

MYDAS. UN PREPROCESADOR PARA ELEMENTOS DE CONTORNO

J. Domínguez*, M. Doblaré*, F. París**, L. Gracia*, J. Cañas**

* Departamento de Ingeniería Mecánica. Centro Politécnico Superior. Universidad de Zaragoza.

** Departamento de Mecánica de los Medios Continuos y Teoría de Estructuras. E.T.S.I.I.S. Universidad de Sevilla.

Resumen

Se presenta un preprocesador bi y tridimensional para elementos de contorno, con generador automático de mallas, haciendo comentario de sus características generales, comunicación con el usuario, descripción de la base de datos y realización de algunos ejemplos.

Introducción

El método de los elementos de contorno (MEC) se ha revelado como un método válido y eficaz como alternativa al método de los elementos finitos (MEF), que ha sido la técnica computacional más profusamente utilizada en ingeniería mecánica en los últimos años. El MEF es ineficiente en muchas aplicaciones de ingeniería y lo que es más importante, de uso tedioso y complicado en la mayoría de los casos.

Los métodos de contorno permiten la reducción de la dimensionalidad del dominio a ser discretizado y la posibilidad de modelar más fácilmente cuerpos con relaciones grandes de volumen-superficie. Aunque la discretización con el MEC es más simple que con el MEF, no por ello deja de ser tediosa y requiere chequeos aburridos en el caso de modelización de cuerpos tridimensionales.

Características generales del programa

El programa MYDAS es un conjunto de herramientas informáticas que permiten hacer de forma

Abstract

A bi and tridimensional preprocessor with automatic mesh generation for the boundary element method (BEM) is presented. Its general characteristics, user interface and data base description are explained. Some examples are also presented.

más fácil, rápida y controlable la definición de un problema bi o tridimensional. El programa contiene además un generador de mallas en dos y tres dimensiones con objeto de facilitar la creación y definición de los dominios continuos que vayan a ser analizados.

MYDAS es eminentemente interactivo con el usuario, si bien para su uso no se requieren conocimientos especiales de programación. La única complejidad del programa reside en la gran variedad de opciones y posibilidades, todas ellas recogidas en el correspondiente manual de usuario.

El programa está desarrollado en lenguajes Fortran y C, si bien ahora se está en el proceso de total desarrollo en C. La parte gráfica está realizada en el estándar gráfico GKS y entorno WINDOWS. Se ha acudido a utilizar siempre estándares tanto gráficos como de lenguajes, con objeto de asegurar una total portabilidad del paquete. Consta de unas 130.000 líneas de programa y 700 subrutinas y funciones. Actualmente se halla implementado en ordenadores VAX 780 y HP 9000 serie 300.

Interfase con el usuario

La comunicación con el usuario se establece básicamente a través de órdenes de entrada, cuya ejecución configura el entorno de trabajo y la definición del problema a estudiar; y de órdenes de salida, que proporcionan información en forma de listas de datos o de gráficas que aparecen por pantalla.

Las órdenes de MYDAS están formadas por varios campos separados por un primer separador (por defecto ";"). El primero de estos campos corresponde siempre al comando a realizar que define la orden. Si dentro de estos campos hay a su vez subcampos, se indican mediante el separador secundario (por defecto ":").

Las órdenes de preproceso son de siete tipos:

- *Ordenes generales.* Son órdenes iniciales para definir el entorno del programa: instalación del mismo, definición del tipo de problema, unidades de medida, reset, shell, etc.

Ej.: `mydas3d>TITLE, problema de contacto` define el título general del problema.

- *Ordenes que afectan a una sola entidad.* Constituyen operaciones de entrada y rectificación de datos que pueden realizarse sobre todas las entidades, como definir, borrar, listar o dibujar. Por entidad se entiende los distintos conceptos básicos que se pueden manejar en el programa por parte del usuario, a saber: sistemas de coordenadas, nudos, elementos, líneas clave, puntos clave, puntos internos, líneas de interés, subregiones, interfases, materiales, áreas de interés, ventanas gráficas, distribuciones de volumen, elementos de convección-radiación y elementos muelle.

