

SUBDIVISION RECURSIVA PARA EL TRAZADO DE CURVAS DE NIVEL

Eduardo E. Laitano
*Grupo de Tecnología Mecánica del INTEC
CONICET - Universidad Nacional del Litoral*

Pierre Beckers, Alain Schyns
*Laboratoire de Techniques Aéronautiques et
Spaciales - Université de Liège (Belgique)*

RESUMEN

En este trabajo se ha estudiado el trazado de curvas de nivel en resultados continuos y discontinuos de cálculos por el método de los elementos finitos. Se muestra que las funciones de interpolación de los desplazamientos, como por ejemplo las funciones de los elementos isoparamétricos, son apropiadas para representar resultados discontinuos como las tensiones. Se presenta un método simple eficiente y preciso utilizando las funciones de interpolación de los elementos isoparamétricos y la subdivisión recursiva.

ABSTRACT

In this paper, the contour mapping of continous or discontinous results from finite element method is studied. We show that the same interpolation functions used for displacements, i.e. the standard isoparametric functions, can be suitable to interpolate discontinous results as tensions. A simple, efficient and accurate method to contour mapping based on isoparametric functions interpolation and recursive subdivision is presented.

INTRODUCCION

Si consideramos la representación de resultados de cálculos por el método de los elementos finitos, debemos destacar que existen dos tipos de resultados, que en general son tratados en forma diferente. Uno de ellos es la representación de la función discretizada, la cual resulta generalmente continua. Podemos dar como ejemplo el caso de los desplazamientos en un modelo cinemáticamente admisible.

El otro caso es el de magnitudes que resultan discontinuas, como por ejemplo las tensiones en el mismo modelo. En este caso, para la representación gráfica de las mismas, generalmente se realiza un alisado. Existen varios métodos, pero el más utilizado por su simplicidad, es el que efectúa el promedio de las tensiones medias de los elementos que concurren a un nodo.

A partir de estos valores nodales, ya sea de tensiones o desplazamientos existen tres posibilidades diferentes: 1) el problema puede ser simplificado dividiendo los cuadriláteros en triángulos y utilizando una interpolación lineal; 2) uno utiliza las funciones de interpolación que definen los desplazamientos en el elemento o 3) se utilizan funciones de un orden superior definidas a partir del elemento y sus vecinos.

En este trabajo se obtienen buenos resultados, utilizando la interpolación del campo discretizado ya sea para la representación de desplazamientos o tensiones.

En el caso de la representación de tensiones, los valores nodales pueden ser obtenidos a partir de valores locales, como por ejemplo: por extrapolación de los valores locales en los puntos de Gauss hacia los nodos. Este método generalmente da buenos resultados en elementos finitos donde los desplazamientos se aproximan por polinomios de orden 2 o superior.

El cálculo de las líneas de nivel se efectúa por subdivisión recursiva de los elementos [1] y subdivisión en triángulos e interpolación lineal en el último nivel de subdivisión. Los errores clásicos producidos por la división en triángulos de elementos bilineales [2] son despreciables ya que el tamaño de estos sub-elementos es mucho menor que el de los elementos originales.

Otros métodos para el trazado de curvas de nivel cuando la función es definida por una interpolación de orden superior pueden ser vistos en [1], [3], [4] y [5].

CALCULO DE LAS COORDENADAS DE LOS SUB-ELEMENTOS Y EVALUACION DE LA MAGNITUD A REPRESENTAR

El método de trazado de líneas de nivel por subdivisión recursiva, tal como se presenta en [1] es simplificado, utilizando la subdivisión en dos direcciones normales entre sí y paralelas a los lados del elemento, subdividiendo los elementos en cuatro sub-elementos. De este modo los sub-elementos mantienen la relación de forma entre sus lados, evitando tener que seleccionar una dirección de subdivisión.

