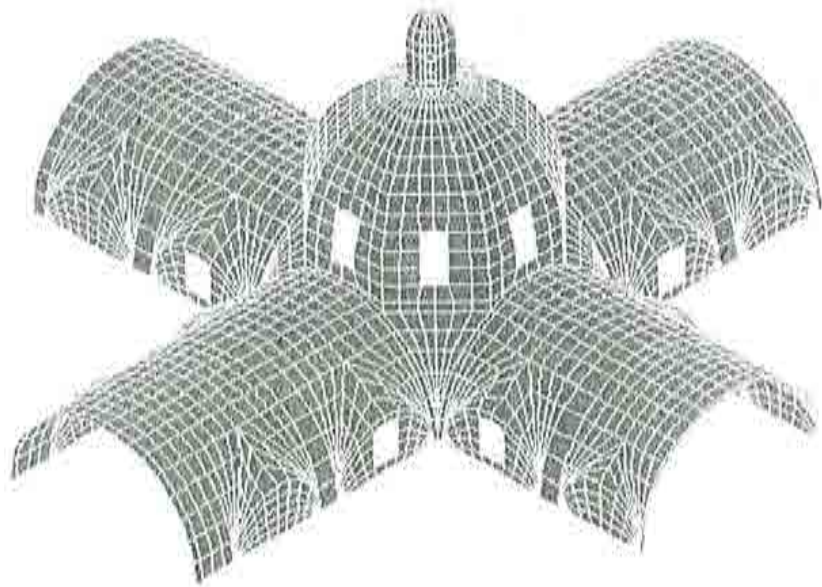


# **Pre y Postprocesadores para el Cálculo de Estructuras**

**Interface GiD-Sap90**

**F. Muñoz Salinas  
J. Maristany i Carreras**



# **Pre y Postprocesadores para el Cálculo de Estructuras**

## **Interface GiD-Sap90**

**F. Muñoz Salinas  
J. Maristany i Carreras**

**Monografía CIMNE N° 52, Julio 1999**

**Colaboradores en la edición:**

Tania Alvarez Sandoval

Juan Giraldo Pereira

Jaume Bennassar i Vicens

Diseño de la cubierta: Jordi Pallí

Primera Edición, Julio 1999

© Los autores

Edita:

Centro Internacional de Métodos Numéricos en Ingeniería

Edificio C1, Campus Norte UPC

Gran Capitán, s/n

08034 Barcelona, España

ISBN: 84-89925-49-6

Deposito Legal: B-36411-99

## ÍNDICE

	<b>Presentación.....</b>	<b>5</b>
<b>1</b>	<b>Introducción.....</b>	<b>7</b>
<b>2</b>	<b>Introducción al GiD Pre y Postproceso</b>	
2.1	El Preproceso.....	11
2.1.1	El modelo matemático.....	13
2.1.1.1	Idealización de la estructura.....	14
2.1.1.2	Modelado del material.....	15
2.1.1.3	Ecuaciones del problema.....	15
2.1.2	Herramientas de uso en la discretización.....	16
2.2	El Postproceso.....	18
<b>3</b>	<b>Estado actual de la informática y su evolución</b>	
3.1	Introducción.....	21
3.2	Modelo matemático.....	22
3.3	Programas para el cálculo de estructuras.....	23
3.4	Métodos numéricos.....	23
3.5	Modelos de tipologías de las estructuras.....	24
3.6	Utilización de los programas de cálculo.....	24
3.6.1	Programas genéricos.....	24
3.6.1.1	Proceso de programación.....	24
3.6.1.2	Hojas de cálculo.....	25
3.6.2	Programas específicos.....	25
<b>4</b>	<b>Introducción al Método de los Elementos Finitos</b>	
4.1	Principios del Cálculo Matricial.....	29
4.2	Aplicaciones al Análisis Matricial de barras.....	30
4.3	Conceptos básicos del Método de los Elementos Finitos.....	35
<b>5</b>	<b>El programa Sap90. (breve introducción.)</b>	
5.1	Introducción.....	39
5.2	Los programas Sap80 y Sap90.....	40
5.2.1	Precauciones al usar Sap90.....	40
5.3	Sus alcances y limitaciones.....	41
5.4	Librerías.....	41
5.4.1	Los módulos de Sap90.....	41
5.5	Pre y Postproceso.....	43
5.5.1	Preproceso del Sap90.....	43
5.5.1.1	Organización del fichero de datos.....	43
5.5.1.2	Fichero de datos Sap90.....	44
5.5.1.3	Opciones de generación de nodos.....	45
5.5.1.4	Restraints (Coacciones rígidas – soportes).....	47
5.5.1.5	Shell (Elementos lámina).....	48
5.5.1.6	Descripción nodal de los elementos shell.....	49
5.5.1.6.1	Generación automática de elementos a partir de un elementos base	49
5.5.1.7	Loads (Cargas nodales – fuerzas y momentos puntuales).....	50
5.5.2	Criterio de signos y de representación de esfuerzos y momentos.....	51
5.5.3	Ejemplo de generación.....	53
5.5.4	Postproceso.....	54
5.6	Instrucciones más comunes del Sap90.....	56
5.6.1	La subrutina "GO".....	56
5.6.2	La secuencia de ejecución.....	56
5.6.3	Los comandos.....	57
5.6.4	La opción Restart.....	58
5.6.5	Las subrutinas del cálculo.....	59
5.6.5.1	Bloque de datos (Frame)	
5.6.5.2	Bloque de datos (Shell)	
5.6.5.3	Bloque de datos (Asolid)	
5.6.5.4	Bloque de datos (Solid)	

<b>6</b>	<b>El programa GiD. (pre y postproceso)</b>	
6.1	Introducción.....	61
6.1.1	Fundamentos de GiD.....	63
6.2	Usos, comandos y alcances del GiD.....	64
6.2.1	Uso de la interface.....	64
6.2.2	Definición de puntos (camino para definir puntos).....	65
6.2.3	Selección de entidades.....	66
6.2.4	Escape: comando de uso continuo (aceptar).....	66
6.2.5	Manual de referencia.....	67
6.2.5.1	Archivos.....	69
6.2.5.2	Creación de entidades geométricas.....	74
6.2.5.3	Eliminación de entidades geométricas.....	78
6.2.5.4	Edición de entidades geométricas.....	78
6.2.5.5	Condiciones de borde.....	81
6.2.5.6	Materiales de base.....	82
6.2.5.7	Datos del problema.....	83
6.2.5.8	Intervalo de datos.....	83
6.2.5.9	Malla.....	84
6.2.5.10	Opciones de visualización.....	88
6.2.5.11	Utilidades.....	94
6.2.5.12	Salir.....	98
	Índice de conceptos.....	99
<b>7</b>	<b>Configuración del GiD para el preproceso</b>	
7.1	Introducción.....	101
7.2	Adaptación de GiD a diferentes programas de cálculo de estructuras.....	103
7.2.1	Datos generales del problema.....	105
7.2.2	Materiales de base.....	106
7.2.3	Condiciones de contorno.....	107
7.2.4	Símbolos gráficos para las condiciones de contorno.....	109
7.2.5	Descripción del formato de escritura.....	110
7.2.6	Descripción de la salida de archivos para el cálculo.....	111
7.2.7	Comandos que devuelven un solo valor.....	112
7.2.8	Comandos que devuelven más un valor a la vez.....	113
7.2.9	Comandos que realizan bucles, condicionales y otros.....	114
7.3	Ejemplo de la configuración del archivo.bas.....	115
<b>8</b>	<b>Configuración de GiD para el postproceso</b>	
8.1	Archivos y ventanas de uso general de GiD en el postproceso.....	119
8.1.1	Archivos (file).....	120
8.1.2	Utilidades (utilities).....	121
8.1.3	Hacer cortes (do cuts).....	122
8.1.4	Opciones (options).....	122
8.1.5	Ventanas (windows).....	123
<b>9</b>	<b>Preproceso, Cálculo y Postproceso de una viga pared (interface asolid)</b>	
9.1	Descripción del problema.....	127
9.2	Generación de la geometría.....	128
9.3	Definición de la malla.....	130
9.4	Configuración del preproceso.....	134
9.4.1	Condiciones de contorno.....	135
9.4.2	Características del material.....	136
9.4.3	Datos del problema.....	137
9.4.4	Intervalo del problema.....	138
9.5	Cálculo del problema.....	138
9.6	Postproceso.....	139
9.6.1	Configuración de los resultados.....	139
9.6.2	Visualización de los resultados tensionales.....	141
9.6.3	Visualización de los resultados de deformacionales.....	148
9.6.4	Herramientas del postproceso.....	149
9.6.5	Salir del postproceso.....	149

<b>10</b>	<b>Preproceso, Cálculo y Postproceso de un forjado (Interface shell)</b>	
10.1	Preproceso.....	151
10.1.1	Generación de la geometría.....	151
10.1.2	Creación de la malla.....	152
10.1.3	Definición de los parámetros para su cálculo.....	153
10.2	Cálculo.....	155
10.3	Postproceso.....	156
<b>11</b>	<b>Ejemplo práctico de la interface 3dshell y 3d solid</b>	
11.1	Interface 3dshell.....	163
11.1.1	Descripción del problema.....	163
11.1.2	Postproceso.....	164
11.2	Interface 3dsolid.....	170
11.2.1	Descripción del problema.....	170
11.2.2	Postproceso.....	171
	Conclusiones.....	180
	<b>Anexos.....</b>	<b>181</b>
	<b>A.1 Anexo 1.....</b>	<b>182</b>
	<b>A.1.1 Archivos de uso para la configuración Asolid2d.gid de la Interface GiD-Sap90</b>	
	A.1.1.1 Archivo Asolid2d.cnd (Archivo de Condiciones)	
	A.1.1.2 Archivo Asolid2d.mat (Archivo de Materiales)	
	A.1.1.3 Archivo Asolid2d.sim (Archivo de Símbolos gráficos para condiciones)	
	A.1.1.4 Archivo Asolid2d.prb (Archivo de Datos del problema)	
	A.1.1.5 Archivo Asolid2d.bas (Archivo para la ordenación de los datos para su análisis)	
	<b>A.2 Anexo 2.....</b>	<b>187</b>
	<b>A.2.1 Archivos de uso para la configuración Shell2d.gid de la Interface GiD-Sap90</b>	
	A.2.1.1 Archivo Shell2d.cnd (Archivo de Condiciones)	
	A.2.1.2 Archivo Shell2d.mat (Archivo de Materiales)	
	A.2.1.3 Archivo Shell2d.sim (Archivo de Símbolos gráficos para condiciones)	
	A.2.1.4 Archivo Shell2d.prb (Archivo de Datos del problema)	
	A.2.1.5 Archivo Shell2d.bas (Archivo para la ordenación de los datos para su análisis)	
	<b>A.3 Anexo 3.....</b>	<b>192</b>
	<b>A.3.1 Archivos de uso para la configuración 3dshell.gid de la Interface GiD-Sap90</b>	
	A.3.1.1 Archivo 3dshell.cnd (Archivo de Condiciones)	
	A.3.1.2 Archivo 3dshell.mat (Archivo de Materiales)	
	A.3.1.3 Archivo 3dshell.sim (Archivo de Símbolos gráficos para condiciones)	
	A.3.1.4 Archivo 3dshell.prb (Archivo de Datos del problema)	
	A.3.1.5 Archivo 3dshell.bas (Archivo para la ordenación de los datos para su análisis)	
	<b>A.4 Anexo 4.....</b>	<b>197</b>
	<b>A.4.1 Archivos de uso para la configuración 3dsolid.gid de la Interface GiD-Sap90</b>	
	A.4.1.1 Archivo 3dsolid.cnd (Archivo de Condiciones)	
	A.4.1.2 Archivo 3dsolid.mat (Archivo de Materiales)	
	A.4.1.3 Archivo 3dsolid.sim (Archivo de Símbolos gráficos para condiciones)	
	A.4.1.4 Archivo 3dsolid.prb (Archivo de Datos del problema)	
	A.4.1.5 Archivo 3dsolid.bas (Archivo para la ordenación de los datos para su análisis)	
	<b>Bibliografía.....</b>	<b>203</b>
	<b>Recomendaciones y precauciones de la interface GiD-Sap90.....</b>	<b>205</b>





## PRESENTACIÓ

Ja fa molt de temps que Francisco Muñoz Salinas va aparèixer en el Departament d'Estructures de l'ETSAB interessant-se pels estudis del tercer cicle i després en qualitat de becari. Jo me'n recordo, ja aleshores, de la seva enorme curiositat, il·lusió i la seva gran capacitat de treball que exhibia. Capacitat, illusió i curiositat que ajuntades a la vegada "mouen muntanyes" com vulgarment es diu. Així va començar en el Departament la singladura d'aquell, en aquest temps aprenent, de les "intrínquils" estructurals dels diferents elements constructius que s'estudien avui en dia.

D'aquella situació original passa a plantejar-se la redacció de la seva tesi doctoral\* qui no ha passat per aquí - que analitza el funcionament dels materials petrís en edificis religiosos, el que li porta inevitablement a profunditzar els coneixements sobre un dels sistemes de càlcul que més han revolucionat darrerament el càlcul de les estructures : el mètode dels elements finits.

Per altre banda, la seva tradicional curiositat el porta a posar-se en contacte amb el *CIMNE*<sup>†</sup> i amb els seus programes.

Es aquí on es troba amb un problema : els programes que tradicionalment es feien servir al nostre Departament per analitzar una estructura a través del mètode dels elements finits : El *Sap*, que s'ha convertit ja en un clàssic, i d'altres programes el *KIRCHOFF*, el *PLASTIC*, etc. dissenyats per un membre d'aquest mateix Departament, no son compatibles amb els que normalment treballa el *CIMNE* : en general programes propis de difícil extrapolació. Lògic si pensem amb la trajectòria absolutament independent de les dues Escoles, Camins i Arquitectura, estúpídament plantejada al meu entendre vista la trajectòria i analitzant els continguts de les seves ensenyances més tècniques.

D'aquí a plantejar-se la possibilitat de fer uns programes que lliguin els programes d'ambdues facultats hi va un pas. El resultat son actualment uns programes que s'utilitzen força al nostra Departament i creiem que, sincerament també se'n pugui aprofitar tothom que vulgui.

D'aquí neix la necessitat de fer aquesta publicació, redactada diria més amb l'ànim de donar a conèixer el treball realitzat i que aquest serveixi per tothom que vulgui més que per motius estrictament professionals i no diguem crematístics<sup>‡</sup>.

---

\* De títol : "Estudio comparativo del comportamiento estructural del Abside de la Sagrada Familia con el Sagrario Lateral de la Catedral de la Ciudad de México".

† Centro Internacional de Métodos Numéricos.



Així ha sortit una publicació que s'ha volgut expressament que no fos gens farragosa, cosa al meu entendre molt difícil de aconseguir i més quan es tracta d'estudiar el que passa amb els elements finits. I penso que ho ha aconseguit plenament tant pel que respecta a l'estructuració del llibre com al seu contingut.

Parlem de la seva estructuració:

El llibre es compon de nou capítols independents :

Els primers, estan dedicats fonamentalment a centrar el problema que més endavant es desenvolupa i a recordar per aquells més profans, com funcionen els actuals mètodes de càlcul informàtics, fent especial incidència amb el mètode dels elements finits. Es tractaria dels capítols primer al cinquè.

A continuació ve un segon grup de capítols, els que van del sisè fins el vuitè. És on es planteja la solució adoptada. Estudia tant la fórmula adoptada en el pre-procés (entrada de dades) com la del post-procés (interpretació de resultats). És el nucli del llibre. És on es pot veure la potència de la solució adoptada i entenent els seus fonaments. És on més tindrem de posar els colzes per entendre fins el final el que ens ha volgut dir Francisco Muñoz.

Finalment s'ha volgut dedicar expressament els dos darrers capítols, el novè i el desè, a veure a la pràctica com funciona aquesta *interface GiD-Sap90* en dos exemples concrets :

En el capítol novè es desenvolupa el càlcul fent servir l'element "ASOLID" per a trobar els esforços i les tensions d'una biga - paret. Recordem que l'estudi d'aquest element constructiu està expressament contemplat amb la nova Instrucció com a "regiones D" (de discontinuïtat) És a dir que no es pot interpretar el seu funcionament utilitzant exclusivament les mateixes hipòtesis : Bernouilli, Navier, etc que es feien servir per analitzar els demés elements estructurals.

En el capítol desè, en canvi, s'utilitza l'element tipus "Shell" per trobar el que passa en un forjat.

D'aquesta forma s'ha combinat primer un aprenentatge en l'utilització d'aquests sistemes de càlcul i d'una localització en l'espai de la seva introducció al mercat, amb un posterior anàlisi molt més profund d'elles seves característiques desenvolupant clarament el mètode d'unió d'ambdós programes : el *GiD* i el *Sap*, amb inclusió dels llistats FONT definitius, cosa que es molt d'agrair per part del futur usuari i investigador; per finalment acabar l'explicació de la manera que tot s'entén millor : amb la redacció d'uns exemples pràctics prou desenvolupats, fonamentalment el primer, on es pot veure la potència de la combinació dels dos programes abans citats.

Jordi Maristany Carreras

<sup>1</sup> Actualment per viure (sobreviure?) de les publicacions que un fa he calculat que es necessarien editar almenys una quinzena com aquesta cada mes. És a dir una cada dos dies (?)

# 1 INTRODUCCIÓN

La irrupción de la informática en el mundo del cálculo y la interpretación de las estructuras de edificación es una realidad que de tan evidente a veces se olvida.

Recuerdo que no hace tanto tiempo los calculistas tenían como principal herramienta de trabajo un bolígrafo "bic" y una calculadora a cada mano. Llenaban páginas y más páginas de papel con cálculos cada vez más sofisticados.

Recuerdo aquellos "artesanos" del cálculo trabajando sobre aquellos contra-vegetales recién salidos de la casa de copias dibujando redondos de armadura metálica, con la plantilla de letras (o a mano los más dotados para la ortografía y la buena letra) rotular miles de escritos y poniendo aquellos "3 φ 16 c/20" que todos recordamos: unos con afinanza y otros con una sensación de alivio nada disimulada.

Hoy en día los monitores de ordenador ocupan las antiguas mesas de dibujo. Los llamados ahora calculistas-informáticos (nunca he acabado de entender cuál es el parecido entre ambas profesiones para que las escriban juntas) se cansan de introducir datos a través de un programa de cálculo que se renueva (y paga) cada año y hasta cada semestre.

El importe que antes se destinaba a salarios de trabajadores, ahora se dedica a la compra y manutención del "hardware - las máquinas - y del software - los programas - . Pero no sólo han cambiado las formas; también se han modificado la manera de interpretar las estructuras, las personas que lo hacen y las conclusiones a las que llegan.

Anteriormente era impensable que una persona que no estuviera bregada en el tema se atreviera a plantearse el cálculo y diseño de una estructura cualquiera. Era consciente de su escasa preparación en la materia cuando se le obligaba a tomar decisiones del estilo de: que coeficiente de pandeo tiene que coger, como afecta la torsión a los elementos secundarios, cuando se tenía que colocar una segunda capa de armaduras, cuál era el sistema de armado más adecuado para el elemento estructural que se estaba diseñando, etc.

El uso de la informática permite hoy obviar a priori muchos de estos interrogantes y el problema parece que "sólo" estriba en introducir los datos correctamente, tal como reza el manual correspondiente al programa de turno que estemos utilizando.

La diferencia respecto a épocas anteriores es que ahora casi cualquiera se atreve a enfrentarse a una estructura de hormigón compleja. Se tiene la impresión de que es sólo un problema de entrada de datos y de interpretación de los resultados. Hay quien cree, aunque se equivoque de medio a medio, que lo importante es estar familiarizado con el programa aunque uno no tenga ni remota idea de lo que es un redondo.

El hecho de no tener que enfrentarse de antemano a dichos interrogantes puede dar la falsa impresión de que quien se equivoca, quien sabe, quien piensa y quien decide no es el calculista sino el programa que utilizamos. ¿A quién no le ha pasado el lamentable percance, demasiado frecuente en nuestro país, que se va a la luz mientras se está calculando una estructura, quedándose lo que se dice "en blanco"? ¿Pero no era el ordenador el que calculaba, pensábamos?

La realidad es bien distinta. A quien se le pueden agrietar las estructuras, quién tendrá que aguantar las protestas de los afectados, quien puestos a pensar en lo peor, tendría que responder incluso de su responsabilidad penal, no sería la máquina, sino aquel pobre iluso que quería pensar que el ordenador lo hacía todo.

Uno descubre, a veces demasiado tarde, que la máquina "sólo" es un instrumento y que siempre es necesario probar, comprobar, cotejar y tomar decisiones, aunque se tenga mucha prisa por entregar los cálculos y los planos acabados.\*

Hay que ser conscientes de que el uso inevitable de la informática en el cálculo y diseño de las estructuras ofrece grandes ventajas pero no está exenta de inconvenientes. Sería como el refrán pero a la inversa: a grandes remedios, grandes males. Tales inconvenientes se pueden producir tanto en la introducción de datos *-preproceso-* como en la interpretación de los resultados obtenidos *-postproceso-*.

Ayudar a que estas dos fases del cálculo se simplifiquen y por tanto que se entiendan mejor es precisamente el objetivo de este libro que el lector, tiene en sus manos.

No se trata ni de un libro tedioso sobre un ensayo teórico sin ninguna aplicación. Ni tampoco de aquellos que vemos con frecuencia expuestos en las librerías. No es un libro superficial que explique de forma genérica los procesos informáticos que se utilizan en la relación usuario-máquina. Se trata mas bien del resultado de una investigación llevada con suma rigurosidad por Francisco Muñoz Salinas por encargo del Departamento de Estructuras de Arquitectura para resolver un problema de uso de los diferentes programas de elementos finitos que se utilizan en el Departamento.

Es evidente que hoy en día el disponer de ordenadores cada vez mas potentes, tanto en lo relativo a la

---

\* Pensemos, que lo que cuesta tiempo y dinero fundamentalmente son las horas del calculista, las horas que el especialista dedica a comprobar la estructura que es el trabajo diario mas rentable y menos vistoso que existe en este proceso.

memoria como a su capacidad de almacenar, permite utilizar programas y sistemas de cálculo impensables hasta hace poco tiempo.

Así se explica el éxito del método de los elementos finitos que aunque requiere una entrada de datos algo complicada todavía, permite analizar las mas variadas tipologías estructurales con la obtención, normalmente dentro del campo elástico, de resultados bastante aproximados siempre que se tenga cuidado de hacer una discretización suficientemente correcta del objeto que se pretende diseñar.

Su principal problema radica en "hacer una entrada de datos" suficientemente sencilla y inteligible así como de conocer la forma de interpretar los resultados que nos facilita el ordenador. En todos los programas de elementos finitos que van apareciendo se aprecia la preocupación por esta cuestión, de ahí que ofrezcan nuevas versiones que contemplan logaritmos para facilitar el trabajo al usuario que introduce los datos o interpreta los resultados.

El inconveniente se deriva de que para cada problema, cada programa y para cada máquina<sup>†</sup> es necesario utilizar un sistema diferente de simplificar la interacción usuario-máquina. Este es el principal escollo a superar porque resulta cada vez mas difícil recordar en cada caso y para cada programa cual es el logaritmo mas adecuado.

Este es el problema que intenta solucionar el programa *GID* de *pre* y *postproceso*: Con un solo sistema de entrada de datos se puede acceder a un gran número de programas de elementos finitos diferentes. Se trata únicamente de haber previsto, haber diseñado el programa de interconexión "*GID*-Programa" que queremos utilizar.

Sólo se trabaja con un solo logaritmo. Sólo hay una única manera de entender los datos y una sola manera de entender e interpretar los resultados. Es fantástico!

Lo que hace falta es diseñar esta interconexión del *GID* con cada programa que se quiera utilizar mas adelante ¿Y por qué no?, donde esta aplicación es mas rentable es en la conexión del *GID* con aquellos programas que han representado auténticos estandartes de mercado como es el caso del *SAP*.

Por este motivo es necesario estudiar a fondo el funcionamiento del *GID*, como trabaja el *SAP90* y acertar con la solución. Esta y no otra ha sido la tarea, el auténtico objetivo que ha guiado a mi apreciado compañero Francisco Muñoz a elaborar un programa de unión, cuya tenacidad ha dado a luz esta publicación.

Espero que saquéis un buen provecho de su lectura porque sinceramente vale la pena.

Barcelona, a 17 de junio de 1999.

Fdo. Jordi Maristany i Carreras

<sup>†</sup> No es lo mismo trabajar con una máquina mínima de Windows 98 que con un terminal de un servidor complicado.



## 2 INTRODUCCIÓN A GiD PRE Y POST PROCESO

### 2.1 El preproceso

En la actualidad, el Cálculo de Estructuras se realiza mediante programas informáticos cada vez más avanzados y sofisticados. Al mismo tiempo se implementan nuevos métodos de cálculo, que nos permiten conocer de forma ágil y fácil el comportamiento estructural de cualquier género de edificio.

El Método de los Elementos Finitos es uno de ellos y gracias a la informática es, actualmente, una de las herramientas más utilizadas para el cálculo de estructuras entre profesionales e investigadores.

Pero, su problema fundamental radica en la discretización de la estructura a estudiar ya que no siempre se trata de estructuras sencillas. Por esta razón, la modelización para su cálculo posterior es compleja y esto implica que se implementen programas de dibujo (*CAD*) para poder realizarla.



Fig. 2.1<sup>\*</sup> Hiperboloide de Revolución (Torroja)  
Tanque elevado de agua

Esto es a lo que llamaríamos el *preproceso*; a la interpretación de entidades de dibujo las cuales a su vez se traducen en datos necesarios (generación de nodos con sus respectivas coordenadas y conectividad de los elementos) para su cálculo posterior.

<sup>\*</sup> Cortesía del Estructuras de E.T.S.A.B.



Estos datos son, como veremos mas adelante, interpretados por un programa de Elementos Finitos, junto con sus respectivos datos estructurales tales como: materiales, condiciones de contorno, etc.

Dentro del cálculo de estructuras, por medio de programas informáticos, en los que se utiliza el Método de los Elementos Finitos, el *preproceso* siempre ha sido un problema difícil de resolver ya que prácticamente todas las estructuras son de geometría variada y en algunos casos complejas, no siempre fáciles de resolver.

Debido a esto, es importante poder realizar un *preproceso* adecuado de estas estructura, antes de proceder al cálculo de ella.

En la mayoría de los programas, el *preproceso* se realiza manualmente insertando los datos de la geometría o con la ayuda de un programa de dibujo. Esto significa dibujar la estructura que queremos estudiar y a su vez realizar una *discretización* de ella (generación de la malla).

Una vez que se tiene esta discretización, normalmente se graban los resultados en un archivo con la extensión *\*.dx* el cual, nos proporcionará nodos, coordenadas de los mismos, elementos de la malla, así como la conectividad entre ellos.

Posteriormente deben de anexarse características de materiales, condiciones de contorno, cargas, método de cálculo, etc.

Para realizar todo este proceso es necesario dedicarle mucho tiempo, además de que puede generarse errores en la discretización de la estructura lo cual, se puede traducir en una mala interpretación de los resultados.

Como se observa en la tabla 2.1, el primer paso a realizar en el *preproceso* es conocer el tipo de estructura a calcular (problema a resolver).

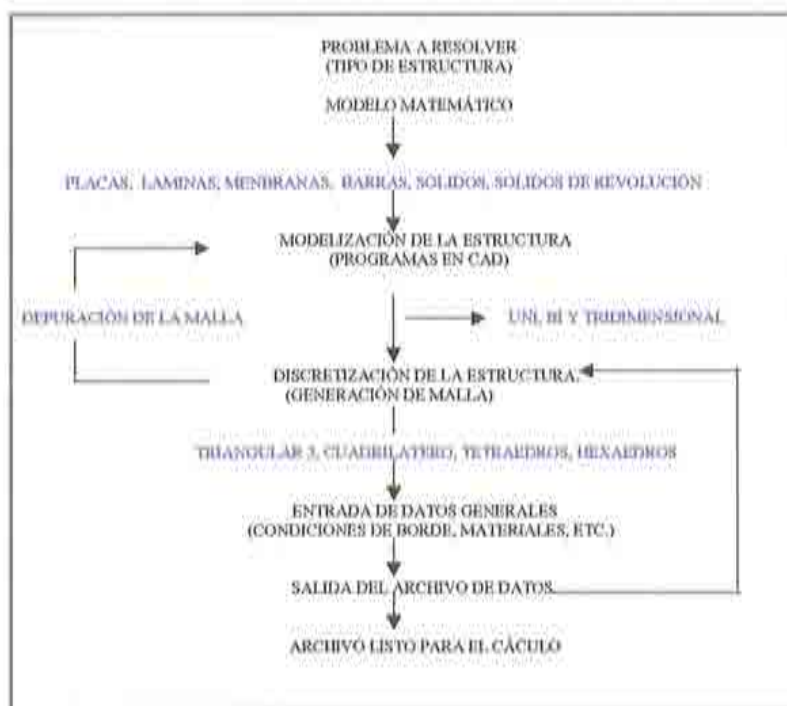
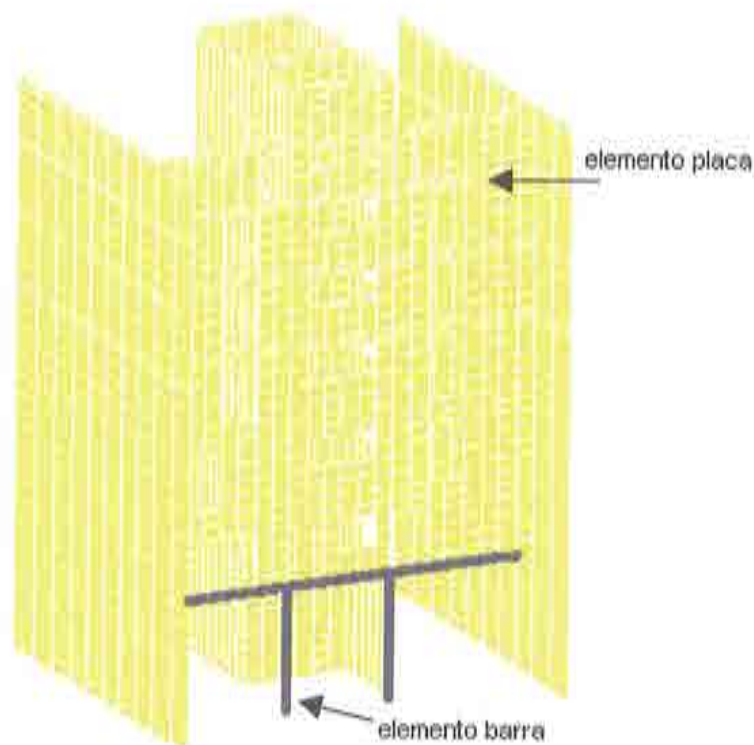


Tabla 2.1 Diagrama del Preproceso de una estructura



Esto significa, seleccionar primero un modelo matemático adecuado para describir su comportamiento estructural<sup>1</sup>.

Una vez que se tiene definido el modelo matemático el siguiente paso sería la discretización (generación de la malla) de la estructura en pequeños trozos que les llamaremos elementos finitos. Los elementos (fragmentos) se conectan entre sí a través de nodos situados en su contorno.



*Fig. 2.2 Discretización de un Edificio de Fábrica  
(Modelización con dos tipos de elemento: barras y placas)*

La generación de malla (discretización), puede estar formada por elementos unidimensionales (frames), bidimensionales (shells, asolids) e incluso tridimensionales (solid); o una combinación de los mismos.

Tal como ya se ha mencionado, la etapa de discretización constituye una parte primordial del *preproceso*.

### 2.1.1 El Modelo Matemático

Aceptando que la mayoría de las estructuras tienen un comportamiento continuo<sup>2</sup> y para conocer su comportamiento estructural, es necesario realizar un análisis más profundo de ellas, para resolver las ecuaciones diferenciales que expresen su equilibrio.

<sup>1</sup> Por ejemplo, si se tiene un edificio soportado solo con muros de carga, podríamos utilizar la teoría de placas o la de la elasticidad tridimensional.

<sup>2</sup> Con excepción de las de barras

El Cálculo Matricial de barras y el Método de los Elementos Finitos para barras y estructuras continuas, son actualmente los métodos de cálculo más utilizados.

En el primer caso<sup>§</sup>, el análisis está destinado al cálculo de las deformaciones elásticas y de los esfuerzos producidos en una estructura de barras, sometidas a unas acciones determinadas.

En lo que respecta al M.E.F.<sup>\*\*</sup>, actualmente es uno de los procedimientos más potentes para el análisis de estructuras de carácter uni, bi o tridimensional sometidas a las acciones más diversas. La gran semejanza entre el cálculo matricial de barras y el M.E.F. facilitan sobre manera el estudio de éste.

En este apartado solo analizaremos los modelos matemáticos relacionados con el M.E.F. Dichos conceptos serán de gran ayuda para el estudio de estructuras de tipologías más complejas.

A partir de esta introducción, el siguiente paso será definir el modelo matemático que se utiliza en el cálculo.

Es importante resaltar que la aplicación de los modelos matemáticos estará en función de la imaginación o la capacidad de abstracción del usuario frente a una estructura real, sea plana o tridimensional.

Para comprender lo que significa el modelo matemático, hemos subdividido la explicación en tres apartados:

- |  |  |
|--|--|
| 1. <i>Idealización de la estructura:</i> | <i>(Vigas, placas, láminas, elasticidad bi y tridimensional, sólidos y láminas de revolución).</i> |
| 2. <i>Modelado del material:</i>         | <i>(Material Elástico lineal o no lineal).</i>   |
| 3. <i>Ecuaciones del problema:</i>       | <i>(Ecuaciones diferenciales).</i>   |

### 2.1.1.1 Idealización de la estructura

Tomemos como ejemplo el análisis de una cubierta de un edificio como el de la Figura 2.2

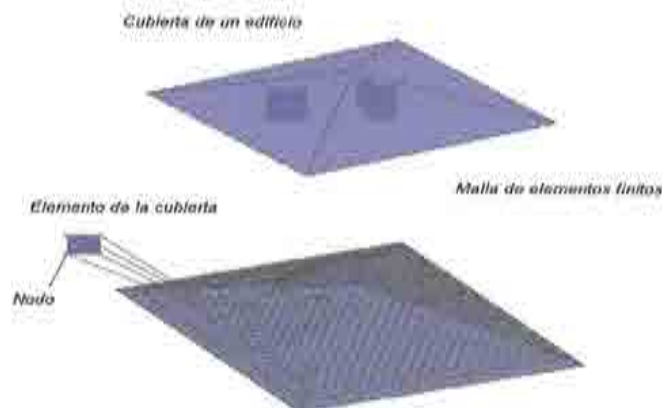


Fig. 2.3 Análisis de una cubierta de un edificio por el M.E.F.

<sup>§</sup> Siempre y cuando la simplificación de la estructura lo permita

<sup>\*\*</sup> Método de los Elementos Finitos (A partir de ahora utilizaremos esta abreviación para simplificar el M.E.F.)

A partir de esta cubierta real, de sus condiciones de borde y de los tipos de carga que actúan sobre ella, es necesario, inicialmente, seleccionar un modelo matemático adecuado para definir su comportamiento<sup>††</sup>.

A título orientativo algunas de las teorías utilizadas en el modelo matemático podrían ser:

- Para Vigas:	<i>Flexión de vigas esbeltas (Teoría de Euler-Bernoulli)</i> <i>Flexión de vigas de Timoshenko.</i>
- Para Placas delgadas:	<i>Teoría de Kirchhoff.</i>
- Para Placas Gruesas:	<i>Teoría de Reissner-Mindlin.</i>
- Para Elasticidad Bidimensional:	<i>Tensión y Deformación plana.</i>
- Para Sólidos:	<i>Teoría de la Elasticidad Tridimensional.</i>

### 2.1.1.2 Modelado del material

Ahora hay que definir las características mecánicas de los materiales de la cubierta, así como la naturaleza de las deformaciones de la misma (pequeños o grandes desplazamientos, análisis en primer o segundo orden, análisis dinámico o estático, etc.).

Una vez se selecciona el modelo matemático ya se procede a discretizar la estructura.

### 2.1.1.3 Ecuaciones del Problema

El siguiente paso del *preproceso* está en la obtención de la matriz de rigidez y del vector de carga para cada elemento discretizado. El hecho de que intervengan integrales sobre el dominio uni, bi, y tridimensional hace que sea más compleja su obtención.

$K$  = *Matriz de Rigidez de la Estructura.*

$f$  = *Vector de Cargas por cada Elemento.*

Una vez que se tienen estos datos, se procede al ensamblaje de las matrices con el vector de cargas en una matriz global en toda la malla de elementos finitos. El sistema de ecuaciones resultante nos permitirá obtener los desplazamientos nodales y las deformaciones.

- *Sistema de ecuaciones resultante  $Ku = f$*

Donde:

*$u$  es la variable incógnita (Movimiento de cada uno de los nodos)*

<sup>††</sup> Por ejemplo, podría utilizarse la teoría de la elasticidad tridimensional o la de láminas planas.

Calculados los movimientos nodales  $a$  ya se pueden obtener las deformaciones y, a continuación, los esfuerzos en cada elemento.

- Relación esfuerzo – deformación:  $\sigma = E\varepsilon$

Donde:

$\sigma$ ..... Es el vector de tensiones  
 $E$ ..... Es el módulo de Elasticidad (características del material)  
 $\varepsilon$ ..... Son los movimientos nodales unitarios

Lógicamente, para poder realizar todas estas operaciones en el *preproceso*, es necesario apoyarse en un programa informático que lo resuelva.

### 2.1.2 Herramientas de uso en la discretización

Como ya se ha comentado, antiguamente cuando se quería discretizar una estructura para su posterior análisis, no había más remedio que hacerlo de forma manual.

Esta discretización se hacía, lógicamente, en estructuras que se podían simplificar en el plano (barras o elementos bidimensionales) ya que era la manera más sencilla de controlar su geometría.

Actualmente, y gracias al avance de la informática, esa tarea es mucho más sencilla. Existen programas gráficos (*CAD*) que nos permiten dibujar cualquier estructura, por compleja que sea, discretizarla, y transformar su geometría (archivo *dxf*) en datos útiles para su cálculo.

Lo más habitual es aprovechar los ficheros de salida de los programas gráficos formateados en un lenguaje estándar, como los *\*.dxf*, que contengan información de la geometría del dibujo y de su discretización.

Este fichero lo lee e interpreta un programa de cálculo que lo traduce en información sobre los nodos (puntos), los elementos de la discretización, así como su conectividad entre ellos.

Finalmente, toda esta información se introduce en un programa de cálculo, ya sea matricial o de elementos finitos, para un análisis posterior.

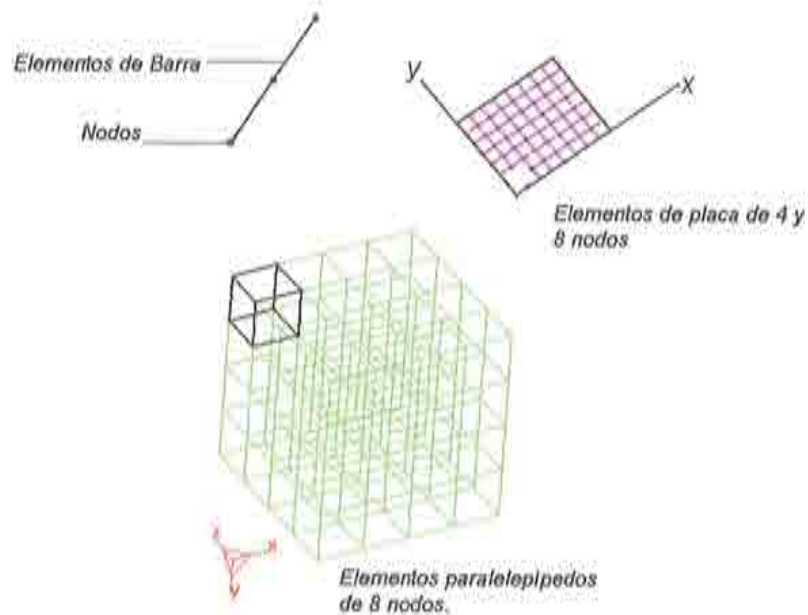
Nos podemos apoyar en diferentes tipos de elementos finitos para discretizar (generar malla) una estructura.

El Programa *Sap90*, que más adelante comentaremos, utiliza:

1. Elementos de barra con dos o más nodos.
2. Elementos rectangulares de cuatro nodos.
3. Elementos rectangulares y paralelepípedos de ocho nodos.
4. Elementos rectangulares de nueve nodos.



*Tipos de elementos finitos  
que emplea el Sap90*



*Fig. 2.4 Tipos de Elementos finitos*

Una vez que se han introducido esos datos (Geometría) en un archivo, se procede a ensamblar el resto de la información, tal como: el tipo de material (Densidad, módulo de Poisson, módulo de Elasticidad, etc.), condiciones de borde (empotramientos, desplazamientos y giros) de la estructura, cargas, peso propio, así como el resto de los datos que requiere el programa para su correcto funcionamiento.

Pero este proceso, que aunque resulta más rápido y eficaz que el anterior, sigue siendo bastante complicado pues continua siendo inevitable el uso de los programas que transforman los archivos de dibujo a uno de texto.

Por esas razones ahora empiezan a popularizarse programas de cálculo que tienen su propio *preproceso* como serían: el *Wineva*, el *Sap90*, el *Sap2000* y el *Ansys*.

Así mismo, la *Interface GID-Sap90* nos permite realizar un *preproceso* mucho más cómodo y dinámico al mismo tiempo que prepara el archivo para la ejecución de su cálculo. Hablaremos de ello mas adelante.

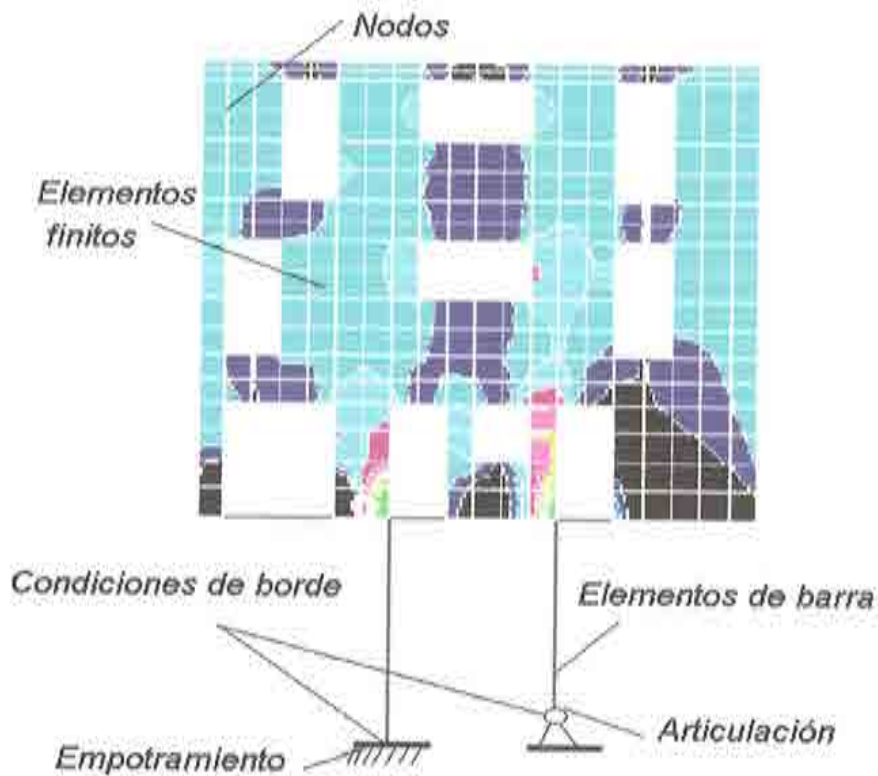


Fig. 2.5 Elementos de una malla

## 2.2 El postproceso

Una vez que la estructura ha sido calculada por algún programa informático ya podemos obtener los resultados. Cada vez más y debido a la cantidad de datos que se manejan, éstos se sustituyen por una adecuada representación gráfica y visualmente más inteligible.

A esto se le llama *postproceso*, a la implementación gráfica de todos y cada uno de los resultados de la estructura calculada.

Este último, al igual que el *preproceso*, requiere de un programa que pueda interpretar de forma gráfica los resultados (esfuerzos, deformaciones, etc.).

Ejemplos de estos resultados son:

- a) *La tipología de la malla de elementos finitos, con sus respectivos elementos, nodos, conectividad entre ellos, así como sus condiciones de contorno.*
- b) *La representación de los desplazamientos y por consiguiente, de la deformación general de la estructura.*
- c) *La visualización de los esfuerzos.*

También es importante que el *postproceso* tenga subrutinas que permitan crear archivos para su impresión o modificación de imágenes.

A partir de esta necesidad, se han creado programas de discretización de ayuda complementaria para los programas de cálculo. Incluso, existen algunos (*Sap90*) que implementan subrutinas de discretización pero aunque limitada a generaciones ortogonales, que siguen siendo muy restringidas.

Actualmente el Centro Internacional de Métodos Numéricos (*CIMNE*) se encuentra investigando este tema y ha creado un programa llamado *GiD*<sup>11</sup>, el cual, conjuntamente con el *Sap90*, nos permiten realizar todo este trabajo de manera sencilla.

La mayoría de los programas comerciales de cálculo de estructuras, contienen su propia representación gráfica; pero pocas veces nos permiten observar los resultados de forma clara y dinámica.

Por ejemplo:

1. *En el Sap90 el postproceso está un tanto mejorado y permite ver con más dinamismo los resultados del cálculo realizado por el mismo programa.*
2. *En el caso de Wineva, su postproceso se limita a la representación en dos dimensiones y solo muestra la información necesaria y utilidades mínimas.*

Actualmente *GiD* es uno de los programas que más prestaciones ofrece en lo que respecta al *postproceso*, ya que permite representar resultados de cualquier programa de cálculo de estructuras. Por esta razón, en este trabajo utilizaremos el *postproceso* de *GiD* para la representación de resultados.

En los capítulos 5 y 6 se explica de manera más detallada los *postprocesos* de *Sap90* y de *GiD*.

Como normalmente este proceso resulta complicado y engorroso, los investigadores cuando utilizan el Método de los Elementos Finitos, se ven en la necesidad de utilizar programas informáticos que integren todos y cada uno de estos procesos (dibujo, discretización, cálculo y representación gráfica de resultados).

---

<sup>11</sup> Programa diseñado por el C.I.M.N.E





### 3 ESTADO ACTUAL DE LA INFORMÁTICA Y SU EVOLUCIÓN EN EL CÁLCULO DE ESTRUCTURAS

#### 3.1 Introducción

Durante el proceso de elaboración de cualquier proyecto arquitectónico, una vez que se definen todas las directrices formales y constructivas, se han de concretar y definir, con una adecuada precisión, los diferentes elementos que la componen (sistemas constructivos, condiciones de borde, cargas, etc.).

En este sentido, muchas de las decisiones se toman a partir de razonamientos lógicos o intuitivamente, pero existen otras que pueden emprenderse por medio de procedimientos de cálculo numérico apoyándose de las ventajas que ofrece la informática y que permiten resultados más exactos del problema abordado.

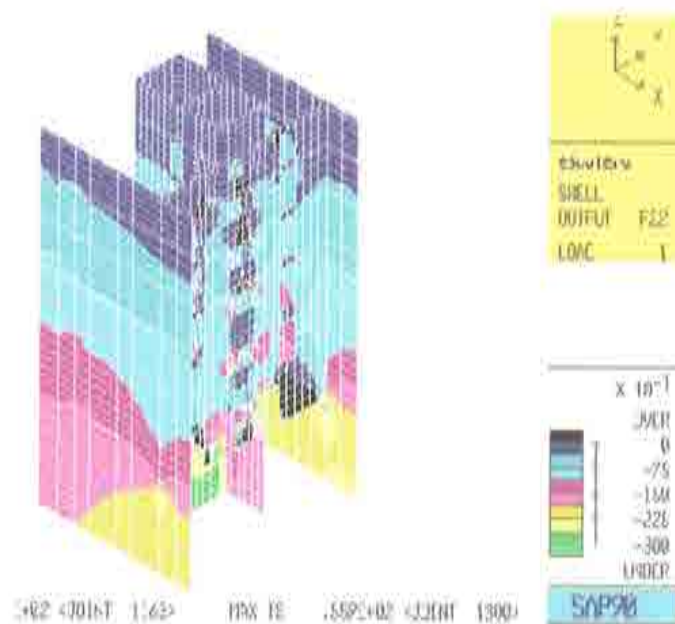


Fig. 3.1 Modelización y Cálculo de un Edificio

Los programas de cálculo de Estructuras tienen como función realizar estos procesos de análisis numérico de forma ágil y rápida; y proporcionar datos que faciliten el conocimiento del comportamiento estructural más aproximado de la estructura, y así facilitar la definición de algunos de los elementos del proyecto, especialmente los que componen el soporte de esta.

En general podemos considerar que estos programas actúan con dos estrategias diferentes:

1. *Predimensionamiento de los elementos estructurales en función de las características definidas por el usuario.*
2. *Estimar el comportamiento de los elementos estructurales proyectados sometidos a determinadas condiciones previstas.*

Cuando se utilizan programas de cálculo se han de diferenciar claramente los resultados que proporciona el ordenador y el comportamiento que tendrá en la realidad el modelo analizado.

Para poder estar totalmente seguros de que los resultados que proporciona el ordenador coinciden con el verdadero comportamiento de la estructura, es necesario tener una idea clara de lo que significa calcular estructuras por medio de un ordenador.

### 3.2 Modelo Matemático

Cuando los programas de cálculo proporcionan los resultados, no quiere decir que en la realidad la estructura se va a comportar tal y como lo determinan estos.

Lo que se obtiene son unos resultados numéricos producto de unos determinados algoritmos de cálculo (Modelo Matemático) sobre unos datos previamente proporcionados por el usuario del programa que, supone, describen numéricamente la situación del problema a resolver.

Existen diversos modelos matemáticos para el cálculo de estructuras que responden a determinados parámetros y condiciones.

1. *Tipologías de problemas: Los modelos matemáticos están pensados para problemas concretos por tanto cada uno de ellos tiene un ámbito de aplicación definida. Es por consiguiente importante conocer el ámbito de aplicación para poder garantizar la veracidad de los resultados.*
2. *Grado de complejidad del método de cálculo: Frecuentemente es posible escoger entre diversos modelos matemáticos para solucionar el mismo problema esto depende de la complejidad del problema así como de la economía del proceso de cálculo.*

En general se pueden obtener resultados más exactos utilizando programas basados en métodos de cálculo más complejos, por ejemplo el Método de los Elementos finitos<sup>\*</sup>, pero normalmente esto es a costa de emplear más tiempo economía - hombre ya que los datos requeridos para su ejecución son mayores y, que gracias a la informática estos se van reduciendo cada vez más.

El Cálculo de Estructuras por medios de programas informáticos, por consiguiente en una tarea que se realiza en tres niveles, de los cuales, los dos primeros se definen como *preproceso*<sup>†</sup> y el tercero como *postproceso*<sup>‡</sup>.

<sup>\*</sup> Ver capítulo 4

<sup>†</sup> Ver Capítulo 2 preproceso

<sup>‡</sup> Ver Capítulo 2 postproceso

1. *Modelización: Seleccionar un programa que utilice el modelo matemático (Cálculo matricial de barras, Método de los Elementos Finitos) más adecuado al problema a abordar y plantearlo en términos de que el programa sea fácil de usar, así mismo que permita una adecuada y lógica discretización de la estructura.*
2. *Aplicación: Utilización del programa, introduciendo los datos necesarios para describir los elementos a calcular y dando las ordenes para realizar el análisis correspondiente y obtener resultados deseados.*
3. *Interpretación: Recepción, comprensión y evaluación de los datos obtenidos, primeramente para verificar su fiabilidad y después para poder utilizarlos en la definición de los elementos de proyecto.*

Para poder realizar estas operaciones con garantía de que los resultados sean correctos es imprescindible que el usuario tenga los conocimientos necesarios sobre el comportamiento y el cálculo de estructuras, ya que si no es así estos programas pueden convertirse en herramientas muy peligrosas en manos inconscientes.

La facilidad cada vez mayor de utilización de éstos aumenta el riesgo de que su uso sea incorrecto ya que el saber introducir los datos no significa que se podrá asegurar la fiabilidad de los resultados.

### 3.3 Programas para el Cálculo de Estructuras

Existe una gran variedad de programas de cálculo de estructuras, como por ejemplo: *WinEva, Sap90, Sap2000, Portics, Plastic, Calsef, Ansys*, etc. Es posible utilizar diferentes criterios para analizar la base matemática y el ámbito de aplicación de ellos.

También es importante evaluar el grado de facilidad en la comunicación entre el usuario y el ordenador y las ventajas que este da al facilitar las tareas y optimizar los procesos.

### 3.4 Métodos Numéricos

Dentro del cálculo de estructuras los métodos más conocidos son:

- a) Cálculo Elástico de elementos independientes
- b) Método de Cross (poco empleado actualmente)
- c) Cálculo Matricial
- d) Cálculo por medio del Método de los Elementos Finitos
- e) Modelos Mixtos

### 3.5 Modelos de Tipologías de las Estructuras

- a) Modelo bidimensional ortogonal (simplificación muy rígida, la cual solo es apta para casos particulares).
- b) Modelo bidimensional libre (simplificación más flexible, útil siempre que la estructura se pueda descomponer en planos).
- c) Modelo tridimensional ortogonal (modelo más complejo y de resultados más exactos pero con la limitación de la ortogonalidad de los elementos).
- d) Modelo tridimensional libre (es el sistema más flexible y que proporciona resultados más precisos).

### 3.6 Utilización de los Programas de Cálculo

En lo que respecta al cálculo de estructuras es posible utilizar *Genéricos* adaptados a unas necesidades concretas, o bien usar *Programas Específicos*, ya preparados para realizar determinados cálculos de estructuras.

#### 3.6.1 Programas Genéricos

Los programas informáticos genéricos aplicados al cálculo de estructuras están pensados para realizar cualquier tarea que tenga que ver con procesos numéricos repetitivos, aplicados a la automatización del cálculo.

Es evidente que para procesos de análisis complejos es preferible poder disponer de programas ya preparados. Pero en algunos casos, de cálculos sencillos de elementos estructurales, es suficiente poder automatizar las operaciones matemáticas repetitivas. De manera que, una vez establecidos los algoritmos numéricos, en función de parámetros variables, estos puedan ser utilizados cada vez introduciendo solamente los nuevos valores correspondientes a los diferentes parámetros.

Esta automatización de las operaciones puede realizarse básicamente por dos métodos:

##### 3.6.1.1 Procesos de Programación

Se trata de elaborar programas compuestos por una lista secuencial de órdenes que vayan ejecutando las operaciones necesarias para realizar los cálculos que resuelvan un determinado problema.

Una vez preparado, el programa requerirá los valores de los parámetros que definirán la tarea que queremos analizar y este realizará de forma automática el resto de las operaciones.

La dificultad del diseño de estos programas depende de la complejidad del modelo matemático, de los cálculos que haya que realizar y del lenguaje de programación que se este utilizando.

### 3.6.1.2 Hojas de Cálculo

Una hoja de cálculo es, como ya se ha comentado, un programa genérico que permite definir y trabajar sobre una estructura matricial de casillas en donde lo que se ve en cada una de ellas puede ser el texto, números que se han introducido previamente o puede ser una relación (Cálculos o Ecuaciones) con otras casillas.

Así, cuando se dan valores a las casillas que han de contener los datos iniciales, automáticamente aparecen los resultados en las casillas en las que su contenido ha quedado definido como una relación numérica con los valores de otras casillas.