Ej.: `mydas2d>df,k,8,40,15` define el punto clave 8 cuyas coordenadas x e y son 40 y 15 respectivamente.

- *Ordenes de operaciones geométricas entre dos entidades.* Son operaciones que se aplican a dos entidades para producir otras como intersectar dos líneas claves para obtener cuatro, etc... y permiten definir una serie de elementos de una entidad a partir de dos de ellos previamente definidos.

Ej.: `mydas2d>it,1,1,2,3` intersecta las líneas 2 y 3

- *Operaciones de mallado.* Permiten mallar una línea o área clave con nudos y elementos o solamente rellenar con nudos.

Ej.: `mydas3d>ms,a,all` realiza el mallado de todas

las áreas claves definidas.

- *Operaciones de activación, cambio de defecto y modificación de cadenas básicas.* Son simples y no tienen un formato general.

Ej.: `mydas3d>activ,df,c` activa la orden de definición de coordenadas.

- *Operaciones con campos.* Sirven para definir un nuevo campo, dibujar su distribución, modificar su nombre o bien asignarle un determinado valor.

Ej.: `mydas2d>efi,n:ten` borra el campo de usuario tensión de la entidad nodo

- *Operaciones sobre ficheros.* Permiten realizar operaciones sobre los ficheros inicialmente definidos en el programa o que intervengan en el proceso.

Ej.: `mydas3d>copy,command.mid,juan.dat` copia el contenido de un fichero a otro

- *Operaciones adicionales de dibujo.* Permiten redefinir límites de ventana, realizar zooms y shrinks de pantalla.

Ej.: `mydas3d>zoom,0,.6,.7,1` realiza un zoom de una parte seleccionada del dibujo

- *Ordenes de distribuciones en el contorno.* Permite definir, borrar, listar y dibujar las condiciones de contorno

Ej.: `mydas3d>lbc,q,sel` lista las condiciones de contorno en flujos de los elementos seleccionados

Base de datos

La filosofía de la base de datos ha sido generar una estructura de datos orientada a objeto. Para cada entidad se define un tipo de registro. Dicho registro incluye todas las características de la entidad y es posible trabajar con la entidad en conjunto como tal. Un trabajo importante se ha dedicado a la definición de los campos que constituyen cada entidad y las relaciones entre campos de distintas entidades. Los registros, de fácil definición a partir del concepto de estructura en lenguaje C, y de más compleja simulación en lenguaje Fortran, se concatenan formando listas configurando el conjunto una estructura dinámica de datos. Utilizando esta posibilidad del lenguaje C es fácil acceder, definir, borrar, las características almacenadas para un elemento integrante de una determinada entidad. Se obtiene de esta forma una cómoda base de datos con la que acceder a cualquier campo o registro definido. Para cada entidad se rellenan campos con los datos introducidos directamente en la orden de definición y otros generados a partir de estos datos previos.

Así por ejemplo, en la entidad nudos del contorno, que son los nudos que definen la geometría y/o las condiciones de contorno, los campos que se almacenan para cada nodo en dos dimensiones, y que constituyen un registro (figura 1), son:

Introducidas directamente de la orden de definición:

Número de usuario del nudo (N).

Coordenadas globales del nudo (X, Y).

Otras variables:

Número de elementos a los que pertenece (NEL).

Puntero al sistema nodal asociado a la salida de las variables adscritas al nudo (*CS).

Código y valor del potencial en un punto (CDP, POT).

Código y valores de los desplazamientos según las direcciones de los ejes coordenados respecto al sistema nodal para cada hipótesis (CDD, D1, D2).