Otra ventaja de la subdivisión en cuatro sub-elementos es la simplicidad en el cálculo de las coordenadas de los nodos de los sub-elementos. Si en el sistema de coordenadas intrínsecas se divide en cuatro elementos iguales y cuadrados, las coordenadas de los vértices de los sub-elementos resultan triviales y las coordenadas globales se calculan a partir de estas por la misma interpolación que la utilizada para los valores a representar.

$$x = \sum_{j=1}^n W_j(\xi, \eta) X_j$$

$$y = \sum_{j=1}^n W_j(\xi, \eta) Y_j$$

$$t = \sum_{j=1}^n W_j(\xi, \eta) T_j$$

SUBDIVISION RECURSIVA

Este método tiene como objetivo obtener una buena aproximación a la isocurva exacta para una función de grado elevado con un mínimo de puntos de evaluación.

El procedimiento de subdivisión recursiva puede resumirse en los pasos siguientes:

- 1- ¿Por el elemento o sub-elemento pasa una isocurva? ¹
NO: El sub-elemento es dibujado con el color que corresponda según el valor de la función en

¹ En la versión original de este método de trazado de isocurvas, se aplica a una interpolación con las funciones de Bezier en la que es muy fácil asegurar si por el elemento no pasa una isocurva por la propiedad llamada de "la envolvente convexa". En nuestro caso este test es una aproximación, ya que en la interpolación de Hermite la función puede tomar valores que superen el máximo o el mínimo de los valores nodales. Pero teniendo en cuenta, que en las soluciones de elementos finitos la curvatura no es en general importante en el interior de un elemento, esta aproximación puede considerarse admisible.

sus nodos². Se pasa al elemento o sub-elemento siguiente, comenzando por el punto 1³.

SI: Se continua con el punto 2.

-2- ¿ Se ha llegado al nivel máximo de subdivisión requerido para el elemento ?

NO: Se pasa al punto 3.

SI: Se dibuja el sub-elemento dividiendo el cuadrilátero en dos triángulos y calculando la isocurva en cada uno por interpolación lineal. Se pasa al elemento o sub-elemento siguiente comenzando por el punto 1.

-3- El elemento o sub-elemento es dividido en cuatro sub-elementos. A continuación cada sub-elemento es tratado como un nuevo elemento, comenzando por el punto 1.

El tratamiento de los elementos de un nivel, y el ascenso al nivel superior, para tratar el elemento siguiente del mismo se produce naturalmente, si uno implementa el método en un lenguaje recursivo. Como ejemplo en el apéndice 1 se da una versión simplificada de la rutina escrita en lenguaje C. El resultado de la subdivisión recursiva puede verse en la figura 1.

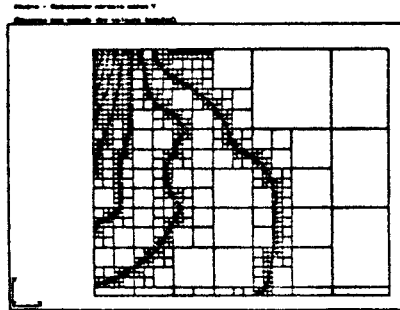


Figura 1 - Sub-elementos generados

TRATAMIENTO DE ELEMENTOS TRIANGULARES

Para el tratado de isocurvas en mallas triangulares, se ha utilizado las mismas rutinas que para las mallas cuadrangulares, repitiendo el último modo. Se debe tener en cuenta, que si el elemento es lineal no se obtienen beneficios con respecto a la interpolación lineal en los triángulos, pero si se obtienen cuando se trata de triángulos orden 2 o superior.

- 2 - En cuanto al tipo de representación de las curvas de nivel, se ha supuesto que el espacio entre 2 curvas es llenado con un color asociado al valor de la función en esta región. Si se dibujan simplemente las líneas de nivel, uno dibuja la frontera o pasa al elemento siguiente si las fronteras de elementos no son dibujadas.
- 3 - En los puntos 1 y 2 se pasa al elemento o sub-elemento siguiente significa:
 - Si el elemento no ha sido subdividido, uno pasa al próximo elemento finito.
 - Si el elemento ha sido dividido, se pasa al próximo sub-elemento del mismo nivel, si existe uno o se asciende un nivel en el árbol de subdivisión, tratando el elemento siguiente de ese nivel.

