera nuevos nodos interpolando una recta entre dos nodos dados.
- INTCircu : Genera nuevos nodos interpolando un arco de circulo entre dos nodos dados.
- Condcont : Ingresas condiciones de contorno.
- Transfor : Transforma la red teniendo en cuenta el contorno definido previamente.
- Plot : Grafica la red en la pantalla e ingresa al modo grafico.
- Help : Ingresas al servicio de ayuda.

--Output : Graba la informacion de la red en un archivo.
--Exit : Salida del programa.
--Smooth : Optimiza la forma de los elementos.
--DENsadapt : Densifica la red, adaptandola a una solucion calculada previamente.
--Merge : Une dos o mas redes.

Las mayusculas en los comandos indican el minimo de letras que se debe dar para identificarlo de los demas. Salvo excepciones obvias los comandos pueden ejecutarse en cualquier orden y tantas veces como se desee.

En general para generar una red se parte de una de forma sencilla, y luego se la va transformando como se desee con los distintos metodos disponibles. Se puede generar varias redes simultaneamente, y, si se desea unir las y formar una red mas compleja.

A modo de ejemplo de la forma de trabajo, en la figura 4, 5 y 6 se pueden ver los comandos necesarios para generar una red que modela el corte de una galeria y la red obtenida.

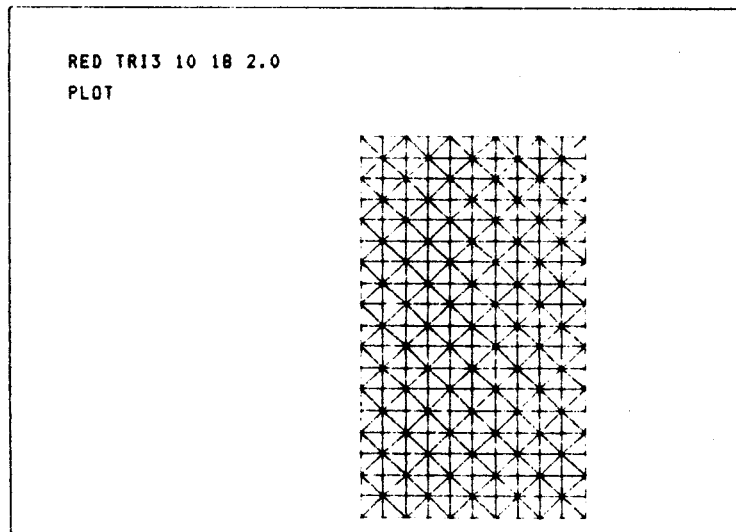
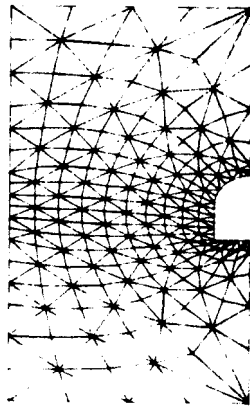


Figura 4: Generacion del corte de una galeria.

```
NOD 1 0. 0. 5 10. 0. 11 10. 9. 77 8.5 9. 143 8.5 11.  
INTR 1:5  
NOD 209 10. 12.5 203 10. 21.5 199 0. 21.5  
INTR PE 5:11  
    5:2 .5 .5  
INTR P 203:209  
    5:2 .5 .5  
INTR 11:77:11  
INTR 77:143:11  
INTR P 1:199:11  
    3*3.0 3*2.0 6*1.0 3*2.0 3*3.0  
INTR 199:203  
INTC 11.0 0.0 143:209:11  
TRAN  
PL
```



```
NO 75 7.95 8.45 141 7.95 11.  
INTR 9:75:11  
INTR 141:75:-11  
INTC 11.0 0.0 207:141:-11  
TRAN  
OUT NOLI
```

Figura 5: Generacion del corte de una galeria (Continuacion)