NODO	N	X	Y	NEL	*CS	CDP	POT	CDD	D1	D2	*CS	CDP
------	---	---	---	-----	-----	-----	-----	-----	----	----	-----	-----

Fig. 1. Detalle de un registro de la entidad nodo

Para la entidad elementos del contorno los campos que se almacenan para cada registro son:

Introducidas directamente de la orden de definición:

Número de usuario del elemento.

Tipo de elemento.

Punteros a los posibles nudos que definen la geometría del elemento.

Código de discontinuidad del elemento.

Otras variables:

Puntero a la interfase a la que pertenece.

Puntero a la línea a la que pertenece.

Puntero al sistema de coordenadas asociado al elemento.

Código y valores de los flujos en los nodos.

Código y valores de las componentes de tensión en cada uno de los nudos del elemento respecto al sistema elemental para cada hipótesis.

Se observa la interrelación de la entidad elemento con las entidades nudos, interfase, líneas, sistema de coordenadas y condiciones de contorno.

Tratamiento gráfico

La comunicación del usuario con el programa se

realiza en la pantalla de la terminal. Si ésta es gráfica, además de los mensajes de información, existen siete ventanas gráficas definidas inicialmente, número que posteriormente puede ser ampliado, con la definición de ventanas sucesivas.

De las ventanas señaladas y según se observará en los ejemplos, la inferior izquierda es la de introducción de órdenes y aparición de los mensajes de información requeridos sobre la base de datos y el desarrollo de ejecución del programa. La superior izquierda es donde van apareciendo, dibujadas, las distintas entidades definidas en el problema y esto si la llave de dibujo está activada. En la parte derecha de la pantalla aparecen otra serie de ventanas auxiliares que permiten conocer el estado de los defectos para las entidades, órdenes

activadas, situación de las banderas gráficas y aparición de determinados contadores.

Generación automática de mallas

Bases de la generación

En el caso bidimensional, la realización del mallado se reduce a una división de las líneas que definen el contorno y que se tratará más adelante.

En tres dimensiones la superficie se divide en macroelementos S_i de tal forma que la unión de todos ellos representen el dominio total S.

$$S = \cup S_i$$

$$S_i \cap S_j = L_{ij}$$

L_{ij} es la interfase entre los elementos i y j. Aunque no es estrictamente necesario se ha decidido subdividir la línea L_{ij} de forma compatible para ambos elementos i y j. La elección de los macroelementos corre de cuenta del usuario y está dictada generalmente por la forma del cuerpo y el catálogo de elementos del programa.

Hay dos posibilidades: trabajar con cada elemento

y subdividir las líneas o bien dividir las líneas y utilizarlas como base para realizar el mallado, una vez establecida la compatibilidad de los puntos generados en la división de los elementos. La primera opción repite trabajo y la segunda necesita más almacenamiento. Se ha utilizado la segunda opción en aras de una mayor rapidez del programa.

Tipología de las líneas

Según se indica en la figura 2 se pueden utilizar cuatro tipos diferentes de líneas

- Línea recta (A,B)
- Línea parabólica (A,B,C)
- Arco circular (A,B,O,n)
- Circunferencia (A,O,n)

La dirección de A a B está relacionada con v por

$$v = \frac{OA \wedge OB}{|OA \wedge OB|}$$

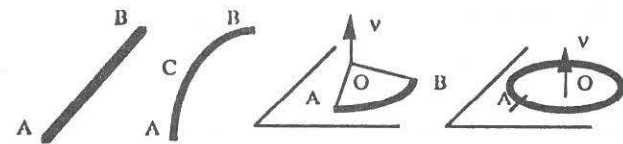


Fig. 2. Tipología de líneas

Subdivisión de las líneas

En adición a los datos previos es necesario especificar el número de subdivisiones en cada línea así como la clase de variación de la subdivisión (densificación) dentro de ella. Siendo ζ la coordenada natural requerida, con sólo el parámetro R, hay una gran familia de posibilidades de densificación.

$$\zeta(J,R) = \bar{\zeta} - \frac{R}{2} (\bar{\zeta}^2 - 1)$$

donde

$$\bar{\zeta} = \frac{2(J-1)}{N} - 1, \quad J = 1, 2, \dots, N+1$$

siendo N el número de subdivisiones de la línea.