CONCLUSIONES

Las funciones de interpolación isoparamétricas, definidas para cada elemento coinciden con la función resultado en el caso de desplazamientos y resultan adecuadas para la interpolación de magnitudes discontinuas, como las tensiones.

La combinación de la subdivisión recursiva de los elementos con la interpolación lineal en el último nivel de subdivisión, provee un algoritmo simple y eficaz para el trazado de curvas de nivel.

Es sumamente fácil de controlar la calidad de la representación obtenida modificando el número máximo de niveles de recursividad. De este modo se reduce el error cometido en la aproximación lineal ya que el tamaño del sub-elemento en el que se efectúa la misma, depende en forma exponencial inversa del número de niveles. Si llamamos L_n al lado de sub-elemento en el que se realiza la aproximación lineal y L al lado del elemento finito, la relación entre ambos para un número de niveles n es:

$$L_{nn} = \frac{L}{2^n}$$

El método presentado es igualmente apto para la representación de magnitudes continuas que para la representación de magnitudes discontinuas.

Este método presenta el inconveniente que cuando el número de elementos es alto, si se demanda un número de niveles de subdivisión elevado, la cantidad de polígonos generado hace que el dibujo sobre una terminal gráfica sea lento; sobre todo en las terminales de velocidad de transmisión más baja.

Para tratar de solucionar este inconveniente ensayamos adicionar los polígonos en un buffer, antes de mandarlos hacia la terminal, en menor número. Pero en la forma que fue implementado, no se obtuvieron ventajas considerables.

El problema de las mallas refinadas, se hace más notable cuando se trata de modelos en los que la relación entre el elemento más grande y el más pequeño es grande. Una forma fácil de solucionar este inconveniente, es utilizando un nivel de recursividad que dependa de la dimensión de cada elemento, evitándose de este modo que los elementos más pequeños sean subdivididos excesivamente.

Como este algoritmo fue inicialmente programado en Fortran, fue necesario idear una serie de artificios para construir y localizar los sub-elementos; en el Apéndice II se presentan algunos detalles referentes al programa Fortran.

ALGUNOS EJEMPLOS

En la figura 2 se muestra la representación de los isovalores de la tensión de Von Mises, sobre un modelo plano de un pórtico sometido a flexión. La malla tiene 8 elementos en el sentido de la altura y 16 en el sentido del largo. La tensión en los nodos ha sido calculada haciendo el promedio entre los elementos que comparten el mismo nodo.

En la figura 3 se observan los mismos resultados para la misma malla pero calculados por extrapolación de valores locales en los puntos de Gauss.