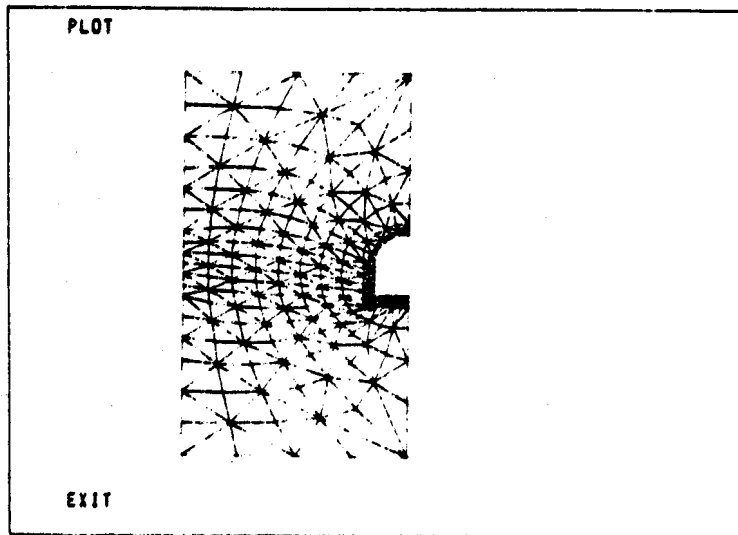


Figura 6: Generacion del corte de una galeria (Continuacion)

INSERCIÓN DE ELEMENTOS

Esta es una idea suamente sencilla y consiste en dividir los elementos, de forma de que sus vecinos queden sin alterar. En la figura 7 se muestra esta division para distintos tipos de elementos.

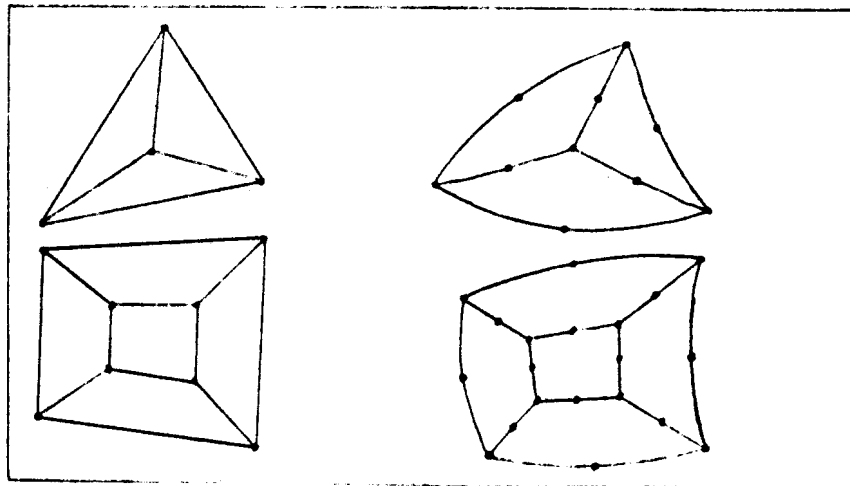


Figura 7: Division de elementos.

Al quedar los elementos vecinos sin alteracion, la implementacion de este metodo es suamente sencilla y bastan unas pocas lineas FORTRAN.

Este metodo es muy practico para densificar localmente (Ver fig. 8); pero presenta dos problemas. Uno es que sucesivas divisiones producen nodos a los que concurren muchos elementos (ver fig. 8), efecto no deseable. Y el otro es que la forma de los nuevos elementos no es la ideal. Este ultimo problema, en algunos casos, puede solucionarse realizando una posterior optimizacion de la forma de los elementos (Ver siguiente punto).

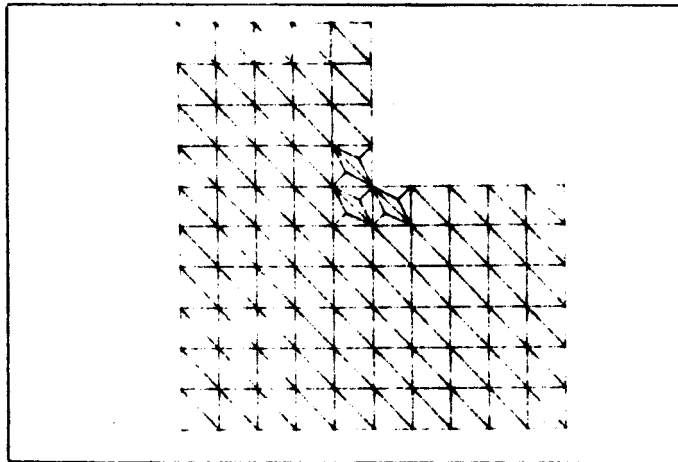


Figura 8: Densificación local en un ángulo.