Después de la subdivisión una transformación inversa produce las coordenadas cartesianas finales

$$x = \frac{1-\zeta}{2} x_A + \frac{1+\zeta}{2} x_B, \quad \text{para líneas rectas}$$

$$x = \frac{1}{2} \zeta(1-\zeta) x_A + (1-\zeta^2) x_C + \frac{1}{2} \zeta(1+\zeta) x_B,$$

para líneas parabólicas.

Los arcos circulares y las circunferencias se interpolan en coordenadas cilíndricas pasando después a cartesianas.

Tipología de los macroelementos

El catálogo de macroelementos es el incluido en la figura 3:

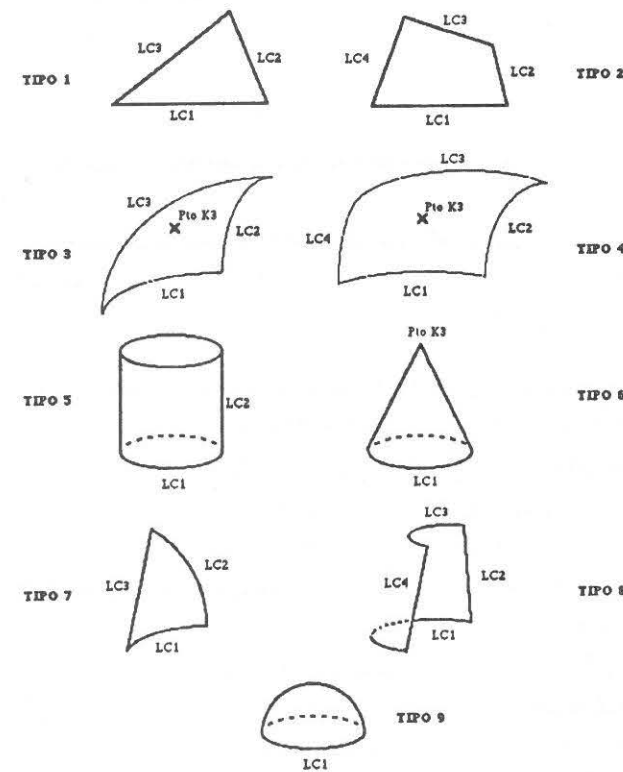


Fig. 3. Catálogo de macroelementos

- Tipo 1. Triangular plano.
- Tipo 2. Cuadrilátero plano.
- Tipo 3. Lagrangiano triangular.
- Tipo 4. Lagrangiano cuadrilátero.
- Tipo 5. Cilindro.
- Tipo 6. Cono.
- Tipo 7. Transfinito triangular.
- Tipo 8. Transfinito cuadrilátero.
- Tipo 9. Semiesfera.

Los tipos 1 y 2 están definidos por líneas rectas. Los tipos 3 y 4 por líneas parabólicas y un punto auxiliar. El tipo 5 por una circunferencia y una línea recta. El tipo 6 por una circunferencia y un punto auxiliar. El tipo 9 por una circunferencia. Los tipos 7 y 8 están constituidos por combinaciones de líneas rectas, parabólicas y arcos de circunferencia, resultando muy versátiles. Algunos de los macroelementos anteriores (todos excepto tipos 3 y 4) no son más que casos particulares de este tipo, pero que se han conservado de forma independiente por su mayor simplicidad, teniendo en cuenta que son los más comúnmente utilizados. (Figura 4).