Por medio de estos mecanismos podemos definir y almacenar secuencias de cálculo entendidas como relaciones numéricas entre casillas sucesivas.

Las hojas de cálculo son adecuadas para procesos secuenciales, cálculos tabulares e iterativos, siempre que el número de repeticiones sea conocido previamente. En cambio tienen dificultades para responder a procesos iterativos variables o de simulación con pérdidas de tiempo y duraciones indeterminadas.

Algunos ejemplos de utilización de hojas de cálculo son:

- a) *Cálculo de tensión en barras.*
- b) *Cálculo de deformación de barras.*
- c) *Cálculo de elementos de cimentación y muros de contención.*
- d) *Cálculo de aislamiento térmico de elementos constructivos compuestos.*
- e) *Cálculo de aislamiento acústico de elementos constructivos compuestos.*
- f) *Cálculo simplificado de pequeñas redes de fluidos.*

### 3.6.2 Programas Específicos

Existen, actualmente, en el mercado una gran variedad de programas específicos de cálculo estructuras, orientados a necesidades y problemas diferentes, entre los que es importante poder seleccionar los más convenientes.

Al evaluar cada uno de ellos es importante contemplar algunos parámetros importantes.

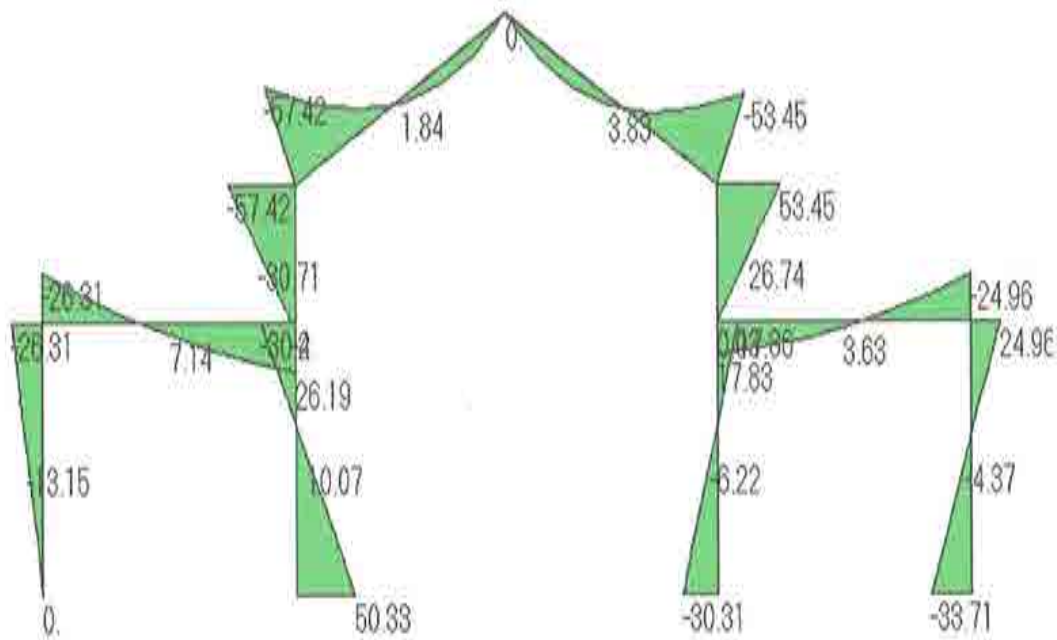
- a) *Modelo o Modelos matemáticos que utiliza y, en consecuencia ámbitos de aplicación así como grado de complejidad y precisión.*
- b) *Facilidad en la utilización (grado de comodidad y eficacia de la interface entre el usuario y el ordenador).*
- c) *Posible compatibilidad entre otras aplicaciones (Posibilidad de transformar información entre el programa de cálculo y otros, por ejemplo: interpretar archivos de dibujo directamente en la ejecución del análisis).*



Es importante conocer algunos programas de Cálculo de Estructuras para analizar que tipo de cálculo realizan y comparar sus interfaces de comunicación con el usuario y las facilidades y la información que estos proporcionan.

- Programa *Portics* de cálculo de pórticos planos; Este programa utiliza el cálculo matricial, pero su *preproceso* es manual y, por consiguiente, lento en su aplicación, sin embargo, es recomendable para estudiantes de los primeros años de formación académica.

- Programa *WinEva* de cálculo de pórticos planos el cual utiliza el cálculo matricial de barras con una interface elemental pero que ya cuenta con una representación gráfica de resultados, así como subrutinas que agilizan el modelado de pórticos ortogonales comunes y crean archivos interactivos.



Momentos (Hipótesis - 1)

Fig. 3.2 Programa WinEva.  
Nave Industrial como pórtico plano



- Programa *Sap90*<sup>8</sup> de cálculo de estructuras más complejas en una, dos y tres dimensiones con la implementación del Método de los Elementos Finitos en análisis lineal, y un *pre* y *postproceso* mucho más completo, además de contar con cálculo sísmico (Este programa será utilizado para la interface que se presenta en esta publicación).

Contiene su propio *pre* y *postproceso*, pero tiene algunas limitaciones dentro de este.

Este programa es actualmente el más utilizado entre los Calculistas, Investigadores y estudiantes debido a que el manejo de este es cómodo.

Pero, siempre se ha de conocer a fondo el programa, los fines de su uso, así como la interpretación de los resultados, pues puede ser peligroso para aquellos que no están familiarizados con los conceptos de Cálculo de estructuras.

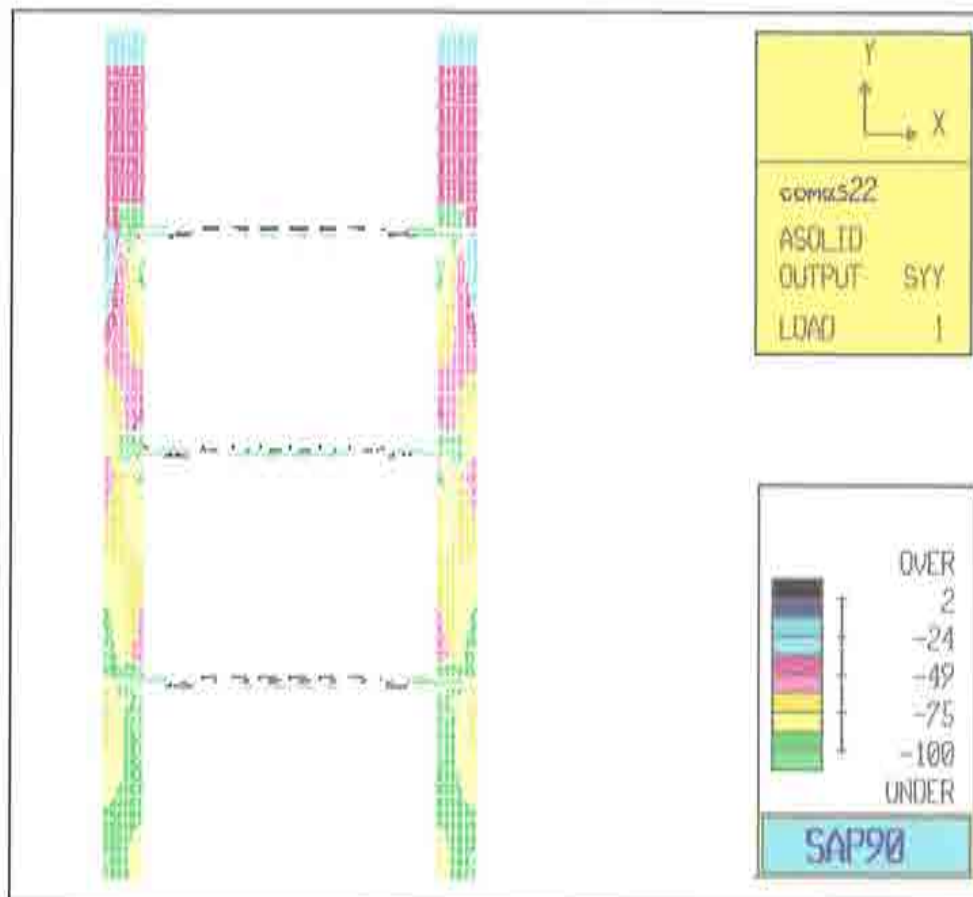


Fig. 3.3 *Sap90*. Cálculo de un Pórtico por tensión plana.

<sup>8</sup> Ver Capítulo 5

- Programa *GiD*\*\* de *pre* y *postproceso* para el Cálculo de estructuras por medio del Método de los Elementos Finitos (Este programa será utilizado para la interfaz que se presenta en esta publicación).

Este programa aparece como una herramienta, muy potente, en el *pre* y *postproceso* del cálculo de estructuras por medio del Método de los Elementos Finitos.

Además implementa programas de dibujo bastante similares al Autocad, lo cual agiliza el uso de este programa.

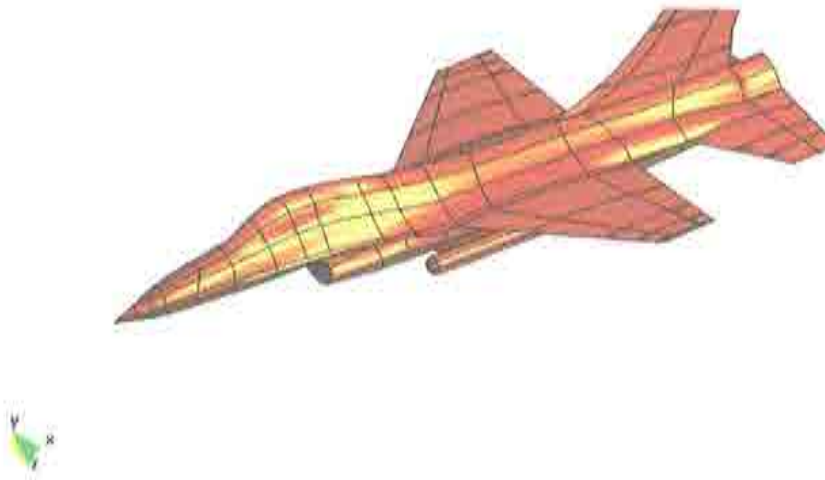


Fig. 3.4† *GiD. Modelización de un Avión*

Al igual que estos programas comerciales, existen muchos más en el mercado.

Como conclusión, se ha de tener en cuenta la enorme evolución que tiene actualmente la informática, así como los programas de cálculo de estructuras que día a día van actualizándose y por consiguiente surgen más con mejores prestaciones.

Por lo cual, es importante que el usuario de éstos, este renovándose al mismo tiempo que la acelerada evolución de la informática.

No hay que olvidar que el uso de estos programas requiere de un conocimiento profundo del comportamiento de las estructuras, sea analítico como geométrico.

\*\* Ver Capítulo 6

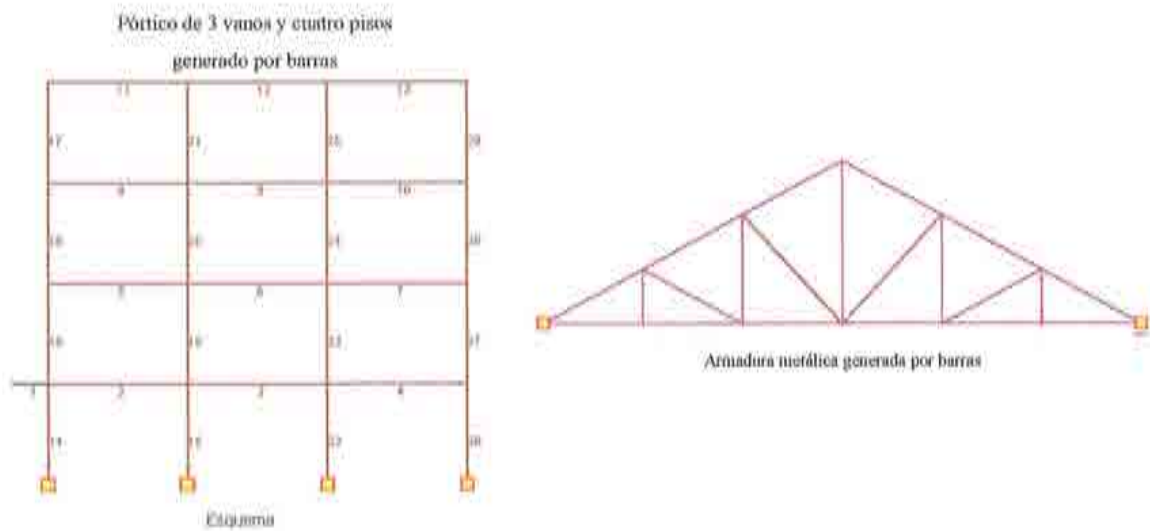
† Imagen del Programa *GiD* del C.I.M.N.E (1998)

## 4 INTRODUCCIÓN AL MÉTODO DE LOS ELEMENTOS FINITOS (M. E. F.)<sup>\*</sup>

### 4.1 Principios del Cálculo Matricial

En diversas ocasiones el Arquitecto se encuentra con el problema de analizar una estructura, siempre y cuando el modelo lo permita, de tipo “malla” compuesta por diferentes elementos acoplados entre sí a través de nudos y sometidos a unas acciones externas.

Ejemplos de este tipo de estructuras (ver figura 4.1), a las cuales se les suele denominar “sistemas discretos”, se pueden encontrar fácilmente en la vida diaria del Arquitecto.



*Fig. 4.1 Sistema discretos*

<sup>\*</sup> Cálculo de Estructuras por el Método de los Elementos Finitos  
Oñate, Eugenio (C.I.M.N.E, 1992)

Se pueden considerar como sistemas discretos, todas las estructuras de barras, tales como los pórticos uni y bidireccionales y los forjados entramados de edificación.

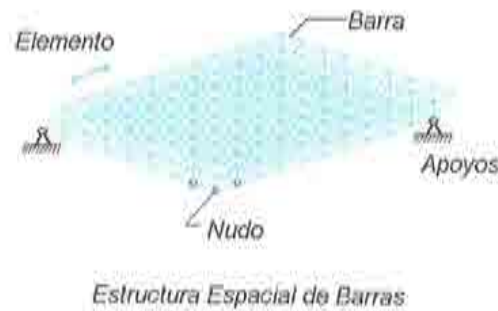


Fig. 4.2 Estructura tridimensional

La mayoría de las estructuras pueden analizarse usando el cálculo matricial el cual, a su vez, guarda una estrecha relación con el Método de los Elementos Finitos. A continuación se exponen las ideas básicas del cálculo matricial de barras, que serán útiles en la introducción a la metodología del análisis de estructuras por medio de M.E.F.

## 4.2 Aplicación al Análisis Matricial de Barras<sup>†</sup>

En general, los métodos de cálculo de estructuras de barras utilizan el análisis matricial de barras. En este capítulo se exponen los conceptos básicos en que se basa.

Las ecuaciones matriciales de una estructura de barras se obtienen a partir del estudio de su equilibrio, respecto a las acciones externas que actúan sobre ella.

Ejemplo Elemental.

a) Versión en Cálculo Directo.

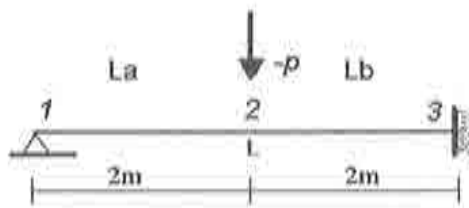
b) Versión en Cálculo Matricial.

a) Versión en Cálculo Directo:

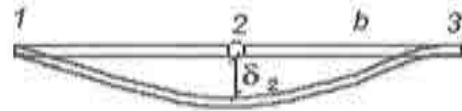
Para esta ejemplo, se necesita plantear el sistema de dos ecuaciones en  $\theta_2$ ,  $\delta_2$  de una forma directa

Primero. Se divide la barra en dos elementos para su análisis, quedando barra a y b.

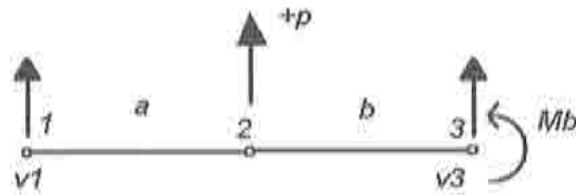
<sup>†</sup> Apuntes de cálculo matricial de barras del Profesor Javier López-Rey Laurens Catedrático de E.T.S.A.B. 1997-98



$$La + Lb = L = 2m$$



Deformada probable



Barra empotrada apoyada con una carga puntual al centro

Criterio de signos :



Regla del sacacorchos

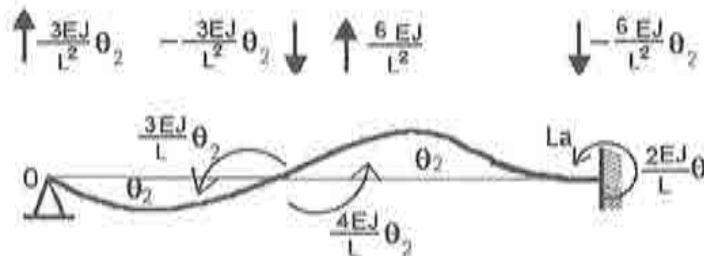
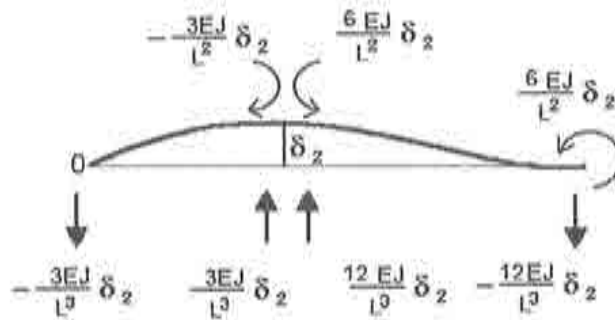
Concavidad



Nudo activo.

Omisión de las ecuaciones de apoyo.

No hay alargamientos ni acortamientos.



$$\Sigma_2 V = 0 \rightarrow +P + \frac{3EJ}{L^3} \delta_2 + \frac{12EJ}{L^3} \delta_2 - \frac{3EJ}{L^2} \theta_2 + \frac{6EJ}{L^2} \theta_2 = 0$$

$$-P = \frac{EJ}{L^3} [3+12] \delta_2 + \frac{EJ}{L^2} [6-3] \theta_2 \rightarrow \boxed{-P = \frac{15EJ}{L} \delta_2 + \frac{3EJ}{L} \theta_2}$$

$$\Sigma_2 M = 0 \quad 0 + \frac{6EJ}{L^2} \delta_2 - \frac{3EJ}{L^2} \delta_2 + \frac{3EJ}{L} \theta_2 + \frac{4EJ}{L} \theta_2 = 0$$

$$0 = \frac{EJ}{L^2} [6-3] \delta_2 + \frac{EJ}{L} [3+4] \theta_2 = 0 \rightarrow \boxed{0 = \frac{3EJ}{L} \delta_2 + \frac{7EJ}{L} \theta_2}$$

La solución de este sistema de dos ecuaciones es:

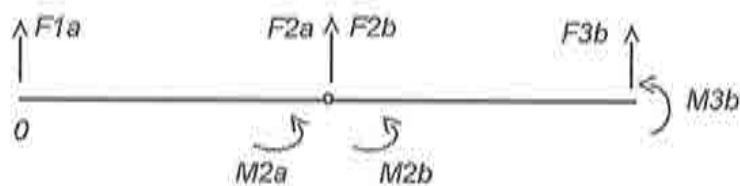
$$\delta_2 = -\frac{7}{12} \frac{P}{EJ}$$

$$\theta_2 = \frac{P}{8EJ}$$

El sistema de ecuaciones de los recuadros puede escribirse de forma matricial. De esta forma

$$\rightarrow \begin{matrix} \begin{matrix} -P \\ 0 \end{matrix} \\ =EJ \end{matrix} \begin{matrix} \begin{matrix} 15/18 & 3/4 \\ 3/4 & 7/2 \end{matrix} \end{matrix} \begin{matrix} \begin{matrix} \delta_2 \\ \theta_2 \end{matrix} \end{matrix}$$

#### b) Versión en Cálculo Matricial



*Ecuaciones de equilibrio en el nudo 2*

$$\Sigma_2 V = 0 \rightarrow P + F_a + F_b + = 0$$

$$F_a + F_b + P = 0$$

$$\Sigma_2 M = 0 \rightarrow 0 + M_a + M_b = 0$$

$$M_a + M_b = 0$$



Matriz de Rigidez de la  
barra a

$$\begin{Bmatrix} F_i \\ F_d \\ M_d \end{Bmatrix} = EJ \begin{bmatrix} 3/L^3 & -3/L^3 & 3/L^2 \\ -3/L^3 & 3/L^3 & -3/L^2 \\ 3/L^2 & -3/L^2 & 3/L \end{bmatrix} \begin{Bmatrix} \delta_i \\ \delta_d \\ \theta_d \end{Bmatrix}$$

Matriz de Rigidez de la  
barra b

$$\begin{Bmatrix} F_i \\ M_i \\ F_d \\ M_d \end{Bmatrix} = EJ \begin{bmatrix} 12/L^3 & 6/L^2 & -12/L^3 & 6/L^2 \\ 6/L^2 & 4/L & -6/L^2 & 2/L \\ -12/L^3 & -6/L^2 & 12/L^3 & -6/L^2 \\ 6/L^2 & 2/L & -6/L^2 & 4/L \end{bmatrix} \begin{Bmatrix} \delta_i \\ \theta_i \\ \delta_d \\ \theta_d \end{Bmatrix}$$

El siguiente paso es: La particularización ó compatibilidad de movimientos, de los nudos comunes de las barras.

Particularización = Compatibilidad de Movimiento

barra a

$$\begin{Bmatrix} F_{1a} \\ F_{2a} \\ M_{2a} \end{Bmatrix} = EJ \begin{bmatrix} 3/8 & -3/8 & 3/4 \\ -3/8 & 3/8 & -3/4 \\ 3/4 & -3/4 & 3/2 \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \theta_2 \end{Bmatrix} = 0$$

barra b

$$\begin{Bmatrix} F_{2b} \\ M_{2b} \\ F_{3b} \\ M_{3b} \end{Bmatrix} = EJ \begin{bmatrix} 12/8 & 6/4 & -12/8 & 6/4 \\ 6/4 & 4/2 & -6/4 & 2/2 \\ -12/8 & -6/4 & 12/8 & -6/4 \\ 6/4 & 2/2 & -6/4 & 4/2 \end{bmatrix} \begin{Bmatrix} \delta_2 \\ \theta_2 \\ \delta_3 \\ \theta_3 \end{Bmatrix} = 0$$

Como para la resolución estricta del sistema de ecuaciones se puede prescindir de las ecuaciones de apoyo, se montara el sistema estricto que depende de  $\theta_2$  y  $\delta_2$  exclusivamente.

Ensamblaje = Ecuaciones de Equilibrio

$$\begin{Bmatrix} F_{2a} \\ + \\ F_{2b} \\ M_{2a} \\ + \\ M_{3b} \end{Bmatrix} = EJ \begin{bmatrix} +3/8 & -3/4 & \delta_2 \\ +12/8 & +6/4 & \theta_2 \\ -3/4 & +3/2 & \\ +6/4 & +4/2 & \end{bmatrix} \begin{Bmatrix} \delta_2 \\ \theta_2 \\ 0 \end{Bmatrix} \rightarrow -P = EJ \begin{bmatrix} 15/18 & 3/4 & \delta_2 \\ 3/4 & 7/2 & \theta_2 \end{bmatrix}$$

Con lo cual llegamos a la misma ecuación matricial, que en el caso del sistema directo de solución:

$$\delta_2 = -\frac{7P}{12EJ} \quad \text{y} \quad \theta_2 = \frac{P}{8EJ}$$



### Obtención de los Parámetros Restantes

#### Proceso de restitución

$$\theta_1 = (\delta_2 - \delta_1) \frac{1.5}{L} - \frac{\theta_2}{2} = -\frac{P}{2EJ}$$

$$F_{1a} = -\frac{3}{8} \times 0 - \frac{3}{8} EJ \times \left(-\frac{7}{12} \frac{P}{EJ}\right) + \frac{3}{4} EJ \frac{P}{8EJ} = \frac{24}{96} P + \frac{3}{32} P = \frac{30}{96} P = \frac{5}{16} P = V_1$$

$$F_{2a} = EJ \left(-\frac{3}{4} \frac{P}{8EJ} - \frac{3}{8} \times \frac{7}{12} \frac{P}{EJ}\right) = -\frac{5}{16} P$$

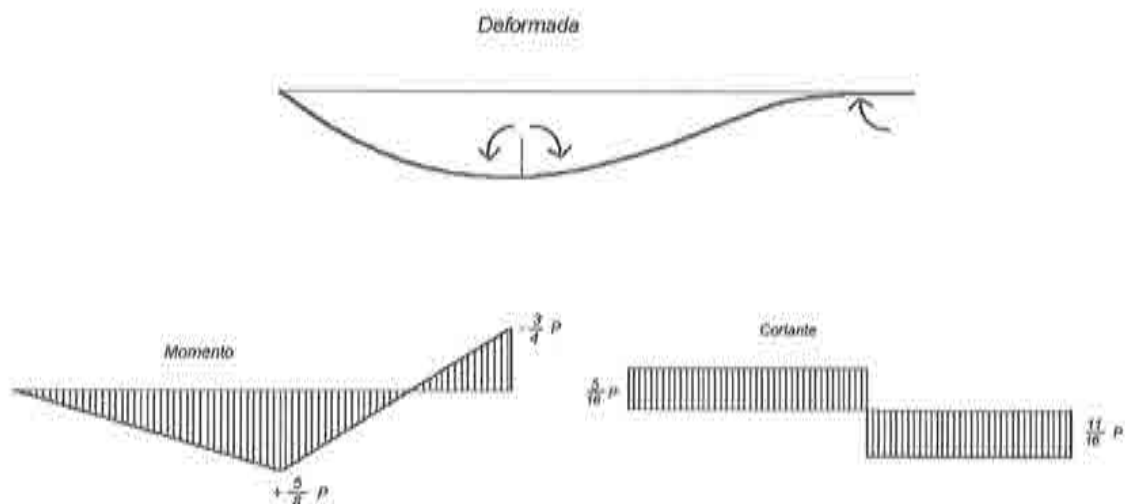
$$M_{2a} = EJ \left(\frac{3}{2} \frac{P}{8EJ} + \frac{3.7}{12.4} \frac{P}{EJ}\right) = P \left(+\frac{9}{48} + \frac{21}{48}\right) = \frac{5}{8} P$$

$$F_{2b} = EJ \left(\frac{6}{4} \frac{P}{8EJ} - \frac{12}{8} \times \frac{7}{12} \frac{P}{EJ}\right) = P \left(\frac{6}{32} - \frac{84}{96}\right) = -\frac{11}{16} P$$

$$M_{2b} = EJ \left(\frac{4}{2} \frac{P}{8EJ} - \frac{6}{4} \times \frac{7}{12} \frac{P}{EJ}\right) = P \left(\frac{12}{48} - \frac{42}{48}\right) = -\frac{5}{8} P$$

$$F_{3b} = -F_{2b} = \frac{11}{16} P$$

$$M_{3b} = EJ \left(\frac{2}{2} \times \frac{P}{8EJ} - \frac{6}{4} \times \frac{7}{12} \frac{P}{EJ}\right) = P \left(\frac{2}{16} - \frac{42}{48}\right) = -\frac{3}{4} P$$



Representación Gráfica de los Resultados del ejemplo

### 4.3 Conceptos básicos del Método de los Elementos Finitos

Dentro del cálculo estructural, las estructuras de barras se pueden expresar con un número mínimo de ecuaciones como es el caso del cálculo matricial, pero en la realidad la mayoría de las estructuras tienen un comportamiento continuo por lo que éste no puede ser expresado con ese mínimo de ecuaciones.

Para poder realizar el análisis más aproximado de estas estructuras es necesaria la integración de las ecuaciones diferenciales de equilibrio de cada una de las partes que forman la estructura.

Dentro de las estructuras continuas más comunes en arquitectura, podemos encontrar muros de contención, depósitos de agua, cúpulas, forjados planos, etc.

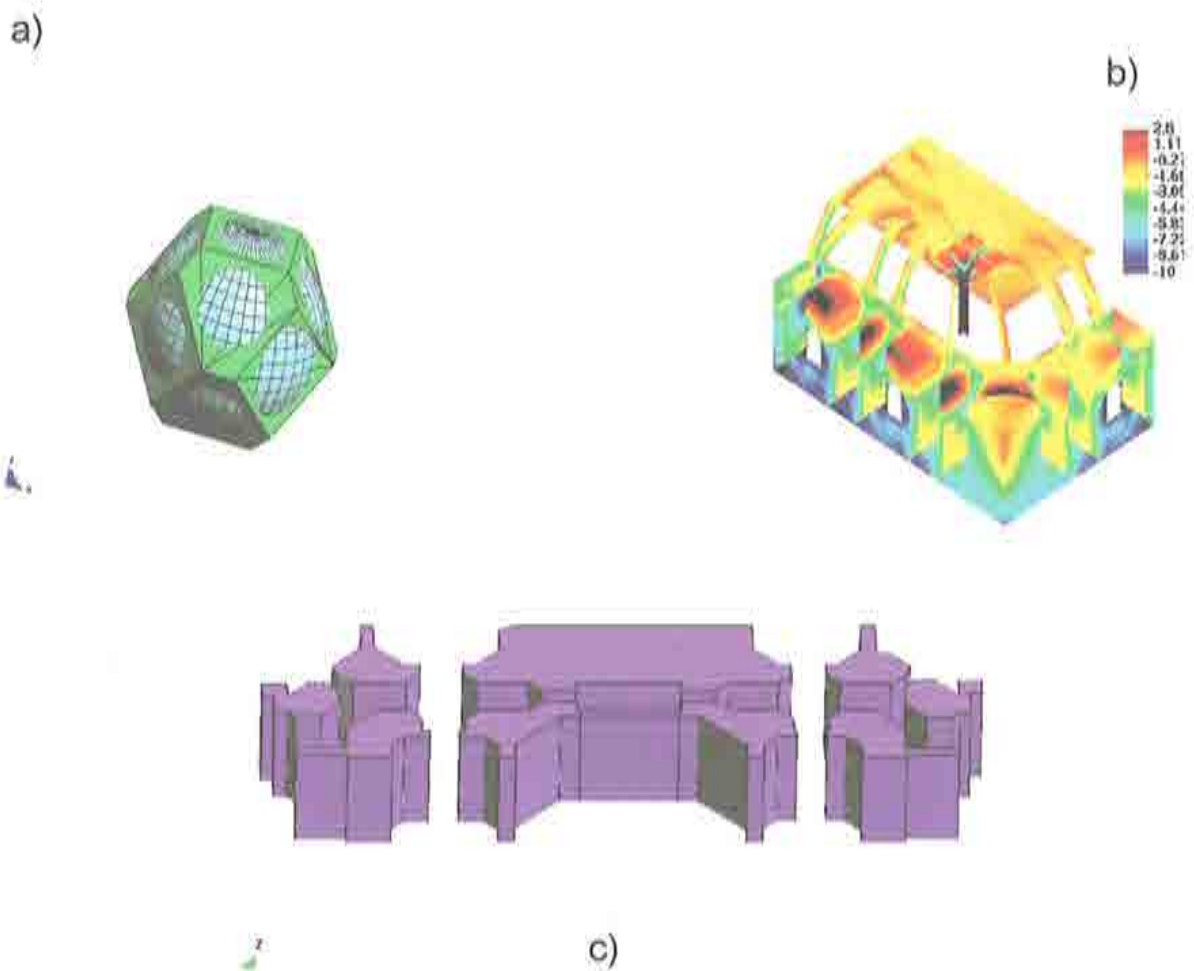


Fig. 4.3 Estructuras continuas:  
 a) Galena b) Casa Bellesguard (Gaudí)<sup>†</sup> c) Esquema de un trozo de la planta de la Sagrada Família

<sup>†</sup> Modelización y Análisis a cargo del equipo de trabajo del Profesor Javier Lopez-Rey

A pesar de que todas las estructuras continuas son tridimensionales existen algunos casos en los cuales se pueden describir sus comportamientos por medio de modelos unidimensionales (tal es el caso de las estructuras de barras), bidimensionales (elasticidad bidimensional, teoría de placas) o sólidos de revolución.

Actualmente, el Método de los Elementos Finitos, es una de las herramientas más potentes para el cálculo de estructuras uni, bi, o tridimensional.

Su empleo, nos permite conocer un comportamiento muy aproximado de las estructuras.

Cualquier usuario de este método puede entenderlo mucho mejor, siempre y cuando tenga unos mínimos conocimientos básicos tanto de cálculo matricial de barras, informática y programación, así como de la teoría de resistencia de materiales.

Todas y cada una de estas herramientas y teorías se integran en el conocimiento del M.E.F.

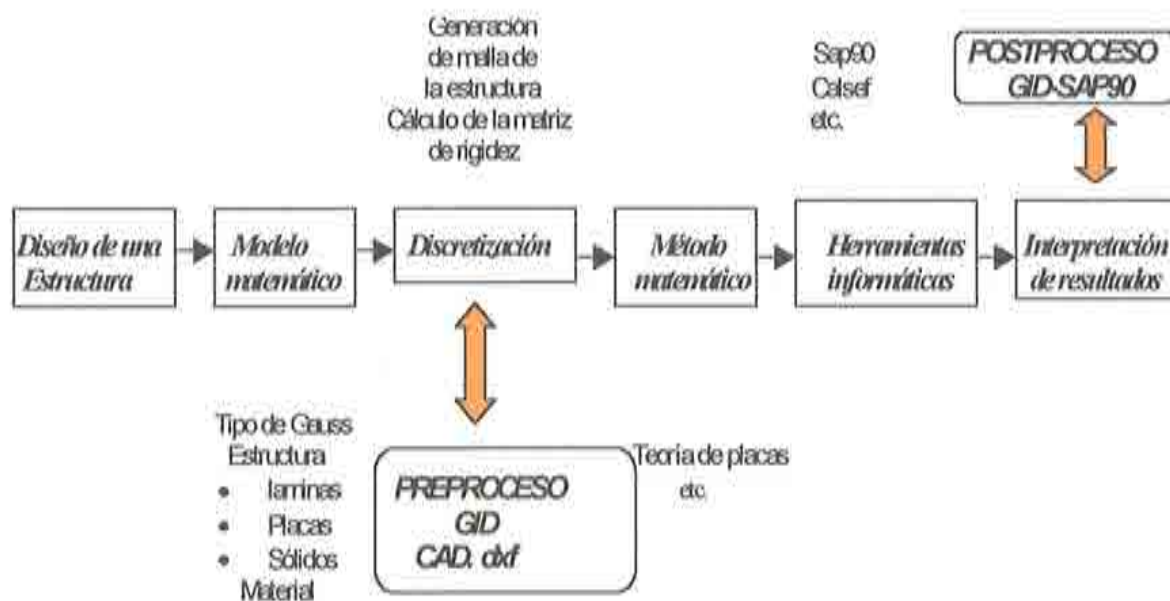
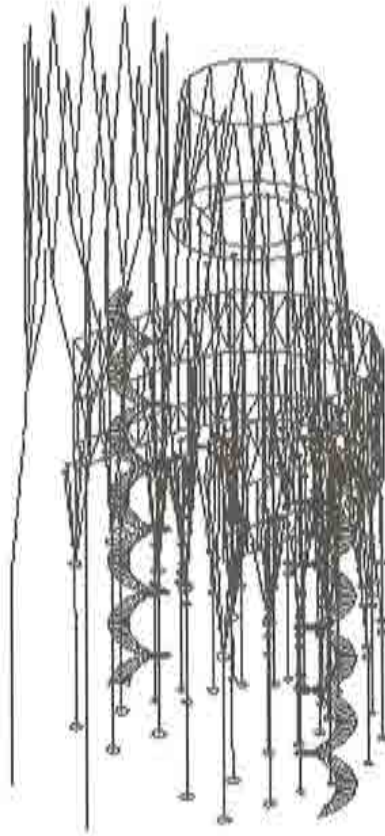
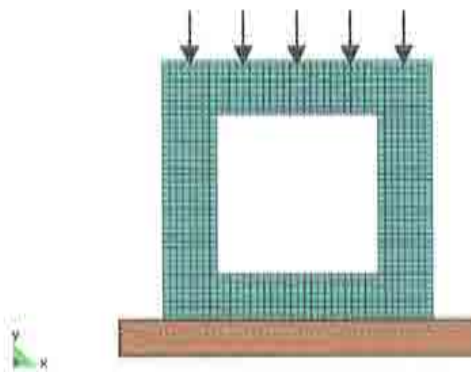


Fig. 4.4 Diagrama del análisis de estructuras por el Método de los Elementos Finitos



*Fig. 4.5 Abside de la Sagrada Família,  
Estructura de Barras*

Ahora consideraremos un ejemplo práctico de las fases de análisis que se realizan, para el cálculo del túnel de la figura 4.6, utilizando el M.E.F.; y veremos su analogía con el cálculo matricial.



*Fig. 4.6 Análisis de un túnel por el M.E.F.*

*Fase 1:* Consideremos un túnel de metro. A partir de aquí es importante seleccionar un modelo a utilizar de acuerdo con el problema.

En este caso, discurriremos la teoría de la elasticidad bi y tridimensional, pero consideraremos la teoría de la elasticidad bidimensional (deformación plana) ya que estamos ante una estructura prismática en la cual una de sus dimensiones (espesor) es mucho mayor que las otras dos y, sobre ella actúan únicamente cargas uniformemente distribuidas a lo largo de toda su longitud y contenidas en planos ortogonales al eje que une los centros de gravedad de sus distintas secciones transversales.

También hay que definir las propiedades mecánicas del material, tipos de cargas, condiciones de apoyo, etc.

*Fase 2:* Teniendo ya el modelo matemático seleccionado, se procede a la discretización de la estructura en porciones llamadas elementos, los cuales están conectados entre sí por nodos (puntos del elemento) toda este conjunto de elementos y nodos se llaman malla.

La malla de elementos finitos, puede estar constituida por elementos de diferente geometría. La fase de discretización es la más importante dentro del *preproceso*.

*Fase 3:* Se obtiene la matriz de rigidez de la estructura que en este caso es más compleja que en estructuras de barra al intervenir integrales sobre el dominio uni, bi o tridimensional.

*Fase 4:* Se realiza el ensamblaje de las matrices de rigidez “k” y el vector de carga “f”.

*Fase 5:* Se resuelve un sistema de ecuaciones de la forma  $Ka = f$  donde la incógnita es “a” (movimiento de los nodos).

*Fase 6:* Una vez calculados los movimientos, se pueden deducir las deformaciones y a su vez las tensiones de los elementos.

*Fase 7:* Para poder resolver de las fases 3 a la 6, es necesario la implementación de un programa informático, ya sea comercial, o de uno realizado en particular para este problema.

*Fase 8:* A partir de la obtención de los resultados numéricos, la fase siguiente es la interpretación gráfica de estos. Para eso es necesario contar con un programa de interpretación gráfica (*postproceso*).

*Fase 9:* Una vez que se estudian los resultados, el usuario deberá decidir si son los óptimos que esperaba, de no ser así, tendrá que regresar a la fase 3 para una nueva discretización de la estructura.

Estas fases se muestran en el diagrama de la figura 4.5

En resumen, estas son las fases que emplea el Método de los Elementos Finitos en el análisis y cálculo de estructuras.

## 5 SAP90 (STRUCTURAL ANALYSIS PROGRAM)\*

### 5.1 Introducción:

El programa *Sap90* aparece como una nueva alternativa en el cálculo de estructuras por medio del método de los elementos finitos, ya que contiene una serie de bloques que permiten realizar un *pre* y un *postproceso* del cálculo.

Este programa surge de la investigación realizada durante varios años en la Universidad de California, Berkeley (1970).

En los años siguientes se continuó la investigación en el área de la formulación de los elementos finitos y las resoluciones de las técnicas numéricas, de la cual surgieron una serie de versiones: *Sap*, *Sap3*, *Solidsap*, *Sap IV*, *Sap 80*, precursores del actual *Sap90*.

El programa ha adquirido un nivel de aceptación muy bueno por parte de los profesionales que trabajan en el campo del análisis estructural.

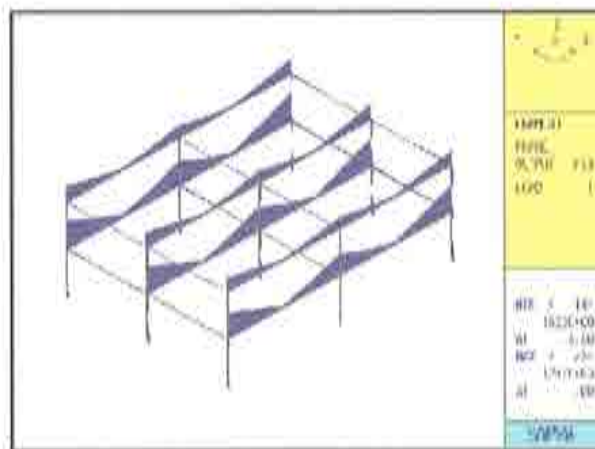


Fig. 5.1 Análisis de un pórtico en 3 dimensiones

\* Sap90 ETABS SAFE  
Computer Software for Structural & Earthquake Developed and Written in U. S. A.



## 5.2 Los programas Sap80 y Sap90

El programa *Sap IV* fue creado hace casi veinte años. Desde entonces se han producido grandes avances en el campo del análisis numérico, mecánica estructural y tecnología de las computadoras.

Dichos avances contribuyeron al desarrollo de *Sap80*, el primer programa de análisis estructural para ordenadores personales, y más recientemente al del *Sap90*.

*Sap90*, representa una nueva tecnología y fue programado por el autor original de la serie de programas *Sap*. El programa no es una modificación o adaptación de *Sap IV*.

La formulación de los elementos y la resolución de las ecuaciones son completamente nuevas.

Todos los datos son introducidos en listas libres de formato. Las opciones de generación están disponibles según la conveniencia del usuario.

Existe la posibilidad de imprimir la geometría deformada y no deformada para la verificación del modelo estructural.

La biblioteca de elementos finitos consiste de cuatro elementos, nominalmente: un elemento barra tridimensional, un elemento tridimensional laminar, un elemento bidimensional de comportamiento plano y un elemento sólido tridimensional.

Las barras bidimensionales, vigas, membranas, láminas a pandeo y los elementos planos y de simetría axial constituyen un subgrupo de dichos elementos.

Todas las opciones geométricas y de carga necesarias asociadas a los elementos han sido incorporadas. Se incluye también un elemento límite en forma de apoyo elástico.

No hay restricciones en la mezcla o combinación de tipos de elementos dentro de un modelo particular.

Las opciones de carga, que incluyen las gravitatorias, térmicas y condiciones de pretensado, se unen a las cargas nodales usuales con fuerzas y desplazamientos especificados. Las cargas dinámicas pueden estar en forma de una base de espectro de respuesta de aceleraciones, o cargas variables con el tiempo y base de aceleraciones.

### 5.2.1 Precauciones al usar Sap90

La aplicación efectiva de un programa, para el análisis de situaciones prácticas, implica tener una considerable experiencia. La fase más difícil del análisis es juntar un modelo que recoja la mayoría de las características del comportamiento de la estructura.

Ninguna computadora puede sustituir el juicio de un ingeniero con experiencia. Se suele decir, acertadamente, que un ingeniero inútil es incapaz de hacer con toneladas de soluciones calculadas por ordenador, lo que un buen ingeniero en el dorso de una carpeta.

La correcta interpretación de las soluciones es tan importante como la preparación de un buen modelo estructural. La verificación de resultados inesperados requiere una buena asimilación de las hipótesis y mecánica básicas del programa.

Las comprobaciones de equilibrio son necesarias no sólo para verificar los resultados, sino también para entender el comportamiento estructural básico.

De hecho, en 1970, la publicación original de *Sap* incluía la siguiente nota:

*"El nombre vulgar de Sap fue elegido para recordar al usuario que este programa, como todos los programas informáticos, carecen de inteligencia. Es tarea del ingeniero idealizar la estructura correctamente y asumir la responsabilidad de los resultados."*

El nombre *Sap* se ha mantenido para este programa exactamente por la misma razón.

### 5.3 Sus alcances y limitaciones

El desarrollo del programa se lleva a cabo en el entorno *ANSI Fortran-77*, lo cual garantiza la compatibilidad tanto con los pequeños ordenadores personales como con las grandes supercomputadoras.

Esta versión del programa fue diseñada para ser usada con el sistema operativo *MS-DOS*. En computadoras con 640K de memoria y 30 MB de disco duro, la capacidad de resolución es de 4.000 nodos (u 8.000 ecuaciones).

Con un disco duro mayor, y utilizando memoria extendida (más allá de 640K), se pueden resolver cálculos muy grandes. Todas las operaciones numéricas son ejecutadas en doble precisión completa de 64-bit.

El programa contiene opciones de análisis estático y dinámico. Dichas opciones pueden ser activadas, de manera simultánea, en la misma ejecución. La combinación de cargas puede, por tanto, incluir resultados del análisis estático y dinámico.

### 5.4 Librerías

Las librerías que contiene *Sap90* consisten en una serie de módulos de programas, (Subrutinas) de las cuales, cuatro definen los modelos matemáticos: *FRAME*, elementos bi y tridimensional (barras), *SHELL*, elementos bi y tridimensional (placa, lámina o membrana), *ASOLID* elementos bidimensionales (deformación y tensión plana) y, *SOLID*, elementos en tres dimensiones (elasticidad tridimensional).

Además de estos módulos, existen otros más que complementan el cálculo de las estructuras, ya sean como pre y post-proceso o como análisis dinámico.

#### 5.4.1. Los módulos de Sap90

El programa *Sap90* es en realidad una combinación de módulos de programa que son ejecutados en una secuencia predefinida. Los módulos, se encadenan a través de una serie de archivos internos que constituyen una base de datos. Cada uno de ellos, realiza una serie de operaciones y actualizaciones durante el proceso. Los archivos internos tienen el mismo nombre que los archivos de entrada de datos, pero, con diferentes extensiones.

En la siguiente tabla se detallan los módulos del programa, incluidos en el paquete completo de esta versión de *Sap90*, y las funciones asociadas de cada módulo. Se enumeran también los nombres de los archivos de salida asociados.

MODULO DE PROGRAMA	FUNCIÓN DEL MÓDULO	ARCHIVO DE SALIDA CREADO
1. Sap90.EXE	Lee, comprueba y tabula todos los datos de entrada.	EXAMPLE.SAP
2. OPTIMIZE.EXE	Optimiza el número de ecuaciones.	EXAMPLE.EQN
3. FRAME.EXE	Formulación de la matriz de rigidez de un elemento barra.	
4. SHELL.EXE	Formulación de la matriz de rigidez de un elementos lámina.	
5. ASOLID.EXE	Formulación de la matriz de rigidez de elementos en comportamiento plano.	
6. SOLID.EXE	Formulación de la matriz de rigidez de elementos tridimensionales.	
7. SOLVE.EXE	Matriz de rigidez y carga.	
8. EIGEN.EXE	Análisis de valores de <i>Eigen</i> .	EXAMPLE.EIG
9. RITZ.EXE	Análisis de vectores de <i>Ritz</i>	EXAMPLE.RIT
10. SPEC.EXE	Análisis de espectro de reacciones.	EXAMPLE.SPC
11. TIMEH.EXE	Análisis de la respuesta temporal.	
12. JOINTF.EXE	Salida de modos, desplazamiento de nodos y reacciones.	EXAMPLE.SOL
13. ELEMFE.EXE	Salida de fuerzas en los elementos nodales.	EXAMPLE.FEF
14. FRAMEFE.EXE	Salida de fuerzas en los elementos barra.	EXAMPLE.F3F
15. SHELLEFE.EXE	Salida de fuerzas en los elementos lámina.	EXAMPLE.F4F
16. ASOLIDFE.EXE	Salida de cargas en los elementos de comportamiento plano.	EXAMPLE.F5F
17. SOLIDFE.EXE	Salida de cargas en elementos tridimensionales.	EXAMPLE.F8F
18. SAPLOT.EXE	Dispositivos gráficos.	
19. SAPTIME	Dispositivos gráficos y creación de salida desde el análisis de la respuesta temporal.	EXAMPLE.HST

*Sap90* dispone además de algunos módulos de programa adicionales, utilizados para el análisis de puentes y análisis de transmisión de calor.

NOTA: Los nombres de algunos módulos del programa *Sap*, son los mismos que los nombres de bloques de entrada de datos de *Sap90*. No permita que esto se convierta en una fuente de confusión.

## 5.5 Preproceso y Postproceso

Este programa contiene un *preproceso*, que permite discretizar una estructura. No obstante, presenta el inconveniente de que sólo lo hace mediante elementos rectangulares (de 4 nodos coplanares) y en figuras regulares. Se puede decir, que dicho *preproceso* es demasiado limitado si consideramos estructuras tridimensionales complejas

### 5.5.1 Preproceso del Sap90

*Sap90* es un programa de cálculo elástico lineal de estructuras, por el método de los elementos finitos, que incluye postprocesadores gráficos para la presentación de resultados.

La preparación de datos para un problema de análisis estructura comprende básicamente:

1. *La descripción de la geometría estructural y de los materiales.*
2. *La definición de los estados de carga para los cuales la estructura precisa ser analizada.*

La correcta interpretación de los resultados es tan importante como la preparación (*preproceso*) de un buen modelo estructural

Las instrucciones originales del programa advierten al usuario que "este programa, como cualquier otro no posee inteligencia. Es obligación del calculista modelizar la estructura correctamente y asumir la responsabilidad de los resultados."

Dentro del *preproceso* existe un bloque llamado Sap90.EXE, el cual revisa los datos de entrada, tales como: numeración de nodos, restricciones de contorno, conexión de los nodos en los elementos, así como otros datos generales y de carga.

#### 5.5.1.1 Organización del fichero de datos

Los datos del problema se organizan en secciones, cada una de las cuales consta de una cabecera y una o varias líneas de datos, en las que se definen los diversos aspectos de la estructura:

(Sin cabecera)	Título
SYSTEM	Datos generales de control del cálculo
JOINTS	Coordenadas nodales
RESTRAINTS	Coacciones rígidas nodales – Soportes

<b>SHELL</b>	Definición y cargas de los elementos lámina, ó
<b>ASOLID</b>	Definición y cargas de los elementos de elasticidad bidimensional
<b>LOADS</b>	Cargas nodales

### Formato libre

Una línea típica de datos numéricos se compone de la siguiente manera:

N 1, N2, N3,... A= A1, A2, A3, ... B= B1, B2, B3,.....

Un bloque de datos numéricos sin identificación, como N1,N2,N3,... debe ser la primera información en la línea.

Un bloque de datos de la forma A= A1, A2, A3... puede estar en cualquier orden o posición en la línea. Los ficheros de datos *Sap90* se preparan en formato libre, ya que los bloques de datos numéricos se identifican por "A=" de manera unívoca.

### Modelización de la estructura

La estructura se define dentro de un sistema tridimensional de coordenadas cartesianas, denominando sistema global de referencia, en el cual queda descrita toda la geometría del modelo.

Los datos relacionados con las características de los elementos se definen, sin embargo, respecto a un sistema local, propio de cada elemento.

#### 5.5.1.2 Fichero de datos Sap90

##### Registro del título

La primera línea del fichero de datos, de la columna 1 a la 70, se imprimirá en la cabecera de cada página del fichero de resultados. El título debería contener la identificación del problema.

##### SYSTEM. Datos generales de control de cálculo

Este bloque de datos define la información de control de cálculo asociada con el análisis estructural.

Formato reducido:

**SYSTEM**

L= NÚMERO DE HIPÓTESIS DE CARGA
---------------------------------

**JOINTS. Nodos - coordenadas nodales.**

Las coordenadas nodales se definen en esta sección respecto al sistema global de referencia. Para simplificar la introducción de coordenadas se han previsto cinco opciones de generación de nodos que se describen en la página siguiente.

Dejar un espacio en blanco al final de este bloque.

Formato:

**JOINTS****OPCIONES DE GENERACIÓN DE NODOS G, Q, F, L, A**

jid	X= x Y= y Z= z	( G = g1, g2, i, r ) ( Q = q1, q2, q3, q4, in, jn ) ( F = f, ni, nj, in, jn )
		( L = l, ni, nj ) ( A = C1, C2, C3, nc, ic, a ) ( S = s )
[ vp ] [ vp ] [ valor previo ]		VALORES POR DEFECTO

COORDENADAS NODALES  
RESPECTO AL SISTEMA  
GLOBAL DE REFERENCIA

Nº ASIGNADO AL NODO

**JOINTS****5.5.1.3 Opciones de generación de nodos G, Q, F, A****Generación lineal unidimensional G**

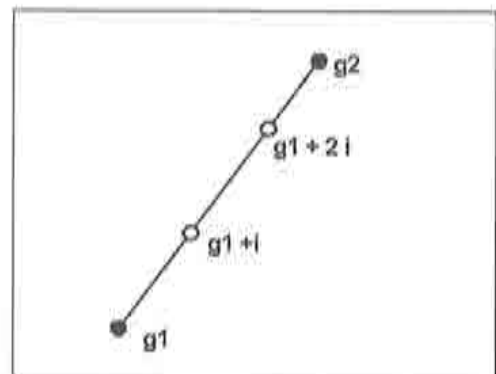
$G = g1, g2, i, (r)$

Para intervalos desiguales en progresión aritmética: Relación entre la longitud del último incremento respecto a la del primero.

Incremento en la numeración de los nodos

Nº del último nodo de la generación lineal

Nº del primer nodo de la generación lineal



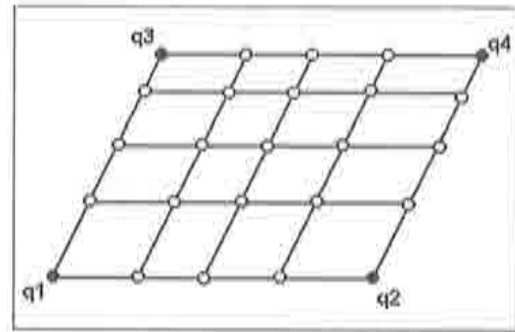


**Generación cuadrilateral bidimensional Q**

$Q = q1, q2, q3, q4, in, jn$

Nº de los nodos de los vértices de la parrilla

Incremento en la numeración de los nodos en la dirección j  
Incremento en la numeración de los nodos en la dirección i

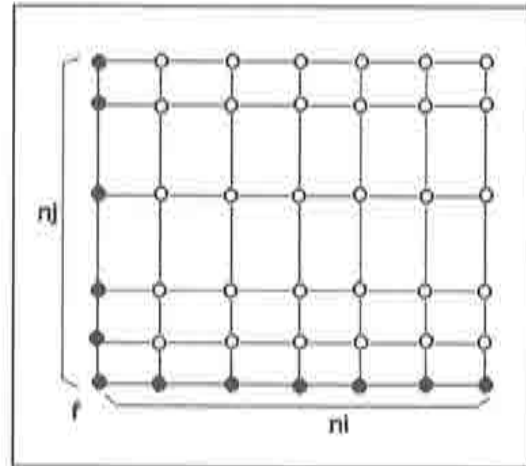


● NODOS DEFINIDOS  
○ NODOS GENERADOS

**Generación frontal F**

$F = f, ni, nj, in, jn$

Incremento en la numeración de los nodos en la dirección j  
Incremento en la numeración de los nodos en la dirección i  
Número de nodos en la dirección j, sin incluir el nodo de origen.  
Número de nodos en la dirección i, sin incluir el nodo de origen.  
Nº del nodo de origen para la generación frontal

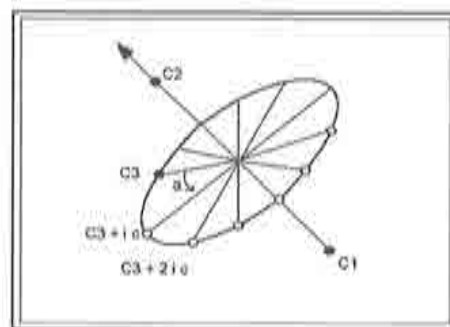


**Generación cilíndrica o axial A**

$A = C1, C2, C3, na, ic, a$

Incremento angular en grados (debe ser menor de 90).  
Incremento en la numeración de los nodos.  
Número de nodos adicionales a generar.