OPTIMIZACION DE LA FORMA DE LOS ELEMENTOS

Ya en el año 79 existía toda una gama de métodos para optimizar la forma de los elementos en una red de Elementos Finitos. Dos de los más utilizados son: Relajación "Laplaciana" en la que cada nodo es ubicado en el centro de los nodos que lo rodean. Y (para el caso de elementos triangulares) un procedimiento iterativo, en el que cada nodo es ubicado en la posición que hace más equiláteros a los elementos a los que pertenece.

El primer metodo esta implementado en ENREDO a traves del comando TRANSFOR . Como el segundo metodo, en algunos casos, presenta ciertas ventajas frente a la transformacion Laplaciana, parecia conveniente incorporarlo al programa.

Tomemos el elemento de la figura 9 en la cual consideramos fijos los nodos "A" y "B", y deseamos re-ubicar al nodo "C". Llamemos "PE" a la posicion de "C" que haria equilatero al elemento.

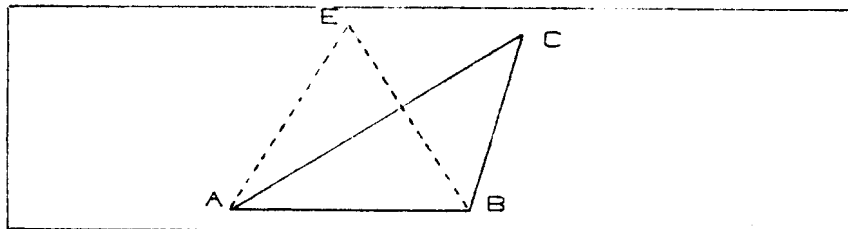


Figura 9: Optimizacion de la forma de un elemento triangular.

Tomemos como medida de la desviacion de la posicion ideal a:

$$D = | PE-PC |$$

Ahora bien, para decidir cual es la posicion de "C" mas adecuada, pedimos que la suma de estas desviaciones sobre todos los elementos a los que pertenece el nodo "C", multiplicadas por un peso conveniente, sea minima:

$$PC \text{ tal que } \sum_i | PE-PC | \cdot W_i = \text{minimo}$$

Un peso conveniente puede ser:

$$W_i = 1/ | PA-PB |$$

De esta forma la influencia de los elementos mayores se asemejara a los de los elementos menores.

Introduciendo la expresion de PE en funcion de PA y PB, y derivando respecto a PC e igualando a cero podemos obtener la expresion para el PC optimo:

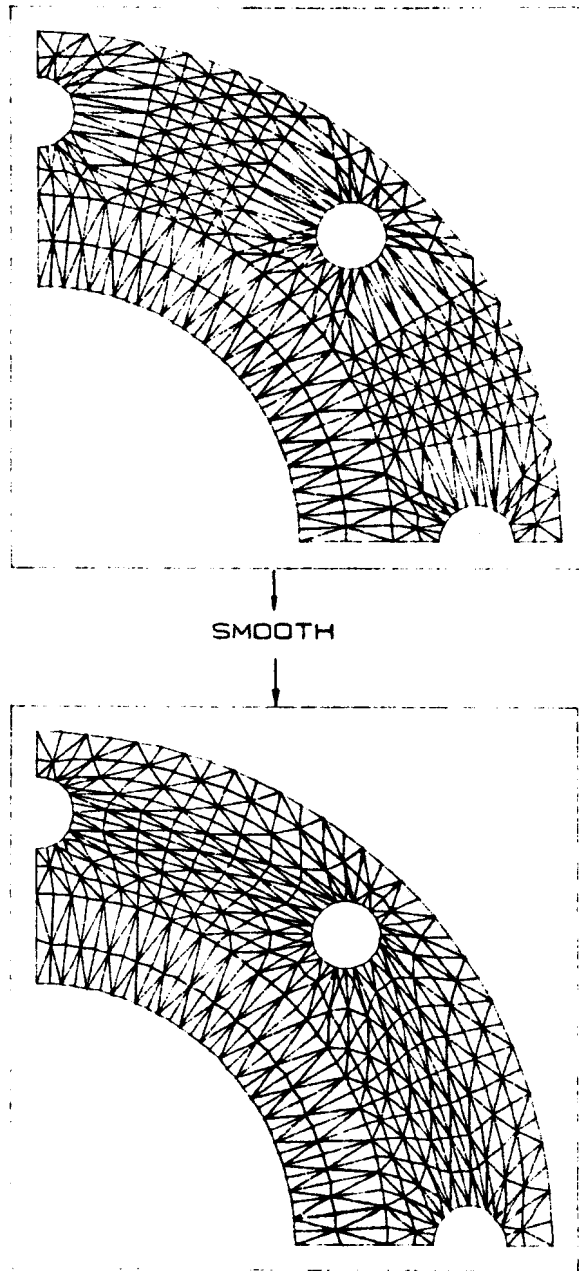


Figura 10: Corte de una brida

$$XC = \sum_i (XA+XB+1.732+(YA-YB)) / 2 + M$$

$$YC = \sum_i (YA+YB+1.732+(XB-XA)) / 2 + M$$

En la figura 10 se muestra la red para el corte de una brida en la que se puede ver el efecto de este comando.

DENSIFICACION ADAPTIVA A UNA SOLUCION

Uno de los comandos disponibles de ENREDO (Transfor) permite, dado una red base y un contorno, transformar a la red base en otra que se adapte al contorno dado, y cuyos nodos internos se distribuyan suavemente. Para ello se resuelven las siguientes ecuaciones diferenciales:

$$\partial(b \partial X / \partial u) / \partial u + \partial(b \partial X / \partial v) / \partial v = 0 \quad \text{con } X/r = X0$$

$$\partial(b \partial Y / \partial u) / \partial u + \partial(b \partial Y / \partial v) / \partial v = 0 \quad \text{con } Y/r = Y0$$

En general el coeficiente "b" es considerado constante a lo largo del recinto, y queda una simple ecuacion de Laplace para cada coordenada.

Ahora bien, si encontramos una forma practica de dar este coeficiente, podriamos densificar de distinta forma dentro del recinto. En general va a ser dificil encontrar una forma funcional que se adapte a nuestros requerimientos, y por otro lado, dejar esa tarea al usuario es una pesima politica.

Sin embargo, el efecto de este coeficiente sobre la densificacion puede ser facilmente explotado si uno dispone de una solucion al problema sobre una red generada previamente; tomando esa red como red base, y utilizando la solucion para determinar el coeficiente "b".

Para este ultimo paso hay varias posibilidades. La que ha sido utilizada en este caso, es calcular los gradientes de la solucion en cada elemento, y tomar el coeficiente "b" proporcional al modulo del gradiente en cada elemento:

$$b = |Grad U| \quad U : \text{solucion aproximada}$$

1 1

De esta forma las zonas de fuerte gradiente quedaran mas densificadas. Otra posibilidad seria estibar el error local cometido y utilizar esta informacion para determinar "b" (se podria tomar proporcional al error). En trabajos del año 82 y 83, Zienkiewicz y colaboradores sugieren un metodo sencillo para estibar este error.

El resultado de este procedimiento, es muy satisfactorio. En las figuras 10, 11 y 12 se puede ver un ejemplo, en el que se muestra la red inicial, la solucion aproximada al problema, y la red adaptada.

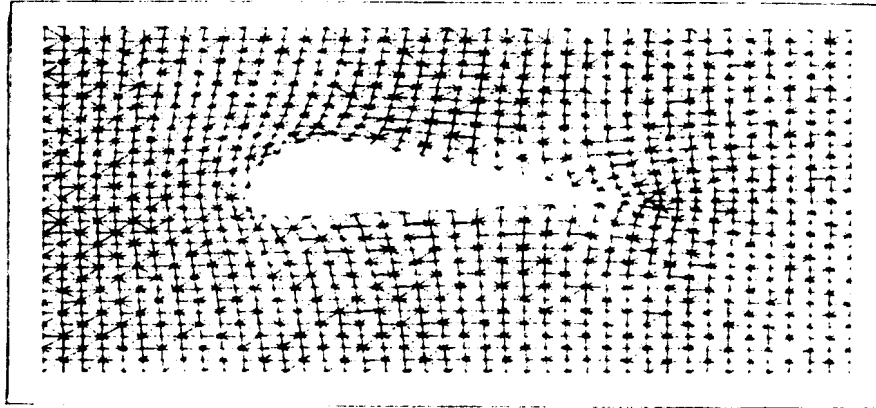


Figura 10: Red inicial. (Perfil alar)