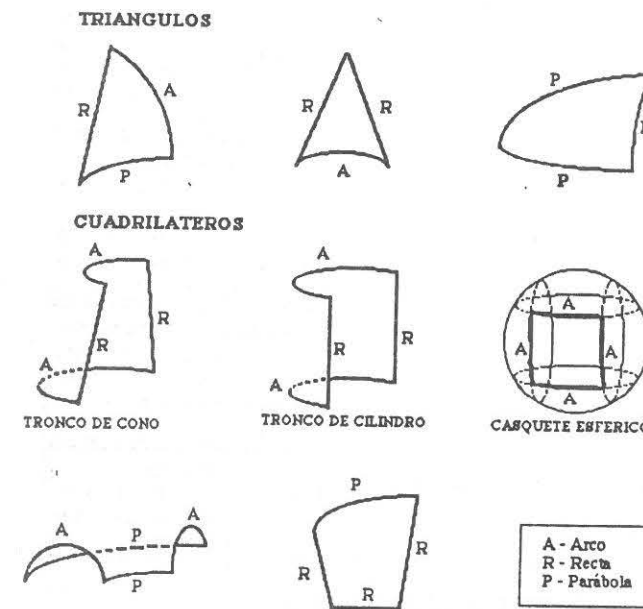


Fig. 4. Ejemplos de macroelementos transfinitos

Interpolación transfinita

La base de la generación es el concepto de elementos isoparamétricos, que después de subdivididos de acuerdo a determinadas reglas, se ajustan a la geometría actual a través del uso de la interpolación transfinita. Un método familiar utiliza la combinación lineal

$$Bf = I_{\zeta} f + I_{\eta} f - I_{\zeta} I_{\eta} f$$

donde

$$I_{\zeta} f(\zeta, \eta) = \sum_{i=0}^n f(\zeta_i, \eta) p_i(\zeta)$$

$$I_{\eta} f(\zeta, \eta) = \sum_{i=0}^n f(\zeta, \eta_i) p_i(\eta)$$

$$I_{\zeta} I_{\eta} f = \sum_{0 \leq i, j \leq n} f(\zeta_i, \eta_j) p_i(\zeta) p_j(\eta)$$

I_{η} y I_{ζ} son las interpoladas transfinitas a f , $I_{\eta} I_{\zeta}$ es un producto tensorial de interpolación usual y p representa los polinomios de interpolación.

Ejemplos de generación automática de mallas

La característica principal del generador es la de permitir dividir macroelementos con distinto número de divisiones en cada línea clave. Para ilustrar esto se añaden dos ejemplos, el primero de ellos el mallado de dos macroelementos con distinto número de divisiones en cada lado (figura 5). El segundo es un mallado de un sólido compuesto por triángulos y cuadriláteros (figura 6).

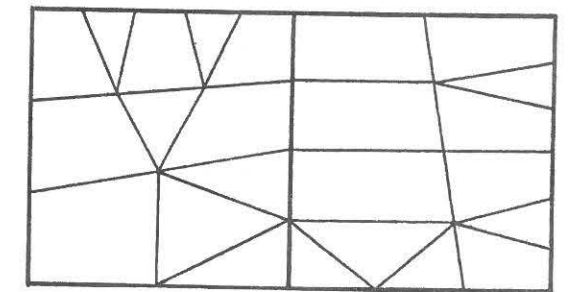


Fig. 5.

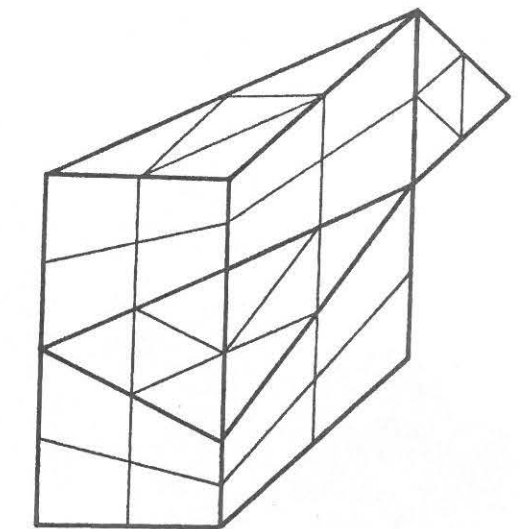


Fig. 6.