Nº de los nodos de la generación cilíndrica.  
C1, C2 Definen la dirección positiva M eje de referencia  
C3 nodo que marca el origen de la generación.



### 5.5.1.4 RESTRAINTS. Coacciones rígidas – soportes

Es el modo de impedir el movimiento de un nodo según uno o varios grados de libertad. El programa calcula las reacciones necesarias en dichos nodos, para mantener nulos sus movimientos.

Cada nodo posee seis grados de libertad, tres desplazamientos UX, UY y UZ, y tres giros RX, RY y RZ, todos ellos definidos respecto al sistema global de referencia. Si en una estructura existen grados de libertad cuya rigidez directa es nula, como por ejemplo los movimientos normales al plano de una estructura de elasticidad bidimensional, debe anularse el movimiento en esas direcciones como si se tratara de una coacción rígida.

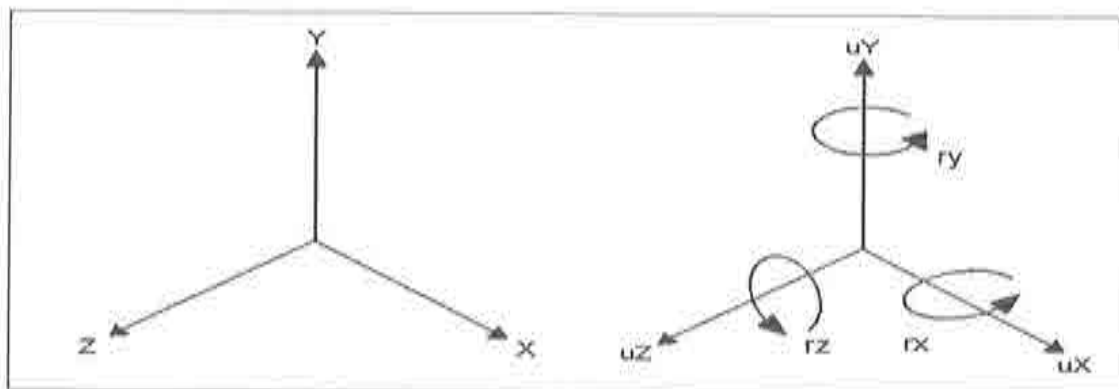
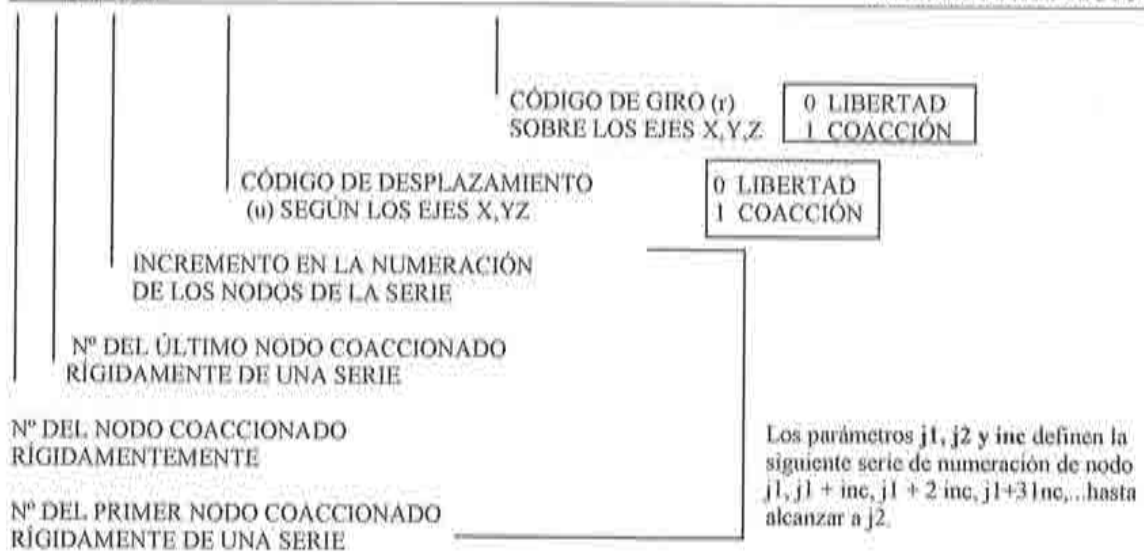
Formato:

#### RESTRAINTS

j1 j2 inc R = rux, ruy, ruz, rrx, rry, rrz

[j1] [j2]

VALORES POR DEFECTO



Criterio de signos positivos para los (Restraints), desplazamientos (U) y giros (R)

### 5.5.1.5 SHELL. Elemento Lámina

Este bloque de datos define las propiedades, la descripción geométrica y las acciones asociadas a un elemento de "cáscara". El elemento *shell* es un elemento laminar plano de cuatro nodos apto para modelizar distintos tipos de estructuras de 2 y 3 dimensiones. A cada nodo le corresponden 6 grados de libertad.

Recuerde que necesario dejar un espacio en blanco al final de cada bloque de datos.

Formato reducido:

#### SHELL

NM= NÚMERO DE MATERIALES	X= x1, x2, ..., xnld	Y= y1, y2, ..., ynld	Z= z1, z2, ..., znld
--------------------------	----------------------	----------------------	----------------------

..... COEFICIENTES GRAVITACIONALES SEGÚN LAS DIRECCIONES DE LOS EJES X, Y, Z PARA CADA UNA DE LAS HIPÓTESIS DE CARGA 1,2,...,NLD (Number of load condition)  
COEFICIENTES PARA LA APLICACION DE PESO PROPIO.

Pueden aplicarse fuerzas de gravedad en las 5 direcciones globales X, Y y Z. Estas cargas resultan de calcular el peso propio de un elemento como el producto del volumen del elemento por el peso específico del material correspondiente. El peso del elemento se distribuye hacia los nodos del elemento después de multiplicar su valor por los coeficientes gravitacionales correspondientes. Si el eje Z está orientado positivamente el peso propio normal de la estructura se especifica mediante Z= -1

Cada material necesita un registro en el que se describen sus propiedades.

El material constitutivo del elemento es isótropo y queda caracterizado por el *módulo de elasticidad longitudinal* (E), el *coeficiente de Poisson* (U) y el *peso específico* (W).

Nº DE IDENTIFICACIÓN DEL MATERIAL			PESO ESPECÍFICO
Nºm	E= MÓDULO DE ELASTICIDAD	U= COEFICIENTE DE POISSON	W= PESO / VOLUMEN

A cada elemento *shell* se le asigna un nº de identificación seguido por su descripción nodal dentro de JQ=.

El comportamiento de las estructuras tipo lámina plana se compone de dos partes: a) El trabajo como membrana (ETYPE=1), entendiéndose como tal, un estado de elasticidad bidimensional de tensión plana, y b) el trabajo como placa (ETYPE=2), que supone un estado de flexión bidimensional. Dependiendo de las características geométricas de la estructura y el tipo de carga, dominará uno u otro tipo de comportamiento, o incluso puede ser necesario tener cuenta ambos (ETYPE=0).

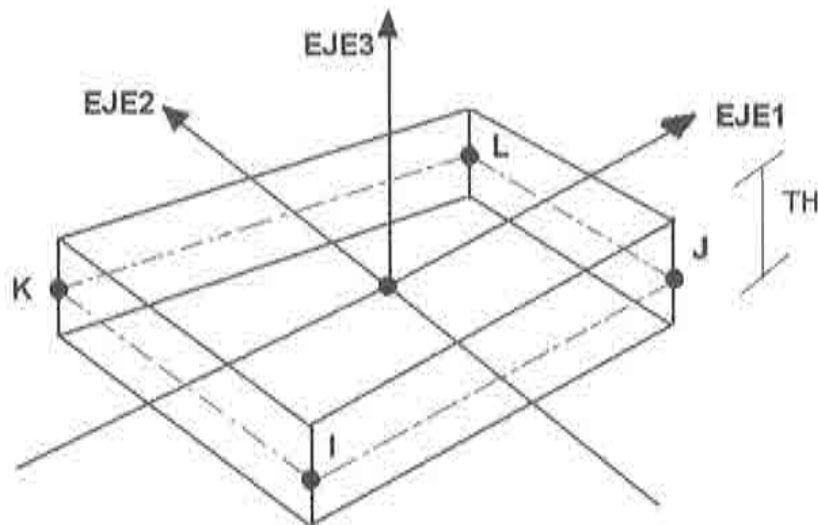
Los elementos tienen espesor constante (TH), si bien el programa permite asignar diferentes espesores para el trabajo a extensión th1 y a flexión th2.

<b>Nº DE IDENTIFICACIÓN DEL ELEMENTO SHELL</b>	<b>TIPO DE ELEMENTO</b>		
Nº el <b>JQ= ji, jj, jk, jl</b> <b>JQ= ji, jj, jk, 1*</b>	<b>ETYPE=0 COMPORTAMIENTO LÁMINA (MEMBRANA+ PLACA)</b>	<b>M=Nº DEL TH=th1,th2 G=g1, g2</b>	<b>2*</b>
<b>DESCRIPCION NODAL DE LOS ELEMENTOS SHELL</b>	6 grados de libertad por nodo (U1, U2, U3, R1, R2, R3) =1 COMPORTAMIENTO DE MEMBRANA 3 grados de libertad por nodo (U1, U2, R3) =2 COMPORTAMIENTO DE PLACA 3 grados de libertad por nodo (R1, R2, U3)	th1 ESPESOR del elemento membrana	th2 ESPESOR del elemento placa

2\* Generación automática de elementos a partir de un elemento base limitada a mallas de numeración regular de nodos y elemento.

#### 5.5.1.6. Descripción nodal de los elementos Shell $JQ= ji, jj, jk, jl$

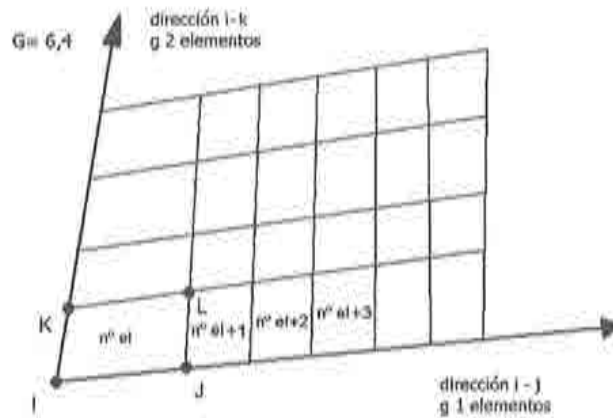
$ji, jj, jk, jl$  Son los números asignados a los cuatro nodos que describen el elemento de lámina, según indica la figura. Se pueden definir elementos triangulares con tres nodos  $JQ= ji, jj, jk$



Descripción nodal del elemento tipo Shell y ejes locales

#### 5.5.1.6.1 Generación automática de elementos a partir de un elemento base. $G= g1, g2$

Pueden generarse los elementos partiendo de un elemento de base previamente definido, en las direcciones de los lados  $i-j$  e  $i-k$ .  $g1$  y  $g2$  son los parámetros que generan la malla de dos dimensiones con  $g1$  elementos en la dirección  $i-j$  y  $g2$  elementos en la dirección  $i-k$ . La numeración nodal de los elementos generados, se forma incrementando el número de nodo del elemento base. Este procedimiento es únicamente válido para la generación de mallas con numeración regular de nodos y elementos.



**5.5.1.7 LOADS. Cargas nodales - Fuerzas y Momentos puntuales**

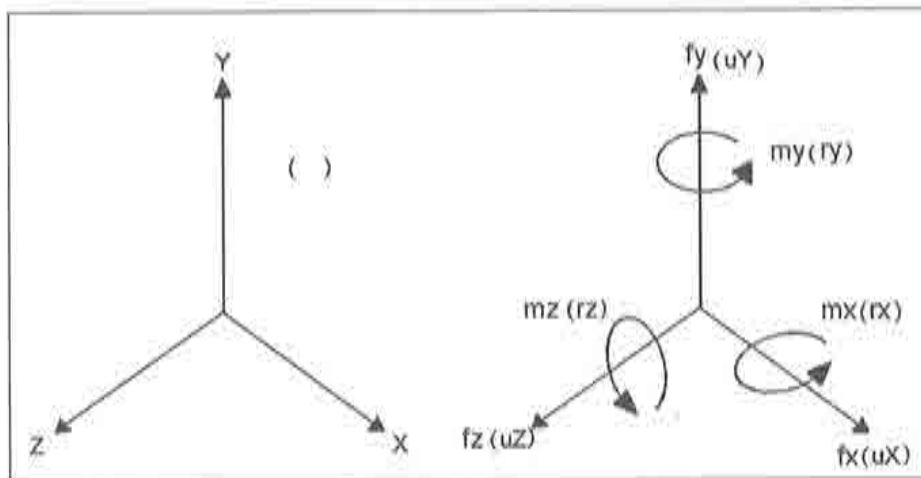
Las cargas nodales consisten en fuerzas y momentos aplicados en los nodos, asociados a una hipótesis de carga. No pueden aplicarse cargas según los grados de libertad coaccionados rigidamente. Las acciones nodales pueden introducirse en cualquier orden, tanto de n° de nodo como de n° de hipótesis de carga. Si se repiten especificaciones de carga para un mismo nodo en una misma hipótesis de carga, el programa aplica al nodo la suma de éstas.

Formato:  
**LOADS**

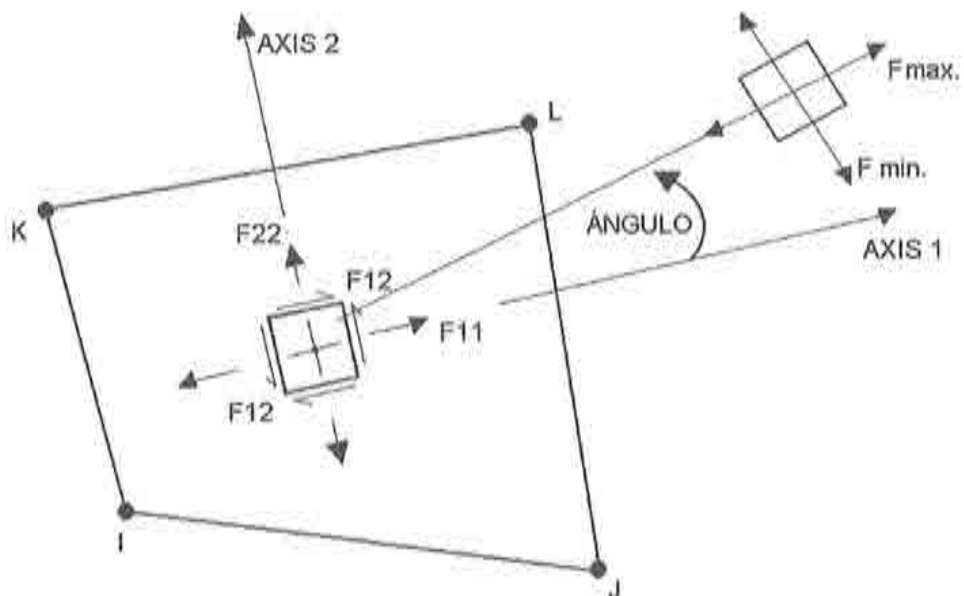
j1	J2	inc	L=1	F= fx, fy, fz, mx, my, mz	VALORES POR DEFECTO
[j1]	[j1]	[1]	[valor previo]		
					VALOR DEL MOMENTO APLICADO SOBRE LOS EJES X, Y, Z
					VALOR DE LA FUERZA APLICADA SEGUN LOS EJES X, Y, Z
					Nº DE LA HIPÓTESIS DE CARGA
					INCREMENTO EN LA NUMERACIÓN DE LOS NODOS DE LA SERIE
					Nº DEL ÚLTIMO NODO DE UNA SERIE SOBRE EL QUE ACTÚA UNA CARGA
					Nº DEL NODO SOBRE EL QUE ACTUA UNA CARGA
					Nº DEL PRIMER NODO DE UNA SERIE SOBRE EL QUE ACTÚA UNA CARGA
					Nº DEL NODO SOBRE EL QUE ACTÚA UNA CARGA

Los parámetros **J1, j2** y **inc** definen la siguiente serie de numeración de nodos **j 1, j1 + inc, j1 + 2 inc, j1 + 3 inc...** hasta alcanzar **j2**

### 5.5.2 Criterio de Signos y de representación de esfuerzos y momentos



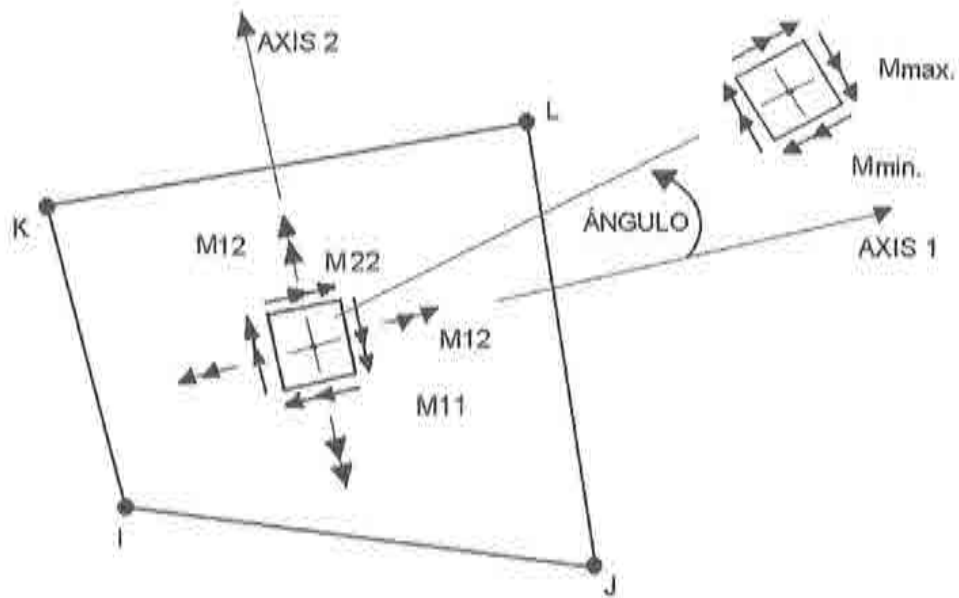
Criterio de signos positivos para la aplicación de cargas en los nodos y para los desplazamientos (u) y giros (r).



#### Esfuerzos y Fuerzas en el plano

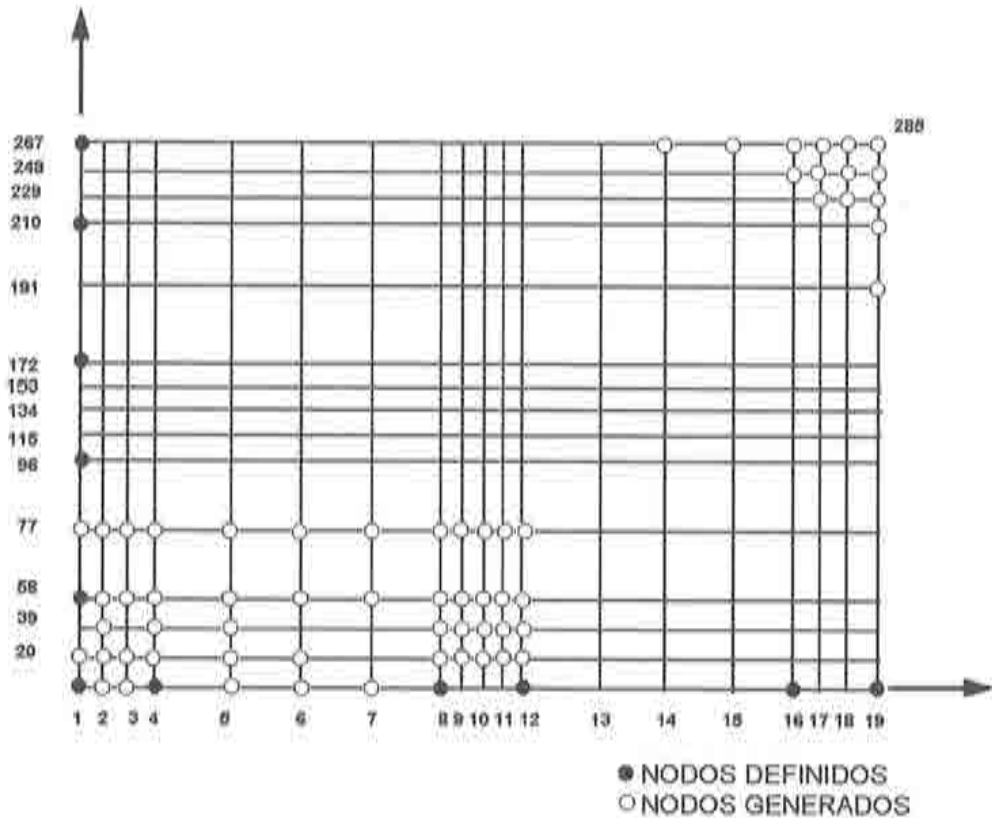
Los esfuerzos  $S_{ij}$  tienen la misma definición que las fuerzas  $F_{ij}$





Momentos fuera del plano

### 5.5.3 Ejemplo de generación (lineal y frontal) de Elementos y de Nodos para una placa apoyada en todo su perímetro y sometida a la acción de su propio peso



SYSTEM

JOINTS

1 X=0.0 Y=0.0 Z=0.0

4 X=1.5 G=1,4,1

8 X=5.5 G=4,8,1

12 X=7.5 G=8,16,1

19 X=13 G=16,19,1

58 X=0.0 Y=1.5 G=1,58,19

96 Y=3.5 G=58,96,19

172 Y=5.5 G=96,172,19

210 Y=7.5 G=172,210,19

267 Y=9.5 G=210,267,19

F=1,18,14,1,19

RESTRAINTS

1 285 1 R=1,1,0,0,1

1 19 1 R=1,1,1,0,1,1

1 267 19 R=1,1,1,1,0,1

19 285 19 R=1,1,1,1,0,1

267285 1 R=1,1,1,0,1,1

SHELL

NM=1 Z=-1

1 E=2.0E6 U=0.0 W=2.45

1 JQ=1,2,20,21 ETYPE=2 M=1 TH=0.25 G=18,14

#### 5.5.4 Postproceso

Una vez que los errores del problema son corregidos y el programa realiza el cálculo, se inicia la fase de *postproceso*, que en el caso de *Sap90* se denomina SAPLOT.EXE.

El *saplot.exe* es una subrutina que implementa de forma gráfica todos y cada uno de los resultados procesados por el programa.

Los ficheros de salida de resultados que presenta el *saplot.exe* son:


1. - Representación gráfica en pantalla de la opción seleccionada.
2. - Opciones de ploteado.
3. - Figura no deformada (identificación de nodos, elementos, conexiones, ejes locales, materiales).
4. - Acciones sobre la estructura.
5. - Deformaciones de la estructura.
6. - Representación gráfica de los esfuerzos.
- 7.- - Influencia en el tipo de línea del ploteado.
8. - Opciones de visualización (puntos de vistas y zoom)
9. - Selección de Elementos
0. - Reset (purgado de la representación)
- x. - Salida del programa

## SAPLOT

Para activar el programa basta con ejecutar el siguiente comando (desde el directorio donde se encuentren los ficheros necesarios):

```
: \> SAPLOT
```

ENTER "SAP90" DATA FILENAME [ ] > FICHERO

Las opciones se pueden seleccionar directamente con el ratón o tecleando el número asignado y pulsando . Las modificaciones de los datos numéricos se introducen por teclado después de seleccionar la opción elegida. En otro tipo de opciones, haciendo clic con el botón izquierdo del ratón van apareciendo las diferentes subopciones.

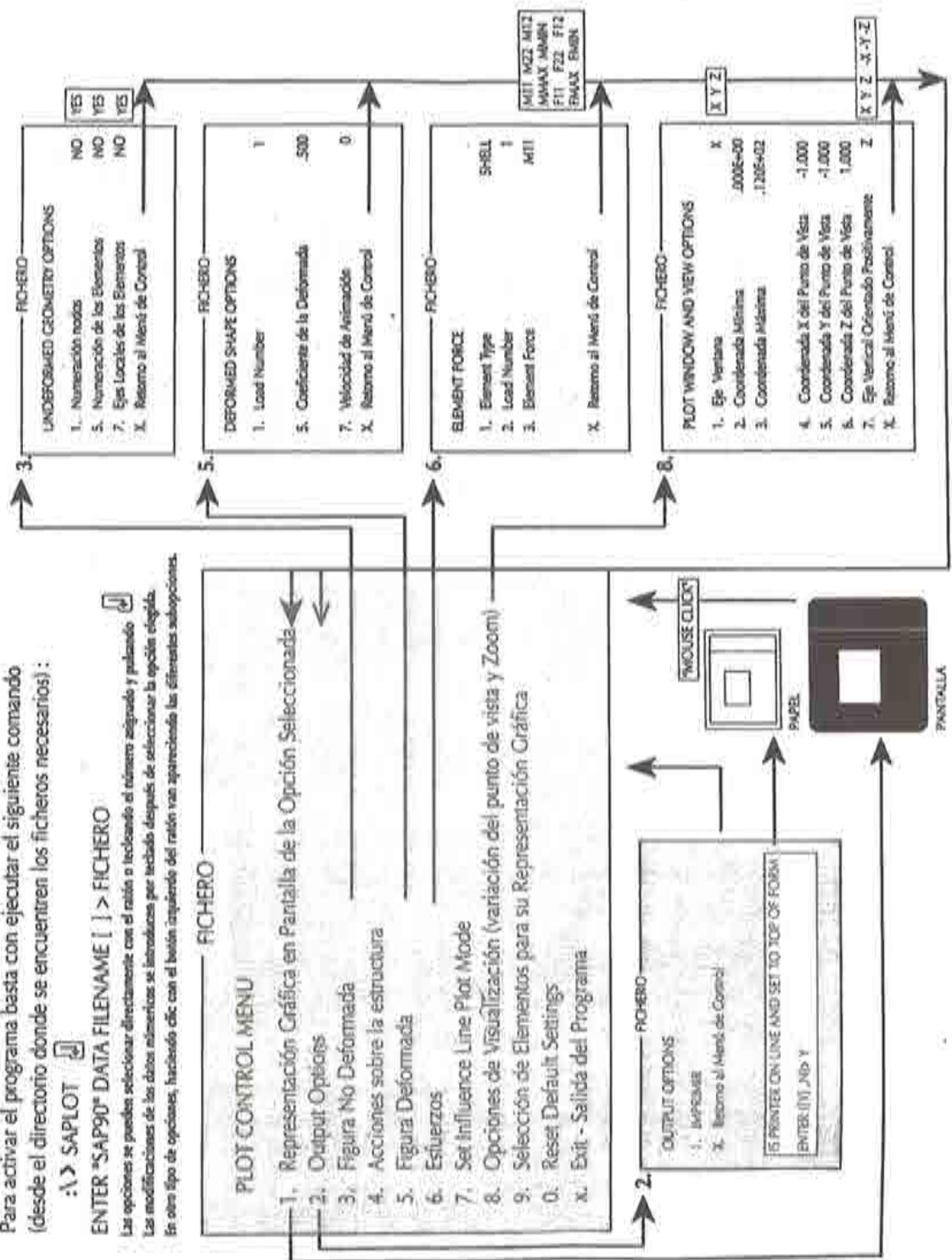
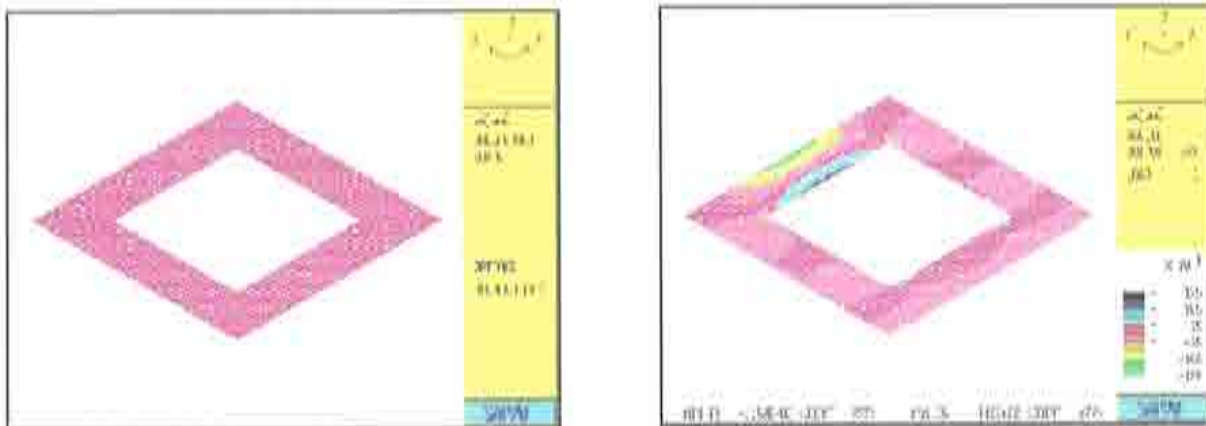


Fig. 5.2

Diagrama de uso de la Subrutina Saplot.exe



*Fig. 5.3*  
*Postproceso Saplot.exe*  
*Representación de un cajón trabajando a deformación plana.*

En el bloque SAPLOT.EXE se obtienen los resultados gráficamente, pero existe una gran limitación a la hora de manipular los resultados, así como una cierta dificultad al trabajar dentro del mismo bloque.

## 5.6 Usos y comandos más comunes del Sap90

### 5.6.1 La subrutina GO

El módulo *Sap90* lee el archivo de entrada y comprueba la compatibilidad de los datos. Este módulo se activa mediante la introducción del comando:

```
C> Sap90 <cr>
```

Si no se detectan errores, el módulo *Sap90* creará un archivo batch GO.BAT que activa los demás módulos de *Sap90* que son necesarios para ese análisis en particular. Los contenidos del archivo GO.BAT dependen de las opciones de entrada que son activadas en un archivo particular de entrada de datos de *Sap90*.

Por consiguiente, sólo serán ejecutados los módulos que son requeridos en realidad para el análisis de un modelo en particular.

El archivo GO.BAT es activado después de la ejecución de *Sap90* mediante la introducción del comando:

```
C> GO <CR>
```

### 5.6.2 La secuencia de ejecución

El programa de análisis estructural *Sap90* requiere que las operaciones analíticas sean realizadas en una rigurosa secuencia predeterminada.

El módulo *Sap90* es ejecutado siempre en primer lugar. Si no se detecta ningún error en este módulo se creará el archivo GO.BAT. Para una ejecución normal (en oposición a lo que sería reiniciar la ejecución) el archivo GO.BAT contendrá algunos o todos los comandos de la siguiente secuencia:

**OPTIMIZE – FRAME – SHELL - ASOLID – SOLID – SOLVE – EIGEN – RITZ – SPEC -  
TIMEH - JOINTF – ELEMf - FRAMEF – SHELLF – ASOLIDF - SOLIDF**

Para reiniciar el modo de ejecución, el archivo GO.BAT contendrá todos o algunos de los siguientes comandos:

**SPEC – TIMEH – JOINTF – ELEMf – FRAMEF – SHELLF – ASOLIDF - SOLIDF**

NOTA: Cada comando en el archivo GO.BAT activa el correspondiente módulo de programa.

### 5.6.3 Los comandos

a) Los comandos FRAME y FRAMEF existirán sólo si el modelo tiene elementos barra. De manera similar, los comandos SHELL y SHELLF sólo existirán si la entrada contiene el bloque de datos de elementos laminares. Este comportamiento es también similar para los comandos ASOLID y ASOLIDF, y SOLID Y SOLIDF.

b) Los comandos OPTIMIZE y SOLVE existirán siempre.

c) Los comandos EIGEN y RITZ existirán sólo si se ha activado un análisis *eigen* o un análisis de vectores de Ritz.

d) Los comandos SPEC o TIMEH existirán sólo si se están llevando a cabo un análisis de espectro de reacciones o un análisis de la respuesta temporal, i.e., el bloque de datos SPEC o el bloque de datos TIMEH existe en el archivo de entrada de datos.

e) El comando JOINT existe siempre.

f) Los comandos ELEMf, FRAMEF, SHELLF, ASOLIDF Y SOLIDF existirán sólo si las cargas son presentes o si el bloque de datos SPEC existe en el archivo de datos.

Este orden es alterado cuando se activa la opción de análisis P-Delta. Son necesarios algunos comandos adicionales para el análisis de puentes y el análisis de transmisión de calor.

El módulo SAPLOT (de dispositivo gráfico interactivo) puede ser activado después de una ejecución exenta de errores del comando *Sap90*, para obtener los gráficos de la geometría indeformada del modelo y de las cargas. Después de una ejecución satisfactoria del comando GO se pueden obtener: Las formas estáticas deformadas, el modo de formas, los diagramas de fuerzas de los elementos barra, y los contornos de carga de los elementos laminares (SHELL), de comportamiento plano (ASOLID), y tridimensionales (SOLID).

El módulo interactivo SAPTIME puede ser activado después de una ejecución satisfactoria de del comando GO para producir los gráficos y la impresión de los análisis de respuesta temporal.



#### 5.6.4 La opción Restart.

La opción *restart* de *Sap90* permite al usuario cambiar ciertos bloques de datos, y obtener salidas asociadas con los datos de entrada cambiados sin afectar al tiempo de montaje y solución de las fases del programa.

Dentro del modo de análisis de datos, sólo pueden ser cambiados o añadidos los siguientes bloques de datos:

- \* La línea de título. (obligatorio)
- \* SYSTEM (obligatorio)
- \* SPEC
- \* TIMEH
- \* COMBO
- \* ENVELOPE
- \* SELECT

Esto permite al usuario salidas para acciones dinámicas adicionales o modificadas, combinaciones de cargas adicionales o modificadas o también listas modificadas de nodos o elementos seleccionados. El resto de datos son tomados como idénticos al los de la ejecución original.

El nombre del archivo de entrada de datos utilizado para reiniciar la ejecución debe ser el mismo que el del archivo de entrada de la ejecución original. Si se desea, el archivo de entrada original puede ser salvado con un nombre diferente utilizando los comandos de *MS-DOS COPY* o *RENAME*.

Dentro del modo restart, el módulo *Sap90* sólo leerá los seis bloques de datos antes mencionados. Los demás bloques de datos existentes en el archivo de entrada de datos no son requeridos y serán ignorados por el programa si existen.

Todos los archivos intermedios creados por la ejecución original deben ser conservados en el disco si se prevé la utilización del modo restart.

Como el nombre del archivo de entrada de datos de reinicio debe ser el mismo que el del archivo de datos original, todos los archivos de salida asociados a la ejecución original será sobrescrito por la ejecución del reinicio. Si los archivos de salida asociados quieren ser conservados, sus nombres deben ser cambiados mediante al utilización del comando de *MS-DOS RENAME*.

### 5.6.5 Las subrutinas de cálculo

#### 5.6.5.1 Bloque de datos FRAME

Este bloque de datos define las propiedades, localizaciones y cargas asociadas con los elementos barra tridimensionales generales que existan en el modelo. Cada barra o viga bidimensional o cada viga tridimensional puede ser considerada como un caso especial de este elemento general. Pase por alto este bloque de datos si no hay elementos barra en el modelo.

#### 5.6.5.2 Bloque de datos SHELL

Este bloque de datos define las propiedades, localizaciones y cargas asociadas con los elementos generales laminares tridimensionales de cuatro nodos. Los pliegues laminares tridimensionales y los elementos membrana son considerados como casos especiales de este elemento general. Pase por alto este bloque de datos si no hay elementos laminares en el modelo.

#### 5.6.5.3 Bloque de datos ASOLID

Este bloque de datos define las propiedades, localizaciones y cargas asociadas con los elementos isoparamétricos de tres a nueve nodos. Este elemento puede ser utilizado para modelar sólidos de simetría axial, estructuras planas tensionadas y estructuras planas sometidas a carga. Todos los elementos deben disponerse paralelamente a los tres planos globales principales (XY, YZ o ZX).

#### 5.6.5.4 Bloque de datos SOLID

Este bloque de datos define las propiedades, localizaciones y cargas asociadas con los elementos sólidos tridimensionales de ocho nodos. Pase por alto este bloque de datos si no hay elementos sólidos en el modelo.

### Comentario Final

*Sap90* se ha convertido en un estándar en el mercado de programas de cálculo de estructuras mediante el *método de los elementos finitos*. Es pues un programa suficientemente comprobado y de amplia aceptación, contando además con un extenso soporte bibliográfico.

Pero tiene algunos inconvenientes, y es que ha quedado bastante desfasado con los nuevos progresos realizados en el mundo de la informática:

1. *Utiliza sólo ocho colores de la escala cromática.*
2. *No emplea más que pantallas de resolución de 640x680 puntos.*
3. *No ha actualizado la salida de impresión.*
4. *No permite entrar dibujos utilizando programas previos de dibujo como autocad.*
5. *Tiene un preproceso (discretización) muy limitado.*



## 6 EL PROGRAMA GiD (PRE Y POSTPROCESO)

### 6.1 Introducción

*GiD* es un programa de interfaz gráfico interactivo que se utiliza para definir, preparar e incluso visualizar todos los datos relacionados con la simulación numérica. Estos datos, permiten definir: la geometría, los materiales, las condiciones de contorno, y toda la información de los resultados a obtener. De la misma manera, el programa puede generar mallas de Elementos Finitos y definir toda la información de simulación numérica en un formato estándar para subsiguientemente, realizar el cálculo. También es posible, dentro del programa, ejecutar la simulación numérica y visualizar la información de los resultados.

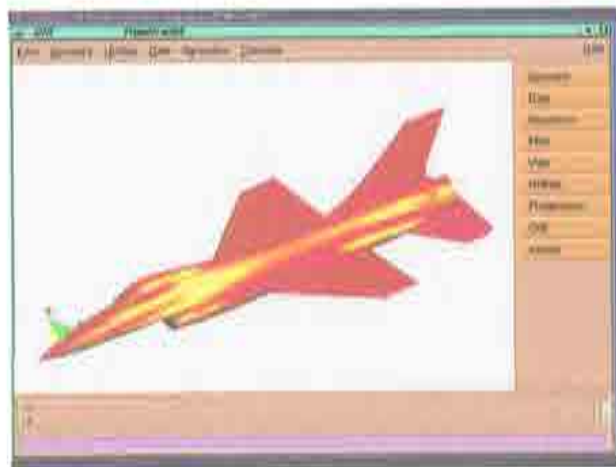


Fig. 6.1 Presentación de GiD\*

\* Imagen de la presentación de GiD por parte del C.I.M.N.E.

*GID* puede ser configurado por el usuario, de manera que pueda generar los datos que se requieren para la solución de sus módulos. La solución de los módulos, puede ser incluida dentro del Software de *GID*.

La forma de trabajo del programa, al definir la geometría, es muy similar al sistema *CAD* (Computer Aided Design) pero, con algunas diferencias. Una de las más importantes es que: la geometría se construye de manera jerárquica.

Esto significa, que una entidad de nivel superior (la línea) es construida sobre otra de nivel inferior (punto); dos (líneas) entidades adyacentes compartirán el mismo nivel inferior.

Todos los materiales, condiciones de contorno, y parámetros de la solución pueden definirse sobre la geometría sin que el usuario tenga algún conocimiento de la malla. La generación de la malla se realiza una vez que el problema se ha definido totalmente.

Las ventajas de trabajar así, radican en que usando los datos asociados a la estructura (tipo de problema, tipo de análisis), las modificaciones pueden realizarse dentro de la misma geometría y el resto de la información será, automáticamente, editada y estará lista para su análisis.

La completa visualización gráfica de la geometría, malla y condiciones, es útil para entender y revisar el modelo antes de la ejecución del análisis (*preproceso*).

Esta misma visualización nos permitirá evaluar de manera gráfica los resultados obtenidos (*postproceso*) después del análisis. Este *postproceso* utiliza una interface, la cual dependerá del tipo de problema y de los resultados obtenidos.

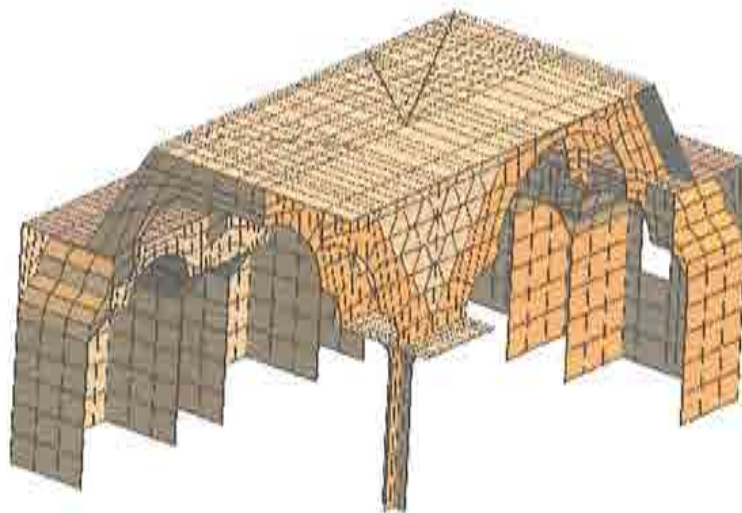


Fig. 6.2 Casa Bellesguard (Antonio Gaudí)<sup>†</sup>

<sup>†</sup> Cortesía del Departamento de Estructuras de E.T.S.A.B. (El análisis del edificio fue dirigido por el Arq. Javier Lopez\_Rey Catedrático de este Departamento).

### 6.1.1 Fundamentos de GiD

El manual de *GiD* está claramente dividido en 5 partes:

1. Primera parte. **Aspectos generales:** Donde el usuario puede encontrar los elementos básicos del programa. Estos, ayudan a obtener la máxima confianza en las acciones interactivas entre el usuario y el sistema.
2. Segunda parte. **Preproceso:** Describe la funcionalidad del *preproceso*. El usuario aprenderá como configurar un proyecto, definiendo todas sus partes (geometría, datos y generación de malla).
3. Tercera parte. **Resolución:** Se refiere al cálculo del proceso. A través de éste, se ejecutará una solución independiente, la cual forma parte del sistema *GiD*.
4. Cuarta parte. **Postproceso,** enfatiza los aspectos relacionados con la visualización de los resultados.
5. Quinta parte: **Utilización:** Explica la forma en que el usuario puede introducir y ejecutar los diferentes módulos, de acuerdo con sus propios requerimientos.

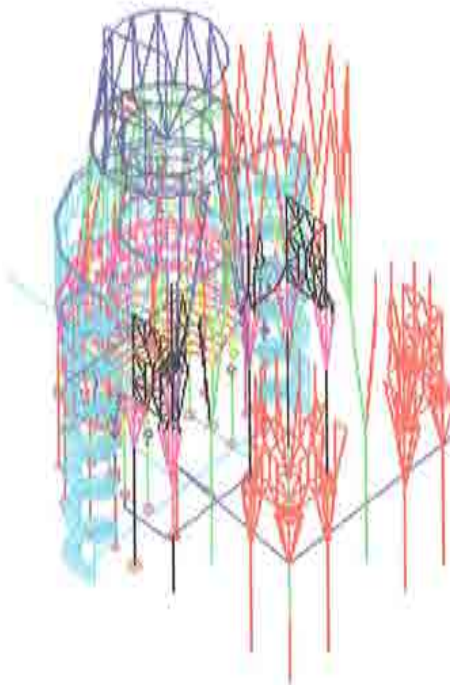


Fig. 6.3  
*Abside del templo expiatorio de la Sagrada Família (Antonio Gaudí)*



## 6.2 Usos, comandos y alcances de GiD

### 6.2.1 Uso de la interface

El uso de la interface, es el medio, que tiene el usuario de *GiD* para interactuar con el programa.

El programa, esta compuesto por: botones, ventanas, iconos, menús, entradas de texto e información gráfica. Puede ser configurado por el usuario de *GiD*, y que puede elegir entre más o menos ventanas.

Toda la información puede ser introducida mediante el teclado, aunque en muchos casos es más útil hacerlo por medio del uso del ratón y/o alguno de los menús.

Todos los comandos pueden ser introducidos tecleando su nombre o parte de él (las letras necesarias para evitar confusiones). Cualquier comando de la columna de la derecha puede ser introducido mediante el nombre escrito allí o parte de él. Los comandos especiales son los de visualización (zoom, rotation, etc.), que pueden ser tecleados o utilizados en cualquier momento durante el trabajo, incluso dentro de otra función.

Los comandos introducidos mediante el teclado son dirigidos por palabras. Esto significa, que es lo mismo escribir el comando completo y después pulsar enter, o escribir parte de él, pulsar enter, y después escribir el resto.

El botón izquierdo del ratón se usa para hacer selecciones (escoger entidades o abrir una caja<sup>†</sup>) y para introducir puntos en el plano  $z=0$ <sup>‡</sup>. El botón central es equivalente a escape<sup>\*\*</sup>. El botón derecho abre un menú con algunas opciones de visualización. Para seleccionar alguna de ellas, utilice el botón derecho o izquierdo. Para rechazarlo, pulse el botón derecho fuera del menú.

Los botones, iconos y menús son activados mediante el botón izquierdo del ratón. Es posible arrastrar desde un menú con el botón central del ratón, para sacarlo y dejarlo en otro lugar. Para hacerlo desaparecer pulse con el botón izquierdo del ratón sobre el botón del menú.

En las ventanas que se utilizan para introducir datos, generalmente, es necesario dar la orden de aceptar antes de cerrar la ventana. En caso contrario, los datos no serán modificados.



Fig. 6.4 Iconos y ventanas de GiD

<sup>†</sup> Ver referencia 6.2.3 Selección de entidades

<sup>‡</sup> Ver referencia 6.2.5.2.1.1 Point creation

<sup>\*\*</sup> Ver referencia 6.2.4 Escape

## 6.2.2 Definición de puntos (camino para definir puntos)

En el programa, existen muchas funciones que piden al usuario un punto. A veces, puede ser un punto existente. Otras, puede tratarse de un nuevo o antiguo punto.

En general, el usuario puede introducir puntos de las siguientes maneras:

1. Pulsando sobre una ventana: Los puntos son seleccionados en la ventana de gráficos en el plano  $z=0$  de acuerdo con las coordenadas visibles en dicha ventana. Si el usuario selecciona cerca de un punto existente, *GiD* pregunta si debe crear un nuevo punto o utilizar el existente.

2. Seleccionando un punto existente: Algunas veces, el usuario es consultado directamente para hacerlo. Si no, cuando se quiera seleccionar un punto existente, el usuario debe pulsar el botón 'join' de la columna de comandos de la derecha o su abreviatura (Control-a). Entonces, este botón se convertirá en 'No Join'. Después de esto, pulse sobre un punto existente para seleccionarlo.

3. Introduciendo puntos por coordenadas: Las coordenadas del punto pueden ser introducidas en la línea de comandos de dos maneras:

- a. En el formato 'x, y, z'.
- b. O en el formato 'x y z'. En ambos casos la z puede ser omitida.

4. Introduciendo las coordenadas locales: Las coordenadas locales, se consideran relativas al último punto "usado". Usado significa: punto creado o seleccionado. Es posible usar el comando 'Utilities Id' para referirse a un punto<sup>11</sup>.

Para referir siempre las coordenadas locales al mismo punto, utilice: 'Options Fixed Relative' al introducir los puntos. El último punto seleccionado o creado antes de usar esto, será el centro de las coordenadas locales. La manera de introducir las coordenadas locales es: '@x, y, z'.

5. Introduciendo coordenadas cilíndricas o esféricas:

- a. Coordenadas cilíndricas: 'r<ángulo, z'
- b. Coordenadas esféricas: 'r<ánguloxy<ánguloz'. En ambos casos z (o el ángulo z) puede ser omitido. Las coordenadas esféricas y cilíndricas pueden ser aplicadas a las coordenadas globales y a las locales.

6. Botón 'base': Si se selecciona el botón 'base'(el botón se convierte en 'No base'), un punto puede ser seleccionado de cualquiera de los otros nodos. Entonces, este punto, en lugar de ser usado, es escrito en la línea de comandos y puede ser editado antes de introducirlo.

Si el botón 'Join' no está o se encuentra en el nodo 'No Join', entonces el usuario puede introducir datos seleccionando un nodo de puntos existentes. Este modo de trabajo por defecto puede ser cambiado mediante 'preferences'<sup>12</sup>

<sup>11</sup> Ver referencia 6.2.5.10.3 Id

<sup>12</sup> Ver referencia 6.2.5.10.1 Variables

### 6.2.3 Selección de entidades

Muchos comandos, necesitan la selección de alguna entidad antes de ser aplicados. El modo de seleccionar entidades es común para todos ellos. Antes de empezar la selección, el usuario es requerido a decidir el tipo de entidad a seleccionar: puntos, líneas, superficies o volúmenes (en algunos casos la decisión se toma según el contexto). Entre cualquiera de estos grupos genéricos, no importa el tipo de entidad (ejemplo: un arco o una polilínea son seleccionados de la misma manera).

Después de ello, si el usuario pica sobre una entidad del grupo deseado, ésta queda seleccionada (su color cambia al rojo), y el usuario es requerido a introducir más entidades. Si el usuario pica lejos de una entidad, se abre una ventana dinámica que se define picando otra vez en otro lado. Todas las entidades que están en parte o totalmente dentro de la ventana, son seleccionadas. Después, el usuario es requerido, una vez más, a introducir más entidades. Si una entidad es seleccionada otra vez, resulta desactivada y su color cambia al normal.

Al mismo tiempo, es posible seleccionar entidades escribiendo su etiqueta en la línea de comandos. Por ejemplo para seleccionar desde la entidad 3 a la 7, use: 3:7 en la línea de comandos.

Existe un caso especial, presente, en algunas ocasiones, en el que cuando se selecciona un tipo de entidad, hay también la posibilidad de elegir 'All' (todo). En este caso sólo es posible la selección mediante la ventana dinámica en la ventana gráfica, y todas las entidades (puntos, líneas, superficies y volúmenes) dentro de la ventana, serán seleccionadas. Para terminar la selección de entidades utilice el comando `escape`<sup>§§</sup>.

Es posible cambiar el modo de selección mediante preferencias<sup>\*\*\*</sup>. Si se elige 'Fast selection', las entidades no cambian de color al ser seleccionadas, y picándolas dos veces no resultan desactivadas.

Atención: Use 'Fast Selection' sólo cuando necesite seleccionar un gran número de entidades (ejemplo: una gran malla). La posibilidad de repetir entidades en la selección es peligrosa.

### 6.2.4 Escape – comando de uso continuo – (aceptar)

El comando 'escape' se puede aplicar:

- a. Pulsando el botón central del ratón.
- b. Pulsando la tecla 'ESC'.
- c. Seleccionando el botón 'escape' de la columna de la derecha.
- d. Escribiendo la palabra 'escape' en la línea de comandos<sup>†††</sup>

<sup>§§</sup> Ver referencia 6.2.4 Escape

<sup>\*\*\*</sup> Ver referencia 6.2.5.10.1 Variables

<sup>†††</sup> Ver referencia 6.2.5.1.9 Batch file

Todas estas posibilidades dan el mismo resultado.

El comando 'escape' es utilizado para salir de menús, acabar selecciones y demás utilidades.

Atención: La palabra 'escape' es una palabra reservada. No puede ser utilizada en ningún otro contexto.