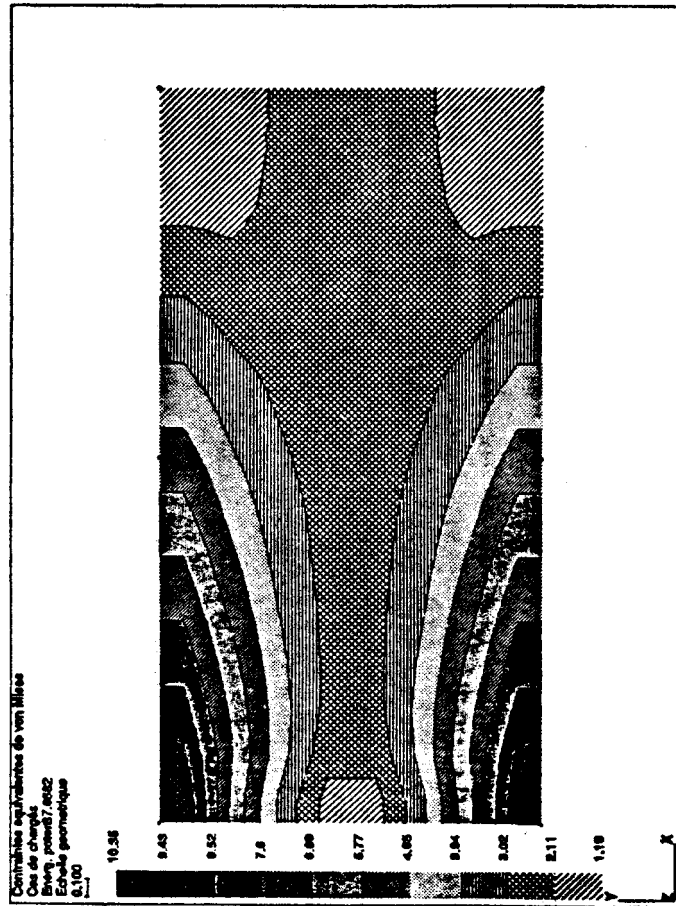


Figura 2 - Prescindo de elementos vecinos

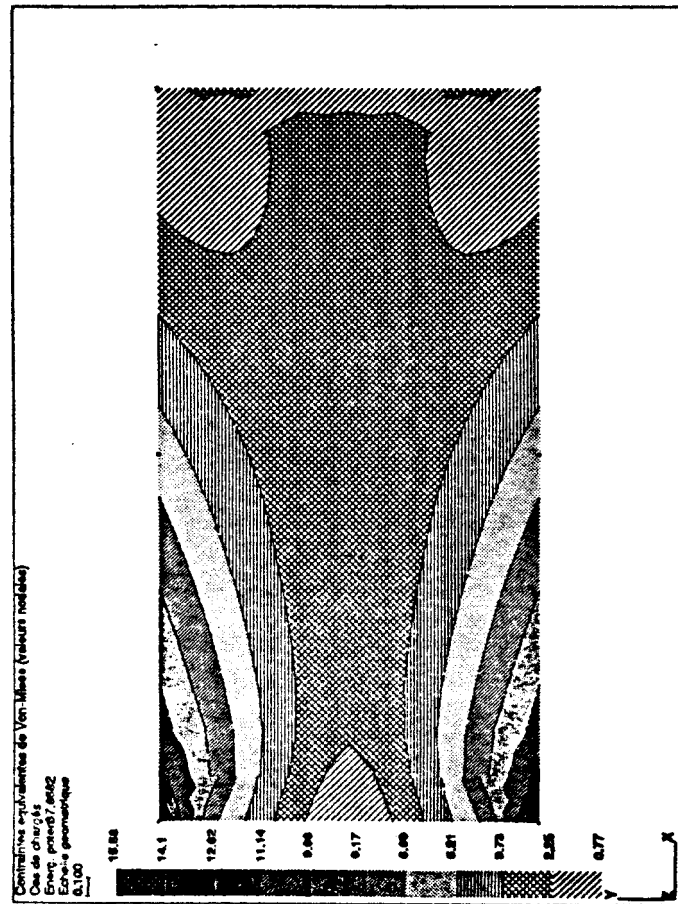


Figura 3 - Valores extrapolados de puntos de Gauss

Apéndice I
Fuente de la rutina en lenguaje C.

```
extern struct Val_MIN_MAX
{
    int nura_iso; /* número de líneas de isovalor */
    float vmin; /* valor mínimo */
    float vmax; /* valor máximo */
} valmax;

extern struct Nivel
{
    int actual; /* contador del número de niveles */
    int maximo; /* nivel de recursividad máximo */
} niv;

/* Dibujo de isovalores */
iso_rec( pnod, xyz, valnod)
int pnod[]; /* punteros a los nodos y valores */
float xyz[3][]; /* matriz de coordenadas de los nodos */
float valnod[]; /* vector de valores nodales */
{
    int pnod1[13];
    float xyz1[3][12];
    float vnod1[12];

    /* La función isover() verifica si por el elemento
    * pasa alguna isocurva. Retorna TRUE o FALSE
    * Utiliza la estructura VAL_MIN_MAX.
    */
    if(! isover( pnod, valnod))
    {
        /* No hay isocurva; dibuja el contorno y
        * lo llena con un color único
        */
        draw_sel( pnod, xyz, valnod[pnod[0]]);
    }
    else if(niv.actual == niv.máximo)
    {
        /* se alcanza el nivel máximo de subdivisión se divide
        * en triángulos y se utiliza la interpolación lineal
        */
        split_hal( pnod, xyz, valnod);
    }
    else
    {
        /* se generan 4 sub-elementos y se llama la función iso_rec en
        * cada uno de ellos. gen_sel(): genera cada sub-elemento, segun
        * una variable que indica de que hijo se trata. Carga en pnod1
        * el nuevo vector de punteros, en xyz1 las nuevas coordenadas
        * de nodo y valnod1 los valores en esos puntos.
        */
        /* Generación sub-elemento 1 */
        gen_sel(1, pnod, xyz, valnod, pnod1, xyz1, valnod1);
        iso_rec(pnod1, xyz1, valnod1);
        /* Generación sub-elemento 2 */
        gen_sel(2, pnod, xyz, valnod, pnod1, xyz1, valnod1);
        iso_rec(pnod1, xyz1, valnod1);
    }
}
```

```
/* Generación sub-elemento 3 */
  gen_sel(3, pnod, xyz, valnod, pnod1, xyz1, valnod1);
  iso_rec(pnod1, xyz1, valnod1);
/* Generación sub-elemento 4 */