Ejemplos del programa MYDAS

1. Bidimensional. Definición de la geometría de un gancho en la cual se ha realizado el mallado con elementos cuadráticos (figura 7).
2. Tridimensional. Definición de la geometría de medio

cigüeñal en la cual se ha realizado un mallado (figura 8). Se han necesitado para su definición 144 puntos clave, 266 líneas clave y 132 áreas clave. El generador de mallas tridimensional ha sido provisto de un faceteador para intentar dar más realismo a las figuras generadas.

Fig. 7

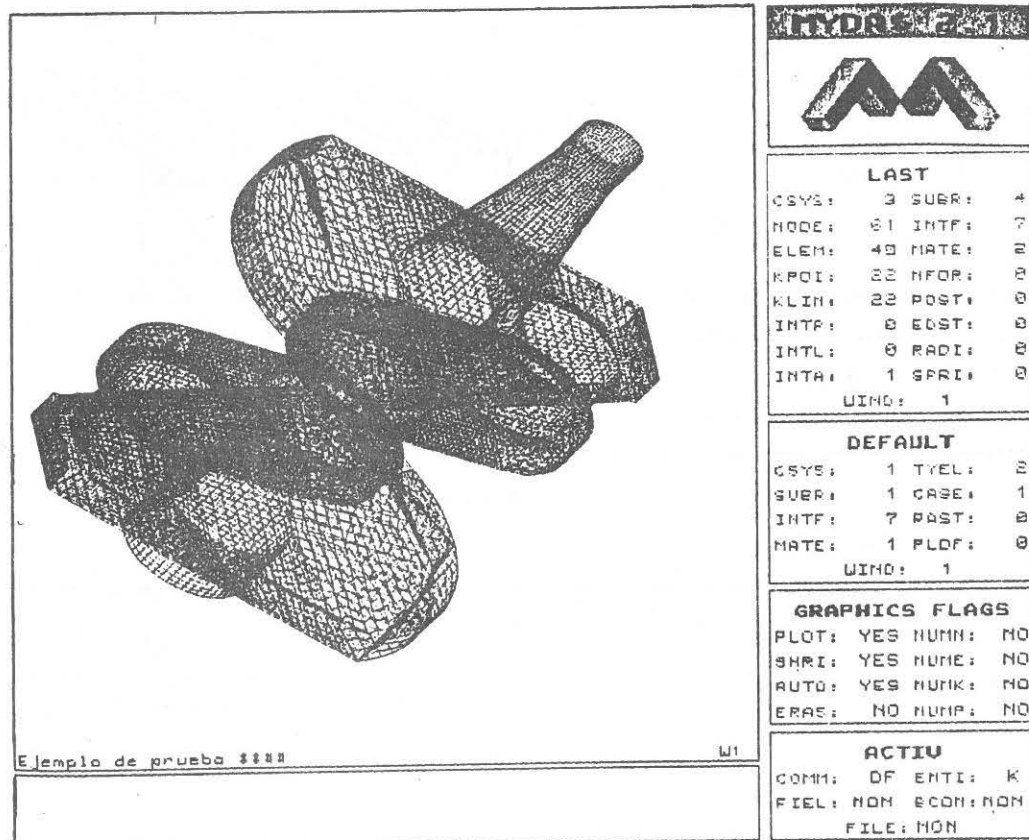
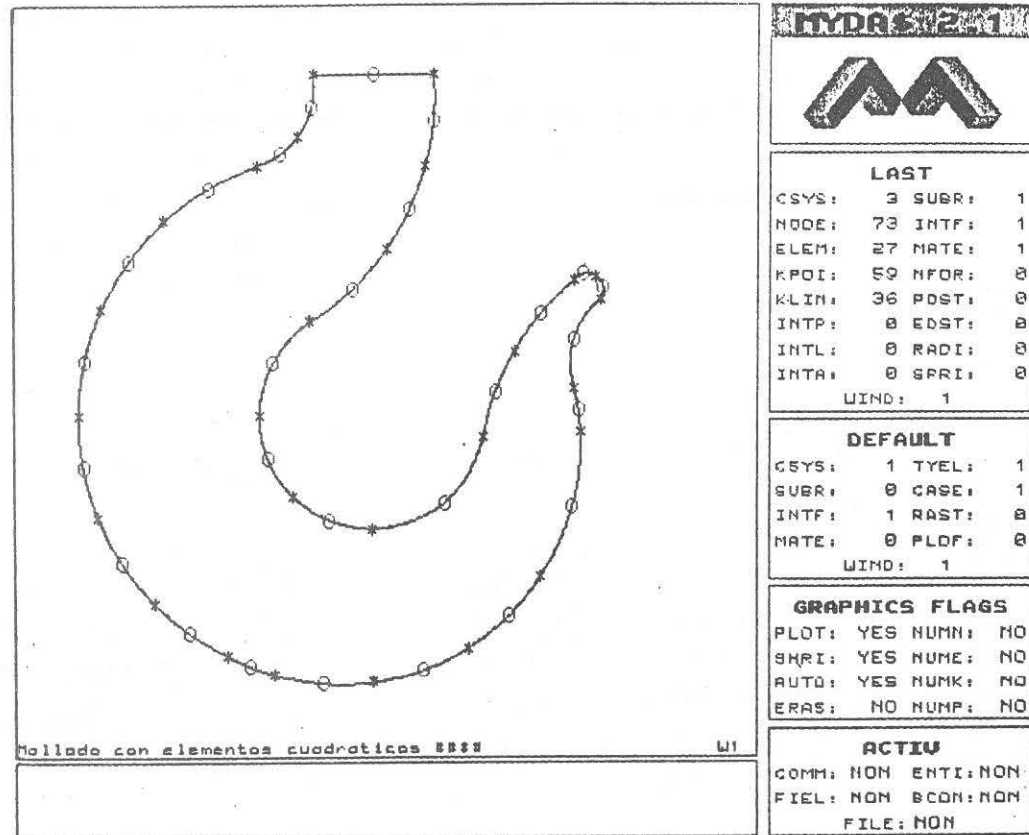