### 6.2.5 Manual de referencias

En este apartado, se ofrece una lista de comandos del programa agrupados por temas. Otra posibilidad para buscarlos es consultando el índice de Conceptos<sup>†††</sup>:

6.2.5.1 Archivos:	(file)
6.2.5.2 Creación de entidades geométricas	(Geometrical entities creation)
6.2.5.3 Eliminación de entidades geométricas	(Deleting geometrical entities)
6.2.5.4 Edición de entidades geométricas	(Editing geometrical entities)
6.2.5.5 Condiciones de borde	(Conditions)
6.2.5.6 Materiales de base	(Materials)
6.2.5.7 Datos del problema	(Problem data)
6.2.5.8 Intervalo de datos	(Interval data)
6.2.5.9 Malla	(Meshing)
6.2.5.10 Opciones de Visualización	(View options)
6.2.5.11 Utilidades	(Utilities)
6.2.5.12 Salir	(Quit)

---

<sup>†††</sup> Ver índice de conceptos: Anexo 6.2

## Lista de comandos del programa

6.2.5.1 - Archivos	(Files)	6.2.5.6 - Materiales	(Materials)
6.2.5.1.1.- Salvar	(Save)	6.2.5.6.1. - Asignar material:	(Assign material)
6.2.5.1.2.- Leer:	(Read)	6.2.5.6.2.- Dibujar material:	(Draw material)
6.2.5.1.3.- Nuevo:	(New)	6.2.5.6.3.- Eliminar material:	(Unassign material)
6.2.5.1.4.- Salvar capa:	(Save layer)	6.2.5.6.4.- Nuevo material:	(New material)
6.2.5.1.5.- Escribir Ascii:	(Write Ascii)	6.2.5.7. - Datos del Problema	(Problem data)
6.2.5.1.6.- Leer malla:	(Mesh read)	6.2.5.8. - Intervalo de Datos:	(Interval data)
6.2.5.1.7.- Leer DXF:	(DXF read)	6.2.5.9. - Malla	(Meshing)
6.2.5.1.8.- Escribir archivo para calcular:	(Write calculations file)	6.2.5.9.1. - Generación de malla:	(Generating mesh)
6.2.5.1.9.- ArchivoBatch	(Batch file)	6.2.5.9.2. - Ver malla:	(Mesh view)
6.2.5.2. - Creación de entidades geométricas	(Geometrical entities creation)	6.2.5.9.3. - Asignar tamaño:	(Assign sizes)
6.2.5.2.1. - Creación de punto:	(Point creator)	6.2.5.9.4. - Malla estructurada	(Structured mesh)
6.2.5.2.2. - Creación de línea recta:	(Straight line creator)	6.2.5.9.5. - Tipo de elemento:	(Element type)
6.2.5.2.3. - Creación de línea en arco:	(Arc line creator)	6.2.5.9.6. - Regenerar:	(Rese)
6.2.5.2.4. - Creación de polígono:	(Spline line creator)	6.2.5.10. - Opciones de visualización	(View options)
6.2.5.2.5. - Creación de Polilínea:	(Polyline creator)	6.2.5.10.1. - Zoom:	
6.2.5.2.6. - Creación de Superficie plana:	(Planar surface creator)	6.2.5.10.2. - Rotación:	(Rotation)
6.2.5.2.7. - Creación de superficie de 4 lados:	(4sided surface creator)	6.2.5.10.2.1. - Rotación dinámica:	(Rotate ScrAxis)
6.2.5.2.8. - Creación de volumen:	(Volume creator)	6.2.5.10.2.2. - Rotación sobre ejes:	(Rotate ObjAxis)
6.2.5.2.9. - Creación de contacto:	(Contact creator)	6.2.5.10.2.3. - Rotación hacia atrás:	(Rotate Trackball)
6.2.5.3. - Eliminación entidades	(Delete)	6.2.5.10.2.4. - Rotación de ángulo:	(Rotate Angle)
- Punto:	(Point)	6.2.5.10.2.5. - Rotación de puntos:	(Rotate points)
- Línea:	(Line)	6.2.5.10.2.6. - Rotación central:	(Rotate center)
- Superficie:	(Surface)	6.2.5.10.3. - Desplazamiento:	(Pan)
- Volumen:	(Volume)	6.2.5.10.4. - Redibujar:	(Redraw)
- Todas las entidades:	(AllTypes)	6.2.5.10.5. - Render:	(Label)
6.2.5.4. - Edición de entidades geométricas	(Editing geometrical entities)	6.2.5.10.6. - Nivel:	(Entities)
6.2.5.4.1. - Mover punto:	(Move point)	6.2.5.10.7. - Entidades:	(Layers)
6.2.5.4.2. - Explotar polilínea:	(Explode polyline)	6.2.5.11. - Utilidades	(Utilities)
6.2.5.4.3. - Editar polilínea:	(Edit polyline)	6.2.5.11. 1. - Variables:	(Variables)
6.2.5.4.4. - Editar línea recta:	(Edit spline)	6.2.5.11. 2. - Renumerar:	(Renumber)
6.2.5.4.5. - Dividir:	(Divide)	6.2.5.11. 3. - Identificación:	(Id)
6.2.5.4.6. - Editar nodos de una línea:	(Join lines)	6.2.5.11. 4. - Lista:	(List)
6.2.5.4.7. - Editar Arco:	(Swap arcs)	6.2.5.11. 5. - Distancia:	(Dist)
6.2.5.5. - Condiciones de borde	(Conditions)	6.2.5.11. 6. - Dibujar normales:	(Draw Normal)
6.2.5.5.1. - Asignar condiciones:	(Assign Condition)	6.2.5.11. 7. - Cambiar sentido:	(Swap sense)
6.2.5.5.2. - Dibujar condiciones:	(Draw Condition)	6.2.5.11. 8. - Copiar:	(Copy)
6.2.5.5.3. - Eliminar condiciones:	(Unassign condition)	6.2.5.11. 9. - Reparar:	(Repair)
		6.2.5.12. - Salir:	(Quit)

Tabla 6.1



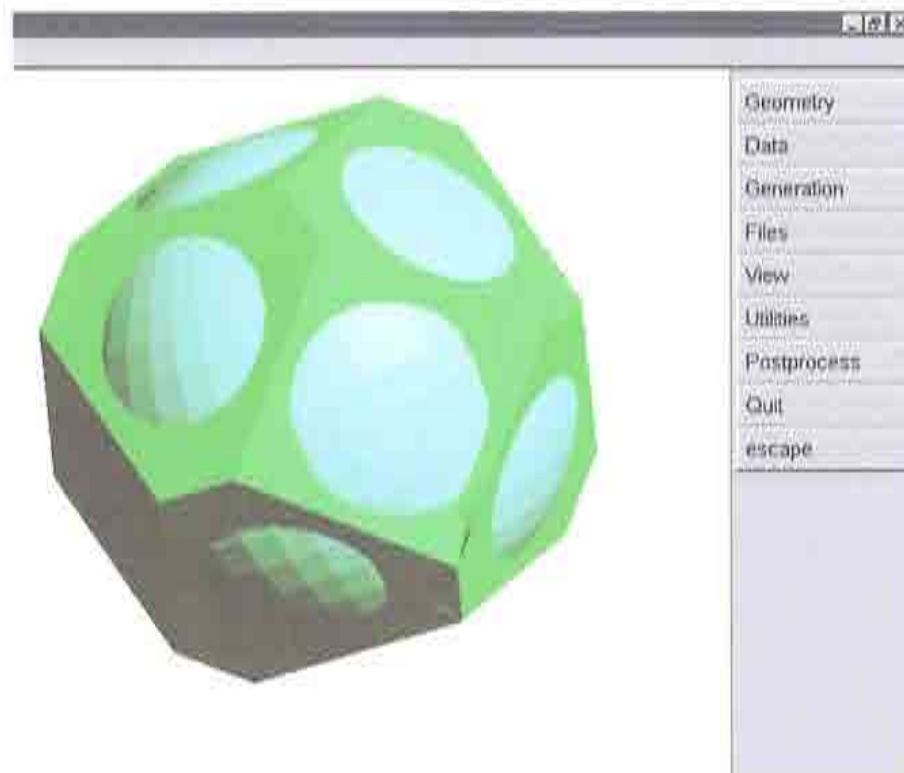


Fig. 6.5 Galena de la Sagrada Familia

### 6.2.5.1 Archivos

#### (Files)

Respecto al tema de los archivos, se incluye la manera usual de salvar, leer información salvada ('Save', 'Read') y otra serie de utilidades como: importar archivos externos, salvar en otros formatos.

#### 6.2.5.1.1 Salvar

#### (Save)

Con la indicación `save project`; se puede salvar toda la información relativa al proyecto; geometría, condiciones, materiales, mallas; en el disco.

*GiD* salva esta información, creando un nuevo directorio de la forma `-gid`, donde `-` representa el nombre del proyecto o directorio y `grava` en él los archivos generados (materiales, condiciones, etc.). Algunos están escritos en formato binario y otros en formato *ASCII*.

Usualmente, es lo mismo utilizar el nombre del proyecto con la extensión `.gid` o sin ella.

Atención: Tenga cuidado al cambiar manualmente los archivos dentro del directorio `proyecto.gid`. Es posible que, al hacerlo, se generen problemas.

Consejo: Salve frecuentemente por precaución.



### 6.2.5.1.2 Leer

(Read)

Con este comando, se puede leer el proyecto salvado con 'Save'<sup>688</sup>. Recuerde que, usualmente, es lo mismo usar el nombre del proyecto con la extensión `-gid` que sin ella.

#### Menú

Save:  
Read:  
New:  
Save layer  
Write Ascii:  
Mesh read:  
DXF read:  
Write calculations file:  
Batch file:

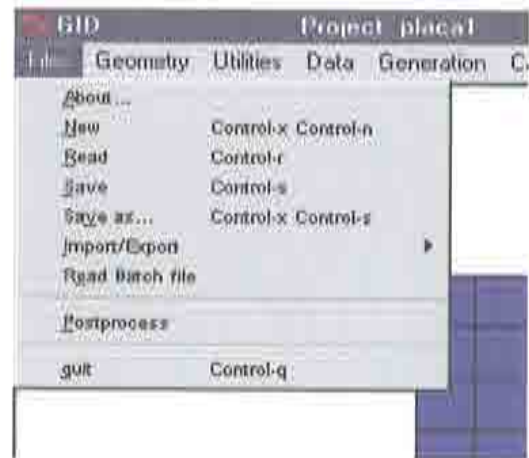


Fig. 6.6 Archivos

### 6.2.5.1.3 Nuevo

(New)

Cuando se selecciona 'New', *GiD* pregunta si debe salvar el archivo, por si ha habido cambios en el trabajo anterior. Después, abre un nuevo archivo sin título.

Si estamos trabajando dentro de un tipo de análisis, este será mantenido, o no, en el nuevo proyecto dependiendo del valor de una variable<sup>689</sup>.

### 6.2.5.1.4 Salvar capa (Save layer)

Con esta opción, sólo las entidades cuyas capas estén activadas ('on') serán salvadas en el proyecto con el nombre que el usuario desee.

Atención: Cuando este proyecto sea leído después, es necesario reparar la geometría<sup>690</sup> una vez, antes de trabajar con él.

<sup>688</sup> Ver referencia 6.2.5.1.1 Save

<sup>689</sup> Ver referencia 6.2.5.10.1 Variables

<sup>690</sup> Ver Referencia 6.2.5.10.8 Repair

## 6.2.5.1.5 Escribir Ascii

## (Write Ascii)

En esta opción, se puede escribir un archivo con toda la información del proyecto. Esta información está dispuesta de tal manera que es fácil de entender, cuando es leído por un editor. Esto, es útil para comprobar la información.

Nota: Este formato ascii es sólo para comprobar información. No puede ser leído otra vez por *GiD*

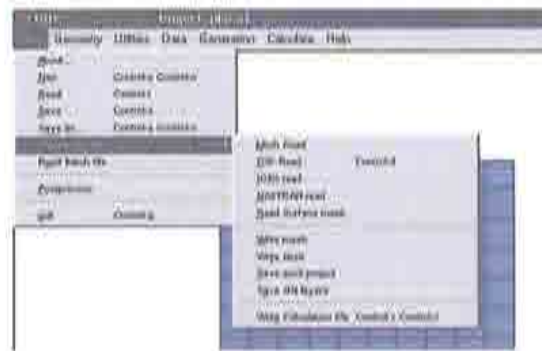


Fig. 6.7 Comandos para importar y exportar

## 6.2.5.1.6 Leer Malla

## (Mesh read)

Con esta opción es posible leer una malla para visualizarla en *GiD*. El formato del archivo que describe la malla debe ser el siguiente:

```

mesh dimension = 3 elemtype tetrahedra nnode = 4
coordinates
1 0 0
2 3 0
3 6 0
4 3 0
5 3 1.5 4
6 3 1.5 -4
7 1.5 0 2
end coordinates
elements
1 1 2 4 5 1
2 2 3 4 5 1
3 1 4 2 6 1
4 2 4 3 6 1
5 1 2 5 7 1
end elements
  
```

Donde 'elemtype' debe ser:

- a. Lineal (Linear)
- b. Triangular (Triangle)
- c. Cuadrilateral (Quadrilateral)
- d. Tetraédrico (Tetrahedra)
- e. Hexaédrico (Hexahedra)



Fig. 6.8 Generación y lectura de malla

Todos los elementos tienen un número opcional después de la definición de las conectividades. Este número designa el material y es útil para dividir la malla en capas, y poder visualizarla mejor.

Si necesita introducir diferentes tipos de elementos, cada uno debe pertenecer a diferentes mallas.

Se pueden introducir más de una malla, escribiéndola una después de la otra, todas ellas en el mismo archivo.

La única diferencia, es que a partir de la segunda malla no se distingue entre 'coordinates' y 'end coordinates' ya que comparten el primer punto de la malla.

-Atención: Antes y después del signo '=' es necesario dejar un espacio.

#### 6.2.5.1.7 Leer DXF (DXF read)

Con esta opción es posible leer un archivo en formato *DXF* versión 12,13 y 14 de *Autocad*.

Las entidades leídas son mayoritariamente todo tipo de líneas, por lo que debe ser evitado trabajar con superficies u objetos, o bien, convertirlos a líneas.

Un importante parámetro a tener en cuenta es cómo deben de ser unidos los puntos. Esto significa que los puntos que son unidos deben ser convertidos en uno solo.

La manera de hacer esto es definir un 'Epsilon'. Los puntos que estén dentro de este radio 'Epsilon' son convertidos en uno solo. Este 'Epsilon' es definido en preferencias<sup>\*\*\*\*</sup>.

Si líneas rectas que comparten ambos puntos, éstos son también convertidos en uno solo.

#### 6.2.5.1.8 Escribir archivo para calcular (Write calculations file)

Este comando escribe el fichero necesario para la ejecución del cálculo

Si *GiD* puede ejecutar el proceso FEM automáticamente, este comando no es necesario. Este comando puede ser útil si queremos ejecutar el proceso del programa fuera de *GiD* o si queremos comprobar este archivo para ver la entrada de datos para los cálculos.

El modo en que este archivo es escrito debe ser definido con anterioridad,<sup>\*\*\*\*</sup>

Cuando se está comprobando la definición de un nuevo tipo, *GiD* ofrece mensajes sobre errores en la configuración. Cuando se localiza el error este comando puede ser utilizado otra vez sin tener que salir del trabajo en curso.

<sup>\*\*\*\*</sup> Ver referencia 6.2.5.10.1 Variables

<sup>\*\*\*\*</sup> Ver referencia 6.2.6 Tipo de análisis

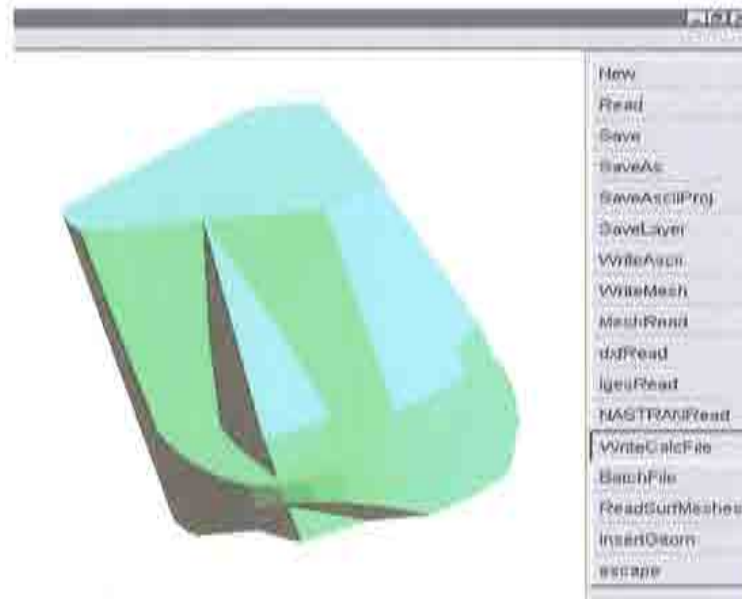


Fig. 6.9 Escritura de archivo para su cálculo

#### 6.2.5.1.9 Archivo Batch (Batch file)

A veces, puede ser interesante no usar *GiD* de manera interactiva. Para hacer esto, los comandos pueden ser escritos en un archivo y *GiD* leerá este archivo y ejecutará dichos comandos.

Los comandos son los mismos que se escriben en la línea de comandos y los mismos de la columna de comandos de la derecha.

Ejemplo: Hemos digitalizado muchos puntos y hemos salvado sus coordenadas en un archivo. Queremos unir estos puntos con líneas rectas para crear el contorno de nuestra geometría.

Para hacerlo, el archivo debe tener este aspecto:

```
geometry create line
3 7 4 5 8
2 5 9
4, 5, 6
...
1 7 0.0
escape
```

Un 'batch file' puede ser cargado en *GiD* mediante la introducción de su nombre con la opción `-b` cuando se abre *GiD*

### 6.2.5.2 Creación de entidades geométricas (Geometrical entities creation)

#### *Create*

*Point creation:*

*Straight line creation:*

*Arc line creation:*

*Spline line creation:*

*Polyline creation:*

*Planar surface creation:*

*4-sided surface creation:*

*Volume creation:*

*Contact creation:*

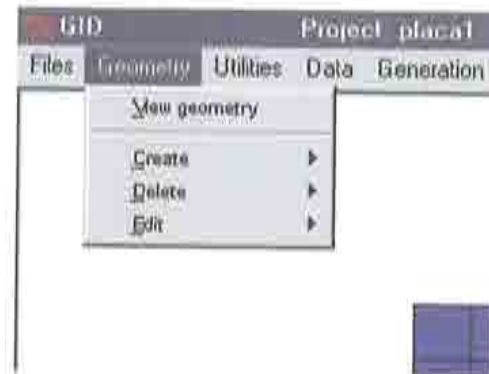


Fig. 6.10 Geometría

#### 6.2.5.2.1 Creación de punto (Point creation)

Para crear un punto individual, es necesario introducirlo del modo habitual. Puede ser usado para unir líneas sobre él.

Atención: Un posible error es intentar crear puntos nuevos uniéndolos a puntos antiguos.

#### 6.2.5.2.2 Creación de línea Recta (Straight line creation)

Para crear una línea recta sólo es necesario entrar dos puntos y continuar introduciendo puntos si se desea seguir definiendo líneas. Cada parte de la línea total creada es una línea independiente.

Es importante advertir que cuando creamos líneas estamos creando también nuevos puntos (sino estamos utilizando puntos ya existentes para definirlos).

La opción 'Close' une el primer punto con el último punto creado al definir varias líneas rectas seguidas.

La opción 'Undo' deshace la creación del último punto (si es nuevo) y de la última línea. Es posible continuar deshaciendo hasta el primer punto.

Si seleccionamos 'Join', esta orden se mantiene para todos los puntos hasta que se selecciona 'No Join'.



### 6.2.5.2.3 Creación de Línea en arco (Arc line creation)

Para crear un arco es necesario introducir tres puntos<sup>\*\*\*\*\*</sup>,

Es importante advertir que cuando creamos un arco estamos creando también nuevos puntos (sino utilizamos puntos ya existentes).

El arco, se genera siempre en sentido antihorario, empieza y acaba en el primer y tercer punto introducidos. El segundo punto es sólo utilizado como referencia y, si no es ya existente, es automáticamente borrado cuando el arco es creado.

Si queremos definir el arco contrario al definido, podemos usar.<sup>†††††</sup>

### 6.2.5.2.4 Creación de Polígono (Spline line creation)

Las splines: son una serie de líneas dentro de una curva las cuales son definidas por una poligonal de control. Esta poligonal esta definida por una serie de puntos en la que el primero y el último punto coinciden con el punto inicial y final de la curva. El resto de puntos no están sobre la curva. Podemos considerar este polígono como una aproximación de la curva pero de un modo simplificado.

Para crear un spline, introducimos los puntos que forman la poligonal de control<sup>‡‡‡‡‡</sup> Antes de introducir el último punto del polígono, seleccione 'LastPoint'.

La opción 'Undo' deshace la creación del último punto. Es posible continuar deshaciendo hasta el primer punto.

Cuando ya se ha creado una spline, todos los puntos interiores (excepto el primero y el último) no son realmente entidades a no ser que fueran puntos ya existentes.

Nota: Existen 4 tipos de spline: uniform, non-rational, cubic B-splines

### 6.2.5.2.5 Creación de Polilínea (Polyline creation)

Las polilíneas contienen una serie de otras líneas de cualquier tipo (incluyendo también otras polilíneas). Cada línea debe compartir uno o dos de sus extremos con los extremos de otras líneas.

Para crear una polilínea, deben ser seleccionadas las rectas que la formarán. No es importante el orden de selección pero sí que todas las líneas compartan sus extremos.

Las polilíneas aparecerán de color verde para mostrar su diferencia con las líneas normales que serán de color azul.

Las polilíneas son usadas ampliamente al crear 4-sided surfaces.<sup>§§§§§</sup>

\*\*\*\*\* Ver referencia 6.2.5.2.1.1 Point creation

††††† Ver referencia 6.2.5.3.7 Swap arcs.

‡‡‡‡‡ Ver referencia 6.2.5.2.1.1 Point creation

§§§§§ Ver referencia 6.2.5.2.1.7 4-sided surface creation



Si borramos una polilínea, todas sus líneas son también borradas. Si la descomponemos<sup>\*\*\*\*\*</sup>, la polilínea se convierte en una serie de líneas independientes unas de otras.

#### 6.2.5.2.6 Creación de Superficie Plana (Planar surface creation)

Una superficie plana es una entidad formada por una serie cerrada de líneas que están todas sobre el mismo plano. Las líneas deben compartir sus extremos.

Para crear una superficie plana, se deben seleccionar las líneas que la formarán. No es importante el orden de selección, pero todas ellas deben compartir sus extremos y formar un contorno cerrado. Si alguna de las líneas no está en el mismo plano, la superficie no es creada.

Es posible hacer agujeros en una superficie plana. Para hacerlo, es necesario crear primero la superficie plana que definirá el contorno exterior. Después de ello, pulse el botón 'Hole' y seleccione la superficie creada. A continuación, seleccione las líneas que forman cada agujero, una por una. Para acabar pulse 'escape'.<sup>†††††</sup>

Si la superficie está contenida en el plano 'z=0', la orientación de la superficie será antihorario en este plano (normal goes to 'z' positive). De otro modo, la orientación es arbitraria.<sup>‡‡‡‡‡</sup>

#### 6.2.5.2.7 Creación de superficie de 4 lados (4-sided surface creation)

4-sided surface es una entidad formada por una serie cerrada de cuatro líneas en el espacio. Su definición matemática es: bilinear Coon surface. La superficie está totalmente definida por el contorno de las líneas, sin ninguna información más en su interior. Esto significa que, a veces, será necesario usar más superficies para obtener una buena definición de la forma.

Para crear una superficie definida por tres líneas, será necesario dividir una de ellas en dos partes.<sup>§§§§§</sup> Entonces es posible crear una superficie de cuatro lados.

Para crear una superficie de cuatro lados, deben seleccionarse las líneas que la formarán. No es importante el orden de selección, pero todas ellas deberán compartir sus extremos y formar un contorno cerrado. Si su creación no es posible, se mostrará información sobre los extremos de las líneas en la pantalla.

Es posible que una o más de las líneas que componen una superficie de 4 lados formen parte de una polilínea.

Para hacerlo, seleccione como una línea la polilínea entera y *GiD* selecciona la línea o líneas requeridas. Con esta posibilidad, se pueden generar superficies no conformadas.

Esto significa que se puede crear en un lado de la polilínea una superficie utilizando toda la línea y en el otro lado crear más de una '4-sided surface' considerando cada una como parte de ella.

\*\*\*\*\* Ver referencia 6.2.5.3.2 Explode polyline

††††† Ver referencia 6.2.4 Escape

‡‡‡‡‡ Ver referencia 6.2.5.10.6 Draw normals

§§§§§ Ver referencia 6.2.5.3.5 Divide

Si seleccionamos más de cuatro líneas, *GiD* buscará automáticamente todas las posibles superficies de cuatro lados que puedan ser creadas con alguna de estas líneas. Esto permite la creación de muchas superficies al mismo tiempo.

El botón 'Automatic' hace el mismo trabajo que la última posibilidad pero con las líneas de todas las capas. Las nuevas superficies son creadas en la capa actual.\*\*\*\*\*

Si la superficie está en el plano ' $z=0$ ', la orientación de la superficie será antihorario en esta plano (normal goes to 'z' positive). De otra manera, la orientación es arbitraria. Ésta puede ser comprobada mediante `xref{Draw Normals}`.

Atención: Al crear varias superficies al mismo tiempo, es posible que resulte creada también alguna superficie no deseada. Es necesario pues comprobar las superficies y después de su creación y borrar las no deseadas.

#### 6.2.5.2.8 Creación de Volumen

( Volume creation)

Un volumen es una entidad formada por una serie cerrada de superficies que comparten sus líneas entre ellas.

Para crear un volumen, deben seleccionarse las superficies que la formarán. No es importante el orden de selección pero todas ellas deberán compartir sus lados y formar un contorno cerrado.

Si existe algún error y el volumen no es creado aparecerá información útil en la pantalla.

Es posible hacer agujeros en un volumen. Para hacerlo, es necesario crear el volumen exterior y el volumen interior como independientes uno de otro. Después de ello, pulse el botón 'Hole' y seleccione el volumen exterior. A continuación, seleccione los volúmenes que forman cada agujero, uno por uno. Para acabar presione 'escape'. !!!!!!!

La orientación del volumen y sus superficies es adaptada automáticamente para que la malla sea correcta.

#### 6.2.5.2.9 Creación de Contacto

(Contact creation)

El contacto es un enlace especial entre dos líneas o dos superficies que están físicamente en el mismo sitio pero en diferentes entidades, como superficies, líneas y puntos. Mediante un contacto, es posible generar elementos que definan un contacto especial entre dos materiales.

Para crear contactos es necesario escoger 'contact surface' o 'contact volume'. El primero es entre líneas. El segundo es entre superficies. A continuación, seleccione líneas en el primer caso, y superficies en el segundo. *GiD* busca automáticamente posibles contactos combinando por pares las superficies elegidas.

\*\*\*\*\* Ver referencia 6.2.5.9.8 Layers

!!!!!! Ver referencia 6.2.4 Escape

### 6.2.5.3 Eliminación de entidades geométricas (Delete)

El borrado de entidades es siempre efectuado a uno de estos niveles: punto, línea, superficie, volumen o todo tipo de entidades y después, se realiza la selección. Después de presionar 'escape', <sup>††††††††</sup> las entidades son borradas.

Si queremos evitar el borrado de entidades seleccionadas, presione 'break'.

Las entidades que tienen sobre ellas otras entidades o tienen condiciones asignadas, no pueden ser borradas.

Ejemplo: Si hemos creado una superficie sobre algunas líneas, es necesario borrar primero la superficie y después las líneas.

### 6.2.5.4 Edición de entidades geométricas (Editing geometrical entities)

Existen algunas opciones de edición para entidades geométricas:

#### *Edit*

- Move point:*
- Explode polyline:*
- Edit polyline:*
- Edit spline:*
- Divide:*
- Join lines:*
- Swap arcs:*

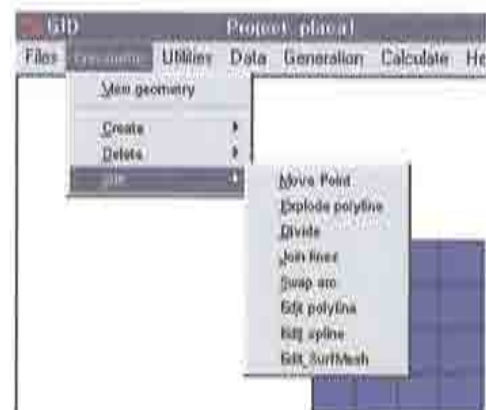


Fig.6.11 Edición de Entidades

#### 6.2.5.4.1 Mover Punto (Move point)

Con este comando se selecciona un punto ya existente. La nueva posición es introducida del modo habitual. <sup>§§§§§§§§</sup> Si la nueva posición es un punto existente (hemos utilizado la orden 'Join'), *GiD* comunica la distancia entre los puntos y pregunta si debe unirlos. Si se contesta que sí, los dos puntos son convertidos en uno solo. Las líneas y superficies son también movidas y deformadas para que continúen sobre el punto.

La opción 'Relative' sirve para definir la distancia de la nueva posición en cada eje de coordenadas.

<sup>††††††††</sup> Ver referencia 6.2.4 Escape

<sup>§§§§§§§§</sup> Ver referencia 6.2.5.2.1.1 Point creation

La opción 'MorePoints' permite al usuario seleccionar más puntos. Estos puntos serán desplazados a igual distancia que el primero. Si los dos extremos de la línea están siendo desplazados, la línea no se deforma, sino que sólo se traslada.

Nota: Join lines; Otra manera de convertir dos puntos en uno solo.

#### 6.2.5.4.2 Explotar polilínea (Explode polyline)

Este comando permite al usuario seleccionar líneas. Dichas líneas son rechazadas si no son polilíneas, si tienen entidades sobre ellas, o condiciones asociadas. Después de la confirmación, el resto de polilínea es descompuesta y convertida en las rectas originales. La polilínea pues desaparece.\*\*\*\*\*

#### 6.2.5.4.3 Editar Polilínea (Edit polyline)

El comando 'EdPoly' solicita la selección de una línea. Si la línea no es polilínea, es rechazada. Después podemos escoger entre varias opciones:

Use Points: Significa que cuando esta polilínea sea definida como malla, habrá al menos un nodo sobre cada punto que define la polilínea, y que son extremos de las líneas interiores.

NoUsePoints: Al generar la malla en la polilínea, el programa no tiene en cuenta los puntos que definen la polilínea y los nodos se sitúan en cualquier lugar. Esta es la opción por defecto.

OnlyPoints: Al generar la malla en esta polilínea, habrá nodos sólo donde existan puntos geométricos.

HidePoint: Es una opción de visualización. Los puntos interiores no son mostrados en la ventana gráfica.

ShowPoints: Los puntos interiores son dibujados en la ventana gráfica. Esta es la opción por defecto.

#### 6.2.5.4.4 Editar línea Recta (Edit spline)

Una vez que la spline es seleccionada, podemos editar de una manera interactiva sus puntos de control.!!!!!!! Podemos activar y desactivar los puntos de control e introducir su nueva posición de la manera habitual.!!!!!!!

#### 6.2.5.4.5 Dividir (Divide)

Con el comando 'Divide' de debe seleccionar una línea. Si esta línea no es una polilínea, el usuario debe introducir el número de subdivisiones. La línea será convertida en una serie de líneas de igual longitud sobre la antigua línea, que desaparece como tal.

\*\*\*\*\* Ver referencia 6.2.5.2.1.5 Polyline creation

!!!!!!! Ver referencia 6.2.5.2.1.4 Spline line creation

!!!!!!! Ver referencia 6.2.5.2.1.1 Point creation

Si es una polilínea, el usuario debe elegir un punto interior existente. La polilínea se, a convertida en dos líneas que podrán ser polilíneas o no.

Atención: El punto interior debe ajustarse a uno perteneciente a la polilínea.\*\*\*\*\*

Atención: Las líneas con entidades sobre ellas no pueden ser divididas.

#### 6.2.5.4.6 Editar Nodos de una Línea (Join lines)

Con el comando 'JoinLines', se deben seleccionar dos líneas. *GiD* comunica la distancia entre los dos extremos más cercanos, dibuja ambos puntos y pide la confirmación. Si una de las líneas es una polilínea, los puntos interiores también son considerados.

Si se acepta, los puntos son convertidos en uno sólo y la línea resulta deformada. El nuevo punto se situará en el lugar del punto de la primera línea.

Nota: Move point: Es otra manera de convertir dos puntos en uno sólo.

La opción 'Interior' busca los puntos más cercanos entre las dos líneas, comunica la mínima distancia entre ellos, los dibuja y pide confirmación.

Si se acepta, las dos líneas se convierten en cuatro, y el segundo punto se desplaza hasta quedar sobre el de la primera línea, convirtiéndose ambos en uno sólo (Si los puntos internos de una o dos de las rectas coinciden con sus extremos, se convertirán sólo en tres o dos nuevas líneas respectivamente). Las polilíneas no aceptan esta opción.

Atención: Las líneas con entidades sobre ellas no pueden ser unidas.

#### 6.2.5.4.7 Editar Arco (Swap arcs)

Con este comando se deben seleccionar líneas. Las líneas que no son arcos son rechazadas. Después de la confirmación, el arco se convierte en otro nuevo con el mismo centro y en el mismo plano, pero opuesto al antiguo, que desaparece. El ángulo del nuevo arco será suplementario al ángulo del antiguo arco.

Atención: Los arcos con entidades sobre ellos no pueden ser cambiados.

---

\*\*\*\*\* Ver referencia 6.2.3.2.1.5 Polyline creation



### 6.2.5.5 Condiciones de borde (Conditions)

Las condiciones son todas las propiedades de un trabajo que realicemos con *GiD*, excepto materiales, que pueden ser asociados a una entidad.

#### *Conditions:*

*Assign Condition:*

*Draw Condition:*

*Unassign condition:*

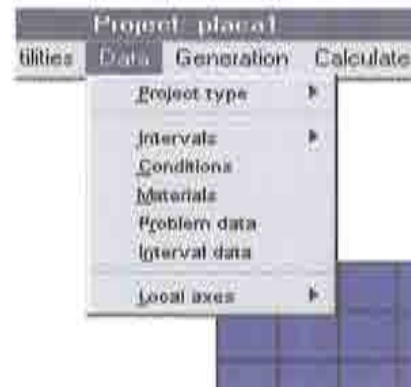


Fig. 6.12 Datos de la condición

#### 6.2.5.5.1 Asignar Condiciones (Assign Condition)

Las condiciones son asignadas a entidades seleccionadas con los campos de valor ya determinados.

Si se asigna desde la ventana gráfica, los campos de valor son cambiados cada vez, y el botón 'Assign' debe ser presionado otra vez.

Si se asigna desde el comando 'AssignCond', 'Change' permite definir los campos de valor. No olvide cambiar estos valores antes de asignar. La opción 'DeleteAll' borra todas las asignaciones de esa condición en particular.

Las condiciones pueden ser asignadas tanto en la geometría como en la malla, pero generalmente es mejor asignarlas en la geometría porque las condiciones son transferidas a la malla. Además en la malla algunas condiciones no pueden ser asignadas correctamente.

**Atención:** Aunque una malla haya sido ya generada, si cambiamos algo en la asignación de condiciones, es necesario generar la malla de nuevo.

#### 6.2.5.5.2 Dibujar Condiciones (Draw Condition)

La opción 'Draw all' dibuja todas las condiciones asignadas a todas las entidades de la ventana gráfica. Dibujar significa que muestra un símbolo gráfico o un número de condición sobre cada entidad que tiene asignada dicha condición.



Si seleccionamos una condición en particular, podemos elegir 'draw' y cualesquiera de los campos desde una o todas las condiciones (Desplazamiento, giros, restricciones en x, y, z).

Si elegimos un sector, el valor de este sector es escrito sobre todas las entidades que llevan esta condición asignada.

#### 6.2.5.5.3 Eliminar condiciones (Unassign condition)

En modo de ventana gráfica, el comando 'UnAssign' se permite al usuario escoger entre desasignar estas condiciones de las entidades que las tienen o desasignar todas las condiciones.

En modo de comando 'UnAssign' la desasignación se realiza para todas las condiciones. Para hacerlo sólo con una condición utiliza 'Delete All'.

#### 6.2.5.6. Materiales de base (Materials)

Cuando exista un cálculo que necesita materiales, hay una base de datos de materiales existentes que pueden ser asignados a las entidades. El usuario puede también crear nuevos materiales derivados de otros existentes y asignarlos también.

*Materials:*

*Assign material:*

*Draw material:*

*Unassign material:*

*New material:*



Fig.6.13 Materiales

#### 6.2.5.6.1 Asignar Material (Assign material)

El material es asignado a las entidades seleccionadas.

Si se asigna desde la ventana gráfica, cada vez el material a asignar cambia, y el botón 'Assign' debe ser presionado de nuevo.

Si se asigna desde la opción de comando 'UnAssignMat' se borran todas las asignaciones de este material en particular.

**Atención:** Aunque una malla haya sido ya generada, si cambiamos algo en la asignación de materiales, es necesario generar la malla de nuevo o asignar materiales directamente a la malla.

#### **6.2.5.6.2 Dibujar Material (Draw material)**

Muestra un número indicando el material seleccionado en todas las entidades a las que ha sido asignado dicho material.

#### **6.2.5.6.3 Eliminar Material (Unassign material)**

El comando 'UnAssign', desasigna todos los materiales de todas las entidades. Para desasignar sólo un material utilice 'UnAssignMat'.<sup>\*\*\*\*\*</sup>

#### **6.2.5.6.4 Nuevo Material (New material)**

Si se usa el comando 'NewMaterial' se crea un nuevo material tomando un material existente como base. Material base significa que el nuevo tendrá los mismos campos que él. Entonces, todos los nuevos valores de los sectores son introducidos en la línea de comandos.

Es posible redefinir un material existente.

Para crear un nuevo material o redefinir uno antiguo en la ventana de materiales, escriba un nuevo nombre o el mismo y cambie algunas de las propiedades. Después pulse 'Accept'.

#### **6.2.5.7 Datos del Problema (Problem data)**

Los datos de un problema, son los datos generales del problema. Esto significa que no están relacionados a una entidad geométrica y no cambian en cada intervalo.

Pueden ser introducidos con el comando 'ProblemData' o en la ventana de datos del problema.

Si se introducen desde la ventana, los datos no son aceptados hasta que se presiona el botón 'Accept'.

Estos datos pueden ser introducidos antes o después de la generación de la malla.

#### **6.2.5.8 Intervalo de Datos (Interval data)**

El intervalo es la manera de separar en diversos grupos. Los datos de intervalo y las condiciones es información que se puede duplicar en cada grupo si se desea. Cuando se define un nuevo intervalo podemos copiar toda la información sobre las condiciones asignadas a las entidades no. Por lo tanto, la manera correcta de trabajar es definir primero todas las condiciones y después crear los nuevos intervalos.

---

<sup>\*\*\*\*\*</sup> Ver referencia 6.2.5.5.1 Assign material

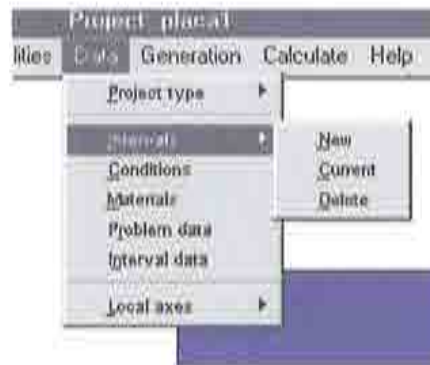


Fig.6.14 Intervalo de Datos

Podemos definir tantos intervalos como se desee con el comando 'ChangeInterval'.

Estos grupos llamados intervalos, pueden ser usados para cambiar algunas condiciones o información como factores de incremento en un análisis de incrementos. Puede ser también útil definir algunos estados de carga para la misma geometría.

Estos datos pueden ser introducidos con el comando 'IntervalData' o en la ventana de datos de intervalo.

Si se introducen en la ventana, los datos no son aceptados hasta que se presiona el botón 'Accept'. Estos datos pueden ser introducidos antes o después de generar la malla.

#### 6.2.5.9 Malla (Meshing)

Generar una malla es el proceso por el cual se obtiene una malla a partir de una geometría definida. Esta malla puede ser usada más tarde para el análisis FEM. Las condiciones asignadas a las entidades geométricas serán transferidas a los nodos y elementos de la nueva. <sup>††††††††††</sup>

La malla y su generación, resulta controlada por algunos valores por defecto que pueden ser cambiados con los comandos que se describen después.

La generación no depende del estado de las capas en el momento de la generación. En todas las capas se generan mallas y todos los nodos y elementos serán colocados en la capa en la que estaba la entidad geométrica original.

Los valores por defecto son:

- a. En una entidad se genera malla si no hay ninguna entidad sobre ella.
- b. Las líneas de la malla están constituidas por elementos de dos nodos. Las superficies de la malla son elementos triangulares no estructurados. El valor por defecto para mallas estructuradas son elementos de cuatro lados. Los elementos de volumen son elementos de cuatro lados no estructurados, y los estructurados son de seis lados.
- c. Los elementos de contacto son por defecto de cuatro lados o prismas.

<sup>††††††††††</sup> Ver referencia 6.2.5.4.1 Assing condition

Todos estos elementos son de tipo lineal.

**Generation:**

*Generating mesh:*

*Mesh view:*

*Assign sizes:*

*Structured mesh:*

*Element type:*

*Reset:*

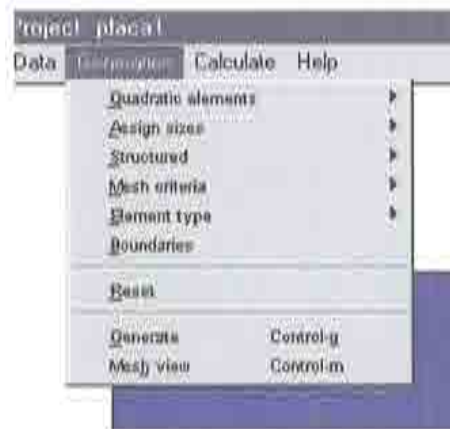


Fig.6.15 Generación de malla

### 6.2.5.9.1 Generación de Malla (Generating mesh)

Cuando todo esté listo, seleccione este comando. Si existe una malla generada previamente, *GiD* pregunta si debe cancelarla. Ello significa hacerla desaparecer (de la memoria, no del disco hasta el próximo salvado).<sup>#####</sup> Si se solicita el nombre de un generador de mallas utilice 'rfast'. Después es requerido a dar un tamaño general a los elementos. Este tamaño será aplicado a todas las líneas, superficies y volúmenes que no tengan otro definido anteriormente.<sup>#####</sup>

Este tamaño es la altura media de los triángulos o cuadriláteros.

### 6.2.5.9.2 Ver Malla (Mesh view)

Cuando una malla ha sido ya generada, esta opción cambia la visualización de la geometría original a la de la malla. El camino contrario, o sea, volver a la geometría, es automáticamente efectuado al seleccionar el menú 'Geometry' o algún comando relacionado con la geometría.

### 6.2.5.9.3 Asignar tamaño (Assign sizes)

El tamaño es la altura media del triángulo o cuadrilátero.

Es posible asignar diferentes tamaños a diferentes entidades de la malla. Esto significa que cerca de estas entidades, los elementos generados tendrán aproximadamente ese tamaño. Todas las entidades que no tienen un tamaño asignado al generar la malla, toman el tamaño por defecto. Los puntos no toman ningún tamaño si no se les asigna.

Si se aplica un tamaño 0, 0 a una entidad, es lo mismo que darle un tamaño libre (se aplica el tamaño por defecto).

<sup>#####</sup> Ver referencin 6.2.5.1.1 Save

<sup>#####</sup> Ver referencin 6.2.5.8.3 Assign sizes



La opción 'ByGeometry' pide al usuario un tamaño máximo y mínimo asigna tamaño a todas las entidades de la forma de la geometría. Esto significa que las superficies pequeñas tendrán elementos más pequeños. Esto sólo modifica los tamaños existentes si el nuevo es más pequeño.

Atención: Tenga cuidado al asignar tamaños grandes a entidades cercanas a otras con un tamaño asignado mucho más pequeño. La generación de malla puede resultar imposible.

Atención 2: Cuando use elementos de contacto,\*\*\*\*\* se deberá utilizar el mismo tamaño para las entidades de contacto y las entidades duplicadas.

#### 6.2.5.9.4 Malla Estructurada (Structured mesh)

Una malla estructurada se define como aquella en la que todos los nodos tienen el mismo número de elementos alrededor de ellos. Esto significa que es una malla regular con bastantes elementos iguales.



Fig.6.16 Estructura de la Malla

El tamaño de los elementos es definido de una manera diferente que el de las mallas no estructuradas.

En este caso, no se define un tamaño, sino que en su lugar, se define el número de elementos que habrá sobre cada línea que se da. Este número debe ser el mismo para líneas opuestas en cada superficie.

Cuando se genera malla en volúmenes, esta definición debe ser la misma para las superficies opuestas.

Para crear mallas estructuradas en superficies, elija 'structured surfaces' y seleccione 4-sided surfaces. Las superficies planas no pueden ser estructuradas. Después de elegir 'escape', es necesario seleccionar una línea o grupo de líneas que pertenezcan a las superficies seleccionadas. Pulse 'escape' para acabar.

Después es necesario determinar el número de elementos definidos sobre este grupo de líneas. Repita la operación hasta el final. Adverta que cuando una línea es seleccionada, todas las líneas opuestas son automáticamente seleccionadas también.

Por defecto, los elementos generados serán cuadriláteros y las líneas sin número asignado tendrán dos elementos sobre ellas.

\*\*\*\*\* Ver referencia 6.2.5.2.1 .9 Contao creation

Para crear mallas estructuradas en volúmenes, elija 'structured volumes' y seleccione volúmenes que tengan seis superficies.

Atención: El usuario debe asegurarse de que todas las superficies que pertenecen a los volúmenes seleccionados son también estructurados y tienen el número correcto y la definición de los elementos para obtener volúmenes coherentes.

Es posible mezclar entidades de una malla estructurada con otras de una malla no estructurada.

Para convertir una entidad estructurada en otra no estructurada, seleccione reset <sup>††††††††††</sup> o asigne un tamaño. <sup>††††††††††</sup>

#### 6.2.5.9.5 Tipo de Elemento (Element type)

Con este comando se realiza la selección del tipo de elemento deseado. Sólo es necesario cuando este tipo es diferente del tipo por defecto. <sup>§§§§§§§§§§</sup>

Los tipos son:

- a. *Lineal:*  
*Para líneas.*
- b. *Triángulos y cuadriláteros:*  
*Para superficies.*
- c. *Tetraedros y hexaedros:*  
*Para volúmenes.*
- d. *Lineal, cuadriláteros, prisma y hexaedros:*  
*Para contactos.*



Fig.6.17 Tipo de Elemento

Por defecto, los elementos son de grado mínimo: Los de 3 nodos son triángulos, los de 4 nodos cuadriláteros, y así sucesivamente.

Para incrementar el grado use el comando 'quadratic'. 'Quadratic' se aplica a todos los elementos del problema. Para utilizar de nuevo elementos lineales use otra vez 'Quadratic'.

Los elementos cuadráticos son:

- Lineal: 3 nodos.
- Triángulo: 6 nodos.
- Cuadriláteros: 8 nodos.

<sup>††††††††††</sup> Ver referencia 6.2.5.8.6 Reset

<sup>††††††††††</sup> Ver Referencia 6.2.5.8.3 Assign sizes

<sup>§§§§§§§§§§</sup> Ver referencia 6.2.5.8 Meshing



Para forzar una entidad, para su generación de malla o no, elija 'Mesh' o 'No mesh' y seleccione la entidad. Esto significa que su malla asociada será parte de la malla final o no, prescindiendo de los valores por defecto.

#### 6.2.5.9.6 Regenerar (Reset)

Con este comando todos los tamaños asignados a las entidades son reseteados. Esto significa que todos ellos son desasignados.

Para desasignar sólo parte de ellos, asigne el tamaño 0. 0<sup>\*\*\*\*\*</sup> a las entidades que deban mantener el tamaño por defecto.

#### 6.2.5.10 Opciones de Visualización (View options)

Los comandos de visualización cambian el modo en que nosotros vemos todo en la ventana gráfica. No cambian nada de la definición de la geometría o cualquier otro dato.

Generalmente, se pueden utilizar mientras se está usando cualquier otro comando sin tener que salir de él. Cuando el comando de visualización está acabado, el otro continúa.

##### Menú:

Zoom:  
Rotation:  
Pan:  
Redraw:  
Render:  
Label:  
Entities:  
Layers:

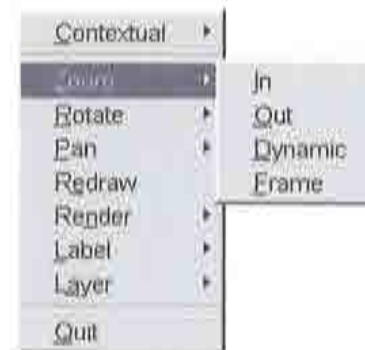


Fig.6.18 Puntos de Vista

##### 6.2.5.10.1 Zoom

Zoom es usado para cambiar el tamaño visualizado del objeto. Esto no deforma los objetos, sólo los hace más grandes o pequeños y cambia la perspectiva de la ventana.

- Zoom in: Seleccione con el ratón la ventana gráfica. Se abrirá un rectángulo dinámico. Seleccionar de nuevo y la visualización cambia de manera que se ve sólo lo que estaba definido dentro del rectángulo.
- Zoom out: Seleccione con el ratón la ventana gráfica. Se abrirá un rectángulo dinámico. Seleccionar de nuevo y la visualización cambia de manera que todo en la ventana se reduce hasta alcanzar el tamaño del rectángulo.
- Zoom frame: Elija un tamaño de visualización de manera que se vea todo lo que hay en la ventana.

\*\*\*\*\* Ver Referencia 6.2.5.8.3 Assign sizes

### 6.2.5.10.2 Rotación (Rotation)

Hay varias maneras de efectuar una rotación. Rotar significa cambiar la vista gráfica de la geometría. La geometría no cambia.

#### *Rotate:*

*Rotate ScrAxes:*  
*Rotate ObjAxes:*  
*Rotate Trackball:*  
*Rotate Angle:*  
*Rotate points:*  
*Rotate center:*

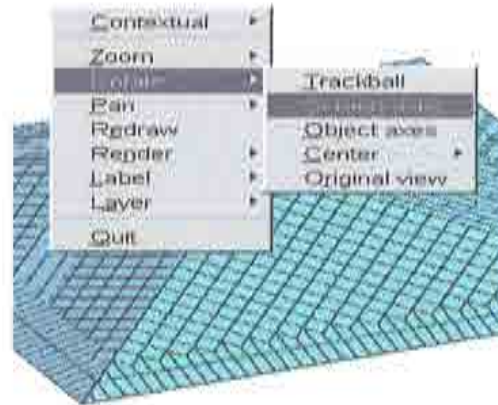


Fig.6.19 Tipo de rotaciones

#### 6.2.5.10.2.1 Rotación Dinámica (Rotate ScrAxes)

Con esta opción se realiza una rotación dinámica sobre ejes relativos a la pantalla, esto es:

- a. 'eje x': Un eje horizontal.
- b. 'eje y': Un eje vertical.
- c. 'eje z': Un eje ortogonal a la pantalla.

Cuando se introduce este comando, el 'eje z' es elegido por defecto, y moviendo el ratón a la izquierda y a la derecha, la geometría rotará alrededor de dicho eje. Pulsando el botón izquierdo del ratón se puede cambiar el eje. Para abandonar esta función use 'escape'.<sup>!!!!!!!</sup>

Para cambiar los ejes es también posible introducir las letras 'x', 'y' o 'z' directamente en la línea de comandos.

Para rotar la geometría una cantidad determinada de grados, introducir el número, positivo o negativo, en la línea de comandos.

<sup>!!!!!!!</sup> Ver referencia 6.2.4 escape

#### 6.2.5.10.2.2 Rotación sobre ejes (Rotate ObjAxes)

Con esta opción se realiza una rotación dinámica sobre ejes relativos al propio objeto. Esto significa que los ejes globales en la posición en la que se encuentran en dicho momento. En otro caso, serán los ejes dibujados en la ventana gráfica.

Cuando se utiliza este comando, el 'eje z' es elegido por defecto y, moviendo el ratón a la izquierda y a la derecha, la geometría rotará alrededor de dicho eje. Pulsando el botón izquierdo del ratón se puede cambiar el eje. Para abandonar la función, use 'escape'. \*\*\*\*\*

Para cambiar los ejes, también es posible introducir las letras 'x', 'y' o 'z' directamente en la línea de comandos.

Para rotar la geometría una cantidad determinada de grados, introducir el número, positivo o negativo, en la línea de comandos.

#### 6.2.5.10.2.3 Rotación hacia atrás (Rotate Trackball)

Con esta opción se realiza una rotación dinámica a la manera de un TRACKBALL DEVICE. Esto significa que cuando se pica sobre un punto de la geometría con el botón izquierdo del ratón y moviéndolo, dicho punto trata de ir detrás de la flecha del ratón. Puede imaginarse como una bola sobre la ventana gráfica que se mueve con el ratón.

El botón izquierdo puede pulsarse en repetidas ocasiones para conectar y desconectar el movimiento. Para abandonar la función, use 'escape'.

#### 6.2.5.10.2.4 Rotación de Angulo (Rotate Angle)

La nueva posición de la geometría después de la rotación puede ser definida a través de un par de ángulos:

- a. El ángulo en el plano 'XY' empezando en el 'eje x'.
- b. El ángulo de elevación respecto al plano 'XY'.

Por ejemplo, la vista inicial (ortogonal al 'eje z' y con el 'eje x' horizontal) puede ser obtenida con:  
rotate angle 270 90

#### 6.2.5.10.2.5 Rotación de puntos (Rotate points)

La nueva posición de la geometría después de la rotación puede ser definida a través de un par de puntos:

- a. El punto 'objetivo', desde el cual se está mirando.
- b. El punto de vista, al cual se está mirando.

---

\*\*\*\*\* Ver referencia 6.2.4 escape

### 6.2.5.10.2.6 Rotación central (Rotate center)

El centro de rotación por defecto se define aproximadamente en el centro de la geometría. Si se desea cambiar este centro, use este comando <sup>#####</sup> e introduzca un punto. Este centro se mantendrá hasta el próximo 'zoom frame' <sup>\*\*\*\*\*</sup>.

### 6.2.5.10.3 Desplazamiento (Pan)

Con este comando la vista de la geometría se desplaza en la ventana gráfica.  
Para hacerlo, pique dos puntos en la ventana gráfica.

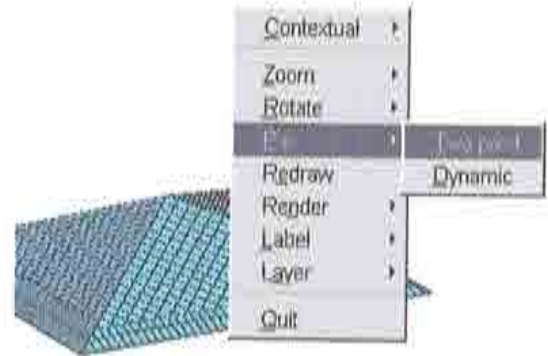


Fig. 6.20 Desplazamientos

### 6.2.5.10.4 Redibujar (Redraw)

Este comando redibuja el modelo en la ventana gráfica.

### 6.2.5.10.5 Render

Con esa opción cambia el modo de vista del modelo. Existen tres opciones:

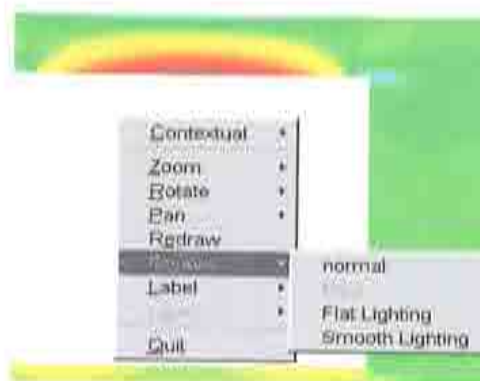


Fig. 6.21 Render

<sup>#####</sup> Ver referencia 6.2.5.2.1.1 Point creation.  
<sup>\*\*\*\*\*</sup> Ver referencia 6.2.5.9.1 Zoom.

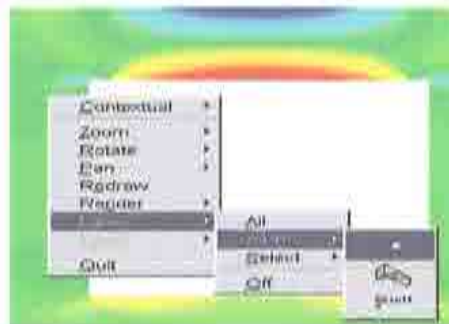


- a. 'Normal': Es el modo habitual de visualizado. La geometría y la malla se ven según las líneas que las definen.
- b. 'Rendering': El modelo es iluminado. Presenta el aspecto de un sólido iluminado por un foco de luz.
- c. 'Polígonos': Es útil sólo para mallas. La malla se representa en un color opaco con las líneas dibujadas sobre ella para poder distinguir los diferentes elementos

#### 6.2.5.10.6 Nivel (Label)

Con esta opción las etiquetas de las entidades son dibujadas o no. Es posible seleccionar varias entidades o aplicar este comando a todas las entidades que aparecen en la ventana gráfica. Para seleccionar algunas entidades elija 'select' antes de aplicar la etiqueta a una de ellas: 'puntos, líneas, superficies o volúmenes'. Después seleccione las entidades de la manera habitual. Las opciones son:

- a. 'All': Todas las entidades de la ventana gráfica mostrarán su etiqueta.
- b. 'Puntos, líneas, superficies o volúmenes': Si 'Select' no se activa, todas las entidades de este tipo en la ventana gráfica mostrarán su etiqueta.
- c. 'Off': Todas las entidades dejan de mostrar su etiqueta.



*Fig.6.22 Niveles*

#### 6.2.5.10.7 Entidades (Entities)

Con esta opción, podemos decidir si algunos de los 'puntos, líneas, superficies o volúmenes no deben ser dibujados. Esto puede ser interesante para hacer el dibujo más rápido o más claro en algún momento.

### 6.2.5.10.8 Capas (Layers)

Las capas son un medio de separar un dibujo complejo en diversas partes. La idea es que cualquier entidad puede pertenecer a una capa o a ninguna. Entonces, es posible ver sólo algunas capas. Es también útil para una mejor selección de entidades en la ventana gráfica.

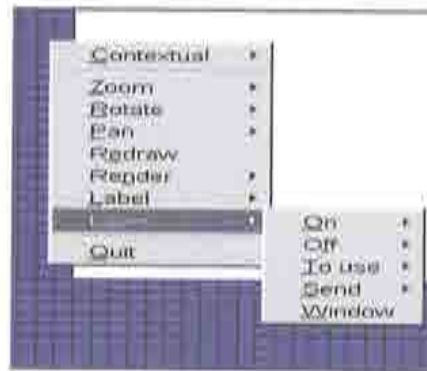


Fig.6.23 Capas

Los comandos relacionados con las capas son:

'New': Para crear una nueva capa, esta será utilizada como capa por defecto hasta el final de la sesión o hasta que es cambiada.

'To use': Selecciona una capa para que sea la capa por defecto. Todas las nuevas entidades serán creadas en esta capa.

'On': Coloca la capa en posición 'On'. Las entidades que pertenezcan a esta capa serán dibujadas y podrán ser seleccionadas en la ventana gráfica.