  gen_sel(4, pnod, xyz, valnod, pnod1, xyz1, valnod1);
  iso_rec(pnod1, xyz1, valnod1);
}
}
```

Apéndice II

Algunos detalles para la programación en un lenguaje no recursivo.

Para la programación de un algoritmo basado en una estructura de árbol en un lenguaje no recursivo, se necesita en cada nivel memorizar el hijo que se trata en ese momento, de modo que al ascender un nivel se sepa cual es el próximo hijo a tratar. En nuestro caso el árbol generado es cuaternario, por lo cual en cada nivel es necesario memorizar cual de los cuatro hijos está siendo tratado.

Para esto se identificó cada uno de los hijos por un número entero según la siguiente convención:

- 1 para el hijo inferior izquierdo.
- 2 para el hijo inferior derecho.
- 3 para el hijo superior izquierdo.
- 4 para el hijo superior derecho.

Luego utilizando un vector entero de tantas componentes como el número máximo de niveles de recursividad, se almacena en cada componente el hijo que está siendo tratado en dicho nivel.

Este mismo vector se utiliza para calcular las coordenadas de los nodos del sub-elemento tratado, en el sistema intrínseco del elemento de base. De este modo para el cálculo de las coordenadas globales de los nodos y de la magnitud a representar para todos los sub-elementos se utiliza las mismas funciones de interpolación.

REFERENCIAS

- 1 - M. Sabin - "A Survey of Contouring Methods" - Computer Graphics Forum 5 (1986) 325-340.
- 2 - E. Laitano - "Visualisation Précise de Résultats de Calcul par Eléments Finis" - Mémoire présenté pour l'obtention du grade de Maîtrise en Sciences Appliquées Année académique 1989-1990 - Laboratoire de Techniques Aéronautiques et Spatiales - Université de Liège.
- 3 - K. J. Berry - "Parametric Finite Element Contour Mapping" -Computer & Structures Vol. 35 N° 3 pp. 249-257, 1990.
- 4 - R. R. Dickinson, R. H. Bartels and A. H. Vermeulen - "Interactive Contouring: The Interactive Editing and Contouring of Empirical Fields" - IEEE Computer Graphics & Applications - pp. 34-43, 1989.
- 5 - Carl S. Petersen - "Adaptive Contouring of three-dimensional surfaces" - Computer Aided Geometric Design 1 pp. 61-74, 1984.