Fig. 8

Conclusiones y desarrollo futuro

Se ha presentado un paquete de preprocesado para análisis mecánico en dos y tres dimensiones con el Método de los Elementos de Contorno caracterizado por su portabilidad, flexibilidad y existencia de un potente generador automático de mallas. El ensamblaje con módulos de análisis fiables ha sido ya realizado.

Actualmente se está trabajando en el módulo de postproceso y en la incorporación de nuevos módulos de análisis.

Bibliografía

- * *Boundary Elements. An Introductory Course.* C.A. Brebbia y J. Domínguez. Computational Mechanics Publications. McGraw-Hill Book Company. 1989.
- * *Computer Graphics Programming. GKS- The Graphics Standard.* G. Enderle, K. Kansy, G. Pfaff. Springer-Verlag. 2ª edición. 1987.
- * *The draftsman's and related equations.* J. Approx. Theory, 1968, 1, 191-208. G. Birkhoff & W. J. Gordon.
- * *Blending-function methods of bivariate and multivariate interpolation and approximation,* SIAM.J. Num. Anal. 1971, 8, 158. W.J. Gordon.
- * *An Introduction to Database Systems.* C. J. Date. The Systems Programming Series. Addison-Wesley. 3ª edición. 1981.
- * *X Window System Technical Reference.* S. Mikes. Addison-Wesley. 1990.