'Off': Coloca la capa en posición 'Off'. Las entidades que pertenezcan a esta capa no serán dibujadas y no podrán ser seleccionadas en la ventana gráfica.

'Freeze': Coloca la capa en situación 'Freeze'. Las entidades que pertenecen a esta capa serán dibujadas pero no podrán ser seleccionadas en la ventana gráfica.

'Delete': Borra la capa. Una capa sólo puede ser borrada si no tiene entidades en ella.

'View': Da información sobre la capa.

'Entities': Se utiliza para cambiar entidades a una nueva capa. No crea nuevas entidades, sólo las mueve de una capa a otra. Cuando se escoge 'point, line, surface, volume or All', la opción 'All' puede ser usada sólo para seleccionar entidades en la ventana gráfica y cambiar todas las entidades de la caja dinámica a la nueva capa.





### 6.2.5.11.1 Variables

Existen algunas definiciones previas o modos de trabajar que pueden ser activadas en *GiD*. Pueden ser activadas de dos maneras: A través de la ventana de 'Preferences' o con el comando 'Variables'. En la siguiente descripción, la opción preferencias se muestran ofreciendo sus variables asociadas.

El primer grupo de preferencias activa diferentes maneras de trabajar con *GiD*.

'Dialog browser: ' Si se activa, el nombre del archivo es requerido al usuario mediante una ventana de exploración. Si no, el nombre del archivo es escrito en la línea de comandos; Variable: 'Dialog Wrowser' values: 1, 0.

'Epsilon DXF: ' Cuando se lee un archivo *DXF*, los puntos más cercanos de dicha distancia son considerados como uno sólo, \*\*\*\*\* Variable: 'EpsilonDXF' value: the same.

'Fast Selection: ' Si se activa, la selección es de tipo rápido. Si no, será de tipo normal. Variable: 'Fast Selection' values 1, 0.

'Create always new point: ' Cambia la manera de introducir los puntos en *GiD*. Las opciones son:

- 'Yes: ' Crea siempre un nuevo punto aunque se intente crear muy cerca de otro ya existente.

- 'Ask: ' Si se intenta crear un nuevo punto cerca de otro ya existente, *GiD* pregunta al usuario si quiere coger el punto antiguo o crear uno nuevo.

- 'No: ' Si se intenta crear un nuevo punto, *GiD* coge automáticamente el antiguo y escribe un mensaje avisando de ello.

- 'Force no: ' Sólo permite coger puntos ya existentes. Puede ser cambiado de forma interactiva activando 'No join' hasta el final de la introducción de puntos. Variable: 'Create Always New Point' values: 1, 0, -1, -2. Default is 'Ask'.

El segundo grupo de preferencias activa diferentes maneras de visualizar el modelo. No cambian ni la geometría ni la información del modelo.

'Light Smoothed Elements: ' Si se activa, cuando la malla es 'renderizada', \*\*\*\*\* la intersección entre elementos con ángulo pequeño será iluminada como si fuera un sólido continuo. Si no, la iluminación será realizada considerando plano cada elemento. Variable: 'Light Smoothed Elems' values: 1, 0.

'Cos Smoothed Elements: ' Si se activa la última propiedad, éste muestra el valor del coseno del ángulo entre las normales de los dos elementos. Si el ángulo actual es más pequeño, la intersección será suavizada. Si no, el borde será dibujado. Variable: 'Cos Smoothed Elems' value: the same.

'Number of lighting subdivisions: Cuando una geometría es 'renderizada', este número activa la calidad de la iluminación. Cuantas más divisiones haya, mayor será la calidad y el dibujo será más lento. Variable: 'Num Of Lighting Subdivisions' value: the same.

\*\*\*\*\* Ver referencia 6.2.5.1.7 DXF read  
\*\*\*\*\* Ver referencia 6.2.5.9.5 Render

'Curve Precision: ' Marca la precisión utilizada al dibujar las curvas. La definición interna de las curvas no cambia. En la ventana de preferencias es posible cambiar de manera dinámica el trazado de una curva con el fin de comprobar su precisión. Variable: 'Curve Precision'. Value: 1 to 0.001 from best to worst.

'Fast rotation: ' Si se activa, podemos escoger entre varias opciones, que serán aplicadas sólo al girar. Estas opciones son:

- 'points, lines, surfaces, volumes: ' Dibuja o no este tipo de entidades al girar. Variables: the same. Values: 0, 1.

- 'Always Geom: ' Si se activa, al girar se muestra la geometría en lugar de la malla. Variable: 'UseAlwaysGeom'. Values: 0, 1.

'Curve Precision: ' Similar a la opción general 'Curve Precision' pero se aplica sólo al rotar. Variable 'FastRotation'. Value: 0, 1.

#### **6.2.5.11.2 Renumerar (Renumber)**

Al crear nuevas entidades, la etiqueta de las nuevas entidades será el número más bajo mayor que cero y que todavía no exista para este tipo de entidades.

Si borramos una entidad, aparece un espacio libre en la lista de etiquetas. Este agujero será reemplazado por nuevas entidades. Si deseamos cambiar las etiquetas de las entidades con el fin de evitar dichos agujeros, podemos renumerar la geometría.

No se producen problemas con los materiales y las condiciones aplicadas a las entidades.

Si estamos en modo de generación de malla, las entidades renumeradas son las de la malla. En este caso, la renumeración no sólo rellena los agujeros en la lista de etiquetas sino que además cambia el número de los nodos con el fin de minimizar la diferencia de números de nodo en cada elemento.

Esto puede ser útil cuando el módulo de cálculo usa métodos de almacenamiento por bandas o siluetas.

'Nota: ' Usualmente no es necesario utilizar este comando porque se aplica automáticamente al generar la malla.

#### **6.2.5.11.3 Identificación (Id)**

Este comando muestra la etiqueta y las coordenadas de un punto existente.

#### **6.2.5.11.4 Lista (List)**

El comando 'List' muestra información sobre las entidades seleccionadas. Dicha información es sólo de lectura. La opción 'Status' ofrece información sobre datos generales interesantes del proyecto.

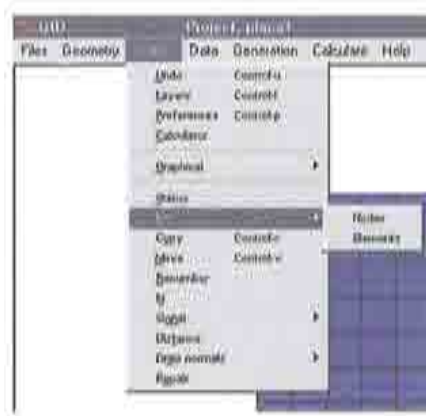


Fig. 6.26 Lista

#### 6.2.5.11.5 Distancia (Dist)

El comando 'Dist' muestra la distancia entre dos puntos existentes.

#### 6.2.5.11.6 Dibujar Normales (Draw Normals)

El comando 'DrawNormals' dibuja la normal de las superficies seleccionadas. Si se escoge volumen se dibujan las normales del volumen. Todas ellas deben señalar hacia el interior de dicho volumen.

La normal de una superficie que pertenezca al plano ' $z=0$ ', por defecto, está orientada hacia el sentido positivo del eje ' $z$ '. Entonces, son definidas en sentido antihorario en 2D.

Los volúmenes son orientados correctamente ignorando la orientación de sus superficies.

#### 6.2.5.11.7 Cambiar Sentido (Swap senses)

El comando 'SwapSenses' cambia el sentido de las normales de las superficies seleccionadas. Deben ser cambiadas una a una, confirmando cada una de ellas.

La opción 'Volume' no debe ser usada.

#### 6.2.5.11.8 Copiar (Copy)

El comando 'Copy' es una función general que permite al usuario seleccionar un grupo de entidades y copiarlas con un movimiento que puede ser de translación, rotación, simetría o escalado.

El proceso es:

Seleccione todas las superficies que desea copiar. Si no hay ninguna pulse 'escape'. Después, seleccione las líneas que desea copiar. Las líneas que soportan las superficies seleccionadas son seleccionadas automáticamente. No es necesario seleccionarlás otra vez. Haga lo mismo para los puntos. Después de presionar 'escape', elija el tipo de movimiento, y después, los parámetros de dicho movimiento. Las opciones son:

- a. Rotación: Es necesario introducir dos puntos o un punto en 2D. Estos dos puntos definen el eje de rotación y su orientación. En 2D el eje va desde el eje definido hacia el sentido positivo del eje 'z'. Introduzca el ángulo de rotación en grados. Éste puede ser positivo o negativo. El sentido es definido por la regla de la mano derecha. En 2D, el sentido es antihorario.
- b. Translación: Es definido por dos puntos. Se pueden obtener movimientos relativos mediante la definición relativa del segundo punto.
- c. Simetría: Es definida por tres puntos que no pueden estar alineados. Dichos puntos forman un plano, que es el plano de simetría. En 2D, el eje de simetría se define mediante dos puntos.
- d. Escala: Es definida mediante un centro y un punto. Cada coordenada del punto es el factor de escala para cada eje 'x, y, z'. Factores de escala mayores que 1, aumentan el tamaño, y menores, lo disminuyen. También pueden ser negativos.

Después de definir el movimiento, *GiD* pregunta al usuario si debe compartir las antiguas entidades o no. Si se responde que sí, el punto nuevo va a parar sobre otro ya existente de la geometría, son convertidos en uno sólo. Las líneas rectas que comparten ambos extremos también son convertidas en una sola. Si se responde que no, todas las nuevas entidades se crean sin tener en cuenta las ya existentes.

#### 6.2.5.11.9 Reparar

(Repair)

Esta opción comprueba la coherencia de la información de la base de datos. Úsela sólo si tiene graves problemas.

#### 6.2.5.12 Salir

(Quit)

El comando 'Quit' es usado para acabar la sesión de trabajo. Si han habido cambios desde el último salvado, *GiD* pregunta si debe salvarlos.

## INDICE DE CONCEPTOS

Ref	Uso de la interfase		* 6.2.1
Ref	Definición de Puntos		* 6.2.2
Ref	Selección de Entidades		* 6.2.3
Ref	Escape		* 6.2.4
Ref	Reference manual		* 6.2.5
Ref	Archivos		* 6.2.5.1
Ref		Save	
Ref		Read	
Ref		New	
Ref		Save layer	
Ref		Write Ascii	
Ref		Mesh read	
Ref		DXF read	
Ref		Write calculations file	
Ref		Batch file	
Ref	Creación de entidades Geométricas		* 6.2.5.2
Ref		Point creation	
Ref		Straight line creation	
Ref		Arc line creation	
Ref		Spline line creation	
Ref		Polyline creation	
Ref		Planar surface creation	
Ref		4-sided surface creation	
Ref		Volume creation	
Ref		Contact creation	
Ref	Eliminación de entidades geométricas	(Delete)	* 6.2.5.3
Ref	Edición de entidades Geométricas	(Editing geometrical entities)	* 6.2.5.4
Ref		Move point	
Ref		Explode polyline	
Ref		Edit polyline	
Ref		Edit spline	
Ref		Divide	
Ref		Join lines	
Ref		Swap arcs	
Ref	Condiciones	(Conditions)	* 6.2.5.5
Ref		Assign Condition	
Ref		Draw Condition	
Ref		Unassign condition	
Ref	Materiales	(Materials)	* 6.2.5.6
Ref	Asign material		
Ref	Draw material		
Ref	Unassign material		
Ref	New material		
Ref	Datos del Problemas	(Problem data)	* 6.2.5.7
Ref	Intervalo de Datos	(Interval data)	* 6.2.5.8
Ref	Malla	(Meshing)	* 6.2.5.9
Ref		Generating mesh	
Ref		Mesh view	
Ref		Assign sizes	
Ref		Structured mesh	
Ref		Element type	
Ref		Reset	
Ref	Opciones de Visualización	(View options)	* 6.2.5.10
Ref		Zoom	
Ref		Rotation	
Ref		Rotate SciAxes	
Ref		Rotate ObjAxes	
Ref		Rotate Trackball	
Ref		Rotate Angle	
Ref		Rotate points	
Ref		Rotate center	
Ref		Pan	
Ref		Redraw	
Ref		Rendai	
Ref		Label	
Ref		Entities	
Ref		Layers	
Ref	Utilidades	(Utilities)	* 6.2.5.11
Ref		Variables	
Ref		Renumber	
Ref		Id	
Ref		List	
Ref		Dist	
Ref		Draw Normals	
Ref		Swap senses	
Ref		Copy	
Ref		Repair	
Ref	Quit		* 6.2.5.12
Fin de la Tabla			

Tabla de referencias: 6.2





## 7. CONFIGURACIÓN DE GiD PARA EL PREPROCESO

### (Configuración para el Análisis: Tipo de Análisis)

#### 7.1 Introducción

Como ya se ha comentado en capítulos anteriores, *GiD* es un programa de *preproceso* gráfico que realiza la preparación de datos y geometría, para, posteriormente procesarlos en un programa de cálculo por medio del Método de los Elementos Finitos (M.E.F.).

Su función, radica en la modelización de la geometría, que define el problema, mediante técnicas de *CAD* (también es posible la importación de la geometría a partir de otros programas alternativos a este mediante archivos de intercambio estándares). Esta geometría debe adaptarse a una serie de particularidades que permitan realizar la posterior generación de la malla.

Además del tratamiento de *CAD*, el programa procesa el resto de la información necesaria para el cálculo, como: la definición de los diferentes materiales, datos accesorios, etc.; así como la definición de algoritmos a usar, datos del tiempo en un análisis dinámico, criterios de convergencia.

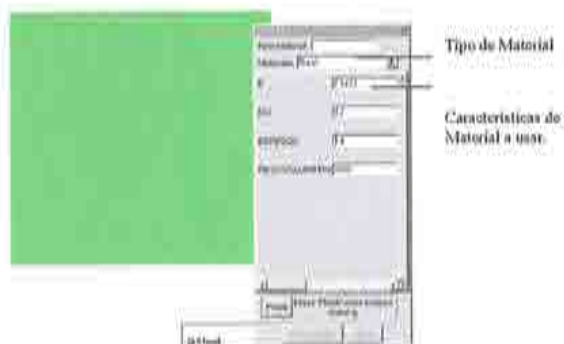


Fig. 7.1 Asignación del Material.

Otro aspecto importante, es la definición de las condiciones de contorno aplicadas a las entidades geométricas y su transferencia a la malla.

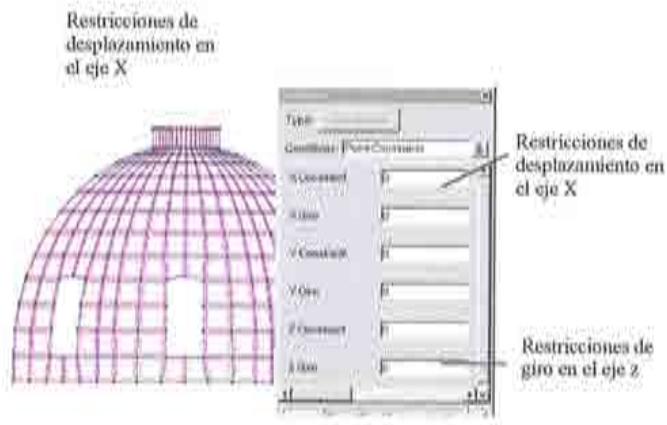


Fig. 7.2 Condiciones de contorno.

En el caso de que *GID* se utilice para un tipo de análisis particular, es necesario predefinir toda la información que le es solicitada al usuario, y precisar el modo en que la información final será dada al módulo del proceso. Para hacerlo, se utilizan ciertos archivos en los cuales se describen las condiciones de contorno, materiales, datos, símbolos y el formato del archivo de salida.

Una vez creada e introducida toda la información, tanto geométrica como de materiales y condiciones de contorno, se genera la malla.

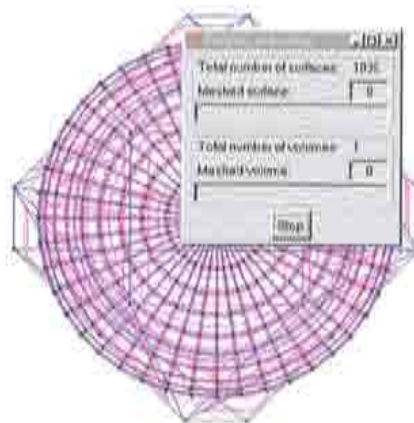


Fig. 7.3 Generación de una malla

Esta generación es automática y el usuario únicamente debe definir el tipo de elementos a usar y la distribución y tamaño que deberán tener.

El último paso es transferir toda esta información al programa de cálculo en un formato que éste pueda comprender.

En esta publicación se ha creado una configuración para la *interface GiD-Sap90*. Sin embargo, puede realizarse cualquier configuración para diferentes programas que utilicen el Método de los Elementos Finitos, ya que la mayoría de ellos utiliza una entrada de datos generalizada.

## 7.2 Adaptación de GiD a diferentes programas de cálculo de estructuras.

Debido al hecho de que *GiD* es un programa de *preproceso* de propósito general, éste debe configurarse para cada tipo de análisis que se desee realizar.

Esta configuración es posible sin necesidad de variar sustancialmente el programa y sin tener que programar o realizar ninguna utilidad adicional.

Se entiende por configurar el programa, a la definición de los datos que son requeridos al usuario, así como a la especificación de los materiales de base que van a ser utilizados.

También hay que dar forma a las condiciones de contorno. Se añade la posibilidad de definir dibujos o esquemas que representan a algunas de las condiciones de contorno.

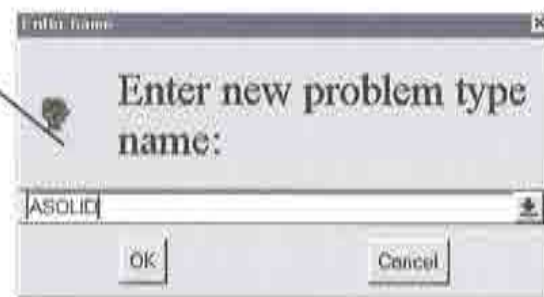
Finalmente, hay que especificar la forma en que se deben escribir los datos generados mediante un fichero de escritura de datos con extensión *\*.bas*. Y que a su vez, estos puedan ser interpretados por el programa de cálculo.

Esta configuración se realiza, primeramente, con la definición del tipo de problema a resolver, más la inclusión de algunos archivos adicionales.

A efectos prácticos esto significa, la creación de un directorio con el nombre del tipo de problema (Asolid, Shell, etc.) y con la extensión *.gid*

Dentro de este directorio se disponen una serie de archivos que conforman la definición del problema. El nombre de estos archivos se forma, en la mayoría de ellos, con el mismo nombre que tienen y con una extensión diferente según la información que contengan.

*En esta ventana se asigna el tipo de problema a resolver. Asolid : Para el caso de deformación ó Tensión plana con Sap90.*



*Fig. 7.4 Asignación del tipo de problema.*

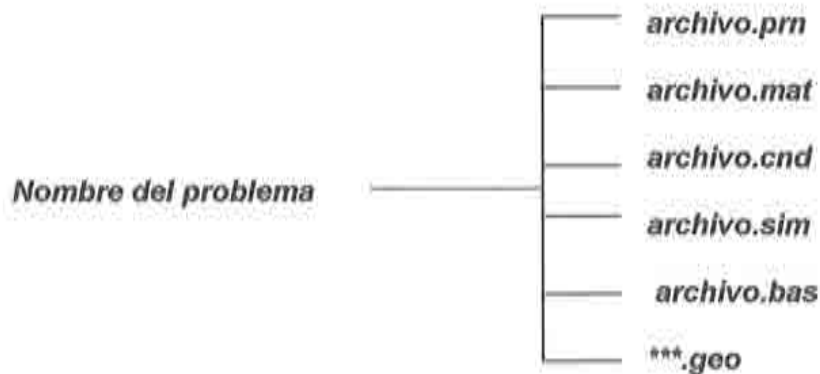
A continuación se da un listado de los archivos contenidos en el directorio que define el problema y su significado.

**Directorio:**

Nombre del Directorio.gid

**Archivos:**

Nombre del Archivo. prb	Datos generales del problema y los datos de intervalo.
Nombre del Archivo. mat	Definición de materiales de base.
Nombre del Archivo. cnd	Definición de las condiciones de contorno.
Nombre del Archivo. sim	Símbolos o dibujos que representan algunas condiciones de contorno.
Nombre del Archivo. bas	Definición de la escritura del fichero de datos requeridos para el cálculo
***.geo	Los nombres con extensión <i>.geo</i> contienen definiciones geométricas de símbolos a utilizar en el proceso.

**Ejemplo de Diagrama de Flujo de un archivo gid:**

*Fig. 7.5 Configuración del Directorio y archivos de un preproceso en GiD.*

De la misma manera, se expondrá la forma y la información que deben contener cada uno de estos archivos. Hay que tener en cuenta que no son todos obligatorios y, por tanto, se definirán o no, en función de las necesidades del problema.

### 7.2.1 Datos generales del problema (Fichero: Nombre del Archivo.prb)

Se entiende por datos generales del problema a toda aquella información necesaria para la ejecución del análisis y que no van relacionados con entidades geométricas particulares.

Por lo tanto, los materiales no entrarían en esta clasificación porque solo se asignan a algunas entidades; al igual que las condiciones de contorno.

Los datos que pueden ser incluidos aquí son tales como: tipo de algoritmo a usar por el programa de cálculo, pasos de tiempo y condiciones de convergencia.

Se puede hacer otra clasificación de estos y dividirlos en datos para todo el problema propiamente dicho y variables para cada intervalo.

Se entiende por intervalo, a una subdivisión del problema que puede tener datos particulares para ella. Normalmente, se podrá definir un caso de carga diferente para cada intervalo o, en problemas dinámicos, en cada intervalo se podrán cambiar las cargas, el paso de tiempo o el tipo de convergencia deseada.

En este fichero se deben definir tanto los datos únicos para todo el problema como las variables para cada intervalo, pero recordemos siempre, que no tengan relación con entidades geométricas individuales.

La manera de hacerlo es dar los campos a rellenar por el usuario mediante una palabra que defina la pregunta a hacerle (o nombre del campo), y un valor por defecto, que podrá ser una palabra.

Se entiende por palabra un conjunto de caracteres sin espacios entre ellos, que pueden representar tanto texto como números.

La forma de disposición de datos será:

*Número de datos generales*

*Nombre del campo-1 valor-defecto-campo-1*

*..... Nombre-campo-NG*

*valor-defecto-campo-NG 1 Número de datos del intervalo Nombre-campo-1*

*valor-defecto-campo-1 .....*

*Nombre-campo-N1 valor-defecto-campo-N1*

Donde el número 1 significa que los datos de intervalo se definen para el intervalo 1, y que luego se copiarán de este a los demás.

#### Ejemplo:

6.....- N<sup>o</sup>. de datos a asignar -  
 TITULO Titulo\_nombre.....- Ejercicio de una placa -  
 Casos\_De\_Carga#CB#(1,2,3,4,5) 1.....- Hipótesis de cargas -  
 Peso\_propio#CB#(X=-1,Y=-1,Z=-1) X=-1.....- Peso propio de la Estructura  
 No\_Materiales#CB#(1,2,3,4,5) 1.....- N<sup>o</sup>. de materiales -  
 Tipo\_Problema#CB#(Tens-Plana,Def-Plana) Def-Plana.....- Modelo matemático -  
 LP#CB#(1,2,3) 1.....- Ejes globales -  
 1.....- Primer intervalo de datos -



### 7.2.2 Materiales de base.

(Fichero: Nombre del Archivo.mat)

Se entiende por material a un conjunto de campos compuestos por el nombre del campo y por su valor. Ambos datos se pueden introducir como palabras en el sentido definido en el apartado anterior.

En este fichero se pueden definir tantos materiales como se quiera y con números diferentes de campos en cada uno de ellos. Estos materiales se podrán más tarde asignar a entidades geométricas.

Se llaman materiales de base porque además de poderse usar directamente, también sirven para que el usuario pueda, durante la ejecución del *preproceso*, crearse nuevos materiales a partir de éstos.

Los materiales derivados que puede crear el usuario están formados por los campos de uno de los materiales de base con los valores cambiados.

El formato es el siguiente:

```

NUMBER: número de MATERIAL: (2) tipo de material
QUESTION: Característica del material (módulo de Elasticidad – E)
VALUE: valor de numérico de la característica
QUESTION: Características del material (módulo de Poisson – UN)
VALUE: valor de la característica
QUESTION: Característica del material (Espesor – TH)
VALUE: valor de numérico de la característica
QUESTION..... VALUE.....
END CONDITION
NUMBER: número de MATERIAL: (2) tipo de material
QUESTION: Característica del material (módulo de Elasticidad – E)
VALUE: valor de numérico de la característica
QUESTION..... VALUE.....
END CONDITION
QUESTION..... VALUE.....

```

Donde las palabras escritas en mayúsculas deben estar en la misma forma y las que están en cursiva deben substituirse por la definición del material. Los números de materiales serán correlativos empezando por el 1. El número de campos no es necesario dárselo al programa. Este, puede ser variable para cada material.

#### Ejemplo:

```

NUMBER: 1 MATERIAL: Acero
QUESTION: E
VALUE: 2.1e07
QUESTION: UN
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 7.85
END MATERIAL
NUMBER: 2 MATERIAL: Hormigón_250
QUESTION: E

```

```

VALUE: 3.0e6
QUESTION: UN
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 2.5
END MATERIAL
NUMBER: 3 MATERIAL: Madera_LaGI24h
QUESTION: E
VALUE: 1.16e6
QUESTION: UN
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 0.38
END MATERIAL

```



*Fig. 7.6  
Definición del Material*

Pueden definirse tantos materiales de base como se quieran, ya que los que no sean asignados a ninguna entidad, no se tendrán en cuenta en la escritura del fichero de entrada del programa de análisis.

Esto es bastante útil, ya que puede crearse una biblioteca de materiales con una gran variedad de ellos.

### 7.2.3 Condiciones de Contorno.

(Fichero: Nombre del Archivo. Cnd)

En este archivo, se incluye toda la información que se asigna a diferentes entidades y pueden tener valores distintos para cada una de ellas. Por ejemplo, en un problema estructural las condiciones de contorno serían las restricciones de desplazamientos y giros; y en un problema de análisis térmico lo serían las temperaturas prescritas o iniciales.

Una característica importante de las condiciones de contorno es que inicialmente debe estar definido, para cada una de ellas, el tipo de entidades sobre las que serán aplicadas. Es decir, si se aplicará sobre puntos, líneas, superficies o volúmenes.

Otra característica, es que las condiciones definidas en este fichero, se podrán aplicar a entidades y valores diferentes, para cada uno de los intervalos antes descritos.

El formato de la información es el habitual, la inclusión de campos con nombre y valor por defecto, ambos formados por palabras, con el mismo formato ya mencionado con anterioridad.

El formato es:

NUMBER: *número de condición* CONDITION: (*nº. de condición de contorno*)  
 CONDTYPE: *tipo de entidad* (puntos, líneas, superficies)  
 QUESTION: *campo de desplazamiento restringido* (dirección del eje restringido)  
 VALUE: *valor de la restricción* (cero libre y uno restringido)  
 QUESTION: *campo de giro restringido* (dirección del giro restringida)  
 VALUE: *valor de la restricción* (cero giro libre y uno restringido).....  
 QUESTION.....VALUE.....  
 END CONDITION

### Ejemplo:

NUMBER: 1 CONDITION: Point-Constraints  
 CONDTYPE: over points  
 QUESTION: X-Constraint  
 VALUE: 0  
 QUESTION: X-Value  
 VALUE: 0  
 QUESTION: Y-Constraint  
 VALUE: 0  
 QUESTION: Y-Value  
 VALUE: 0  
 QUESTION: Z-Constraint  
 VALUE: 0  
 QUESTION: Z-Value  
 VALUE: 0  
 END CONDITION  
 NUMBER: 2 CONDITION: Line-Constraints  
 CONDTYPE: over lines  
 QUESTION: X-Constraint  
 VALUE: 0  
 QUESTION: X-Value  
 VALUE: 0  
 QUESTION: Y-Constraint  
 VALUE: 0  
 QUESTION: Y-Value  
 VALUE: 0  
 QUESTION: Z-Constraint  
 VALUE: 0  
 QUESTION: Z-Value  
 VALUE: 0  
 END CONDITION  
 NUMBER: 3 CONDITION: Face-Load  
 CONDTYPE: over lines  
 QUESTION: Axis\_definition:#CB#(GLOBAL,LOCAL)  
 VALUE: GLOBAL  
 QUESTION: X\_preassure:  
 VALUE: 0  
 QUESTION: Y\_preassure  
 VALUE: 0  
 END CONDITION

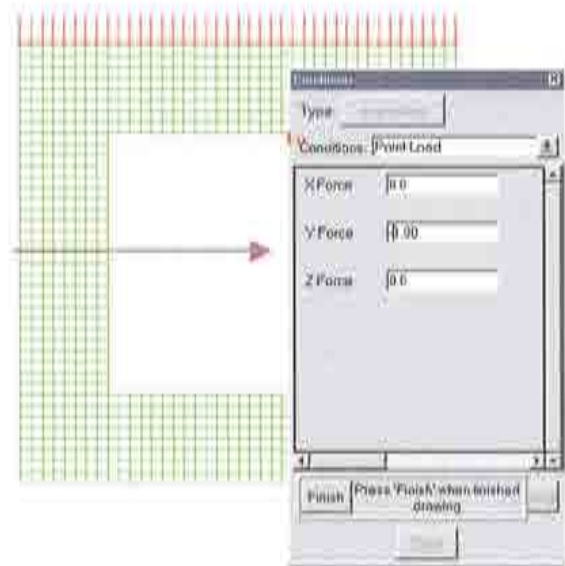


Fig. 7.7 Asignación de Cargas

```

NUMBER: 5 CONDITION: Point-Load
CONDTYPE: over points
QUESTION: X-Force
VALUE: 0.0
QUESTION: Y-Force
VALUE: 0.0
QUESTION: Z-Force
VALUE: 0.0
END CONDITION
NUMBER: 5 CONDITION:.....
CONDTYPE:.....
QUESTION:.....

```

#### 7.2.4 Símbolos gráficos para las condiciones de contorno. (Fichero: Nombre del Archivo. Sim)

Si se desea, se pueden definir símbolos o dibujos para representar gráficamente algunas de las condiciones de contorno durante la etapa de preproceso.

Estos dibujos aparecerán sobre cada una de las entidades que posee la condición de contorno requerida, cuando el usuario así lo requiera.

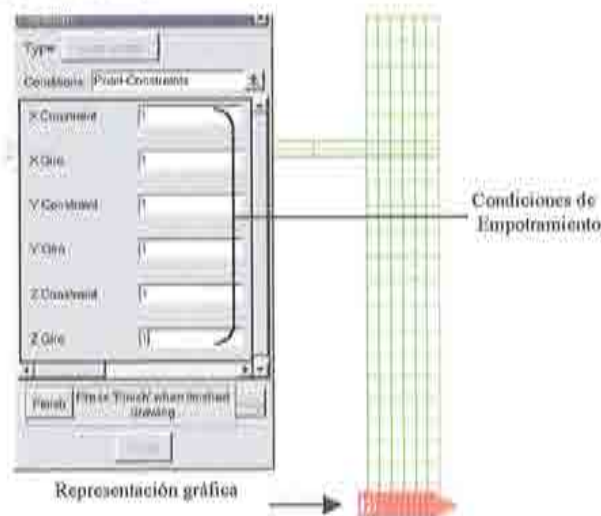


Fig. 7.8

*Símbolos gráficos en la asignación de condiciones de contorno*

La orientación en el espacio también podrá ser definida en este fichero en función de los valores de los datos de la condición de contorno.

Para cada condición de contorno que se desee, se le pueden asignar uno o más dibujos y la elección entre uno u otro dependerá de una expresión en la que se pueden incluir los valores de los datos de esa condición



Para las condiciones sobre línea o superficie puede definirse el símbolo como local. En ese caso no se tiene en cuenta el vector de orientación, y ésta viene definida por la orientación de la línea o superficie. Si la definición es global, se debe dar un vector que defina un giro en el espacio.

Las componentes de este vector son una expresión en la que se puede incluir los valores de los datos de la condición. El giro viene definido por la rotación desde el vector (1,0,0) hacia el vector definido.

Si la definición es local, se alinea la normal a la entidad, con el vector (1,0,0) del símbolo.

Se entiende por expresión, a un conjunto de operaciones escritas con el formato del lenguaje C, y que puede usar los valores de los diferentes campos de la condición a dibujar, mediante la expresión: cond (i,REAL) (en este caso significa el valor del campo i de la condición, considerado como un número real). Para más información, ver el apartado dedicado al fichero \*.bas\*

El fichero se define en la siguiente forma:

```
Cond nombre\_condicion N§ de símbolos *** condición para dibujarse
vector-X vector-Y vector-Z nombre Fichero de dibujo ... ***
condición para dibujarse vector-X vector-Y vector-Z nombre Fichero de
dibujo ...
```

Siendo \*\*\*:    global                    local

El nombre Fichero de dibujo debe ser el nombre de un fichero de un dibujo creado con el mismo *GiD* (extensión *.geo*) y debe darse con el path completo.

Las reglas para crear el dibujo son: El origen es el punto (0,0,0) y el vector de referencia es el eje (1,0,0).

El tamaño es indiferente pues después se escalará automáticamente. En dibujos que luego tengan que usarse como símbolos, no debe haber información de capas ni de generación de malla.

Una vez hecho el dibujo, este se salva y se sale del programa, dentro del directorio creado: Nombre del Archivo.gid, debe haber un fichero : Nombre del Archivo. geo.

Se copia este fichero donde se desee (normalmente en el directorio de definición del tipo de problema) y se pone su nombre con todo el path en la línea correspondiente del fichero: Nombre del Archivo. sim.

### 7.2.5 Descripción del formato de escritura      (Fichero: Nombre del Archivo. bas)

Una vez generada la malla, asignadas las condiciones de contorno y definidos el resto de datos necesarios para el programa de cálculo, es necesario disponer de un fichero que pueda ser leído por el programa de cálculo.

Para poder hacer esto sin necesidad de un programa independiente que haga de interface, se dispone de la posibilidad de crear un fichero llamado: Nombre del Archivo.bas, en el cual se describe el formato y la disposición de los datos del problema.

\* Ver apartado 7.2.4 Descripción del formato de escritura

El fichero: Nombre del Archivo.bas, sería como un conjunto de macros, o sencillo lenguaje de programación, que será leído e interpretado cada vez que sea necesario escribir el fichero para el cálculo.

La ventaja de esto, es que cuando se prepara este fichero, en caso de error, sólo hay que corregirlo y volver a intentar escribir con *GiD* un ejemplo, sin necesidad de recompilaciones ni de salir y volver a entrar al programa. Esto permite una velocidad muy rápida en la corrección de errores.

### 7.2.6 Descripción de la salida del archivo para el cálculo

En general, todo lo que se escriba en este fichero, excepto los llamados comandos, se reproducirá exactamente en el fichero de salida. Los comandos son palabras que comienzan con el signo \*.

Cuando interese reproducir un \* (asterisco) en el fichero de salida, será necesario escribir dos de contiguas (\*\*). Mezclado con el texto a reproducir, se inserta los comandos. Cada uno de estos comandos devuelve alguno o varios de los datos tratados en el preproceso.

Otros comandos imitan las estructuras tradicionales para hacer bucles o condicionales. También se pueden crear variables para manejo de datos. Haciendo un símil con un lenguaje de programación, las diferencias serían las siguientes:

Al estar orientado a la escritura, el texto se reproduce directamente, sin instrucciones de impresión.

Los bucles no contienen índices. Al indicar a un bucle sobre qué tiene que iterar, el programa ya conoce el número de iteraciones a realizar. Por otra parte, las variables contenidas dentro del bucle y que dependan de él, automáticamente van cambiando de valor según la iteración.

Los comandos se pueden subdividir en cuatro tipos: Los que devuelven un valor. Este valor puede ser un entero, un real, o una cadena. El valor dependerá de algunos parámetros que permita el comando y de la posición del comando dentro de un bucle o después de hacer un set de algunos otros parámetros.

Estos comandos se pueden intercalar entre el texto y escribirán su valor en el lugar que les corresponda. También pueden usarse dentro de una expresión como en el caso de un condicional. En este caso es obligatorio especificar si son enteros o reales excepto en el caso de *stemp* y *stcasecmp*. Cuando estén en el interior de una expresión, no se pondrá el \* (asterisco) delante del comando.

Los que devuelven más de un valor a la vez. Tienen un uso similar a los del apartado anterior, excepto en los que no pueden ser usados en expresiones. Pueden devolver varios valores diferentes, uno detrás de otro, según algunas características del problema.

Los comandos que realizan bucles, condicionales, crean variables o que definen algunas especificaciones como: condición a usar, tipo de elemento a usar, etc.

Estos comandos deben comenzar al principio de una línea y nada de lo que haya en esa línea se imprimirá en el fichero de salida. Detrás del comando, en la misma línea, puede haber otras palabras o comandos que acaben de definir lo que deseamos hacer.

Otros, como: \ que evitan el salto de línea. Así, la línea que se está escribiendo continua siendo definida en la línea siguiente del fichero: Nombre del Archivo\*.bas.



El/los argumentos que se asignan a un comando se escriben inmediatamente detrás de él y entre paréntesis. En caso de haber más de uno, se separan por comas. También pueden ponerse los paréntesis sin ningún argumento dentro.

Esto es útil en el caso de querer escribir algo inmediatamente después del comando, sin ninguna separación en medio.

Los argumentos pueden ser: números reales, enteros, la palabra INT o la REAL (en mayúsculas o minúsculas) que significan que el valor a que se refiere el comando debe considerarse como entero o real, respectivamente.

En algunos casos especiales se permiten otros tipos de argumentos como: el tipo de elemento, definido por su nombre, en el comando `*set elem` o una cadena de caracteres incluida dentro de dobles comillas (") para las instrucciones de C, `strcmp` y `strcasemp`.

También es posible, en algunos casos, en lugar de poner el número de orden del campo a imprimir, poner el nombre del campo.

#### 7.2.7 Comandos que devuelven un solo valor.

En general, es indiferente escribir un comando en mayúsculas o minúsculas o con una mezcla de ambas. A continuación se da un listado de los comandos con las especificaciones que corresponden a cada uno de ellos.

`*npoin`, `*ndime`, `*nnode`, `*nelem`, `*nmats` devuelven, respectivamente, el número de puntos, la dimensión, el número de nodos del elemento con mayor número, el número de elementos y el número de materiales.

Están considerados como enteros y no llevan argumentos, excepto `*nelem`, que puede llevar. `*nmats`, devuelve el número de materiales que están efectivamente asignados a alguna entidad, no todos los definidos.

`*GeneralData`. Debe llevar un argumento entero que especifica el número de campo a imprimir. Este será el valor de uno de los campos definidos como datos fijos para todo el problema (independientes del intervalo, ver fichero: Nombre del Archivo.prb).

También puede darse como argumento el nombre del campo. Pueden darse los argumentos INT o REAL que especifiquen que es un entero o real. En caso contrario lo imprimirá como una cadena de caracteres.

Dentro de una expresión, es obligatorio dar uno de ellos, excepto para los comparadores de C: `strcmp`, `strcasemp`. La numeración de los campos empieza en 1.

`*IntvData`. Igual que el anterior excepto que el campo será uno de los definidos como información variable con el intervalo (ver fichero: Nombre del Archivo.prb).

Este comando debe estar dentro de un bucle sobre intervalos y automáticamente irá cogiendo el valor adecuado para cada iteración.

`*MatProp`. Igual que los anteriores pero debe estar dentro de un bucle sobre los materiales.

Devuelve la propiedad cuyo número de campo viene definida por el parámetro.

\* ATENCION: si se trabaja con materiales con diferente número de campos, hay que controlar mediante condicionales la no-impresión de campos inexistentes.

\*cond. Igual que los anteriores pero debe haberse notificado, anteriormente mediante el comando \*set Cond, de que condición se está tratando. Puede estar dentro de un bucle sobre los diferentes intervalos, en caso de que las condiciones varíen para cada uno de ellos.

\*ElmsNum, \*NodesNum, \*MatNum, \*ElmsMat estos comandos devuelven, respectivamente, el número de elemento, el del nodo, el del material y el del número del material asignado al elemento. Todos ellos deben estar dentro de un bucle adecuado para ellos y varían automáticamente con cada iteración. Son considerados como enteros y no pueden llevar ningún argumento.

Los números de materiales serán reordenados de manera que empezarán por 1 y continuarán de forma ascendente: Nombre del Archivo hasta el máximo número de materiales asignados a alguna entidad.

\*CondNumEntities. Como los anteriores, pero debe haberse seleccionado previamente una condición con \*set cond.

Devuelve el número de entidades que contiene una condición.

\*LoopVar. Debe estar dentro de un bucle y devuelve, como entero, lo que se considera como variable interna del loop. Esta variable toma el valor de 1 en la primera iteración y va aumentando de unidad en unidad para cada una de las iteraciones.

Se puede indicar a que bucle nos estamos refiriendo dando como argumento uno de los valores 'elms, nodes, materials, intervals'. En caso contrario y si hay bucles anidados, se usa el valor del bucle más interno.

\*Operation. Devuelve el resultado de una expresión aritmética que debe estar escrita entre paréntesis inmediatamente después del comando.

Esta operación estará definida en formato C, y puede contener cualquiera de los comandos que devuelven un valor. Se puede forzar la devolución de un entero o real con los conocidos parámetros INT o REAL. Si no, devuelve lo que corresponda en función del cálculo.

NOTA: no puede haber espacios entre los comandos y el paréntesis que contiene a los parámetros.

### 7.2.8 Comandos que devuelven más de un valor a la vez

Estos comandos devuelven varios valores en un orden prefijado, escribiéndolos uno detrás de otro. No pueden ser usados dentro de expresiones. A continuación se da una lista de ellos con su descripción:

\*NodesCoord. Escribe las coordenadas del nodo. Debe estar dentro de un bucle sobre nodos. Las coordenadas son tratadas como números reales.

Se escribirán 2 ó 3 coordenadas según sea el número de dimensiones del problema.

\*ElemsConec. Escribe las conectividades del elemento, o sea, una lista de los nodos que contiene; manteniendo el sentido adecuado para cada caso (antihorario en 2D, o los estándares en 3D. Para láminas, el sentido viene definido por el usuario; para el caso de la interface con *Sap90* se utilizará la conectividad correspondiente a éste).

Las conectividades son tratadas como números enteros.

\*GlobalNodes. En una condición que se haya definido sobre las caras de elementos, este comando devuelve los nodos que pertenecen a esa cara para el elemento que corresponda (del bucle sobre elementos). El sentido es el mismo que para las colectividades del elemento. Los valores devueltos son tratados como enteros.

\*LocalNodes. Igual que el anterior pero el valor devuelto es la numeración local del nodo dentro del elemento en cuestión (o sea, entre 1 y Nnode).

### 7.2.9 Los comandos que realizan bucles, condicionales y otros

Estos comandos deben escribirse a comienzo de línea y el resto de dicha línea se dedicará a modificadores de ellos. No debe escribirse texto adicional.

Para la realización de bucles se usarán: \*loop, \*end. Un bucle consta de una línea que empieza por \*loop y que contiene otra palabra que define sobre qué se hace el bucle; unas líneas en medio que son las que se repetirán tantas veces como convenga según esté definido el bucle y cuyos comandos irán cambiando de valor para cada iteración, si es necesario; y una línea que empiece por \*end, que indica el fin del bucle.

Después de \*end, puede escribirse lo que sea y no será tenido en cuenta.

Después del comando \*loop deberá escribirse una de las palabras siguientes: -elems, nodes, materials, intervals. Las cuales significan, respectivamente, que el bucle va a iterar sobre los elementos, nodos, materiales o intervalos. Los bucles pueden anidarse uno dentro de otro.

El bucle sobre materiales sólo iterará sobre los materiales que hayan sido efectivamente asignados a una entidad, aunque hayan más definidos.

Los números de los materiales estarán ordenados a partir del uno y alternativamente. Si un comando que depende del intervalo está situado fuera de un bucle, tomará por defecto el intervalo 1 (uno).

Después del comando \*loop y de su variable, si esta es elems o nodes, se puede poner un modificador entre estos dos: \*all, \*OnlyInCond. La primera significa que se va a iterar sobre todas las entidades, la segunda significa que sólo se iterará sobre aquellas entidades que pertenezcan a la condición.

Esta condición ha de estar definida previamente con \*set cond. Por defecto, entiende que se va a iterar sobre todas las entidades.

Los condicionales se construyen mediante los comandos \*if, \*else, \*endif. El formato es una línea que comience con \*if seguida de una expresión encerrada entre paréntesis. Esta expresión estará escrita en sintaxis del lenguaje C y podrá contener cualquiera de los comandos que devuelvan un valor

En este caso se escribirán sin el \* delante, y debe definirse si hay que tratarlos como enteros o como reales. (Excepción: strcmp, strcmp). Si la expresión es cierta, se imprimirán todas las líneas hasta que se encuentre el correspondiente \*else o \*endif.

El comando \*else es opcional y provoca que todas las líneas entre este y su correspondiente \*endif, sean escritas en caso de que la condición dada por \*if sea falsa. Si existe, \*else debe estar entre \*if y \*endif. Los bucles pueden estar anidados.

En caso de que se desee poner una condicional en medio de una línea, se escribirá en líneas separadas y se hará uso del comando \*\

El comando \*set sirve para tres funciones: -\*set cond -\*set elems. (esta se comentará en el apartado dedicado a operaciones con varios tipos de elementos). -\*set var.

Cuando se quiera utilizar una condición en concreto, debe definirse primero la condición a usar y entonces, está se usará hasta que se defina una de nuevo. Si se realiza dentro de un bucle sobre intervalos, se escogerán las entidades correspondientes al intervalo adecuado. En caso contrario, se escogerán las entidades del intervalo 1.

El motivo es que al señalar al programa que se va a usar una condición, se crea una tabla que permite saber el número de entidades a las que se ha asignado esa condición. Para crear esta tabla es necesario especificar si la condición va a usarse sobre nodos o sobre elementos.

### 7.3 Ejemplo de la configuración del archivo. bas

CALSEF: PROGRAMA PARA CALCULO DE SOLIDOS Y ESTRUCTURAS POR EL M. E. F.

```

$-----
ESTATICO-LINEAL, EN SOLIDOS
$-----
$NUMERO DE PROBLEMAS:
NPROB = *GenData(1)
$-----
$ TITULO DEL PROBLEMA
TITULO= *GenData(2)
$-----
$DIMENSIONES DEL PROBLEMA
DIMENSIONES : NPNOD= *npoin ,   NELEM= *nelem ,   NMATS= *nmats ,   \
              NNODE= *nnode ,   NDIME= *ndime ,   \
              NCARG= *GenData(3) , NGDLN= *GenData(4) , NPROP= *GenData(5) , \
              NGAUS= *GenData(6) , NTIPO= *\  

*if(strcmp(GenData(7),"Tens-Plana")==0)
1 *\  

*endif
*if(strcmp(GenData(7),"Def-Plana")==0)
2 *\  

*endif
*if(strcmp(GenData(7),"Sol-Revol")==0)
3 *\  

*endif
*if(strcmp(GenData(7),"Sol-Tridim")==0)

```

```

4 *\
*endif
*if(strcmp(GenData(7),"Placas")==0)
5 *\
*endif
*if(strcmp(GenData(7),"Laminas-Rev")==0)
6 *\
*endif
, IWRIT= *\
*if(strcmp(GenData(8),"Si")==0)
1 ,\
*endif
*if(strcmp(GenData(8),"No")==0)
0 ,\
*endif
      INDSO= *GenData(9), *\
*Set Cond Surface-Constraints *nodes *or(1,int) *or(3,int) *or(5,int)
*Add Cond Line-Constraints *nodes *or(1,int) *or(3,int) *or(5,int)
*Add Cond Point-Constraints *nodes *or(1,int) *or(3,int) *or(5,int)
NPRES= *CondNumEntities
$-----
GEOMETRIA
$ CONECTIVIDADES ELEMENTALES
$ ELEM. MATER. SECUENCIA DE CONECTIVIDADES
*loop elems
  *elemsnum *elemsmat *elemsConec(invert)
*end elems
$ COORDENADAS DE PUNTOS NODALES
$ NODO COOR.-X COOR.-Y COOR.-Z
*loop nodes
*format "%6i%15.5f%15.5f%15.5f"
  *NodesNum *NodesCoord
*end
FIN_GEOMETRIA
CONTORNO CONDICIONES
$ NODOS RESTRINGIDOS Y PRESCRIPCIONES
$ NODO VALORES PRESCRITOS
*loop nodes *OnlyInCond
*format "%5i%f%f%f%f%f"
  *NodesNum *cond(1) *cond(2) *cond(3) *cond(4) *cond(5) cond(6) *\
*if(GenData(4,int)>=3)
*cond(5,int) *\
*endif
*if(GenData(4,int)==5)
0 0 *\
*endif
*cond(2,real) *cond(4,real) *\
*if(GenData(4,int)>=3)
*cond(6,real) *\
*endif
*if(GenData(4,int)==5)
0 0 0 0 *\

```



```

*endif
*end
FIN_CONTORNO CONDICIONES
$-----
PROPIEDADES MATERIALES
$ PROPIEDADES DE LOS MATERIALES PARA MULTILAMINADO
*loop materials
  Material *matnum *MatProp(1) *MatProp(2) *MatProp(3) *MatProp(4)
*end
FIN_PROPIEDADES MATERIALES
$-----
*loop intervals
FUERZAS
  TTULO: *IntvData(1)
$ TIPO DE CARGA
*Set Cond Face-Load *elems *CanRepeat
*if(CondNumEntities(int)>0)
DISTRIBUIDA UNIFORMEMENTE EN LADOS
$ CARGAS UNIFORMEMENTE REPARTIDAS EN LADOS DE ELEMENTOS
$ NUMERO DE NODOS EN LADO
NODGE = 2
$ LADOS CARGADOS Y CARGAS ACTUANTES
*loop elems *OnlyInCond
  ELEMENTO=*elemsnum(), CONECTIV *globalnodes
  *cond(1) *cond(1) *cond(2) *cond(2)
*end
FIN_DISTRIBUIDA UNIFORME EN LADOS
*endif
*Set Cond Point-Load *nodes
*if(CondNumEntities(int)>0)
PUNTUAL EN NODOS
*loop nodes *OnlyInCond
  *NodesNum *cond(1) *cond(2) *\
*if(GenData(4,int)>=3)
*cond(3) *\
*endif
*if(GenData(4,int)=5)
0 0 *\
*endif
*end
FIN_PUNTUAL EN NODOS
*endif
*if(strcasecmp(IntvData(2),"Si")==0)
PESO_PROPIO
*endif
FIN_FUERZAS
$-----
$ARCHIVO DE POSTPROCESO
FEMV = *IntvData(3)
$-----
*end
FIN_CASEF

```



## 8. CONFIGURACIÓN DE GiD PARA EL POSTPROCESO

### 8.1 Archivos y ventanas de uso general del GiD en el postproceso.

Como ya se ha comentado en apartados anteriores, el análisis estructural se puede subdividir en tres procesos primordiales:

1. *El preproceso..... (la discretización y el modelo matemático)*
2. *El Cálculo de la estructura..... (el uso de programas informáticos)*
3. *El postproceso..... (visualización de los resultados)*

En este capítulo se describirán las utilidades que contiene el *postproceso* de *GiD* y la forma en que el usuario puede configurarlo para cualquier problema en particular.

Sabemos que el cálculo de estructuras es una tarea de gran responsabilidad para el profesionalista que se dedica a ello.

Por esa razón, es importante que éste sepa interpretar los resultados gráficos (*postproceso*) generados en el análisis ya que de ello depende el buen funcionamiento de la estructura.

Por consiguiente, es esencial contar con un buen programa que nos permita visualizar, correctamente, los resultados obtenidos.

*GiD*, al igual que con el *preproceso*, cuenta con un postprocesador bastante dinámico, el cual puede ser configurado, por el usuario, y adaptado a diferentes salidas de datos de programas de cálculo de estructuras.



Fig. 8.1 Comandos de GiD en el postproceso.

### 8.1.1 Archivos (file)

Esta opción, tiene el mismo nombre y funciona de manera similar al *preproceso*.

A continuación serán resumidos de manera breve todos y cada uno de los comandos de este apartado.

La ventana de archivos (*file*) contiene:

#### - Archivo:

- *Read*: Lee archivos generados en el cálculo.
- *Read cut*: Lee cortes generados en la estructura.
- *Save cut*: Salva cortes generados en el proceso.
- *Print to file*: Salva imágenes, creadas dentro del postproceso, en varios formatos (*ps, eps, tiff*).
- *Preproceso*: permite regresar a la configuración del preproceso.
- *Quit*: permite salir del programa GiD.

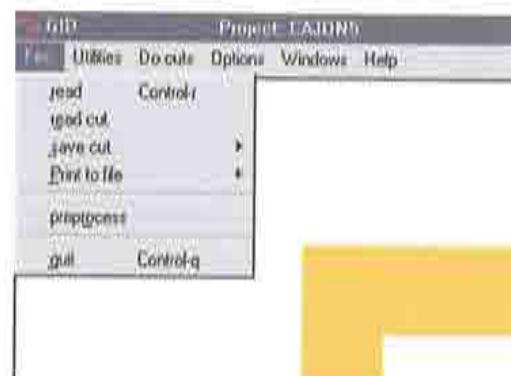


Fig. 8.2 Ventana de archivos (*file*)

### 8.1.2 Utilidades (utilities)

Esta opción nos ayuda a obtener toda la información acerca de las entidades utilizadas dentro del *postproceso*.

#### Utilidades:

- **Id:** Identifica cualquier nodo de la malla creada, y muestra su numeración así como las coordenadas del mismo.
- **Signal:** indica las características de la entidad seleccionada por el usuario.
- **Distance:** determina la distancia entre dos puntos.
- **Graphica:** ayuda a configurar la pantalla en función de los menús a utilizar.



Fig. 8.3 Ventana de utilidades (utilities)

- **Preferences:** permite seleccionar la forma de trabajo dentro de la visualización del problema.

Dentro de esta ventana se encuentran los subcomandos:

- a) **General:** ayuda a crear nuevos puntos, ventanas y archivos batch.
- b) **Graphical:** genera curvas de precisión para rotación e iluminación en la representación.
- c) **Meshing:** configura el tipo de malla en superficies y volúmenes.
- d) **Postproceso:** configura el número de colores a utilizar en la representación, los parámetros de máximos y mínimos de los resultados, así como el número de vectores en los colores.

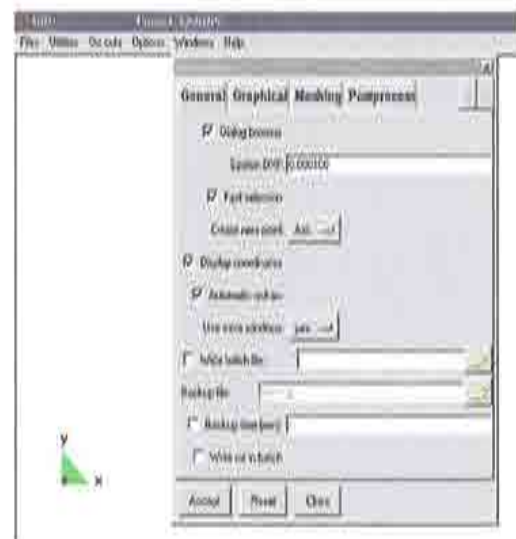


Fig. 8.4 Ventana de preferencias (preferences)



### 8.1.3 Hacer cortes (do cuts)

Con el comando *do cuts* se pueden realizar cortes perpendiculares en la malla generada para poder optimizar la visualización de los resultados.

#### - Hacer cortes:

- *Cut plane*: Realiza cortes en el plano. Estos, pueden ser generados por dos o tres puntos.