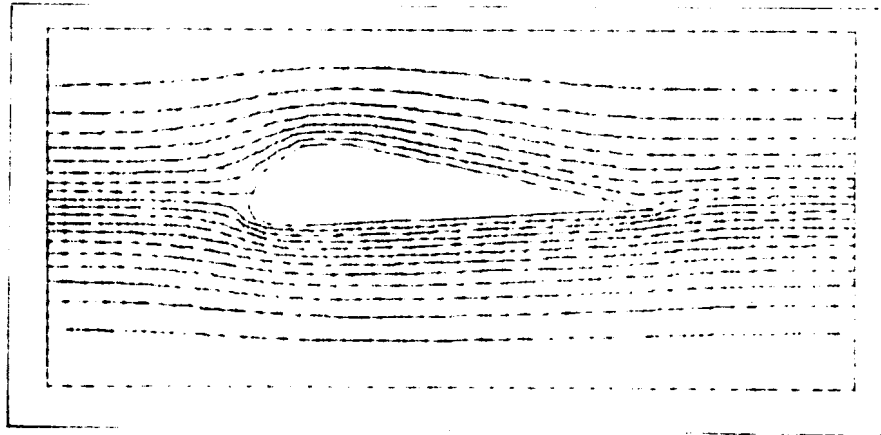


Figura 11: Solucion del problema con la red inicial.

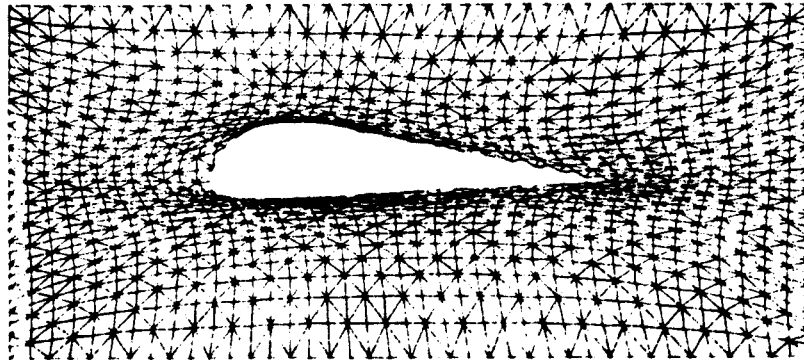


Figura 12: Red adaptada a la solución.

DIGITALIZACION DE UN NODO

Esta idea es sumamente sencilla, y de fácil implementación. Solo es necesario disponer de alguna facilidad gráfica interactiva, preferiblemente de buena resolución.

Esta técnica es especialmente útil para corregir errores en la ubicación de algunos nodos de una red ya generada, evitándose de esta forma re-hacer una generación.

En las figuras 14, 15 y 16 se muestra el proceso de corrección en un nodo de una red que modela el corte de una biela. En primer lugar se amplía la zona donde se encuentra el nodo en cuestión; luego se lo selecciona, y finalmente con el cursor se indica la posición definitiva.

Desde el punto de vista de la programación, solo hay que adjudicarle al nodo seleccionado las coordenadas del cursor:

$$X(N) = X_{\text{cursor}}$$

$$Y(N) = Y_{\text{cursor}}$$

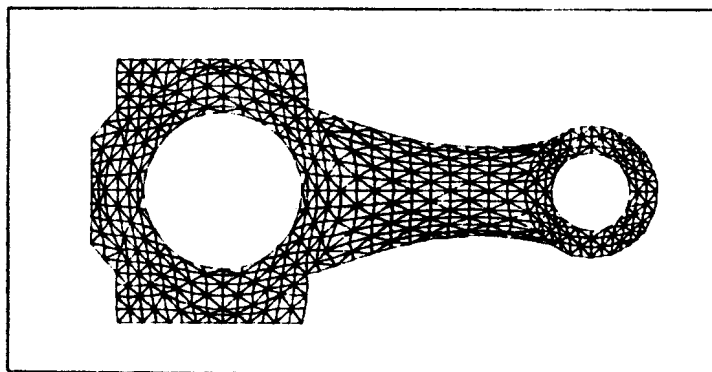


Figura 14: Corte de una biela (sin correccion)