- *Divide mesh*: genera divisiones en la malla en dos partes en su plano y puede ser en dos o tres puntos.

- *Divide sets*: al igual que los dos comandos anteriores divide los sets en el plano por dos o tres puntos.

- *Sucesion*: permite al usuario hacer secciones paralelas entre ellas, definiendo un eje de dirección normal a los cortes, y el número de divisiones que quiera realizar a lo largo de éste.

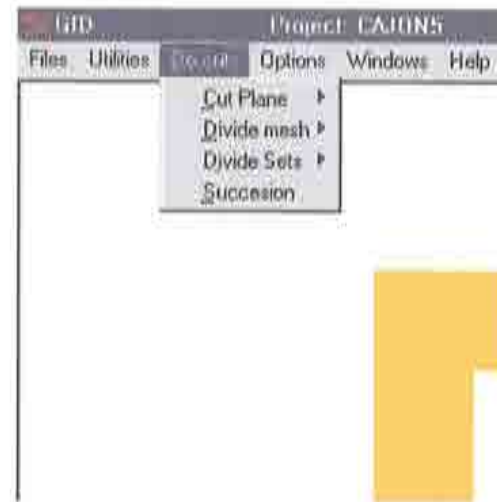


Fig. 8.5 Ventana de Cortes (do cuts)

### 8.1.4 Opciones (options)

A partir de esta ventana el usuario puede seleccionar opciones relativas a la configuración de los resultados, como por ejemplo: la gama de colores de la representación de los resultados o la visualización de vectores.

#### - Opciones:

- *Geometry*: esta opción se subdivide en dos subcomandos.

a) *Border angle*: visualiza una línea de borde entre aquellas caras que forman un ángulo mayor o igual a aquel que define el usuario.

b) *Massive body*: permite al usuario definir, si desea o no, que el interior de la geometría se dibuje o no.

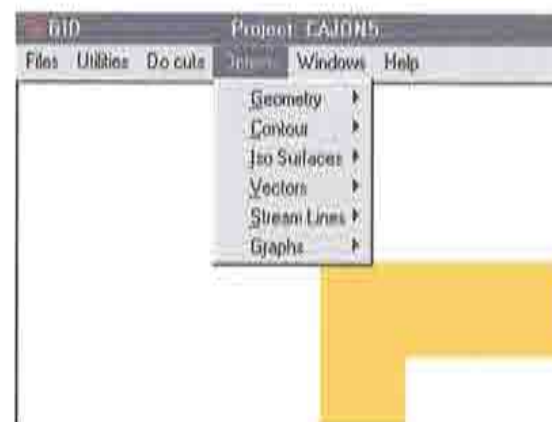


Fig. 8.6 Opciones (options)

- **Contour:** define, el número de colores que desee en la visualización, los intervalos de los resultados. También se puede cambiar la escala de colores de acuerdo con las necesidades del usuario. Y parametrizar los máximos y mínimos en la representación.

- **Iso Surfaces:** crea isosuperficies de valores determinados de los resultados y define el tipo de iluminación (render) que se desea o si se quiere transparente.

- **Vectors:** define el número de colores para los vectores que definen los resultados, elige la posición del vector respecto del nodo.

- **Stream lines:** en las trayectorias de las partículas se muestran o no las etiquetas de los tiempos y el tamaño de la línea de trayectoria.

- **Graphs:** define las opciones que se requieren a la hora de visualizar los resultados gráficos, muestra o no las gráficas, superpone o no la rejilla, borra gráficos, define el estilo de las nuevas gráficas, cambia el estilo.



Fig. 8.7 Comandos de opciones (options)

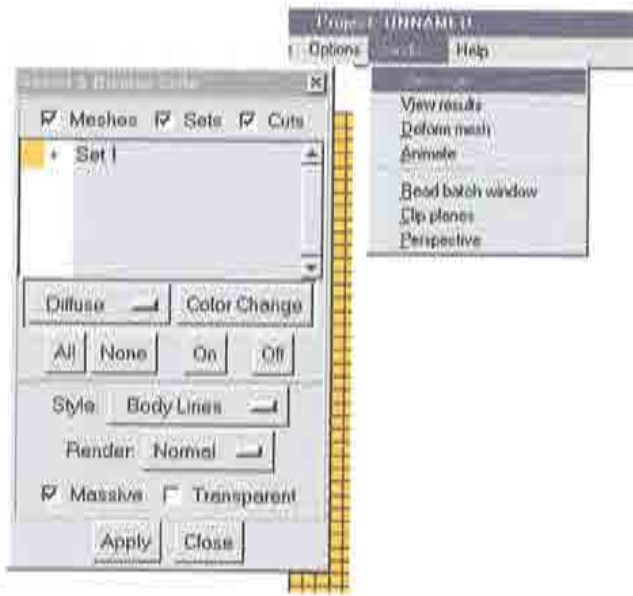
### 8.1.5 Ventanas

### (Windows)

El icono Windows ofrece al usuario la posibilidad, mediante diversas ventanas, de acceder a los resultados del cálculo de forma activa.



Fig. 8.8 Comandos de ventanas (windows)



### - Ventanas:

- *View style*: accede a la ventana auxiliar que permite al usuario escoger la forma de visualización de las mallas (*mesh*), los pasos dentro del análisis (*sets*) y cortes (*cuts*) dentro del postproceso.

Dentro de esta ventana se puede configurar la forma en que se desean ver, tanto las mallas, set o corte. Se pueden presentar de manera difusa, con renders, transparente o en masa.

Fig. 8.9 Comandos de ventanas (windows)

- *View Results*: este icono permite al usuario acceder a una ventana auxiliar donde puede seleccionar qué tipo de análisis y pasos, de los resultados desee visualizar.

- *No Result*: opción para que se visualicen los resultados del cálculo.

- *Display vectors*: se selecciona qué resultados se desean visualizar a través de vectores.

- *Contour Fill*: se selecciona qué resultados se desean visualizar en bandas de colores.

- *Contour Lines*: se escoge qué resultados se desean visualizar en isolíneas de colores.

- *Sh Min Max*: muestra el sitio en donde se encuentra el máximo y el mínimo numérico de los resultados que se elijan.

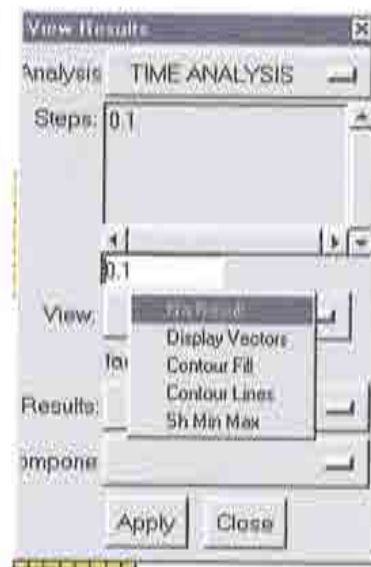


Fig. 8.10 View Result (windows)

- *Deform Mesh*: visualiza la deformada de la malla según un campo de desplazamiento.

Esta ventana contiene diversos subcomandos para visualizar, ya sea la geometría original o la deformada:

1. Main Geometry
  - Original
  - Deformation
  - Results
  - factor
2. Show Geometry
  - off
  - Original
  - Deformation; analysis, steps, results, factor.



Fig. 8.11 Deform Mesh (windows)

- *Animate*: ventana de animación que permite visualizar un resultado a través de los pasos de un análisis, así como guardar fotos de cada paso o un fichero de animación en formato mpeg



Fig. 8.12 Animate (windows)

- *Read batch windows*: el usuario puede ejecutar la animación de un archivo en formato batch creado con anterioridad.

Funciona como un reproductor de video y puede configurarse, de tal manera que el usuario puede definir tiempo de iniciación y de terminación.



Fig. 8.13 Read batch windows (windows)



- *Clip planes*: genera planos de corte de visualización dinámicos paralelos a la pantalla que permiten ver el interior, tanto de la geometría como de la malla.

De igual manera, el usuario puede generar una configuración de acuerdo a sus necesidades de visualización.



Fig. 8.14 Clip planes (windows)

- *Perspective*: permite cambiar entre perspectiva normal y axonométrica.

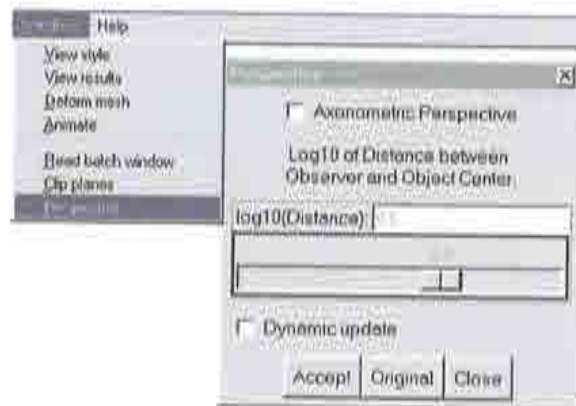


Fig. 8.15 Perspectiva (windows)

Estos son, de manera general, los comandos más comunes y usados en *GiD* durante el *postproceso*.

En los capítulos 9 y 10 se realizarán dos ejemplos prácticos, en los cuales utilizaremos la mayoría de los comandos del *postproceso*.<sup>1</sup>

<sup>1</sup> Dentro del directorio de *GiD* se encuentra un archivo llamado *info*, el cual detalla de manera más amplia las funciones del *postproceso*.

## 9 PREPROCESO, CÁLCULO Y POSTPROCESO DE UNA VIGA PARED (interface asolid)

### 9.1 Descripción del problema

En este capítulo, aplicaremos un ejemplo práctico para mostrar la *interface GiD-Sap90* para el caso de Deformación Plana (*Asolid*). Es de aclarar que esta interface con Sap90 solo se plantea para un análisis lineal, por lo que no se considero el estudio dinámico, ni térmico.

El ejemplo práctico será un muro pared de Hormigón armado rectangular de 5.00 mts. de largo por 5.00 mts. de altura, con un espesor de 0.10 mts.

El muro pared tendrá una carga, uniformemente repartida, en su parte superior y paralelo al plano de 1.5 T/ml. Y se encontrará totalmente empotrado en su parte inferior.

Las características del Material (hormigón) son:

Módulo de Elasticidad.....  $E = 3.0E6 \text{ T/m}^2$

Módulo de Poisson.....  $\nu = 0.2$

Espesor.....  $TH = 0.10 \text{ mts.}$

Peso Volumétrico.....  $W = 2.35 \text{ T/m}^3$

Carga.....  $F = 1.5 \text{ T/ml}$

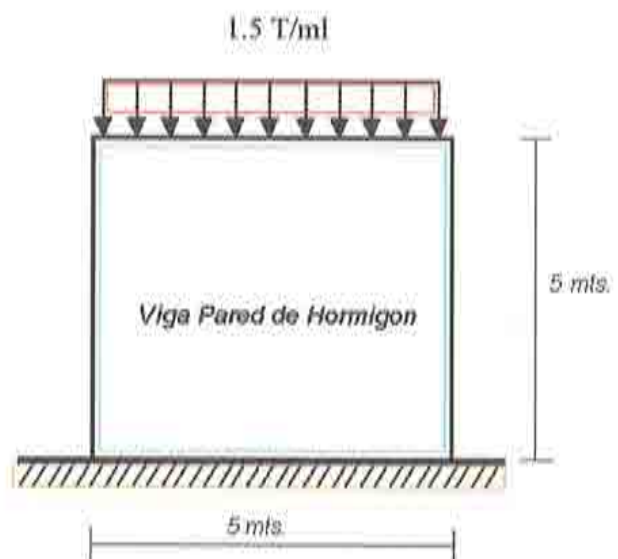


Fig. 9.1 Ejemplo práctico.  
Viga pared



En este problema utilizaremos la teoría de la Elasticidad Bidimensional, y en particular la de Tensión plana, debido a sus características.

Una vez que se tiene definido el problema, el siguiente paso es representar gráficamente la estructura. Y para ello existen varias herramientas que pueden ser utilizadas.

## 9.2 Generación de la geometría

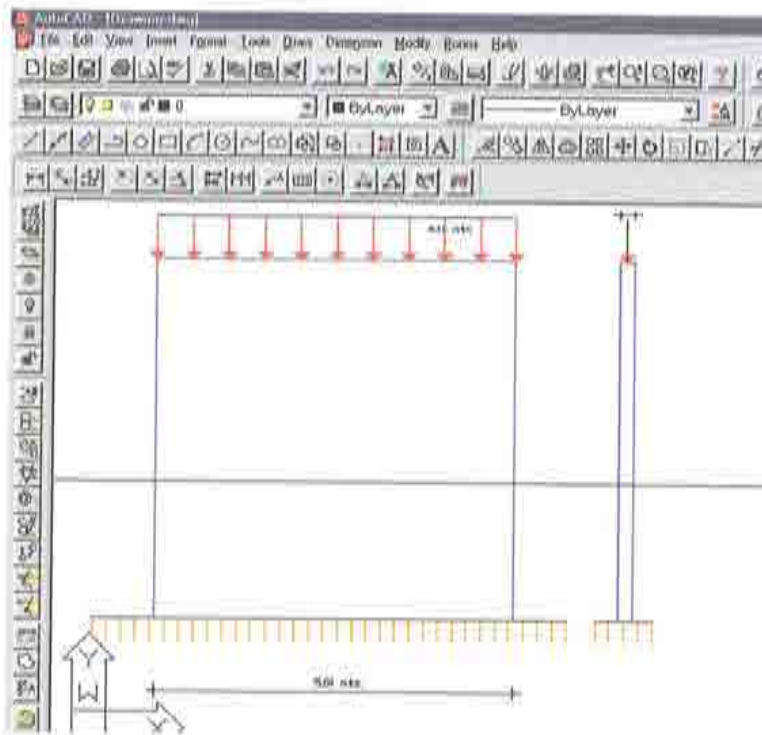
Como ya se comentó en el apartado anterior, existen varios programas informáticos para representar la geometría del problema a resolver. Tales como: Cualquier programa de dibujo (*CAD*) que permita exportar la geometría en un fichero con formato *dxf*.

Ejemplo de estos programas son: El *autocad* en sus versiones 12,13,14; el *3D studio*, etc.

En esta publicación nos centraremos solo en dos:

1. *La generación de la estructura por medio del programa Autocad (12,13,14)*
2. *Las propias herramientas de dibujo que posee el GiD.*

En el primer caso, el usuario puede comenzar a dibujar la estructura, como si se tratase de cualquier dibujo a representar. Toda vez que se tiene terminado el dibujo (la estructura ya sea en una, dos o tres dimensiones) el siguiente paso es exportarlo a un formato *dxf* y llevarlo al directorio donde se desee iniciar el proceso de análisis.



*Fig. 9.2 Representación gráfica del problema en Autocad.*

La utilización de programas de dibujo (*CAD*) dependerá de la complejidad del problema, ya que contienen muchas herramientas que facilitan la generación del dibujo, por complejo que sea.

En el segundo caso, las herramientas, aunque escasas, de *GiD* nos permiten generar geometrías básicas y sencillas. Para el ejemplo que nos atañe, es posible trabajar directamente en el programa.

A Continuación, se detallarán todos y cada uno de los pasos a seguir en la solución del problema.

#### \* Generación de un archivo:

- A partir de que se accede a *GiD*, lo primero es crear un proyecto nuevo.

Al nombre del archivo (proyecto) por defecto, se le asignará la extensión \*.gid.

Este, funciona como un directorio en el cual, conforme se vayan asignando todos los parámetros, se irán almacenando subarchivos, los cuales contendrán:

- \* Las condiciones de contorno
- \* Las características de los materiales
- \* Y, demás datos necesarios para su cálculo



Fig. 9.3 Generación de un archivo GiD.

Ahora, se esta en condiciones de comenzar a crear la geometría de la estructura. De manera breve describiremos la generación de esta, desde la creación de un punto hasta la generación de la malla.

#### \* Generación de puntos:

- Comenzamos por crear un punto, el cual se puede generar desde el teclado, dando las coordenadas de este, en el orden  $x, y, z$ . (es importante recordar que este procedimiento se realiza para geometrías sencillas y regulares).

De esta manera, se continuará hasta haber completado todos los puntos básicos de la geometría

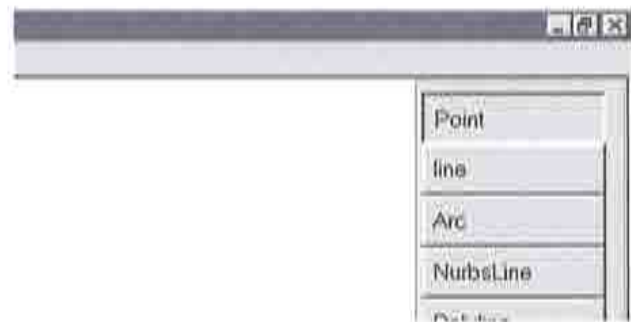


Fig. 9.4 Creación de los puntos

**\* Generación de líneas:**

- Una vez que se tienen los puntos, crearemos las líneas que rigen esta geometría.

Esto se realiza uniendo todos los puntos ya existentes.

Cabe recordar, que las últimas versiones de GiD permiten generar líneas curvas, las cuales podrán ser utilizadas para la generación de elementos de malla con determinada curvatura.

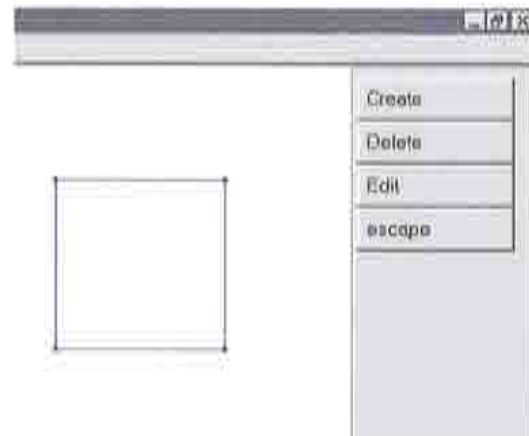


Fig. 9.5 Creación de líneas.

Debido a que es una geometría regular, la generación de esta geometría es bastante sencilla. Recordemos que la mayoría de las estructuras son de formas más complejas pero, depende mucho de la abstracción que el usuario le de a la solución del problema, así como del modelo matemático a usar.

### 9.3 Definición de la malla

**\* Generación de superficies:**

- A partir de este momento, tenemos de manera general creada la geometría del problema a resolver.

Y para crear la malla en este programa, es importante crear tantas superficies como mallas deseemos crear.

En la ventana de Geometry - Create se busca la creación de superficies planas. El programa le solicitará al usuario que seleccione las líneas para su generación.

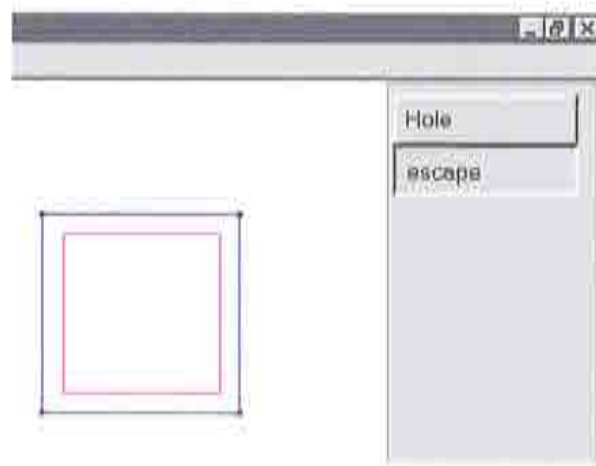


Fig. 9.6 Creación de superficies.



**\* Generación malla:**

- Resumiendo, después de haber creado la geometría y la superficie, que nos permitirá generar la malla, iniciaremos a detallar las instrucciones necesarias para su generación.

Dentro del menú *Generation*, el usuario puede generar una malla, editarla y escoger las opciones de generación de malla que se desee.

Dentro de este comando, la primera opción que aparece es *Quadratic elements*, aquí se describe el grado del elemento que se desee. Se puede seleccionar elementos cuadráticos, cuadráticos de 9 nodos ó normales.

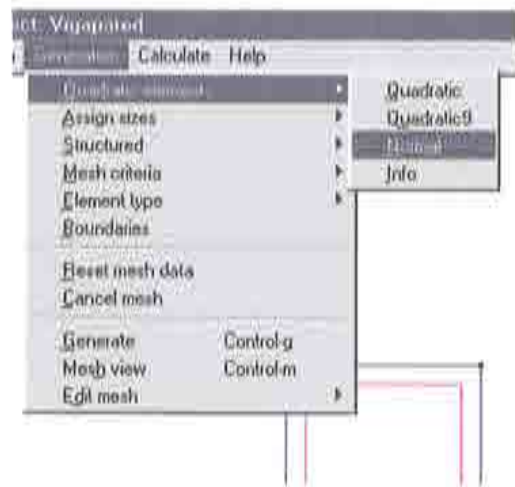


Fig. 9.7 Grado del elemento.  
(Generación de malla)

**\* Asignar tamaño:**

- Esta opción, permite asignar a las entidades del problema un tamaño concreto de elemento para la malla no estructurada.

Esta asignación se puede dar a puntos, líneas, superficies, volúmenes, por su geometría.

Para nuestro problema, no utilizaremos este comando ya que iremos directamente a definir una malla estructurada.

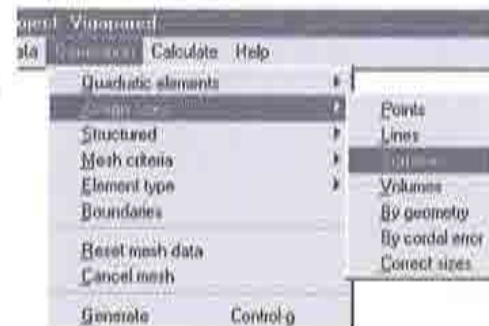


Fig. 9.8 Asignar tamaño.  
(Generación de malla)

**\* Estructuración de malla:**

- *Structured*, nos permite definir una malla estructurada, ya sea por líneas, superficies o volúmenes.

Por ejemplo: Nosotros tenemos una sola superficie por lo tanto asignaremos ésta con el comando *surfaces*. A continuación, aparecerá en la pantalla la pregunta del número de particiones.

Cabe recordar, que se pueden generar particiones en ambos sentidos, es decir, que podemos seleccionar una línea vertical y asignarle más particiones que a la horizontal. De esta manera podemos densificar la malla en el sentido que más nos convenga.

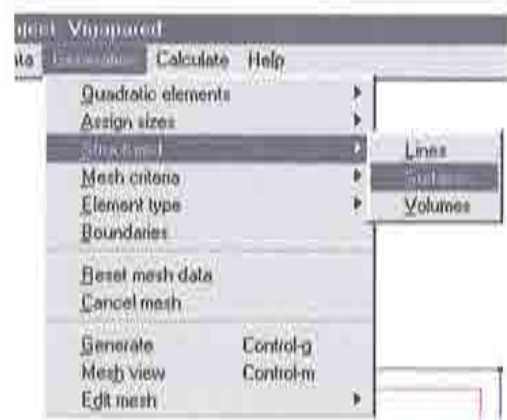


Fig. 9.9 Estructuración de malla.  
(Generación de malla)

**\* Criterio de malla:**

-*Mesh criteria:* nos permite escoger el criterio de malla ya sea por defecto, mallar determinadas entidades (líneas, superficies o volúmenes) o no mallar otras.

Al igual que los comandos anteriores, siempre después de definir con el comando el tipo de criterio, el programa exige que se asignen con el ratón todas las entidades a trabajar.

Para nuestro problema asignaremos una partición en ambos sentidos de 20 x 20.

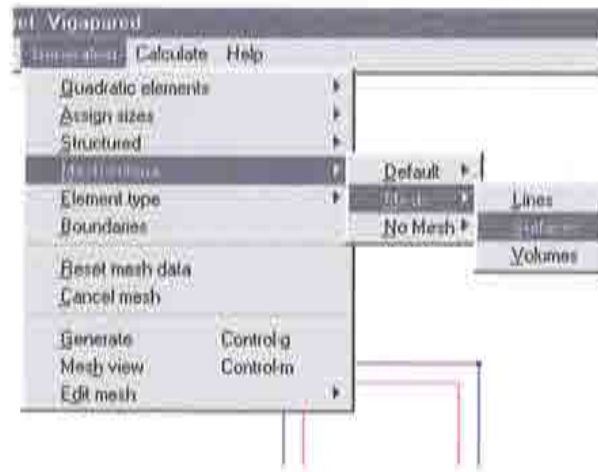


Fig. 9.10 Criterio de malla.  
(Generación de malla)

\* **Nota:** es importante mencionar que el usuario puede ir generando capas para almacenar toda la información, ya sea de líneas, superficies, volúmenes o mallas.

Este comando, funciona de la misma manera que cualquier programa (CAD)

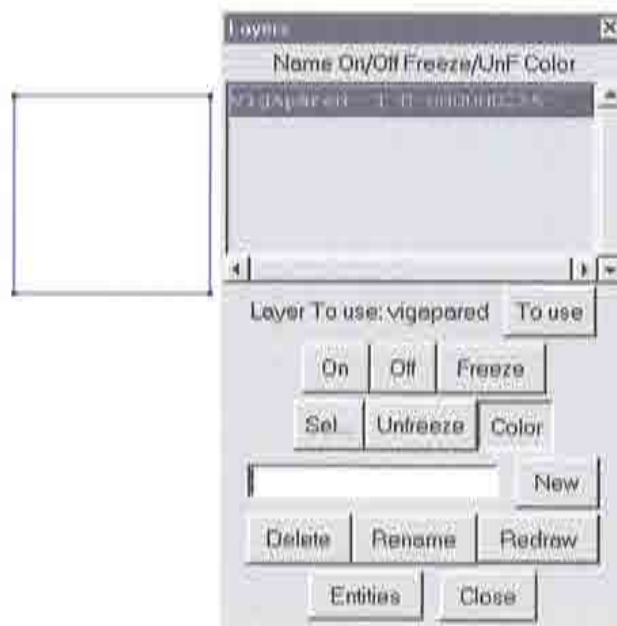


Fig. 9.11 Creación de capas.  
(Generación de malla)



**\* Tipo de elemento:**

- **Element type:** Permite asignar un tipo de elemento determinado a una entidad en particular. Los elementos van desde lineales, triangulares, cuadrilaterales, tetraedros, hexaedros, ó simplemente en puntos.

Para el caso que nos atañe, utilizaremos elementos cuadrilaterales de 4 nodos. Si se esta trabajando en elasticidad bidimensional (asolid), el usuario debe de recordar que el Sap permite el uso de 9 nodos.

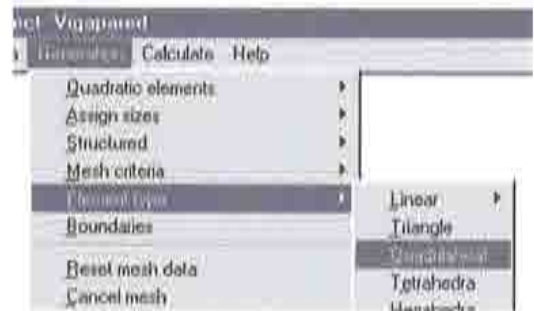


Fig. 9.12 Tipo de elementos.  
(Generación de malla)

**\* Herramientas de ayuda en la generación:**

- **Boundaries:** muestra el contorno de la malla.
- **Reset mesh data:** borra la información asignada a la malla durante su generación.
- **Cancel mesh:** cancela la malla generada anteriormente.

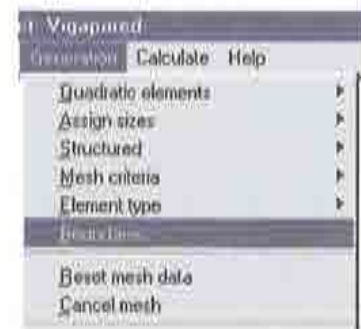
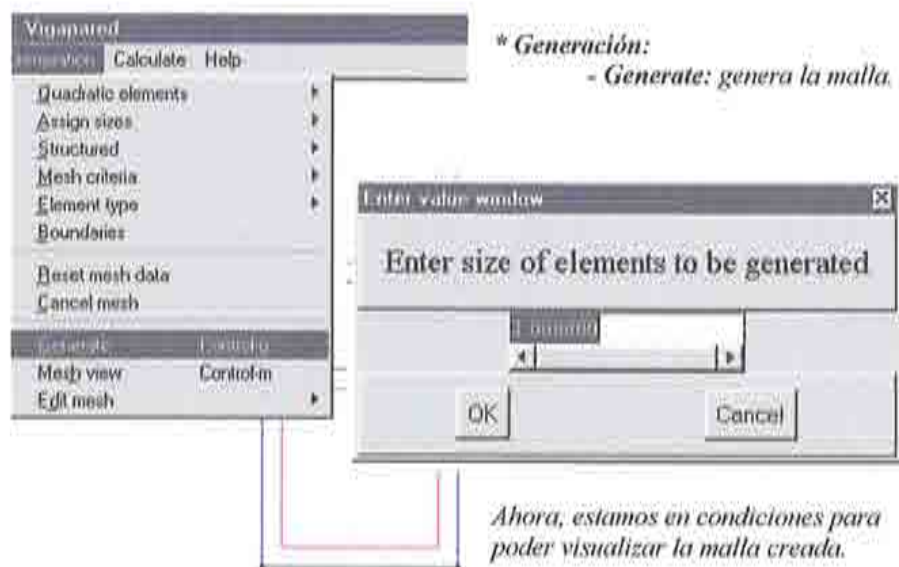


Fig. 9.13 Herramientas de ayuda.  
(Generación de malla)

Esta opción, pregunta al usuario el tamaño del elemento para su generación.

En la mayoría de los casos se asigna la unidad por defecto, ya que la malla ya ha sido estructurada anteriormente.



**\* Generación:**

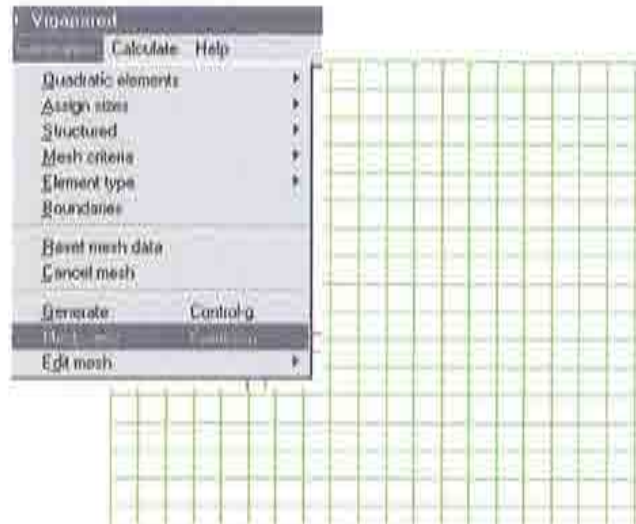
- **Generate:** genera la malla.

Ahora, estamos en condiciones para poder visualizar la malla creada.

Fig. 9.14 Generación de la malla.

**\* Visualización de la malla:**

- *Mesh view: muestra en pantalla la malla generada,*



*Fig. 9.15 Visualización de la malla.  
(Generación de malla)*

Hasta aquí hemos creado la geometría, su superficie y la malla del problema a resolver. A partir de ahora queda configurar el problema. Esto quiere decir, que tenemos que asignar todas y cada una de las condiciones de borde, materiales, datos del problema y los intervalos de éste.

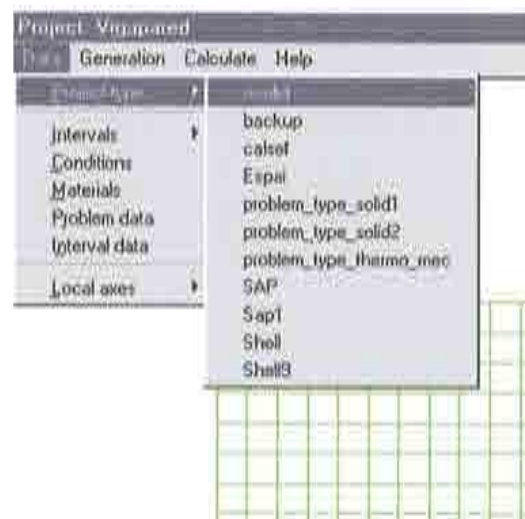
## 9.4 Configuración del preproceso

**\* Definición del tipo de problema:**

- *Data: este comando, carga el tipo de problema a resolver (asolid para tensión ó deformación plana y shell para teoría de placas).*

El usuario debe, previamente, crear todos los archivos necesarios que definen el tipo de problema, tales como: *Asolid.mat* (define las características de los materiales a emplear), *Asolid.cnd* (define las condiciones de borde del problema), *Asolid.prd* (define los datos del problema, así como los intervalos del mismo), *Asolid.bas* (define la escritura del archivo para su cálculo), *Asolid.sim* (define la geometría de las condiciones de borde que se deseen representar en el proceso).

Si el programa lee todos estos archivos, le informa al usuario que esta en condiciones de iniciar el proceso.



*Fig. 9.16 Tipo de problema.  
(Configuración para el cálculo)*

**\* Nota:**

*Toda vez que se asigna el tipo de problema, el programa avisa al usuario que perderá toda la información creada anteriormente.*

*Con lo cual, se deberán de asignar de nuevo todos los datos del problema.*



*Fig. 9.17 Información sobre datos.  
(Configuración para el cálculo)*

**9.4.1 Condiciones de contorno.****\* Condiciones de contorno:**

- **Conditions:** en este apartado, se incluyen todas las condiciones de contorno que rigen el problema.

*Por ejemplo:*

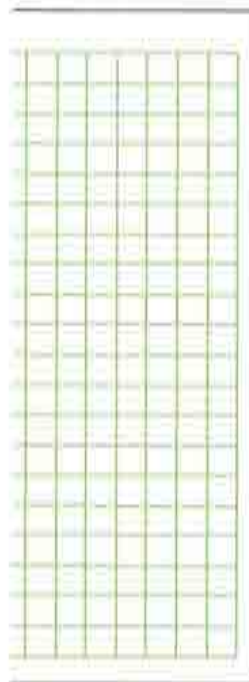
*Para el problema que estamos abordando, los únicos nodos que irán con restricciones totales son los que se encuentran en la base del muro pared. El resto de nodos irán con la condición: 0,0,0,0,0,0 que representa que se encuentran libres de cualquier restricción.*

*Esto es con lo que respecta a grados de libertad de los nodos.*

*En el mismo comando, podemos asignar las cargas, ya sean por puntos o por líneas.*

*Nosotros aplicaremos una carga uniformemente repartida a lo largo de toda la parte superior de la estructura de 1.5 t/ml.*

*La forma de fijar estas condiciones es con el ratón y a continuación dar la instrucción de finalizar (finish)*



*Fig. 9.18 Asignación de condiciones de borde.  
(Configuración para el cálculo)*

**\* NOTA:**

Es importante recordar que la asignación de las condiciones se puede hacer o ya sea por ventana o por teclado.

De esta misma manera se pueden dibujar las condiciones (previa configuración del archivo asolid.sim) o desasignar todas las condiciones que se hayan realizado con anterioridad.



Fig. 9.19 Condiciones de borde.  
(Configuración para el cálculo)

#### 9.4.2 Características de material

**\* Asignación de los materiales:**

- **Materials:** la asignación del material o los materiales se hace de la misma manera que en condiciones.

- De nuevo le recordamos al usuario que la configuración de los archivos, para el preproceso, se debe de realizar previamente y puede incluir tantos materiales como desee.

Para la viga pared en estudio, asignaremos hormigón como material base del problema.

El usuario deberá asignar los elementos que correspondan a este material, y que en nuestro caso serán todos.

Una vez asignados los materiales, es posible visualizarlos para evitar errores en la generación.

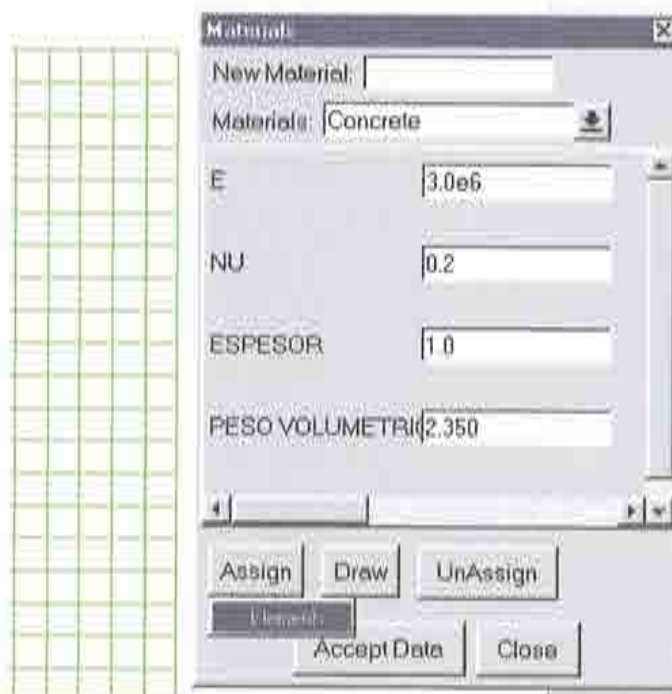


Fig. 9.20 Asignación del material.  
(Configuración para el cálculo)



**\* Nota:**

Al igual que en las condiciones, las características del material se pueden asignar (por teclado o por ventana) se pueden crear nuevos materiales, representarlos gráficamente en colores y se pueden desasignar de una configuración anterior.

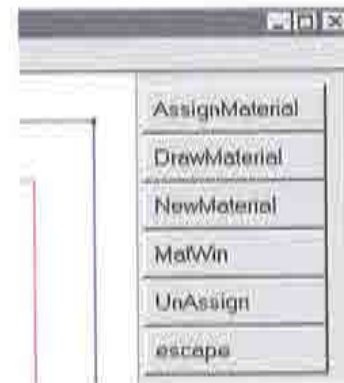


Fig. 9.21 Materiales.  
(Configuración para el cálculo)

**9.4.3 Datos del problema**

En este subapartado, se van a asignar todos los datos requeridos por el programa (Sap90) tales como:

- Peso propio.
- Título del problema.
- Hipótesis de carga.
- N° de materiales.
- Tipo de problema (Asolid ó Shell)



Fig. 9.22 Datos del problema.  
(Configuración para el cálculo)

**\* Asignación de datos del problema:**

- **Problem Data:** Como ya se comentó, aquí asignaremos todos los datos necesarios para el cálculo.

Como muestra la figura 9.22, problem data nos permite definir el resto de la información que requiera, ya sea el Sap90 o cualquier otro programa de cálculo de estructuras que se desee utilizar.

Toda esta información será recogida por el archivo Asolid.has, el cual escribirá un archivo para su análisis.

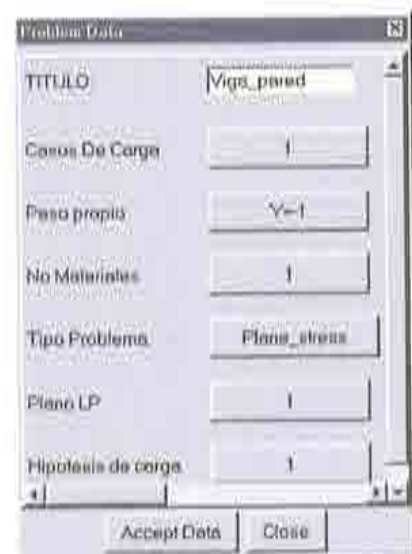


Fig. 9.23 Asignación de los datos del problema.  
(Configuración para el cálculo)



#### 9.4.4 Intervalos del problema

Para este problema (Asolid.gid) solo fue necesario configurar el archivo Interval Data con una instrucción, y fue la de solicitarle al programa que escriba un archivo para su cálculo.

Pero, el usuario puede configurar este archivo, con tantos datos como desee, y en función de sus requerimientos.

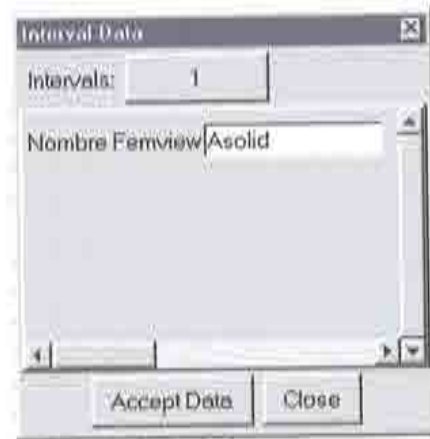


Fig. 9.24 Intervalos del problema.  
(Configuración para el cálculo)

A partir de este momento, estamos en condiciones de enviar el problema a calcular.

#### 9.5 Cálculo del Problema

##### \* Calculate:

Para realizar esta operación, el usuario debe, previamente, configurar un archivo con extensión .bat.

Este archivo debe de contener todas las instrucciones para direccionar la ejecución del análisis.

Esta instrucciones deben de contener:

- La orden de buscar el programa de cálculo que realizara el análisis.
- la Subrutina que configure el archivo de resultados (\*.flavia.res).
- Y la instrucción que direcciona este archivo de nuevo al problema en cuestión.

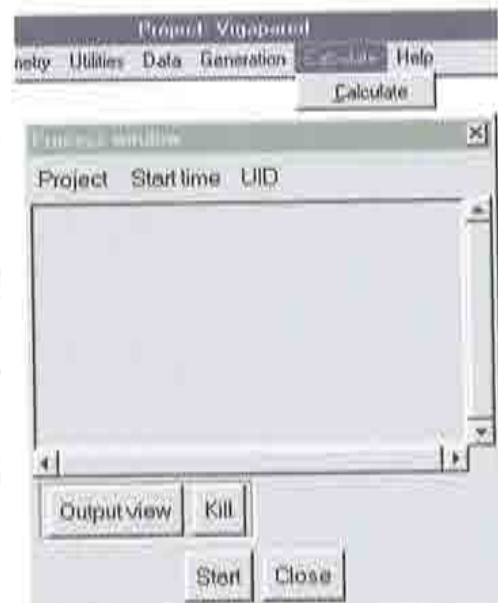


Fig. 9.25 Cálculo de la estructura.  
(Configuración para el cálculo)

## 9.6 Postproceso

En este momento, estamos en condiciones de poder visualizar los resultados del análisis. Para poder acceder a los resultados es necesario seleccionar el icono de *Postproceso*.

Automáticamente el programa accede al *postproceso*, y es ahí donde podremos obtener los resultados gráficos del problema.

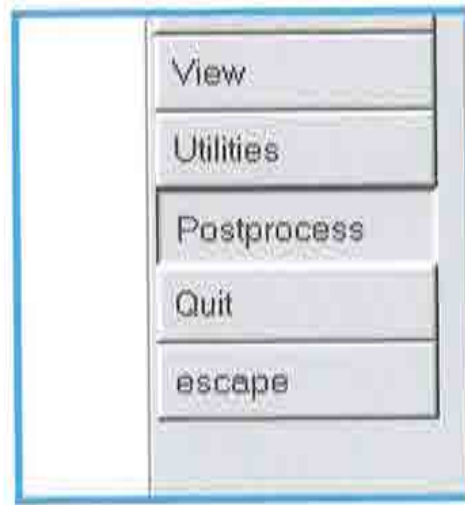


Fig. 9.26 Postproceso

### 9.6.1 Configuración de los resultados

Dentro del *postproceso* podemos configurar los resultados obtenidos, ya sean colores, escala de la deformación y valores numéricos.

Es importante señalar, que el usuario debe de conocer los programas de cálculo a utilizar, así como los resultados que desea obtener.

**\*Nota:** Nunca podemos olvidar, que toda persona que pretenda realizar el análisis de cualquier estructura debe poseer los conocimientos numéricos necesarios. Ya que sería de desagradables consecuencias interpretar de manera incorrecta los resultados, lo cual se traduciría en un grave peligro para la construcción de la estructura.

#### \*Utilidades del postproceso:

- *Utilities:* al igual que en el preproceso, la ventana de utilidades nos permite identificar nodos y elementos de la malla generada y la distancia entre ellos. También, podemos configurar todos los datos relacionados con los resultados (General, Graphical, Meshing y postprocess) utilizando *preferences*.



Fig. 9.27 Utilidades del Postproceso

#### \* Ventanas del postproceso:

- *Windows:* este comando nos permite visualizar todas las ventanas, las cuales contienen los resultados gráficos.

- *View style:* permite configurar los datos relacionados con la representación de la geometría, tales como: renders, visualización de la malla, cortes realizados a la estructura, así como los pasos en el cálculo.

- *View results:* muestra una ventana con los resultados que deseamos visualizar.

- *Deform mesh:* permite configurar todos los resultados relacionados con la malla y su deformación.

El resto de comandos, ayuda a configurar imágenes, por ejemplo: de animación, perspectivas, etc.



Fig. 9.28 Ventanas del Postproceso

**\*Nota:** Si el usuario desea conocer más información, sobre todos y cada uno de los comandos, así como de su configuración, puede acceder a la página Web del GiD.

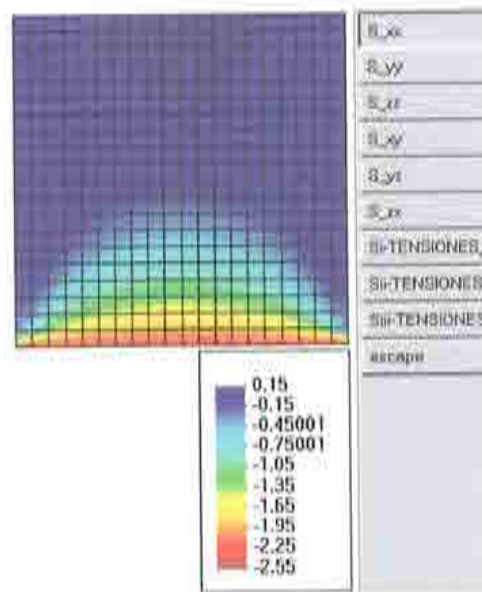
<http://www.gatxan.upc.es>

### 9.6.2 Visualización de los resultados tensionales.

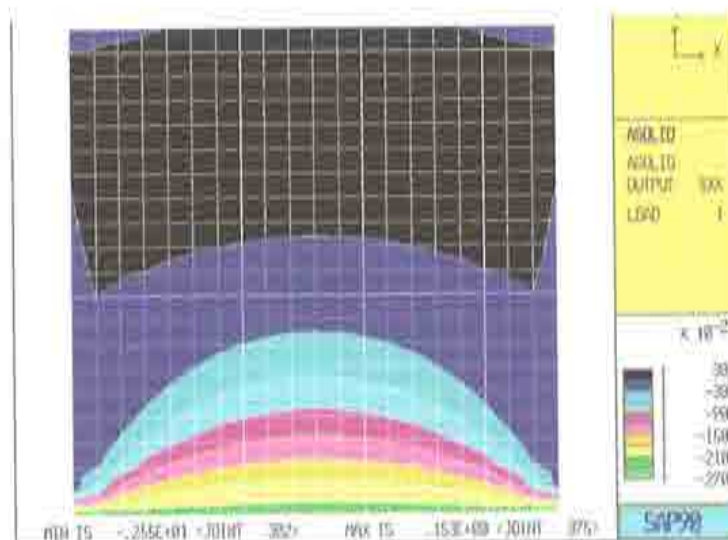
En este apartado, se muestran los resultados gráficos del análisis de la viga pared sometida a una carga uniformemente repartida y empotrada en su base. Recordemos que en este ejemplo utilizamos la teoría de la Tensión Plana y el cálculo se realizó con la *interface GiD – Sap90*. Y para este caso en particular, la *interface* de *ASOLID.GiD*.

Como forma de análisis comparativo de resultados, se mostrara el resultado tensional del *postproceso* de *GiD* y a continuación el de *Sap90*, para cada componente.

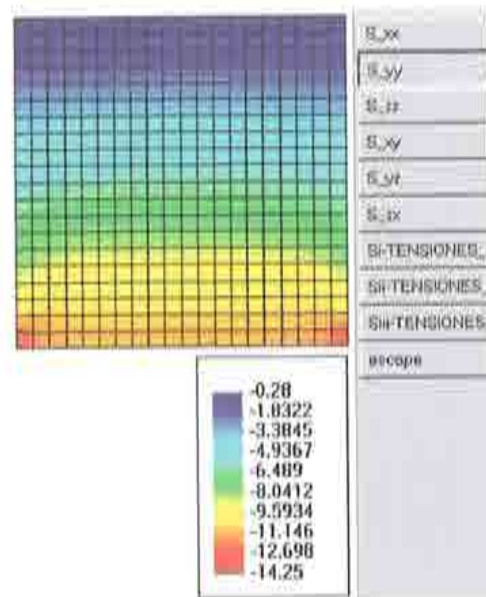
#### \* VALORES TENSIONALES



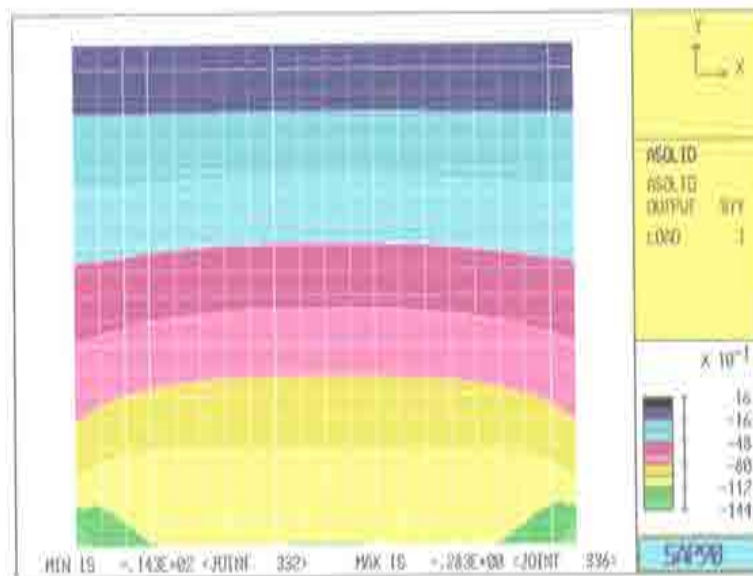
Tensiones SXX en el *postproceso* de *GiD*, con la *interface GiD – Sap90*



Tensiones SXX en el *postproceso* de *Sap90*

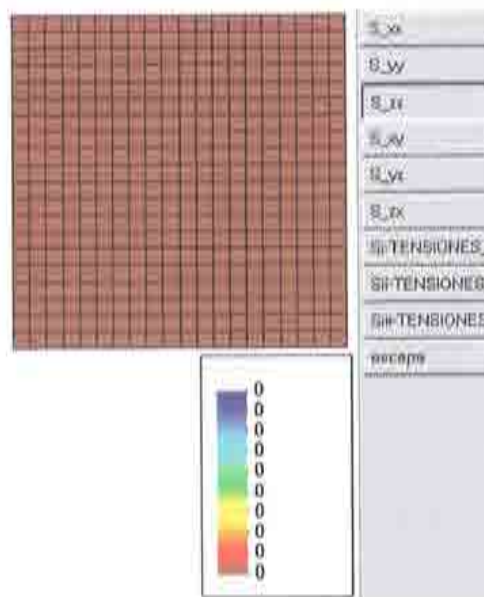


**Tensiones SYX en el postproceso de GiD, con la interface GiD – Sap90**

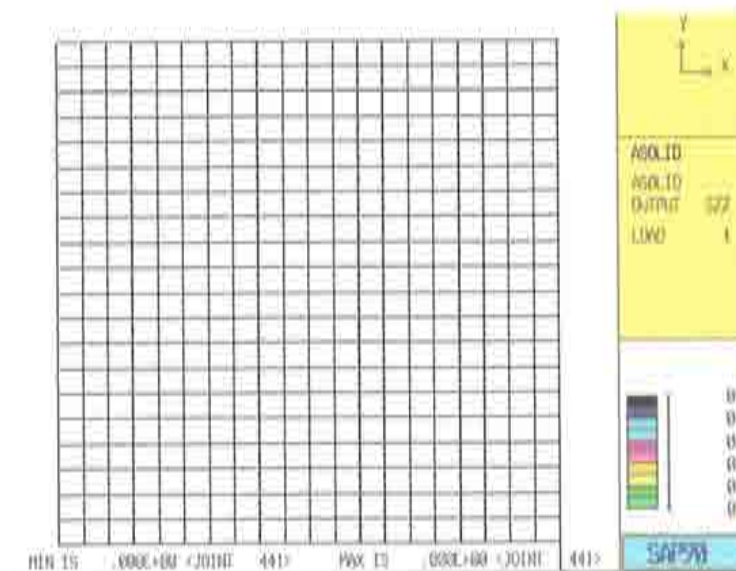


**Tensiones SYX en el postproceso de Sap90**

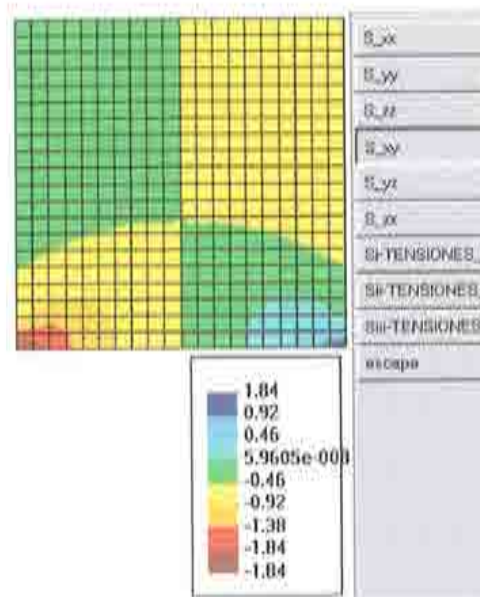




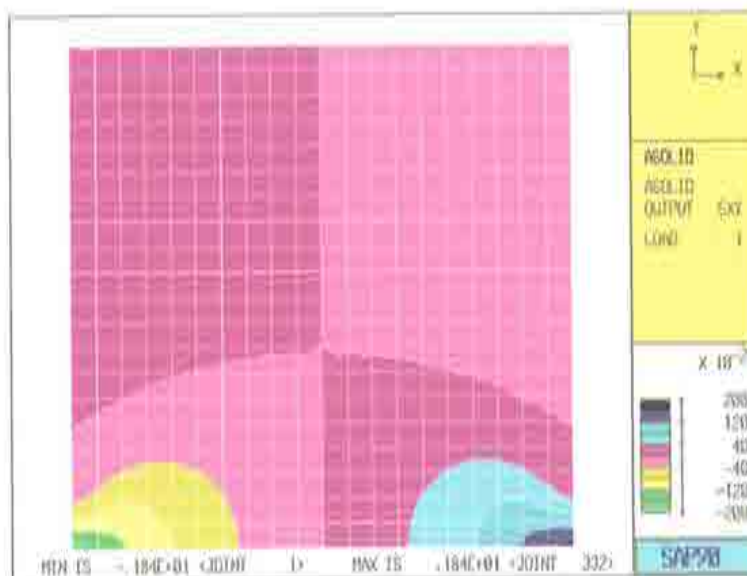
**Tensiones  $SZZ$  en el postproceso de *GiD*, con la interface *GiD – Sap90***



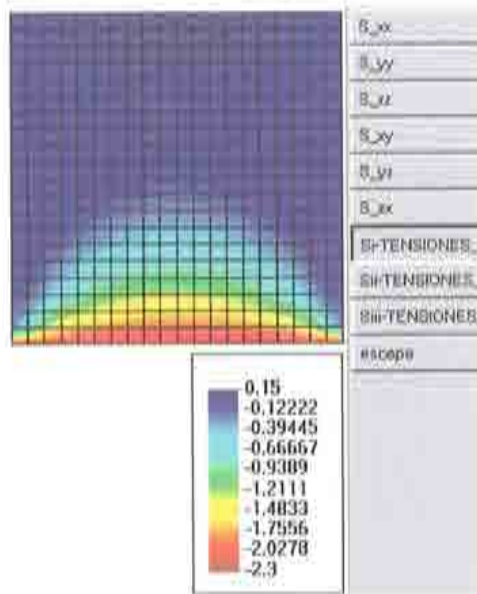
**Tensiones  $SZZ$  en el postproceso de *Sap90***



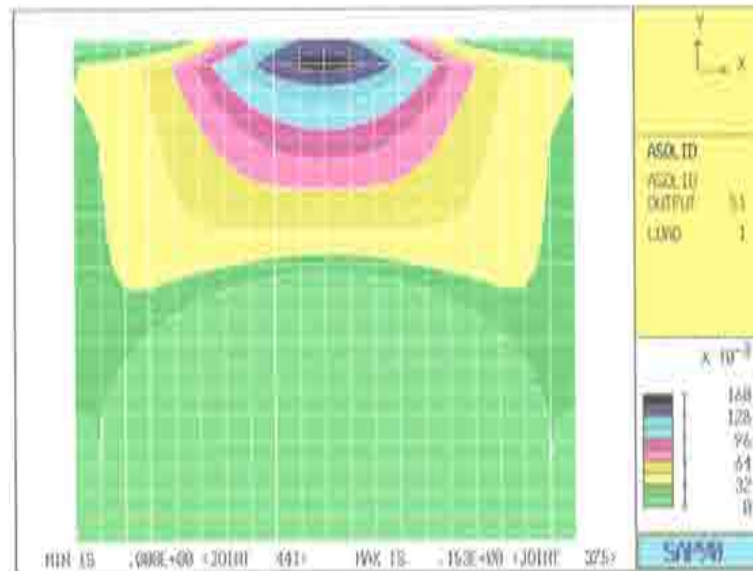
**Tensiones SXY en el postproceso de GiD, con la interface GiD – Sap90**



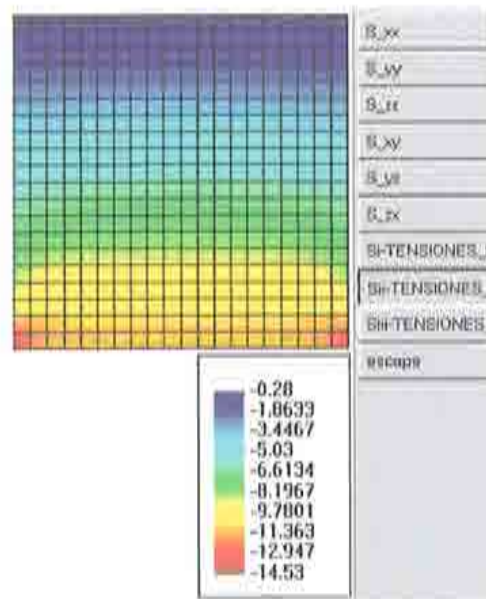
**Tensiones SXY en el postproceso de Sap90**



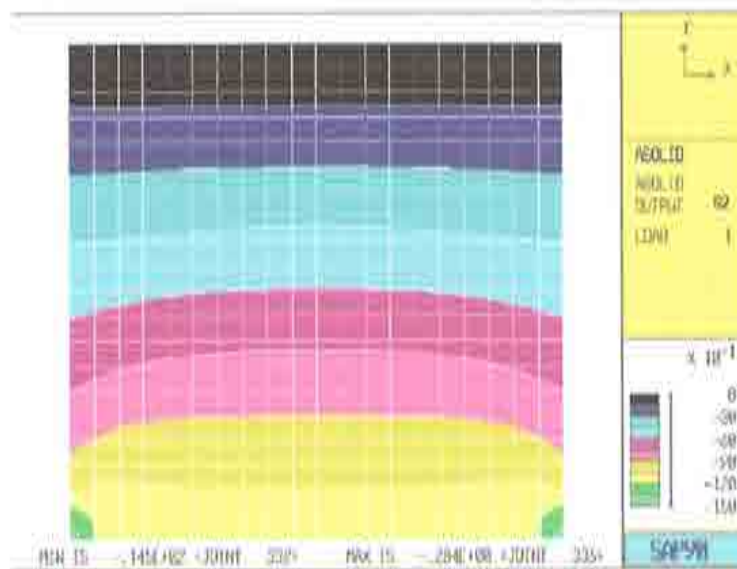
Tensiones Si en el *postproceso* de *GID*, con la *interface GID – Sap90*



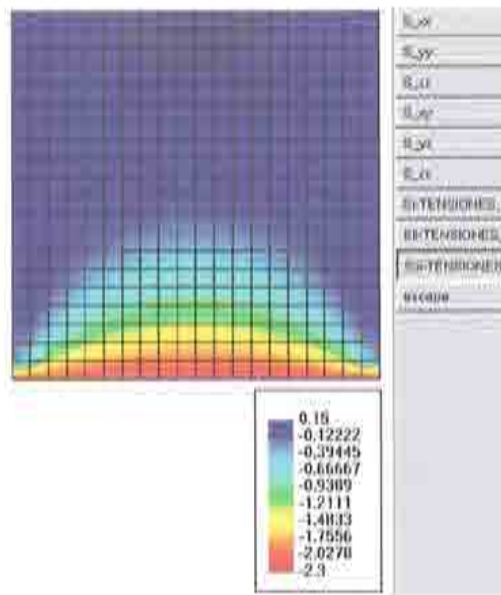
Tensiones S1 en el *postproceso* de *Sap90*



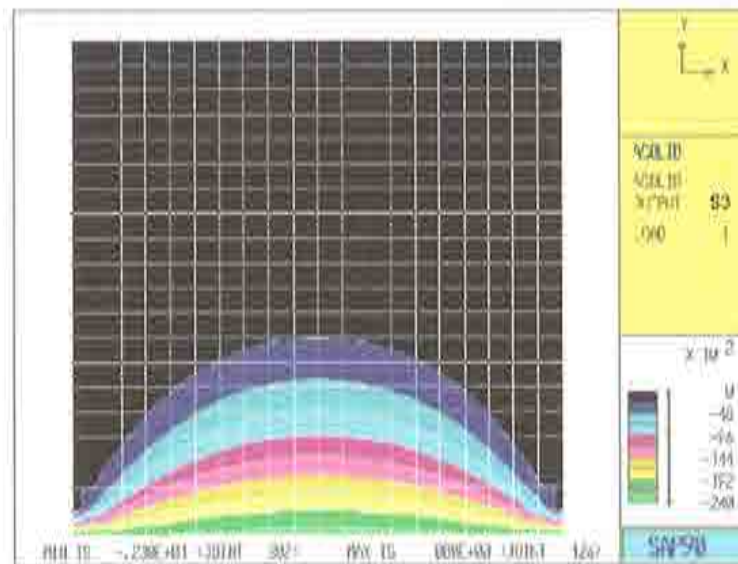
Tensiones  $S_{11}$  en el *postproceso* de *GiD*, con la *interface GiD – Sap90*



Tensiones  $S_2$  en el *postproceso* de *Sap90*



Tensiones S11 en el *postproceso* de *GiD*, con la *interface GiD-Sap90*



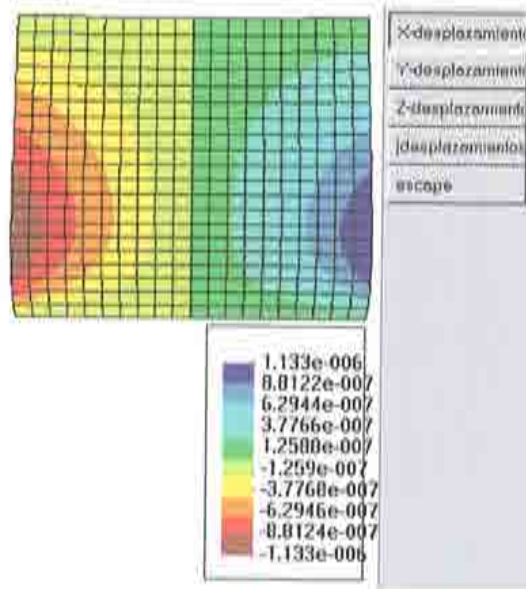
Tensiones S3 en el *postproceso* de *Sap90*



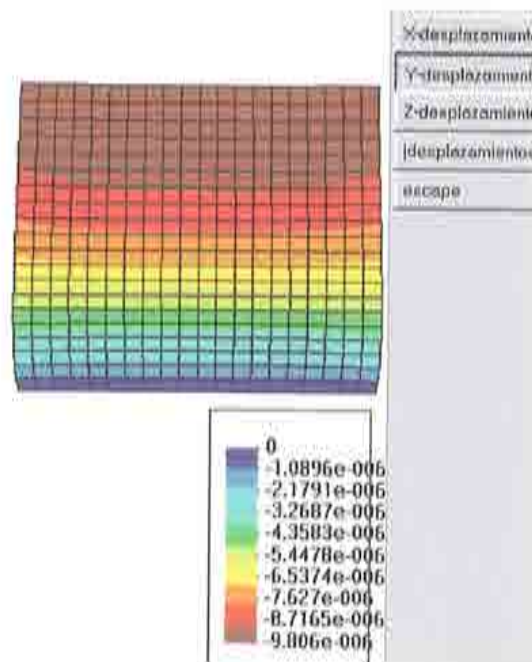
### 9.6.3 Visualización de los resultados deformacionales.

Aquí solo anexaremos los valores de deformación y la misma deformada de la Viga pared.

#### \* VALORES DEFORMACIONALES



#### Deformada y desplazamientos en el eje X



#### Deformada y desplazamientos en el eje Y

#### 9.6.4 Herramientas del postproceso

Dentro del *postproceso* podemos realizar:

- *Salvar archivos de imagen con varias extensiones.*
- *Hacer Animaciones de la estructura y sus resultados.*
- *Renders.*
- *Comentarios.*
- *Hacer cortes en la malla.*
- *Designar variables.*

Existen más opciones dentro del *postproceso*, pero solo hemos enumerado las más comunes y útiles en este trabajo.

#### 9.6.5 Salir del postproceso.

Toda vez que hemos obtenido los resultados deseados, el usuario puede salir del programa con solo picar el icono de *quit*.

En el caso que los resultados no hayan sido los adecuados, el usuario puede regresar al *preproceso* e iniciar de nuevo el análisis de la estructura.



## 10 PREPROCESO, CÁLCULO Y POSTPROCESO DE UN FORJADO (interface shell)

### 10.1 Preproceso

En el capítulo anterior analizamos un Viga pared, por medio de la teoría de Elasticidad Bidimensional (*interface Asolid.gid*) concretamente, con la de Tensión Plana. En éste mismo capítulo se detallo, de manera amplia, todo el procedimiento a seguir dentro de la *interface*, así como los resultados obtenidos.

A continuación, se presenta la *interface*, para el caso de *Shell.gid*, de Teoría de Placas. En este capítulo vamos a analizar una lámina (shell), utilizando la teoría de placas. Y mostraremos lo fácil y rápido que es este proceso.

Consideremos un forjado de 5 x 5 mts. que se encuentra empotrado en dos de sus lados paralelos. El material de este será de hormigón armado, con un espesor de 0.10 mts. y solo se calculara a peso propio.

#### 10.1.1 Generación de la geometría

- La generación de esta geometría es muy sencilla, así que lo haremos desde el preproceso de *GID*.

Recordemos, que vamos a generar un cuadrado simple con dimensiones de 5 x 5.

La partición (generación de malla) será de 20 por 20 elementos.

Utilizaremos la Teoría de placas.

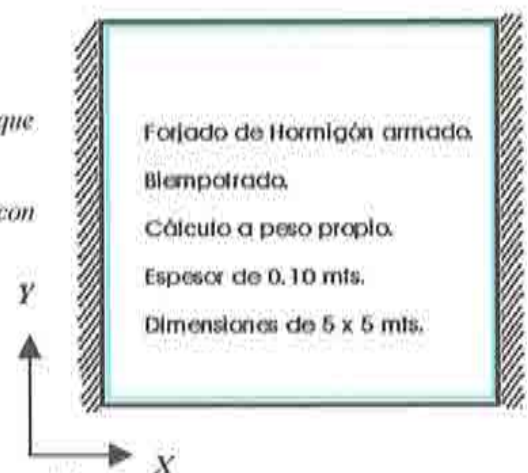


Fig. 10.1 Generación de la Geometría.  
(Forjado 5 x 5)

Una vez que se entra al programa *GiD*, vamos a generar la geometría. Primero, recordemos que hay que generar los puntos de contorno. Después, construiremos el perímetro del cuadrilátero uniendo los puntos generados (también es posible generar líneas directamente). No hay que olvidar que hay que darle nombre al problema.

A partir de este momento, estamos en condiciones para crear la malla.

### 10.1.2 Creación de la malla

- *Creación de Superficie:* Una vez que se ha creado la geometría es necesario crear una superficie, la cual nos permitirá generar la malla.

El usuario debe de acceder a la ventana *Geometry* y seleccionar la opción *4 sided surface*. A continuación, es necesario seleccionar la geometría y el programa hará la generación de la superficie.

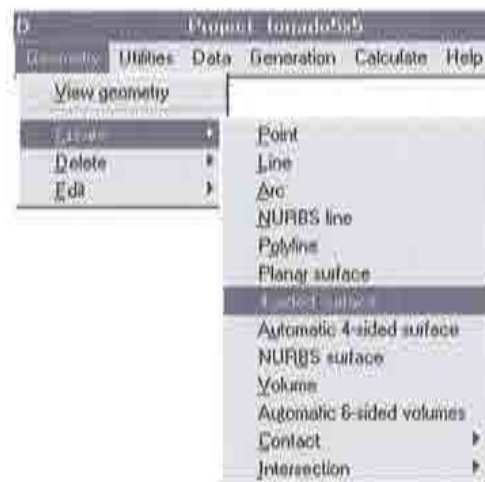


Fig. 10.2 Generación de la Superficie.  
(Forjado 5 x 5)

- *Generación de malla:* toda vez que tenemos la geometría y la superficie, podemos comenzar a generar la malla para realizar el cálculo.

Iniciaremos por seleccionar la ventana de *Generation* y seleccionar la opción *Quadratic elements* donde distinguiremos el tipo de elemento a utilizar.

Los elementos pueden ser:

- Quadratic
- Quadratic 9
- Normal

\* Además podemos obtener información sobre el tipo de elemento seleccionado.

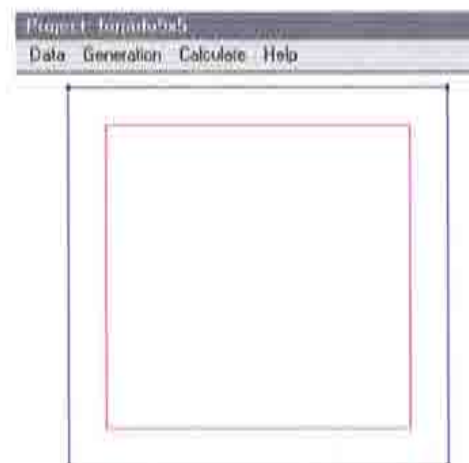


Fig. 10.3 Generación de la malla.  
(Forjado 5 x 5)



- **Parámetros de la malla:** en este análisis generaremos una malla de cuatro nodos y una partición de 20 x 20.

Los pasos a seguir son:

1º Estructurar la malla con superficies y solicitarle al programa la partición necesaria.

2º Seleccionar el tipo de elemento, que en nuestro caso será el cuadrilátero.

3º Generar la malla.

Ahora solo queda solicitarle al programa que nos muestre la malla.

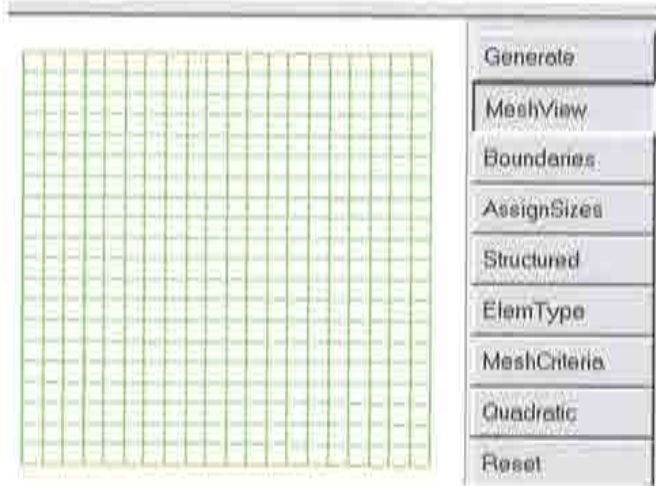


Fig. 10.4 Parámetros de la malla.  
(Forjado 5 x 5)

### 10.1.3 Definición de los parámetros para su cálculo

A continuación, tenemos que dar los parámetros del problema para su análisis:

- Primero, tenemos que seleccionar el tipo de problema a resolver, que para nuestro caso será el de Shell.gid.

El programa leerá los datos de Shell.gid y nos informará si todo está correcto.

Lo siguiente será, definir todos los datos que se requieran para el cálculo, tales como: condiciones de borde, tipo de material y los datos del problema.



Fig. 10.5 Parámetros del análisis.  
(Forjado 5 x 5)

- **Condiciones de borde:** en este análisis sabemos que el forjado se encuentra biempotrado, por lo que las condiciones de borde serán:

- Los nodos de los bordes paralelos al eje y tendrán la condición de restricción total en lo que respecta a giros y desplazamientos (1,1,1,1,1,1).
- El resto de los nodos, serán libres sin ninguna restricción (0,0,0,0,0,0).
- Aquí no aplicaremos carga alguna, ya que el análisis será solo a peso propio.

Sin embargo, la interface Shell permite aplicar cargas puntuales y momentos a los nodos.

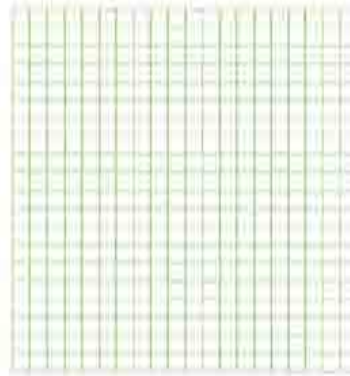


Fig. 10.6 Condiciones de borde.  
(Forjado 5 x 5)

- **Características del material:** el usuario debe de seleccionar la ventana de materiales y a continuación escoger el tipo de material a utilizar.

\* Recordemos que la selección de los elementos debe de hacerse con la malla generada.

Para nuestro problema utilizaremos el material de hormigón armado, el cual ya se encuentra configurado en el archivo *shell.mat*

El forjado es de hormigón armado y tiene un espesor de 0.10 mts.

\* No hay que olvidar que se pueden configurar nuevos materiales dentro de esta ventana, sin haberlo configurado previamente en el archivo *mat*.

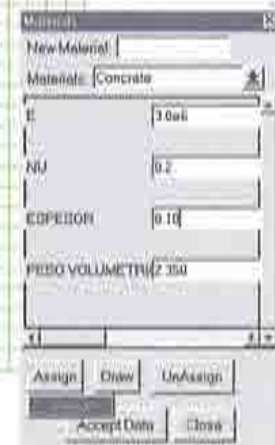
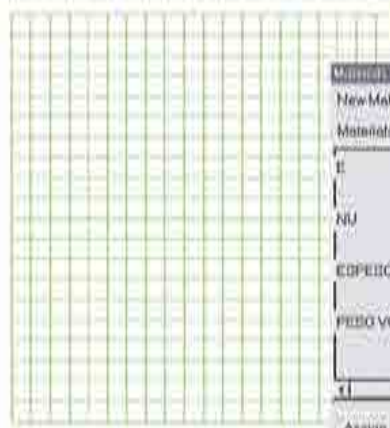


Fig. 10.7 Características del material.  
(Forjado 5 x 5)

- **Datos del problema:** en esta ventana, se van a incluir todos los datos restantes del problema, tales como:

*Peso propio.*  
*Título del problema.*  
*Tipo de problema.*  
*Número de materiales etc.*

No hay que olvidar, que cada programa de análisis de estructura tiene sus propios parámetros y que el usuario puede configurarlos en el archivo. prb.

\* *Nota:* Todos estos datos serán recogidos y ordenados gracias al archivo.bas, el cual generará otro que será el utilizado por el programa Sap90 para su análisis.

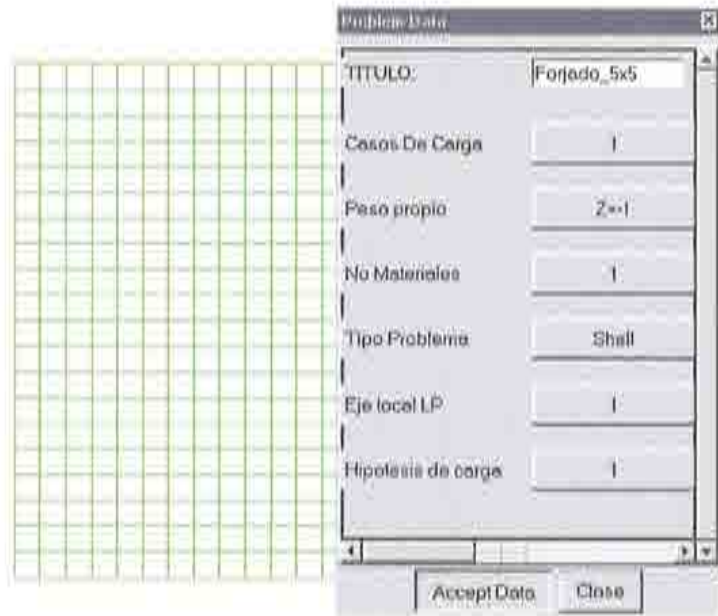


Fig. 10.8 Datos del problema.  
 (Forjado 5 x 5)

## 10.2 Cálculo

Ahora solo nos queda seleccionar la ventana *Calculate* y el programa *GID* iniciará el cálculo conjuntamente con el *Sap90*. Y, el usuario solo tiene que acceder al *postproceso* para visualizar los resultados gráficos.



Fig. 10.9 Análisis del problema.  
 (Forjado 5 x 5)

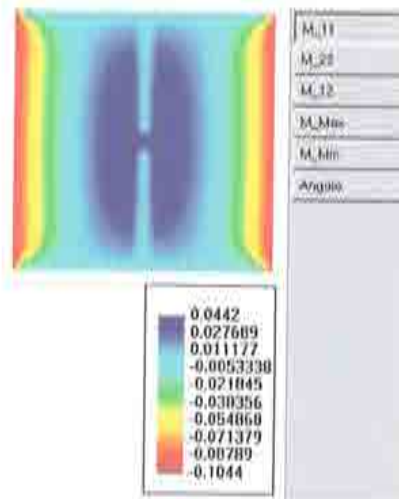
### 10.3 Postproceso

Como puede apreciarse, todo el *preproceso* realizado hasta el momento ha sido de forma sencilla y dinámica. Esto, muestra la eficacia del programa *GiD* en la fase de la preparación de los datos y la generación de malla, previa al análisis de la estructura. Este análisis, es realizado con programas informáticos, ya sean comerciales o creados por el usuario.

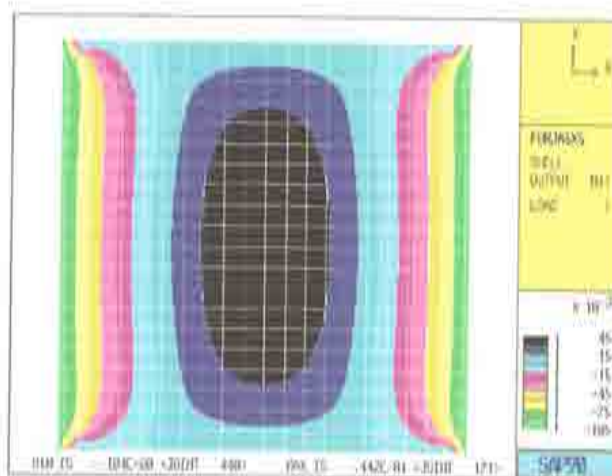
Finalmente, mostraremos los resultados obtenidos en el análisis y la facilidad con que se pueden configurar en el *postproceso* y manipular los datos generados de manera gráfica.

Al igual que en el capítulo 9, se presentaran los resultados del análisis en los dos formatos, el *postproceso* de *GiD* y el de *Sap90*.

#### ESFUERZOS OBTENIDOS

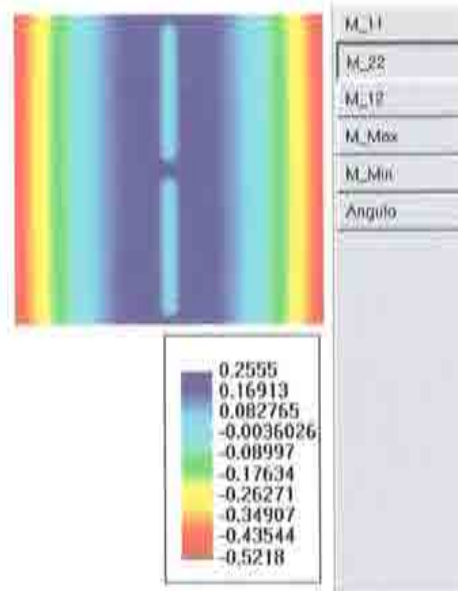


Momento 11 del *postproceso* de *GiD*

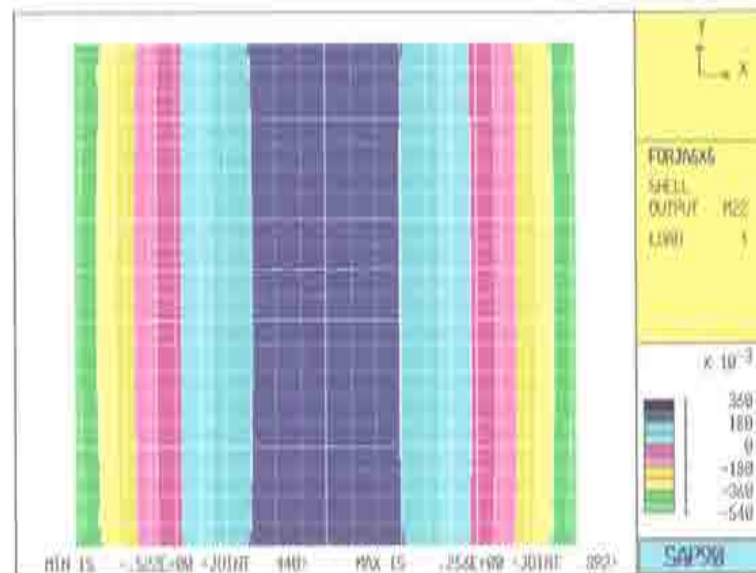


Momento 11 del *postproceso* de *Sap90*



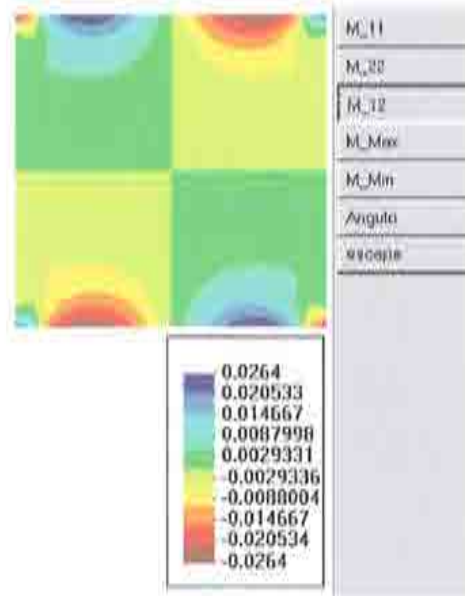


**Momento 22 del postproceso de GID**

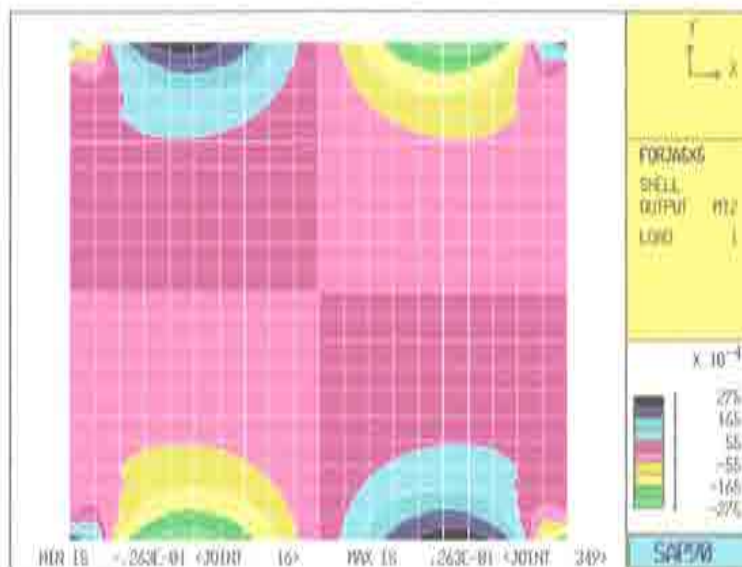


**Momento 22 del postproceso de Sap90**

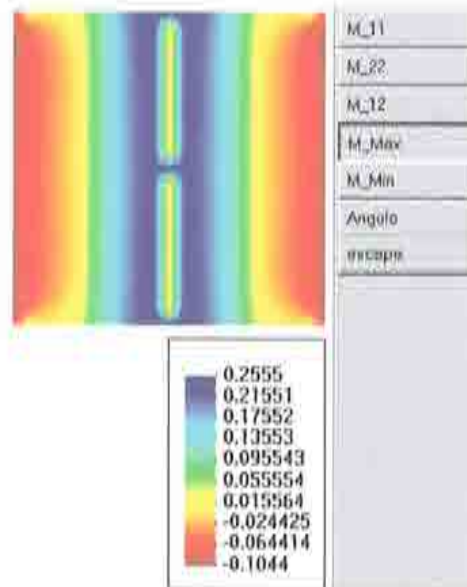




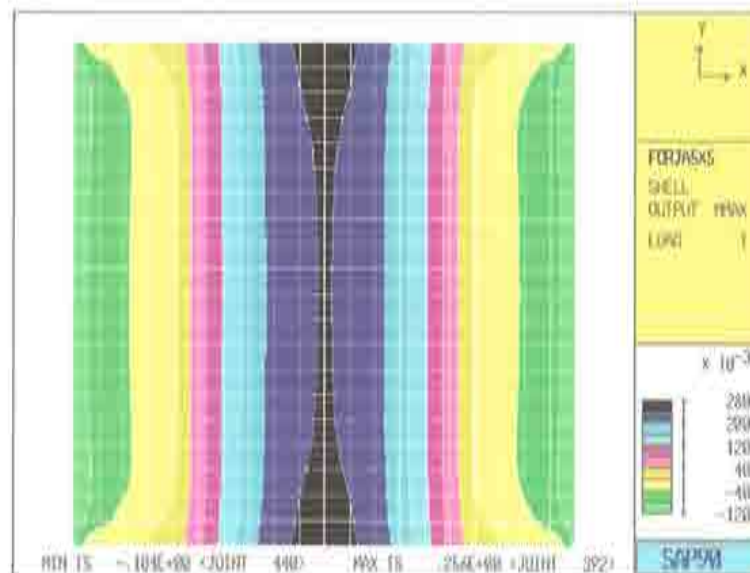
**Momento 12 del *postproceso* de *GiD*.**



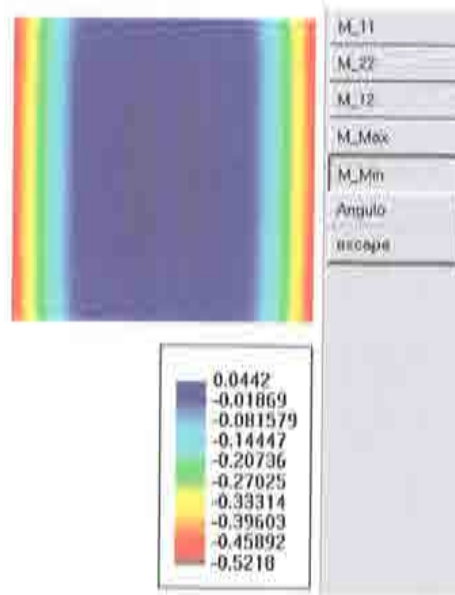
**Momento 12 del *postproceso* de *Sap90*.**



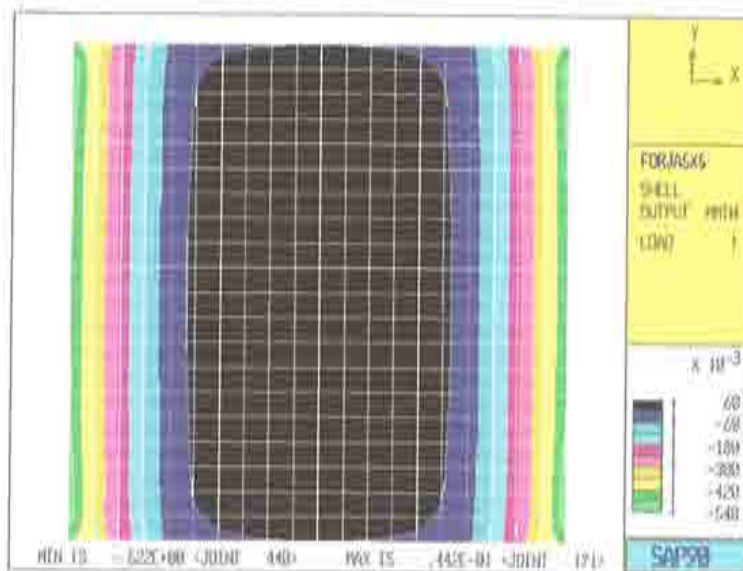
**Momento Máximo del *postproceso* de GID**



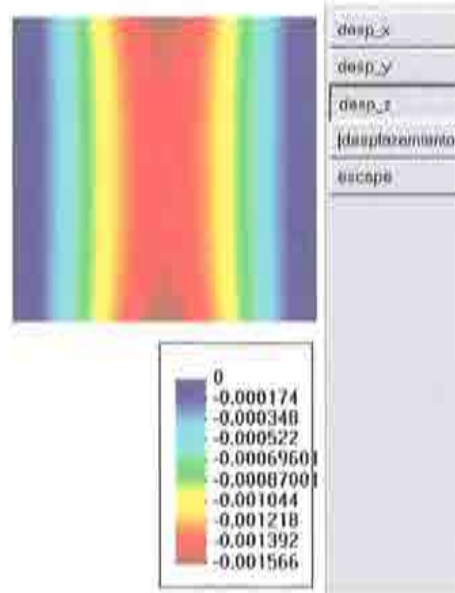
**Momento Máximo del *postproceso* de Sap90**



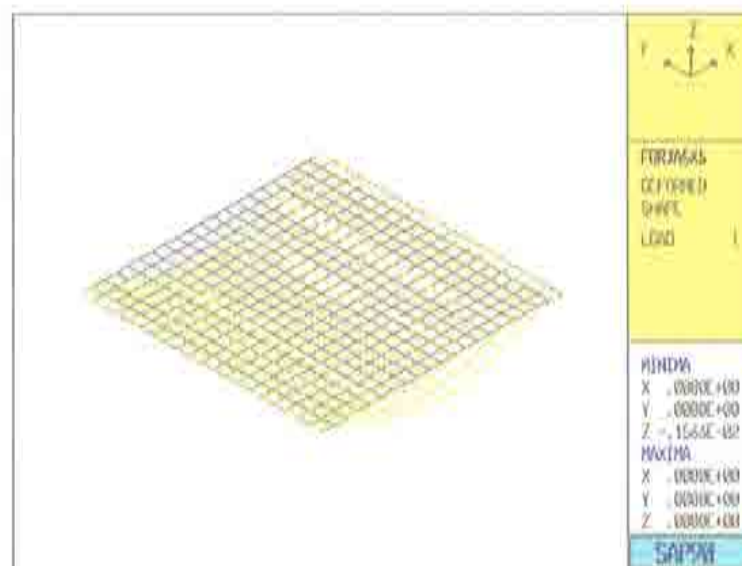
**Momento Mínimo del *postproceso* de GID**



**Momento Mínimo del *postproceso* de Sap90**



Valores de Deformación del *postproceso* de *GiD*



Valores de Deformación del *postproceso* de *Sap90*





## 11 EJEMPLO PRÁCTICO DE LA INTERFACE 3DSHELL Y 3DSOLID (Cubierta Central del Sagrario de la Catedral de México D.F.) (Pared3d)

### 11.1 Interface 3dshell

### (Cubierta Central del Sagrario, México)

En este capítulo solo se muestran los resultados del análisis y la comparación entre el postproceso gráfico de *Gid* y el de *Sap90*. Cabe mencionar, que el análisis del preproceso y cálculo se detallan ampliamente en los capítulos 9 y 10, y la configuración de los archivos de esta interfaz se muestran en el Anexo 3.

#### 11.1.1 Descripción del problema

Se trata de la cubierta central del Sagrario de la Catedral de la Cd. de México, que fue construida a partir del segundo tercio del siglo XVI, sobre ruinas de la cd. prehispánica (Tenochtitlan).

En el análisis de este problema solo se considera el peso propio de la estructura, y las características mecánicas del material son las de fábrica convencional. En este estudio solo se realiza un análisis estático lineal.

#### \* Características del material (fábrica):

Módulo de elasticidad.....  $E = 5.0e5 \text{ T/m}^2$

Módulo de Poisson.....  $\nu = 0.2$

Espesor.....  $TH = 0.30 \text{ mts.}$

Peso Volumétrico.....  $W = 1.8 \text{ T/m}^3$

Tipo de elemento.....  $Etype = 0$   
(membrane plus bending)

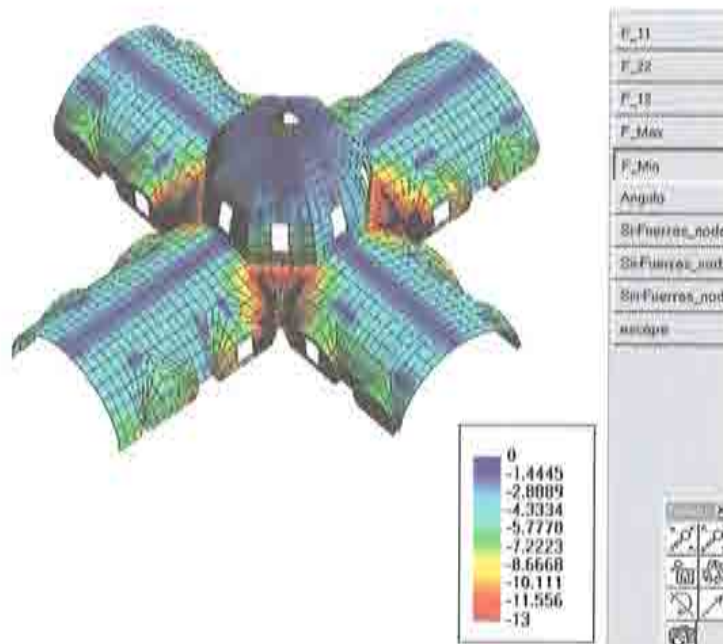


Fig. 11.1 Cubierta del Sagrario  
(Análisis a peso propio)

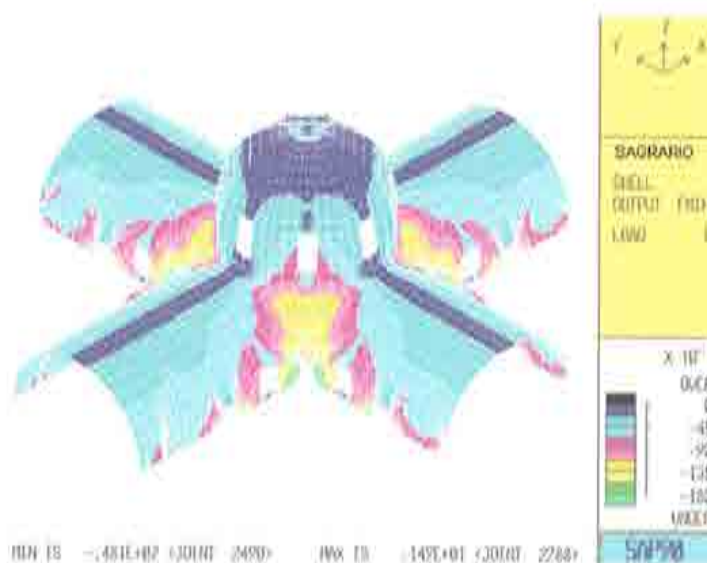
### 11.1.2 Postproceso

Como ya se comento en el subapartado anterior, solo incluiremos aquí los resultados gráficos del análisis.

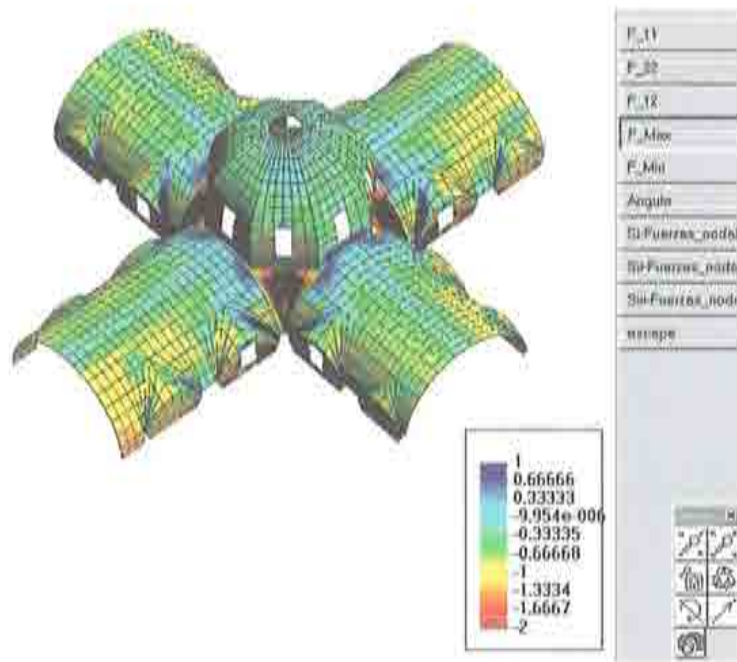
#### FUERZAS OBTENIDAS



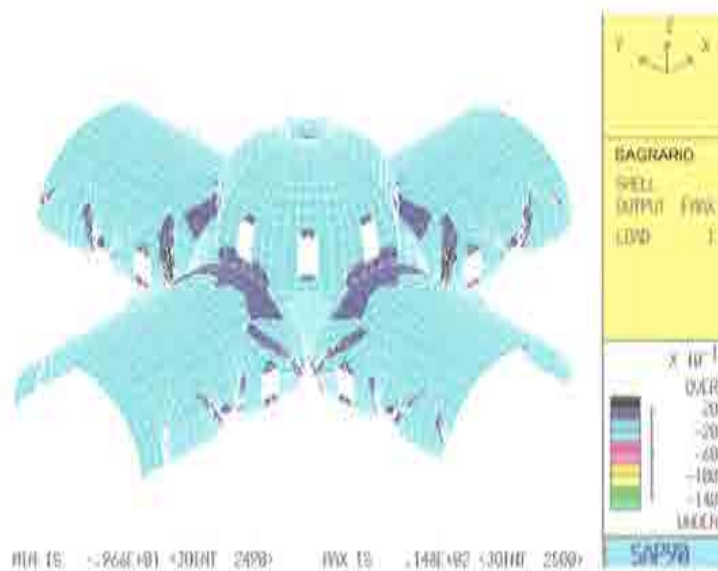
Fuerzas Mínimas del *postproceso* de *GiD*.



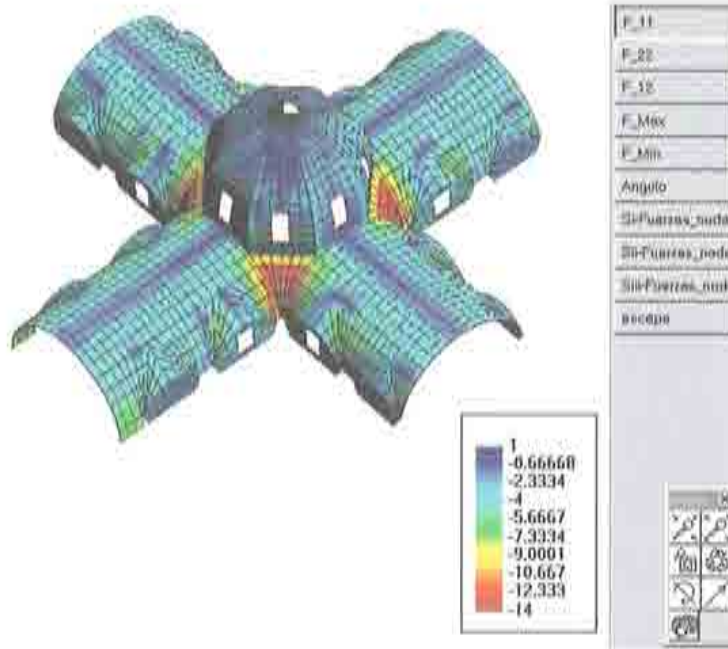
Fuerzas Mínimas del *postproceso* de *Sap90*



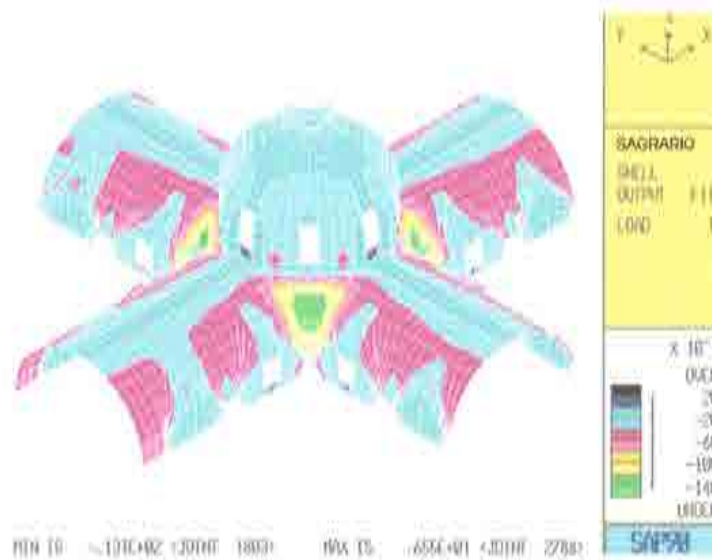
**Fuerzas Máximas del *postproceso* de GID**



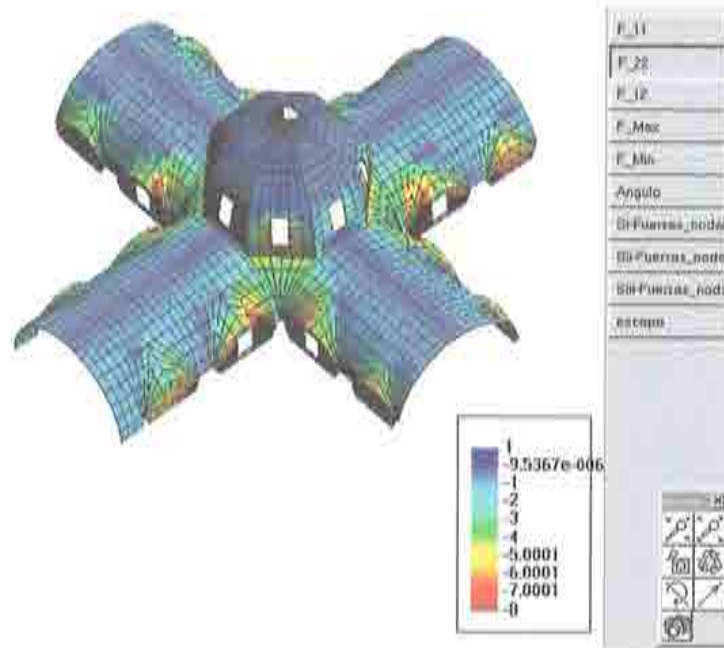
**Fuerzas Máximas del *postproceso* de Sap90**



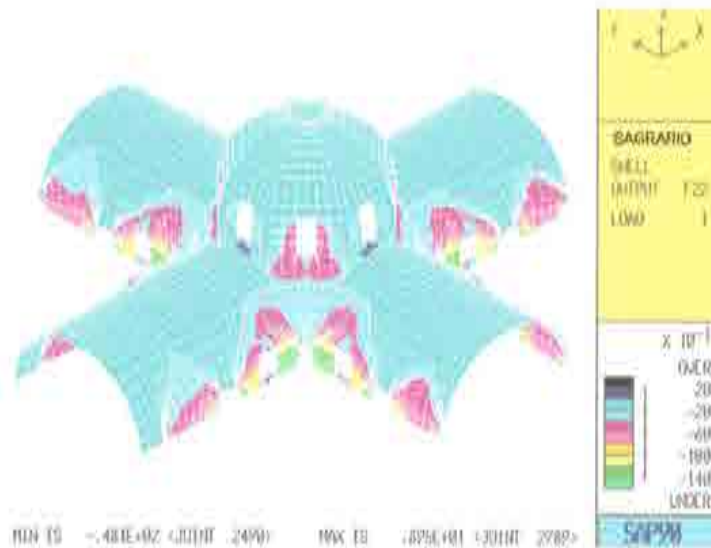
Fuerzas 11 del *postproceso* de *GiD*



Fuerzas 11 del *postproceso* de *Sap90*

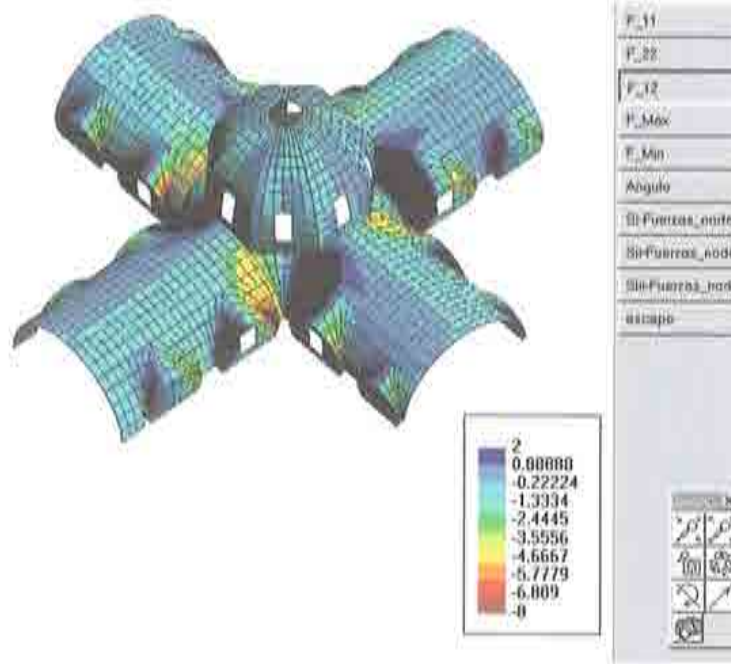


**Fuerzas 22 del postproceso de GID**

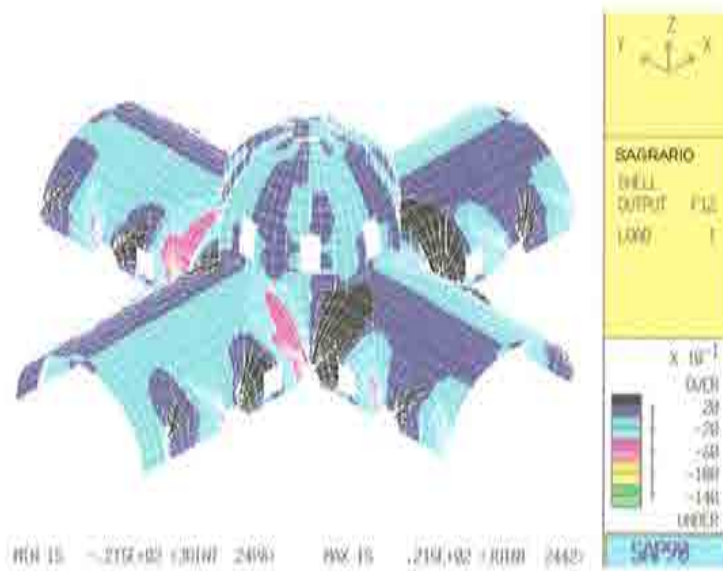


**Fuerzas 22 del postproceso de Sap90**

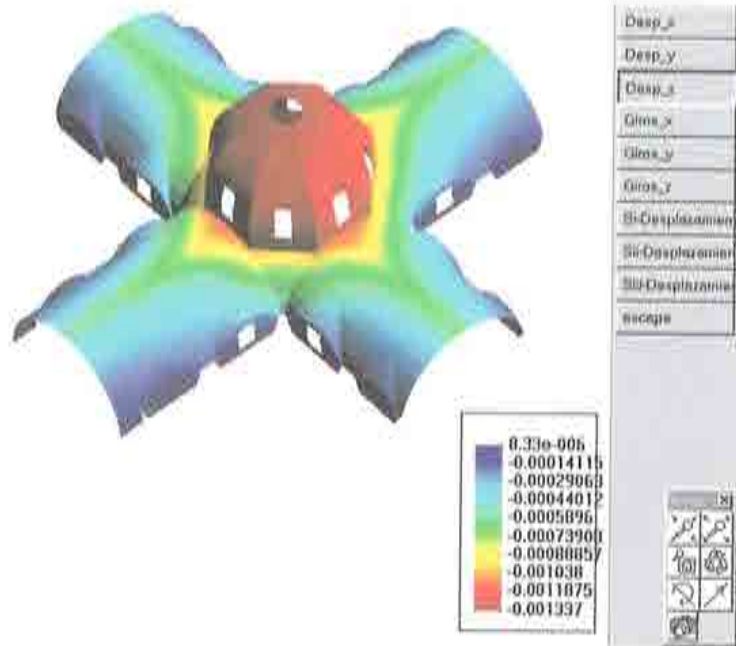




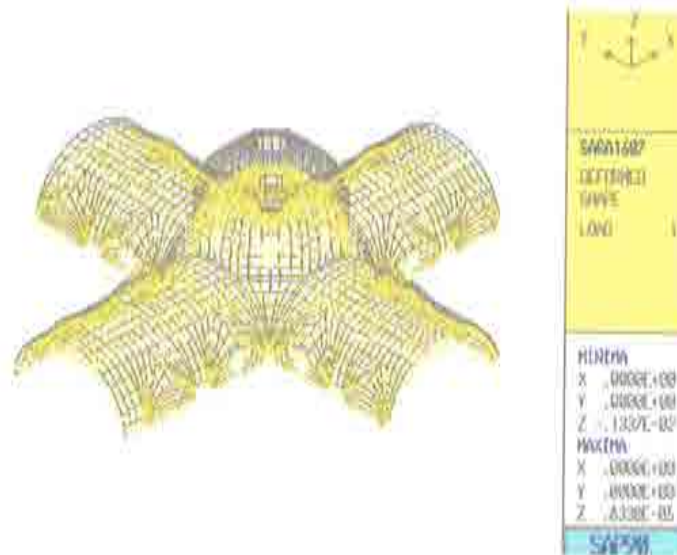
Fuerzas 12 del *postproceso* de GiD



Fuerzas 12 del *postproceso* de Sap90



Valores de Deformación del *postproceso* de *GID*



Valores de Deformación del *postproceso* de *Sap90*

## 11.2 Interface de 3dsolid (Pared3d con carga uniforme)

Como se explico en el apartada 11.1, aqui solo se muestran los resultados gráficos de la interface 3dSolid de *GID-Sap90*. Los archivos de la configuración se muestran en el anexo A4.

Es importe recordar, que el usuario puede configurar los archivos para cualquier tipo de problema y ante cualquier programa de análisis de estructuras que tengan formatos similares a *Sap90* y *Calsef*, por ejemplo.