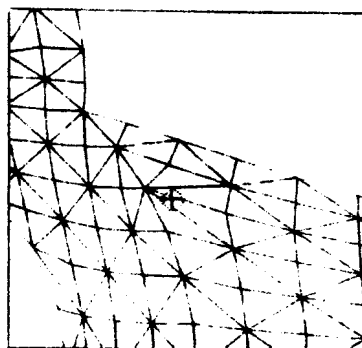


Figura 15: Aplicacion de la zona de correccion

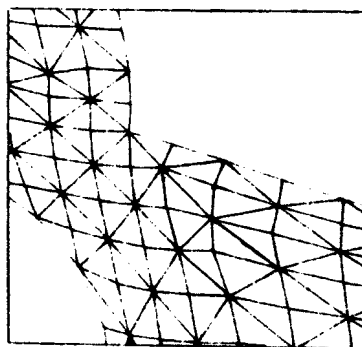


Figura 16: Aplicacion de la zona corregida

PEGADO DE REDES

Esta tecnica es usada ampliamente en varios generadores de redes, ya sea en dos o tres dimensiones. La idea consiste en generar varias redes (en general sencillas), y luego unir las para formar una sola.

Las tecnicas de pegado se podrian clasificar en dos tipos bien diferenciados:

- Pegado logico
- Pegado geometrico

En el primer caso el usuario dice que nodos de tal red se pegan con que nodos de tal otra red. En este caso, las coordenadas de la segunda red seran modificadas de forma que se verifique el pegado y mantenga su forma relativa.

En el segundo caso el programa busca automaticamente los nodos con coordenadas coincidentes entre las redes a pegar. Para ello el usuario debe dar una cota, de forma que los nodos cuyas coordenadas difieran en menos de esta cota se consideren coincidentes.

Cada metodo tiene sus pro y sus contras. El pegado logico requiere bastante informacion de parte del usuario, y ademas su implementacion es relativamente compleja; por el otro lado, reduce las posibilidades de error, ya que los nodos son forzados a pegarse explicitamente. En cambio en el pegado geometrico, nodos aparentemente coincidentes pueden no estar efectivamente unidos si la cota dada es demasiado pequena, y este error es imposible de detectar desde el programa; se le esta exigiendo entonces al usuario una especial atencion cuando se usa este metodo, lo que le resta cierta confiabilidad. A favor de el pegado geometrico hay que remarcar el poco volumen de datos que requiere (solo una cota), y su facilidad de implementacion.

En el caso de pegado geometrico en el espacio habria que sumar otro serio problema, para buscar los nodos coincidentes hay que barrer todos los nodos de todas las redes para cada nodo en particular; o sea que se deben ejecutar tres o cuatro sentencias aproximadamente $N*(N/2)$ veces, siendo N el numero de nodos total. Esto consume un tiempo de ejecucion inaceptable en un programa interactivo. Si nos limitamos a dos dimensiones, las redes solo se pueden pegar por los nodos del borde, lo que reduce drasticamente la busqueda.

En mi caso he implementado un pegado geométrico, debido a que su implementación era más sencilla. En las figuras 17 y 18 se muestra un ejemplo. Para el usuario simplemente debe generar las distintas redes por separado sin hacer ninguna mención especial y en el orden que se desee. El programa, cada vez que se ejecuta un comando REDBASE guarda en disco la información de la red generada previamente. Si en algún momento se ejecuta el comando MERGE, se recuperan todas las redes ya generadas y se las pega entre sí. Se pueden realizar varios pegados por separado si se desea.