### 11.2.1 Descripción del problema

Se trata de un muro de hormigón armado de  $5.00 \times 5.00$  mts. Con un espesor de  $0.20$  mts, una carga uniformemente repartida de  $1.0$  T/ml y empotrado en su base

El estudio se basa en un análisis estático- lineal, se considera el peso propio, así como la carga.

Las características mecánicas del material son las de un hormigón convencional.

\* Características del material:

Módulo de elasticidad.....  $E = 3.0e6$  T/m<sup>2</sup>

Módulo de Poisson.....  $U = 0.2$

Espesor.....  $TH = 0.30$  mts.

Peso Volumétrico.....  $W = 2.35$  T/m<sup>3</sup>

Carga.....  $F = 1.00$  T/ml

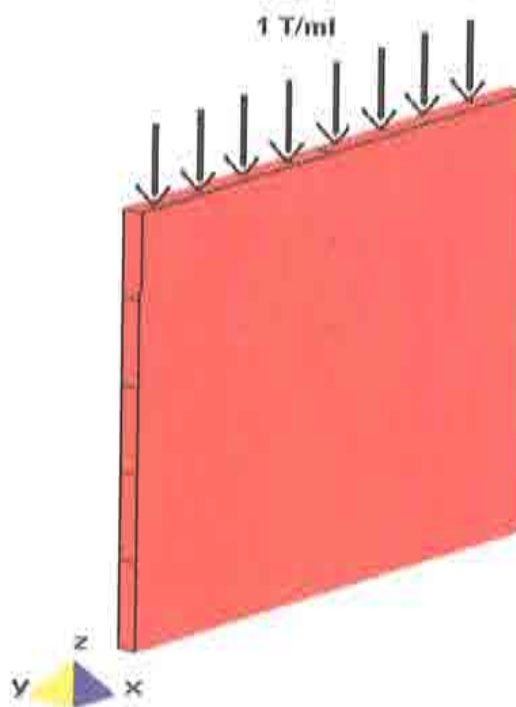
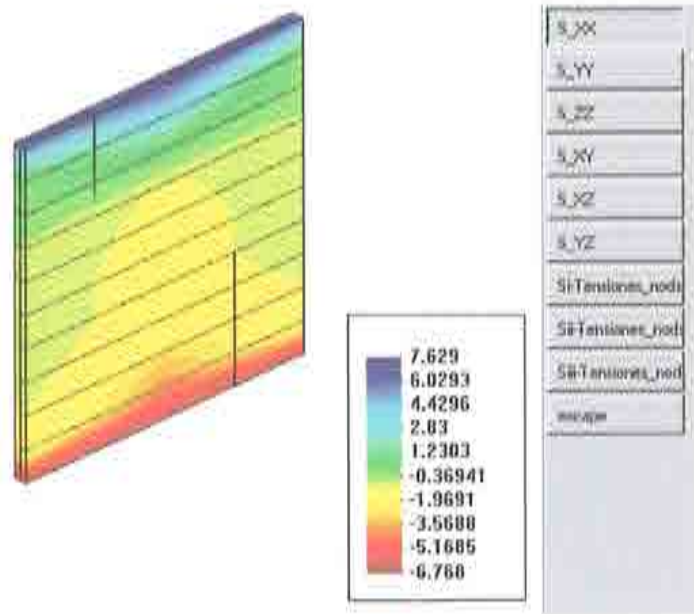


Fig. 11.2 Pared3d  
(Análisis a peso propio y carga uniforme de 1 T/ml)

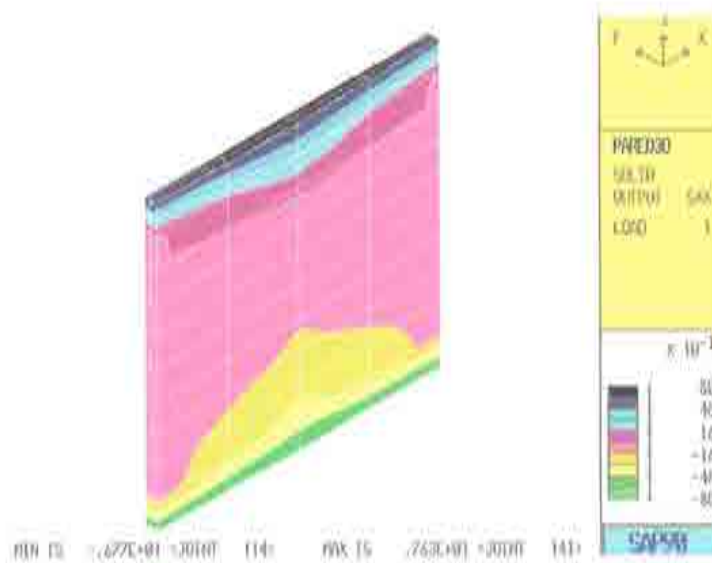
### 11.2.2 Postproceso

Resultados gráficos del análisis de la interface 3dsolid de *GiD-Sap90*

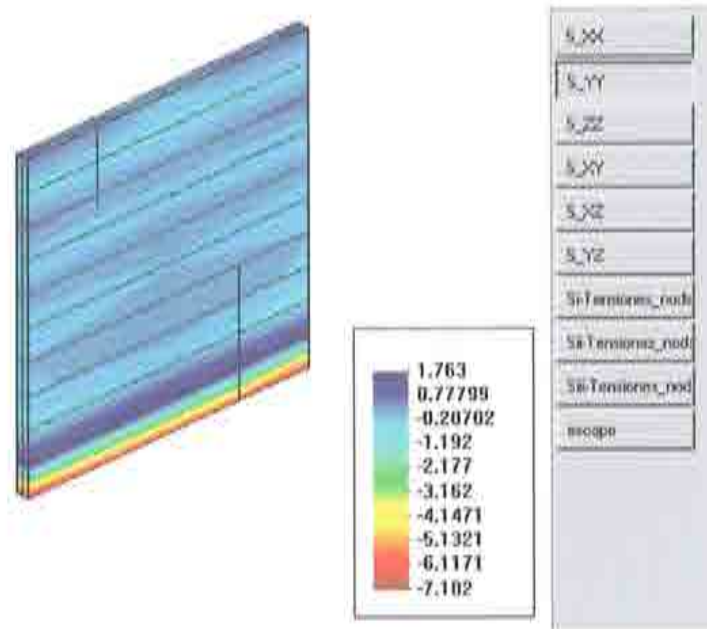
#### TENSIONES OBTENIDAS



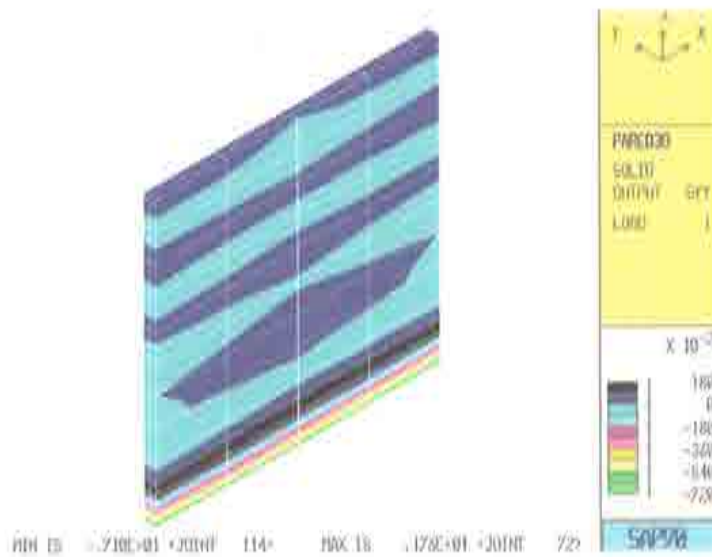
Tensiones S XX del *postproceso* de *GiD*



Tensiones S XX del *postproceso* de *Sap90*

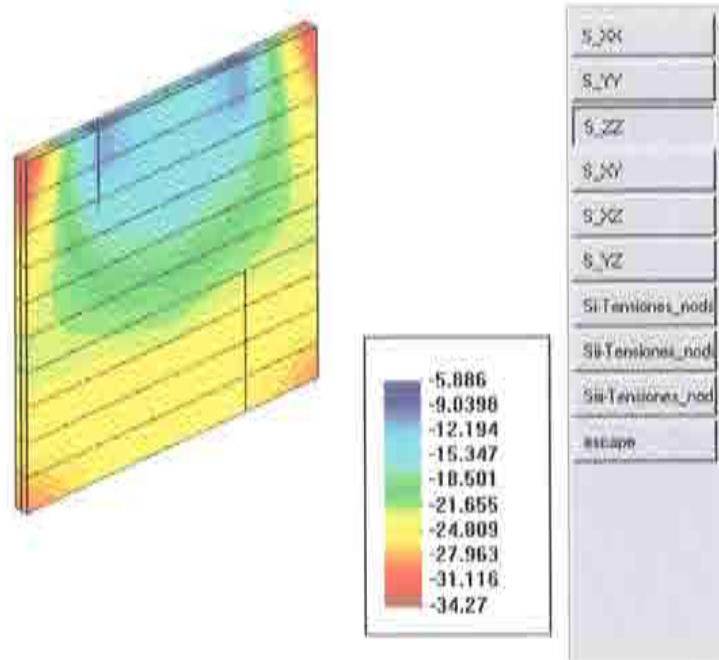


Tensiones S YY del *postproceso* de GiD

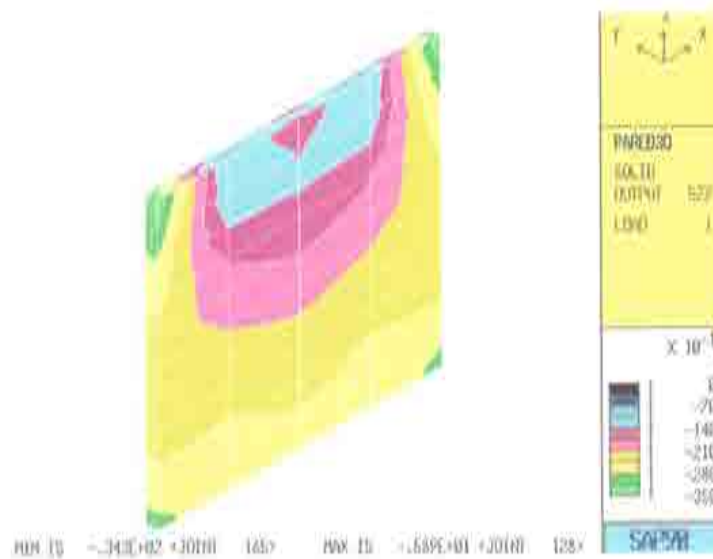


Tensiones S YY del *postproceso* de Sap90

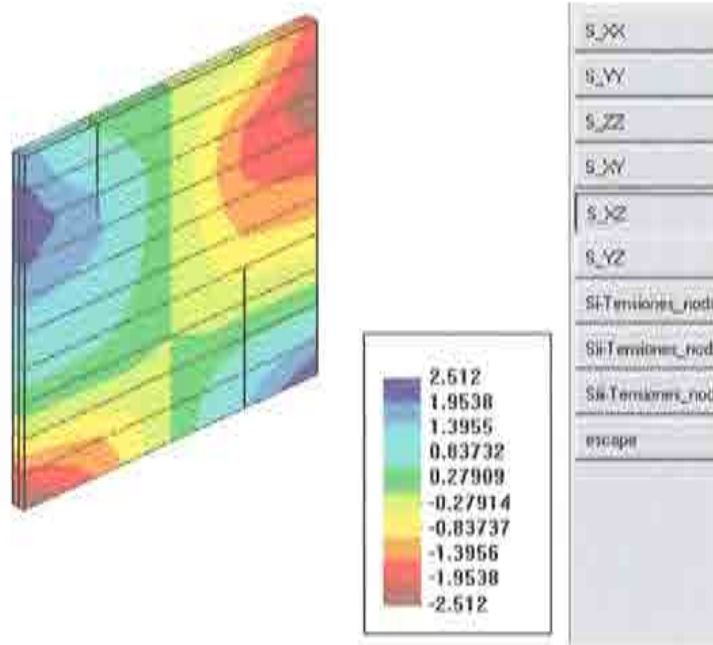




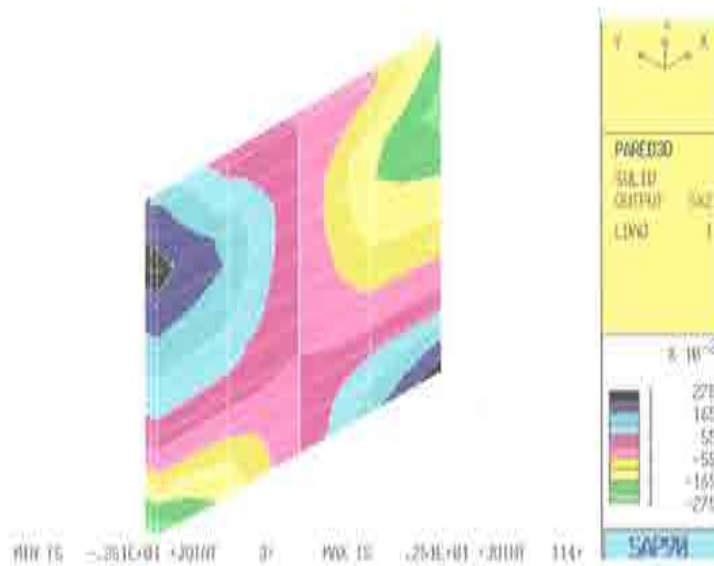
**Tensiones S ZZ del postproceso de GiD**



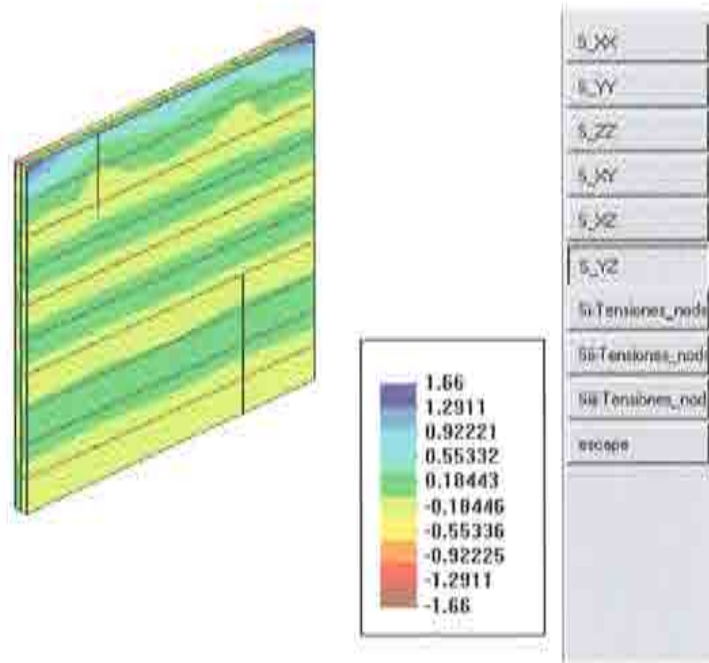
**Tensiones S ZZ del postproceso de Sap90**



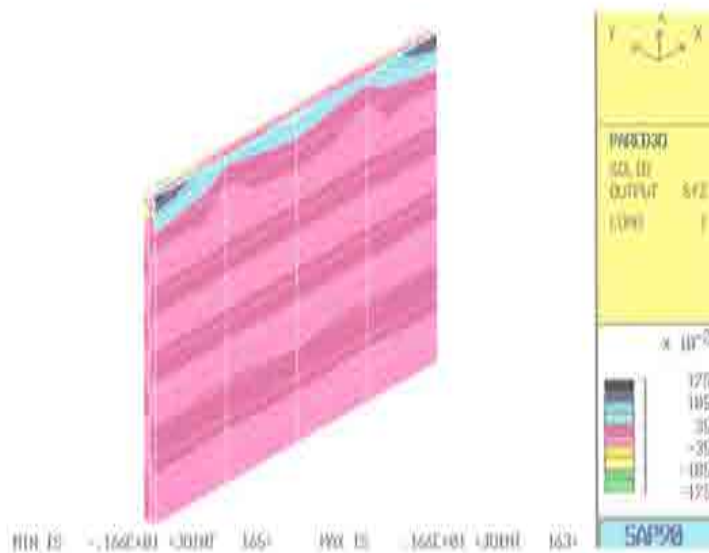
Tensiones S XZ del *postproceso* de GiD



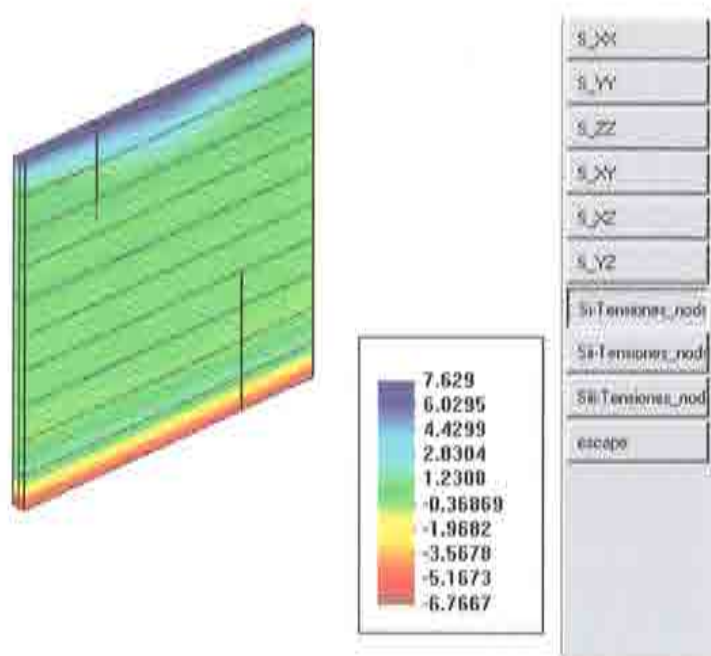
Tensiones S XZ del *postproceso* de Sap90



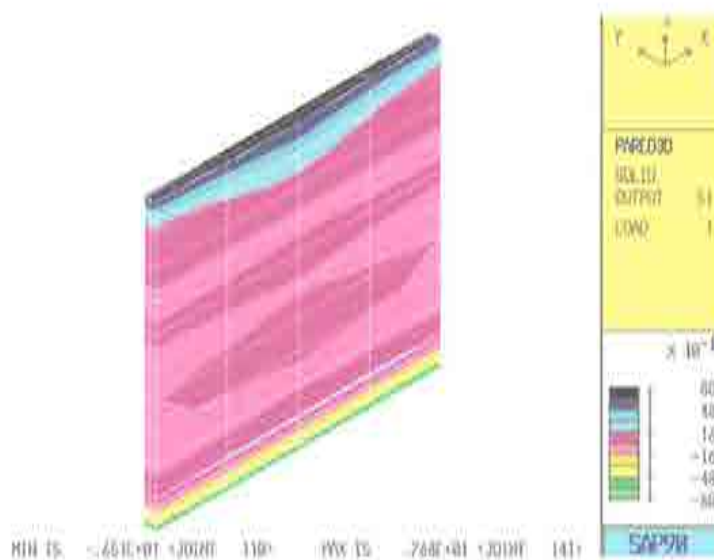
Tensiones S YZ del *postproceso* de GID



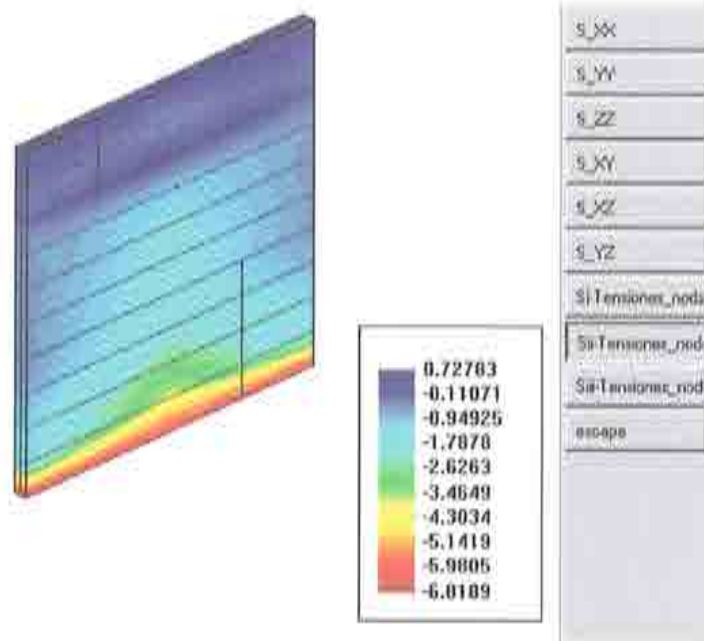
Tensiones S YZ del *postproceso* de Sap90



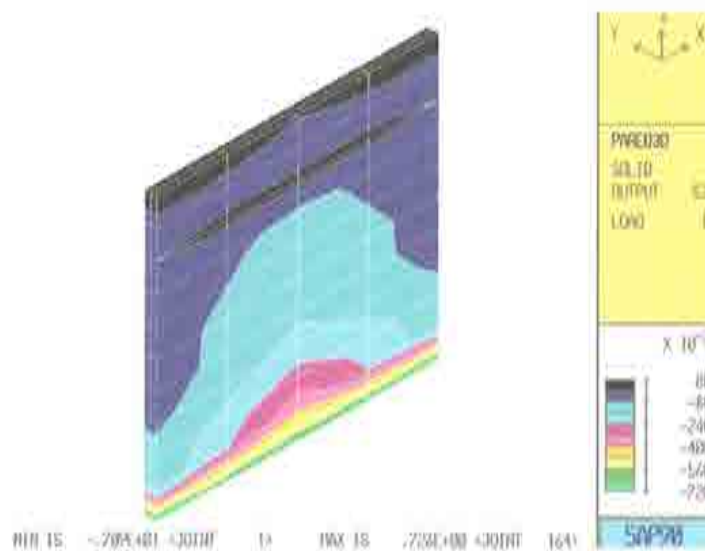
Tensiones S I del *postproceso* de *GiD*



Tensiones S I del *postproceso* de *Sap90*

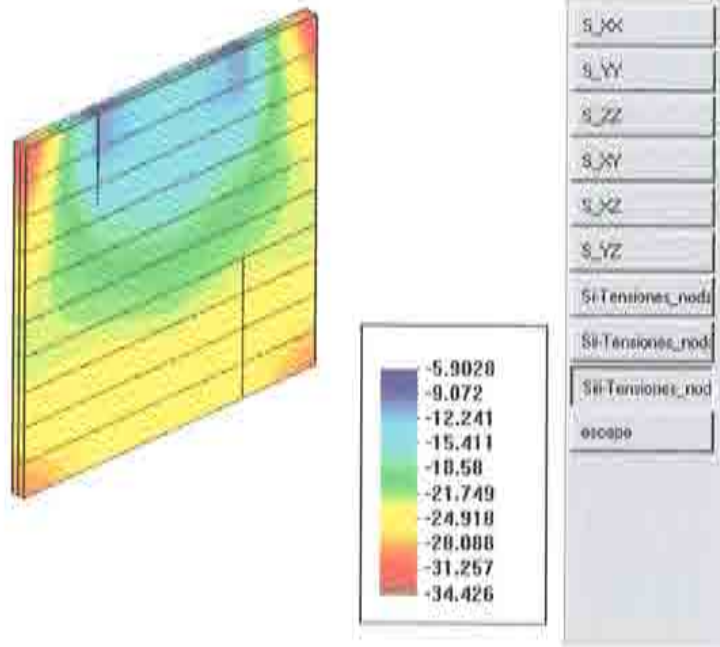


Tensiones S II del *postproceso* de *GiD*

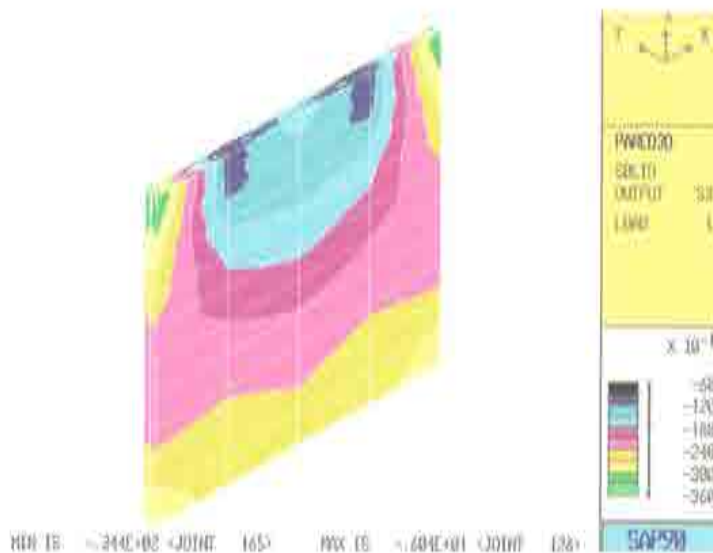


Tensiones S II del *postproceso* de *Sap90*

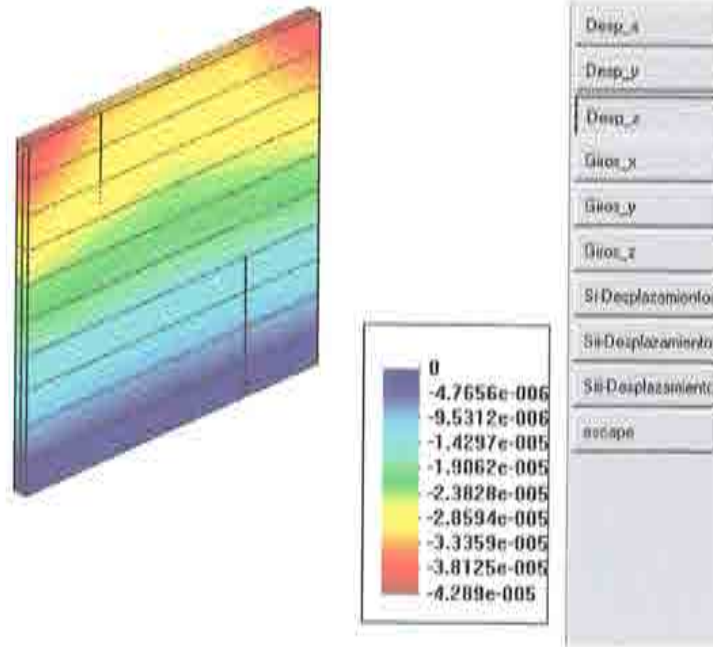




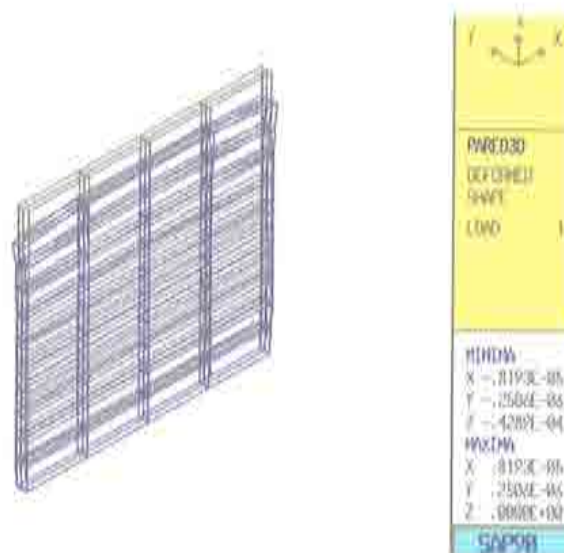
Tensiones S III del *postproceso* de *GiD*



Tensiones S III del *postproceso* de *Sap90*



Valores de Deformación del *postproceso* de GiD



Valores de Deformación del *postproceso* de Sap90

## Conclusiones

Como se puede observar, la integración (*interface*) de los programas, por un lado el *pre* y *postproceso* y por el otro el análisis de estructuras por medio del M.E.F., *GiD* y *Sap90* nos permiten de forma ágil y sencilla realizar el análisis de cualquier género estructural, por compleja que sea.

Por esta razón, en el Departamentos de Estructuras en la Arquitectura apostamos por esta *interface* y esperamos, conjuntamente con el Centro Internacional de Métodos Numéricos (*CIMNE*), que esta publicación sea de gran utilidad para cualquier usuario de programas informáticos que realicen el análisis de estructuras, sea cual sea.

## Anexos

### A.1 Anexo 1

#### A.1.1 Archivos de uso para la configuración Asolid.gid de la Interface GiD-Sap90

- A.1.1.1 Archivo Asolid2d.cnd ( Archivo de Condiciones).
- A.1.1.2 Archivo Asolid2d.mat ( Archivo de Materiales).
- A.1.1.3 Archivo Asolid2d.sim ( Archivo de Símbolos gráficos para condiciones).
- A.1.1.4 Archivo Asolid2d.prn ( Archivo de Datos del problema)
- A.1.1.5 Archivo Asolid2d.bas ( Archivo para la ordenación de los datos para su análisis)

### A.2 Anexo 2

#### A.2.1 Archivos de uso para la configuración Shell2d.gid de la Interface GiD-Sap90

- A.2.1.1 Archivo Shell2d.cnd ( Archivo de Condiciones).
- A.2.1.2 Archivo Shell2d.mat ( Archivo de Materiales).
- A.2.1.3 Archivo Shell2d.sim ( Archivo de Símbolos gráficos para condiciones).
- A.2.1.4 Archivo Shell2d.prb ( Archivo de Datos del problema)
- A.2.1.5 Archivo Shell2d.bas ( Archivo para la ordenación de los datos para su análisis)

### A.3 Anexo 3

#### A.3.1 Archivos de uso para la configuración 3dshell.gid de la Interface GiD-Sap90

- A.3.1.1 Archivo 3dshell.cnd ( Archivo de Condiciones).
- A.3.1.2 Archivo 3dshell.mat ( Archivo de Materiales).
- A.3.1.3 Archivo 3dshell.sim ( Archivo de Símbolos gráficos para condiciones).
- A.3.1.4 Archivo 3dshell.prb ( Archivo de Datos del problema)
- A.3.1.5 Archivo 3dshell.bas ( Archivo para la ordenación de los datos para su análisis)

### A.4 Anexo 4

#### A.4.1 Archivos de uso para la configuración 3dsolid.gid de la Interface GiD-Sap90

- A.4.1.1 Archivo 3dsolid.cnd ( Archivo de Condiciones).
- A.4.1.2 Archivo 3dsolid.mat ( Archivo de Materiales).
- A.4.1.3 Archivo 3dsolid.sim ( Archivo de Símbolos gráficos para condiciones).
- A.4.1.4 Archivo 3dsolid.prb ( Archivo de Datos del problema)
- A.4.1.5 Archivo 3dsolid.bas ( Archivo para la ordenación de los datos para su análisis)

## A.1 Anexo 1

### A.1.1 Archivos de uso para la configuración Asolid2d.gid de la Interface GiD-Sap90

#### A.1.1.1 Archivo Asolid2d.end (Archivo de Condiciones).

```
NUMBER: 1 CONDITION: Puntos-Restringidos
CONDTYPE: over points
QUESTION: X-Restraints
VALUE: 0
QUESTION: X-Giro
VALUE: 0
QUESTION: Y-Restraints
VALUE: 0
QUESTION: Y-Giro
VALUE: 0
QUESTION: Z-Restraints
VALUE: 0
QUESTION: Z-Giro
VALUE:0
END CONDITION
NUMBER: 2 CONDITION: Puntos-Cargas-y-Momentos
CONDTYPE: over points
QUESTION: X-Force
VALUE: 0
QUESTION: Y-Force
VALUE: 0
QUESTION: Z-Force
VALUE: 0
QUESTION: X-Momento
VALUE: 0
QUESTION: Y-Momento
VALUE: 0
QUESTION: Z-Momento
VALUE: 0
END CONDITION
```



**A.1.1.2 Archivo Asolid2d.mat (Archivo de Materiales).**

```
NUMBER: 1 MATERIAL: Steel
QUESTION: E
VALUE: 2.1e07
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.0
QUESTION: PESO_VOLUMETRICO
VALUE: 7.800
END MATERIAL
NUMBER: 2 MATERIAL: Concrete
QUESTION: E
VALUE: 3.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 2.350
END MATERIAL
NUMBER: 3 MATERIAL: Madera
QUESTION: E
VALUE: 1.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 0.800
END MATERIAL
NUMBER: 4 MATERIAL: Fundición
END MATERIAL
.....
.....
NUMBER: 18 MATERIAL: Fabrical4
QUESTION: E
VALUE: 5.0e5
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.15
QUESTION: PESO_VOLUMETRICO
VALUE: 20.2
END MATERIAL
```

### A.1.1.3 Archivo Asolid2d.sim (Archivo de Símbolos gráficos para condiciones).

```

cond Point-Constraints
3
global
cond(int,5)
1
0
0
apoi3d.geo
global
cond(int,1) && cond(int,3)
1
0
0
apoi.geo
global
cond(int,1) || cond(int,3)
cond(int,3)
cond(int,1)*(-1)
0
apoi-rodillo.geo
cond Point-Load
1
global
1
cond(real,1)
cond(real,2)
cond(real,3)
fletxa.geo
cond Face-Load
3
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))<1e-5
1
0
0
Normal.geo
local
fabs(cond(real,1))<1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Tangencial.geo
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Normal-Tangen.geo
cond Surface-Load
1
local
1
-1
0

```

```
0
Normal-Superf.geo
cond Line-Constraints
1
local
1
1
0
0
apoiol.geo
```

#### A.1.1.4 Archivo Asolid2d.prb (Archivo de Datos del problema)

```
TITULO: Title_name
Casos_De_Carga#CB#(1,2,3,4,5) 1
Peso_propio#CB#(X=-1,Y=-1,Z=-1) X=-1
No_Materiales#CB#(1,2,3,4,5,6,7,8,9,10) 1
Tipo_Problema#CB#(Plane_strain,Plane_stress) Plane_strain
Plano_LP#CB#(1,2,3) 1
Hipotesis_de_carga#CB#(1,2,3,4,5) 1
1
1
Nombre_Femview Asolid
```

### A.1.1.5 Archivo Asolid2d.bas (Archivo para la ordenación de los datos para su análisis)

```

*GenData (1)
SYSTEM
L=*GenData (2)
JOINTS
*loop nodes
*format "%3i X=%6.3f Y=%6.3f"
*NodesNum *NodesCoord
*end

*Set Cond Puntos-Restringidos
RESTRAINTS
*loop nodes *OnlyInCond
*format "%1i R=%1i,%1i,%1i,%1i,%1i,%1i"
*NodesNum
*cond(1,int)*cond(3,int)*cond(5,int)*cond(2,int)*cond(4,int)*cond(6,int)
*end

ASOLID
NM=*GenData (4) *if (strcmp (GenData (5), "Plane_strain")==0)
ETYP= 1 *\
*endif
*if (strcmp (GenData (5), "Plane_stress")==0)
ETYP= 2 *\
*endif
*GenData (3)
*loop materials
*matnum W=*MatProp (4)
E=*MatProp (1) U=*MatProp (2)
*end
*loop elems
*loop materials
*elemsnum JQ=*elemsConec (4) *elemsConec (1) *elemsConec (3) *elemsConec (2) *\
M=*elemsmat TH=*MatProp (3) LP=*GenData (6)
*end materials
*end elems

*Set Cond Puntos-Cargados *nodes
*if (CondNumEntities (int)>0)
LOADS
*loop nodes *OnlyInCond
*NodesNumL=*GenData (7) F=*cond (1), *cond (2), *cond (3), *\
cond (4), *cond (5), *cond (6)
*end
*endif

```

## A.2 Anexo 2

### A.2.1 Archivos de uso para la configuración Shell2d.gid de la Interface GiD-Sap90

#### A.2.1.1 Archivo Shell2d.end (Archivo de Condiciones).

```
NUMBER: 1 CONDITION: Puntos-Restringidos
CONDTYPE: over points
QUESTION: X-Restraints
VALUE: 0
QUESTION: X-Giro
VALUE: 0
QUESTION: Y-Restraints
VALUE: 0
QUESTION: Y-Giro
VALUE: 0
QUESTION: Z-Restraints
VALUE: 0
QUESTION: Z-Giro
VALUE: 0
END CONDITION
NUMBER: 2 CONDITION: Puntos-Cargas-y-Momentos
CONDTYPE: over points
QUESTION: X-Force
VALUE: 0
QUESTION: Y-Force
VALUE: 0
QUESTION: Z-Force
VALUE: 0
QUESTION: X-Momento
VALUE: 0
QUESTION: Y-Momento
VALUE: 0
QUESTION: Z-Momento
VALUE: 0
END CONDITION
```



**A.2.1.2 Archivo Shell2d.mat (Archivo de Materiales),**

```
NUMBER: 1 MATERIAL: Steel
QUESTION: E
VALUE: 2.1e07
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.0
QUESTION: PESO_VOLUMETRICO
VALUE: 7.800
END MATERIAL
NUMBER: 2 MATERIAL: Concrete
QUESTION: E
VALUE: 3.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 2.350
END MATERIAL
NUMBER: 3 MATERIAL: Madera
QUESTION: E
VALUE: 1.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 0.800
END MATERIAL
NUMBER: 4 MATERIAL: Fundición
END MATERIAL
.....
.....
NUMBER: 18 MATERIAL: Fabrical4
QUESTION: E
VALUE: 5.0e5
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.15
QUESTION: PESO_VOLUMETRICO
VALUE: 20.2
END MATERIAL
```

**A.2.1.3 Archivo Shell2d.sim (Archivo de Símbolos gráficos para condiciones).**

```
cond Point-Constraints
3
global
cond(int,5)
1
0
0
apoi03d.geo
global
cond(int,1) && cond(int,3)
1
0
0
apoi0.geo
global
cond(int,1) || cond(int,3)
cond(int,3)
cond(int,1)*(-1)
0
apoi0-rodillo.geo
cond Point-Load
1
global
1
cond(real,1)
cond(real,2)
cond(real,3)
fletxa.geo
cond Face-Load
3
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))<1e-5
1
0
0
Normal.geo
local
fabs(cond(real,1))<1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Tangencial.geo
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Normal-Tangen.geo
cond Surface-Load
1
local
1
-1
0
```

```

0
Normal-Superf.geo
cond Line-Constraints
1
local
1
1
0
0
apoiol.geo

```

#### A.2.1.4 Archivo Shell2d.prb (Archivo de Datos del problema)

```

7
TITULO: Title_name
Casos_De_Carga#CB# (1,2,3,4,5) 1
Peso_propio#CB# (X=-1,Y=-1,Z=-1) X=-1
No_Materiales#CB# (1,2,3,4,5,6,7,8,9,10) 1
Tipo_Problema#CB# (Shell,Membrane_behavior,Plate_bending) Shell
Eje_local_LP#CB# (0,1,2,3,-1,-2,-3) 1
Hipotesis_de_carga#CB# (1,2,3,4,5) 1
1
1
Nombre_Femview Asolid

```

### A.2.1.5 Archivo Shell2d.bas (Archivo para la ordenación de los datos para su análisis)

```

*GenData(1)
SYSTEM
L=*GenData(2)
JOINTS
*loop nodes
*format "%3i X=%6.3f Y=%6.3f Z=0.0"
*NodesNum *NodesCoord
*end

*Set Cond Puntos-Restringidos *nodes
RESTRAINTS
*loop nodes *OnlyInCond
*format "%1i R=%1i,%1i,%1i,%1i,%1i,%1i"
*NodesNum
*cond(1,int)*cond(3,int)*cond(5,int)*cond(2,int)*cond(4,int)*cond(6,int)
*end

SHELL
NM=*GenData(4) *GenData(3)
*loop materials
*matnum E=*MatProp(1) U=*MatProp(2) W=*MatProp(4)
*end
*loop elems
*loop materials
*elemsnum JQ=*elemsConec(4)*elemsConec(1)*elemsConec(3)*elemsConec(2)
*if(strcmp(GenData(5),"Shell")==0)
ETYPE= 0 *\
*endif
*if(strcmp(GenData(5),"Membrane_behavior")==0)
ETYPE= 1 *\
*endif
*if(strcmp(GenData(5),"Plate_bending")==0)
ETYPE= 2 *\
*endif
M=*elemsmat TH=*MatProp(3) LP=*GenData(6)
*end materials
*end elems

*Set Cond Puntos-Cargados *nodes
*if(CondNumEntities(int)>0)
LOADS
*loop nodes *OnlyInCond
*NodesNumL=*GenData(7) F=*cond(1),*cond(2),*cond(3)*\
*cond(4),*cond(5),*cond(6)
*end
*endif

```

## A.3 Anexo 3

### A.3.1 Archivos de uso para la configuración 3dshell de la Interface GiD-Sap90

#### A.3.1.1 Archivo 3dshell.cnd (Archivo de Condiciones).

```
NUMBER: 1 CONDITION: Puntos-Restringidos
CONDTYPE: over points
QUESTION: X-Restraints
VALUE: 0
QUESTION: X-Giro
VALUE: 0
QUESTION: Y-Restraints
VALUE: 0
QUESTION: Y-Giro
VALUE: 0
QUESTION: Z-Restraints
VALUE: 0
QUESTION: Z-Giro
VALUE: 0
END CONDITION
NUMBER: 2 CONDITION: Puntos-Cargados
CONDTYPE: over points
QUESTION: X-Force
VALUE: 0.0
QUESTION: Y-Force
VALUE: 0.0
QUESTION: Z-Force
VALUE: 0.0
QUESTION: X-Momen
VALUE: 0.0
QUESTION: Y-Momen
VALUE: 0.0
QUESTION: Z-Momen
VALUE: 0.0
END CONDITION
```

**A.3.1.2 Archivo 3dshell.mat (Archivo de Materiales).**

```
NUMBER: 1 MATERIAL: Steel
QUESTION: E
VALUE: 2.1e07
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.0
QUESTION: PESO_VOLUMETRICO
VALUE: 7.800
QUESTION: Eje_local_LP
Value:0
END MATERIAL
NUMBER: 2 MATERIAL: Concrete
QUESTION: E
VALUE: 3.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 2.350
QUESTION: Eje_local_LP
Value:0
END MATERIAL
NUMBER: 3 MATERIAL: Madera
QUESTION: E
VALUE: 1.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 0.800
QUESTION: Eje_local_LP
Value:0
END MATERIAL
NUMBER: 4 MATERIAL: Fundición
END MATERIAL
.....
.....
NUMBER: 18 MATERIAL: Fabrical4
QUESTION: E
VALUE: 5.0e5
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.15
QUESTION: PESO_VOLUMETRICO
VALUE: 20.2
QUESTION: Eje_local_LP
Value:0
END MATERIAL
```



### A.3.1.3 Archivo 3dsbell.sim (Archivo de Símbolos gráficos para condiciones).

```

cond Point-Constraints
3
global
cond(int,5)
1
0
0
apolo3d.geo
global
cond(int,1) && cond(int,3)
1
0
0
apolo.geo
global
cond(int,1) | | cond(int,3)
cond(int,3)
cond(int,1)*(-1)
0
apolo-rodillo.geo
cond Point-Load
1
global
1
cond(real,1)
cond(real,2)
cond(real,3)
fletxa.geo
cond Face-Load
3
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))<1e-5
1
0
0
Normal.geo
local
fabs(cond(real,1))<1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Tangencial.geo
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Normal-Tangen.geo
cond Surface-Load
1
local
1
-1
0

```

```
0
Normal-Superf.geo
cond Line-Constraints
1
local
1
1
0
0
apoiol.geo
```

#### A.3.1.4 Archivo 3dshell.prb (Archivo de Datos del problema)

```
6
TITULO: Title_name
Casos_De_Carga#CB#(1,2,3,4,5) 1
Peso_propio#CB#(X=-1,Y=-1,Z=-1) X=-1
No_Materiales#CB#(1,2,3,4,5,6,7,8,9,10) 1
Tipo_Problema#CB#(Shell,Membrane_behavior,Plate_bending) Shell
Hipotesis_de_carga#CB#(1,2,3,4,5) 1
1
1
Nombre_Femview Asolid
```

### A.3.1.5 Archivo 3dshell.bas (Archivo para la ordenación de los datos para su análisis)

```

*GenData(1)
SYSTEM
L=*GenData(2)
JOINTS
*loop nodes
*format "%3i X=%6.3f Y=%6.3f Z=%6.3f"
*NodesNum *NodesCoord
*end

*Set Cond Puntos-Restringidos *nodes
RESTRAINTS
*loop nodes *OnlyInCond
*format "%11 R=%11,%11,%11,%11,%11,%11"
*NodesNum
*cond(1,int)*cond(3,int)*cond(5,int)*cond(2,int)*cond(4,int)*cond(6,int)
*end

SHELL
NM=*GenData(4) *GenData(3)
*loop materials
*matnum E=*MatProp(1) U=*MatProp(2) W=*MatProp(4)
*end
*loop elems
*elemsnum JQ=*elemsConec(4)*elemsConec(1)*elemsConec(3)*elemsConec(2)
*if(strcmp(GenData(5),"Shell")==0)
ETYPE= 0 *\
*endif
*if(strcmp(GenData(5),"Membrane_behavior")==0)
ETYPE= 1 *\
*endif
*if(strcmp(GenData(5),"Plate_bending")==0)
ETYPE= 2 *\
*endif
M=*elemsmat *\
*loop materials
*if(elemsmat==matnum())
TH=*MatProp(3) LP=*MatProp(5)
*endif
*end
*end elems

*Set Cond Puntos-Cargados *nodes
*if(CondNumEntities(int)>0)
LOADS
*loop nodes *OnlyInCond
*NodesNum L=*GenData(7)
F=*cond(1),*cond(2),*cond(3),*cond(4),*cond(5),*cond(6)
*end
*endif

```

## A.4 Anexo 4

### A.4.1 Archivos de uso para la configuración 3dsolid de la Interface GiD-Sap90

#### A.4.1.1 Archivo 3dsolid.end (Archivo de Condiciones).

```
NUMBER: 1 CONDITION: Puntos-Restringidos
CONDTYPE: over points
QUESTION: X-Restraints
VALUE: 0
QUESTION: X-Giro
VALUE: 0
QUESTION: Y-Restraints
VALUE: 0
QUESTION: Y-Giro
VALUE: 0
QUESTION: Z-Restraints
VALUE: 0
QUESTION: Z-Giro
VALUE: 0
END CONDITION
NUMBER: 2 CONDITION: Puntos-Cargas-y-Momentos
CONDTYPE: over points
QUESTION: X-Force
VALUE: 0
QUESTION: Y-Force
VALUE: 0
QUESTION: Z-Force
VALUE: 0
QUESTION: X-Momento
VALUE: 0
QUESTION: Y-Momento
VALUE: 0
QUESTION: Z-Momento
VALUE: 0
END CONDITION
```

**A.4.1.2 Archivo 3dsolid.mat (Archivo de Materiales).**

```
NUMBER: 1 MATERIAL: Steel
QUESTION: E
VALUE: 2.1e07
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 7.800
QUESTION: Masa
VALUE: 0.0
QUESTION: Modulo_de_CortanteXY
VALUE: 0.0
QUESTION: Modulo_de_CortanteYZ
VALUE: 0.0
QUESTION: Modulo_de_CortanteZX
VALUE: 0.0
END MATERIAL
NUMBER: 2 MATERIAL: Concrete
QUESTION: E
VALUE: 3.0e6
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 1.0
QUESTION: PESO_VOLUMETRICO
VALUE: 2.350
QUESTION: Masa
VALUE: 0.0
QUESTION: Modulo_de_CortanteXY
VALUE: 0.0
QUESTION: Modulo_de_CortanteYZ
VALUE: 0.0
QUESTION: Modulo_de_CortanteZX
VALUE: 0.0
END MATERIAL
.....
.....
NUMBER: 18 MATERIAL: Fabrical4
QUESTION: E
VALUE: 5.0e5
QUESTION: NU
VALUE: 0.2
QUESTION: ESPESOR
VALUE: 0.15
QUESTION: PESO_VOLUMETRICO
VALUE: 20.2
QUESTION: Masa
VALUE: 0.0
QUESTION: Modulo_de_CortanteXY
VALUE: 0.0
QUESTION: Modulo_de_CortanteYZ
VALUE: 0.0
QUESTION: Modulo_de_CortanteZX
VALUE: 0.0
END MATERIAL
```

### A.4.1.3 Archivo 3dsolid.sim (Archivo de Símbolos gráficos para condiciones).

```
Archivo 3dsolid.sim

cond Point-Constraints
3
global
cond(int,5)
1
0
0
apoio3d.geo
global
cond(int,1) && cond(int,3)
1
0
0
apoio.geo
global
cond(int,1) || cond(int,3)
cond(int,3)
cond(int,1)*(-1)
0
apoio-rodillo.geo
cond Point-Load
1
global
1
cond(real,1)
cond(real,2)
cond(real,3)
fletxa.geo
cond Face-Load
3
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))<1e-5
1
0
0
Normal.geo
local
fabs(cond(real,1))<1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Tangencial.geo
local
fabs(cond(real,1))>1e-5 && fabs(cond(real,2))>1e-5
1
0
0
Normal-Tangen.geo
cond Surface-Load
1
local
```



```
1
-1
0
0
Normal-Superf.geo
cond Line-Constraints
1
local
1
1
0
0
apoiol.geo
```

#### A.4.1.4 Archivo 3dsolid.prb (Archivo de Datos del problema)

```
5
TITULO: Title_name
Casos_De_Carga#CB#(1,2,3,4,5) 1
Peso_propio#CB#(X=-1,Y=-1,Z=-1) X=-1
No_Materiales#CB#(1,2,3,4,5,6,7,8,9,10) 1
Hipotesis_de_carga#CB#(1,2,3,4,5) 1
1
1
Nombre_Femview Asolid
```

**A.4.1.5 Archivo 3dsolid.bas (Archivo para la ordenación de los datos para su análisis)**

```
*GenData(1)
SYSTEM
L=*GenData(2)
JOINTS
*loop nodes
*format "%3i X=%6.3f Y=%6.3f Z=%6.3f"
*NodesNum *NodesCoord
*end

*Set Cond Puntos-Restringidos *nodes
RESTRAINTS
*loop nodes *OnlyInCond
*format "%1i R=%1i,%1i,%1i,%1i,%1i,%1i"
*NodesNum
*cond(1,int)*cond(3,int)*cond(5,int)*cond(2,int)*cond(4,int)*cond(6,int)
*end

SOLID
NM=*GenData(4) *GenData(3)
*loop materials
*matnum W=*MatProp(4) M=*Matprop(5)
E=*MatProp(1) U=*MatProp(2) G=*MatProp(6), *MatProp(7), *MatProp(8)
*end
*loop elems
*elemsnum
JQ=*elemsConec(6)*elemsConec(7)*elemsConec(5)*elemsConec(8)*elemsConec(2)*e
lemsConec(3)*elemsConec(1)*elemsConec(4) M=*elemsmat
*end

*Set Cond Puntos-Cargas-y-Momentos *nodes
*if(CondNumEntities(int)>0)
LOADS
*loop nodes *OnlyInCond
*NodesNum L=*GenData(5)
F=*cond(1),*cond(2),*cond(3),*cond(4),*cond(5),*cond(6)
*end
```



**BIBLIOGRAFÍA:**

**CALDERBANK, V. J.**

\*Programación en Fortran.  
( Incluye fortran 77).  
Colección CIENCIA INFORMATICA,  
Editorial Gustavo Gili, S. A.

**GARCÍA DE ARANGOÁ, ANTONIO.**

\*Elasticidad  
( Teoría y Experimental), Fundamentos aplicaciones.  
Editorial DOSSAT, S. A.  
Madrid 1945.

**HUGHES, THOMAS J. R.**

\*The finite element method linear static and Dynamic  
Finite element analysis Thomas J: R: Hughes.  
Publicacion : Englewood Cliffs, N. J: Prentice-Hall International.

**INTEMAC No 17, Cuadernos**

\*TEMA: Contribución al estudio de las inclusiones de sillería en el  
comportamiento de los muros de mampostería en la rehabilitación de edificios históricos .José Ma.  
Izquierdo Bernardo de Quiros (Ingeniero de Caminos)

**LOVE, A. E. H.**

\*A Treatise on the Mathematical theory of Elasticity, Inc. New York, 1944

**MELI PIRALLA, ROBERTO.**

\*Diseño Estructural  
Editorial LIMUSA, México 1987

**OÑATE, E., HANGANU, A. BARBAT, A., OLLER, S. , VITALJANI, R. Y SAETTA, A.**

\*Structural analysis and durability assesment of historical constructions  
using a finete element damage model, structural analysis of historical  
construccion, Editor: P. Roca, CIMNE, BARCELONA.

**MANUAL DE GID**

\*Manual de Referencias  
<http://www.cimne.upc.es>

**SAP 90 ETABS SAFE.**

\*Computer Software for  
Structural & Earthquake Engineering.  
Developed and written in U. S. A.

**SEMINARIO INTERNACIONAL SOBRE ANÁLISIS ESTRUCTURAL DE  
CONSTRUCCIONES HISTÓRICAS.**

\*Posibilidades de las técnicas numéricas y experimentales  
E.T.S.E.C.C.P., U.P.C.  
Barcelona, 8, 9, 10 de Noviembre 1995.

**TIMOSHENKO, S. P.; JAMES M. GERE.**

\*Mecánica de Materiales.  
México, UTEHA, 1974

**TIMOSHENKO, S. P. Y GOODIER, J.N.**

\*Theory of Elasticity, 3a edición  
Mc Graw-Hill book Co., Inc., New York, 1970

**TODHUNTER, I.**

\* A History of the theory of elasticity and of the strength of materials:  
From Galilei to lord Kelvin / Isaac Todhunter ; edited and completed for  
the syndics of the Cambridge University Press by Karl Pearson.  
1 : Galilei to Saint-Venant: 1639-1850.  
New York: Dover Publications, (1960)

**TODHUNTER, I.**

\* A History of the theory of elasticity and of the strength of materials:  
From Galilei to lord Kelvin / Isaac Todhunter ; edited and completed for  
the syndics of the Cambridge University Press by Karl Pearson.  
2 : Saint -Venant to Lord Kevin  
New York: Dover Publications, (1960)

**TORROJA, EDUARDO.**

\*Elasticidad.  
Con aplicación a la técnica de la construcción  
Numerario de la Real Academia de Ciencias  
Editorial Dossat S. A.  
Segunda Edición.

**LOPEZ-REY LAURENS, JAVIER**

\*Apuntes de cálculo matricial de barras.  
Catedrático de Departamento de Estructuras en la Arquitectura.  
E.T.S.A.B. Universidad Politécnica de Catalunya.

## Recomendaciones y precauciones de la Interface GiD-Sap90

### Recomendaciones:

\* Toda vez que se realiza todo el preproceso, se recomienda que el usuario copie un archivo de cálculo directamente de la ventana *WriteCalcFile* en el directorio *Sap90* y ejecutar la subrutina *Sap90.exe*. Esto le permitirá al usuario saber si cometió algún error durante el proceso. Si no es así, se esta en condiciones de realizar el análisis.

### Precauciones:

\* Es importante que todas las normales de las superficies y de la malla se encuentren en la misma dirección. De lo contrario, al realizarse el análisis mediante *Sap90* los ejes locales de la geometría no estarán correctamente orientados; y podrían generarse errores de representación gráfica importantes.

\* Esta interface se confeccionó solo para análisis elástico lineal y no se consideró el problema dinámico ni térmico. Pero, el usuario puede realizar cambio de manera fácil y sencilla. Solo es necesario anexar tantos datos, como lo requiera el análisis en la configuración original de la interface.





## **AGRADECIMIENTOS**

A mi Esposa, por su comprensión y apoyo incondicional.

A mi Padre y Hermanos.

Al Profesorado del Departamento de Estructuras en la Arquitectura de la E.T.S.A.B de la U.P.C.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT).