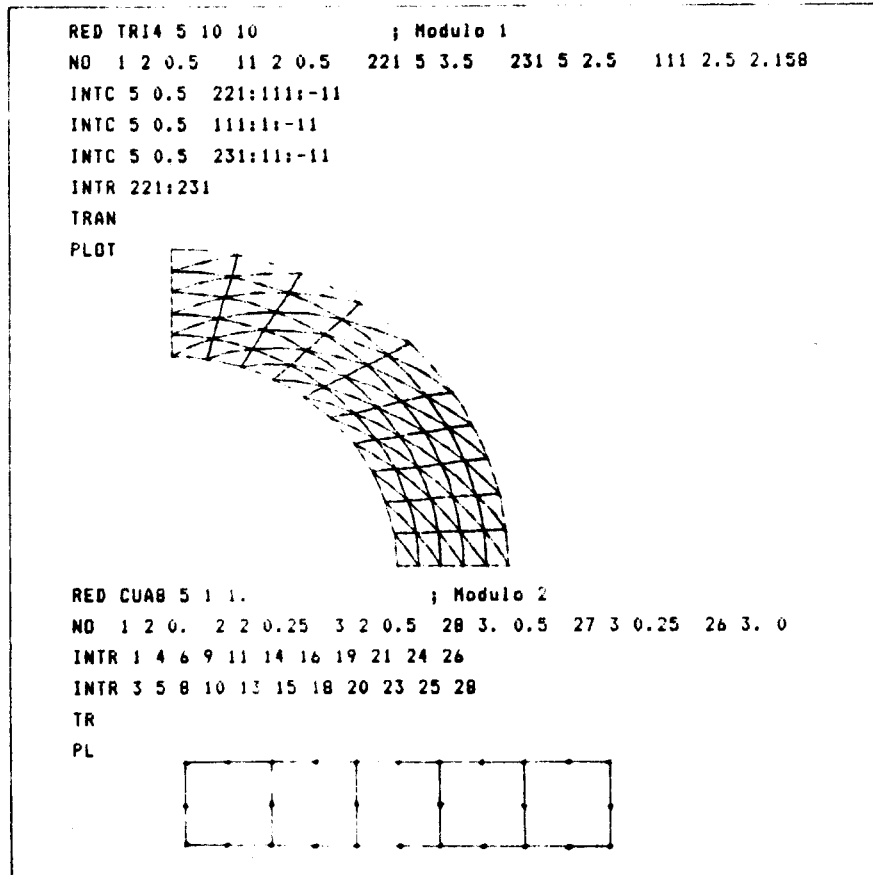
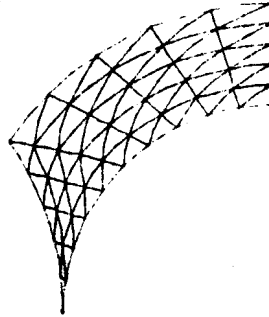


Figura 17: Pegado de redes.


```
RED TRI4 5 10 10.          ; Modulo 3  
NO 1 0 3.5 11 0 2.5 231 1 0.5 111 2.5 2.1  
INTC 0 0.5 111:1:-11  
INTC 0 0.5 231:11:-11  
INTC 5 0.5 111:231:12  
INTR 1:11  
DELE 51 61:63 71:75 81:87 91:99  
TR  
PL
```



```
MERGE 0.001  
PL
```

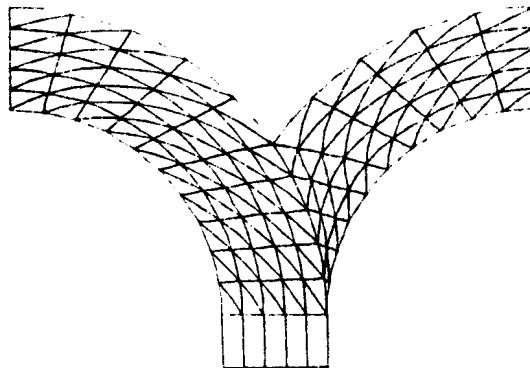


Figura 18: Pegado de redes

REFERENCIAS

- (1) Venere M., "ENREDD- Generador de redes de elementos finitos para recintos irregulares de dos dimensiones", ENIEF 86.
- (2) Sanchez Sarmiento G., "REDEF- Esquema simple para la generacion automatica de redes de elementos finitos bidimensionales.", CNEA-NT 16/80.
- (3) Villarino E., "Paquete de rutinas para el ingreso de datos con fomato libre", Publicacion interna CNEA.
- (4) Lo S. H., "A new mesh generation scheme for arbitrary planar domains", Int. J. Numerical Method in Engineering, vol 21 1985.
- (5) Zienkiewicz and Babuska, "Accuracy estimates and adaptive refinement, in finit element computation", 1986 Jhon Wiley & sons
- (6) Kleinstreuer C., "A triangular finite element mesh generator for fluid dynamic system of arbitrary geoetry", Int. J. Numerical Method in Engineering, vol 15 1980.