

Desarrollo y aplicación de un modelo de elementos finitos para cálculo de vigas con materiales compuestos laminados basado en la teoría de Timoshenko

A. Llanos
E. Oñate
F. Zárata

Desarrollo y aplicación de un modelo de elementos finitos para cálculo de vigas con materiales compuestos laminados basado en la teoría de Timoshenko

A. Llanos
E. Oñate
F. Zárate

Publicación CIMNE N°-398, June 2013

RESUMEN

Actualmente existen programas de MAT-fem para calcular vigas de material homogéneo, según la teoría de Euler-Bernoulli o de Timoshenko. El objetivo de la presente tesina es conseguir un nuevo programa capaz de calcular vigas de materiales compuestos con la teoría de Timoshenko.

En el capítulo 1, se introducen los materiales compuestos describiendo las ventajas y aplicaciones que tienen hoy en día en el ámbito de la Ingeniería Civil. En el capítulo 2 y 3 se expone la teoría de las vigas planas laminadas y, en particular, de las vigas compuestas de Timoshenko. El capítulo 4 explica cómo se ha llevado a cabo la implementación numérica y da paso al capítulo 5, en el que se exponen varios ejemplos. Por último, en el capítulo 6, se exponen las conclusiones que se extraen del trabajo realizado y de los resultados obtenidos.

Finalmente se presentan los anejos. Primero, en el anejo 1, se muestra un ejemplo real de refuerzo de viga mediante materiales compuestos. En el segundo anejo se presentan más ejemplos de los expuestos en el quinto capítulo y, por último, se termina la tesina con el anejo 3, que presenta los distintos archivos de programación utilizados para el cálculo de vigas compuestas de Timoshenko.

ABSTRACT

Nowadays, there are programs offered by MAT-fem that calculate Euler-Bernoulli beams and Timoshenko beams for homogeneous material. The aim of this research is to achieve a new programme capable of calculating composite beams with Timoshenko theory.

In chapter 1, composite materials are named and their advantages and disadvantages in the Civil Engineering field are described. In chapters 2 and 3, the composite laminated plane beams theory is explained and, especially, the composite laminated Timoshenko beams. In chapter 4 the numerical implementation is illustrated and it introduces the chapter 5, where several examples are mentioned. Finally, in chapter 6 the conclusions deduced in this research are put forward.

Lastly there are the appendixes. In appendix 1, we include a real example of a reinforced beam with composite material. In the second one, we present more cases adding what is said in chapter five and, finally, the numerical files which calculate the composite Timoshenko beam programme are included in appendix 3.

CONTENIDO

RESUMEN.....	i
ABSTRACT.....	iii
CONTENDIO.....	v
LISTA DE VARIABLES.....	vii
ÍNDICE DE FIGURAS, FOTOS Y TABLAS.....	xi
1 INTRODUCCIÓN.....	1
2. VIGAS PLANAS LAMINADAS.....	2
2.1 INTRODUCCIÓN.....	2
2.2 PROPIEDADES DE LOS MATERIALES COMPUESTOS.....	2
2.3 TEORÍAS DE VIGAS COMPUESTAS LAMINADAS.....	4
2.3.1 Teoría de Euler-Bernoulli y teoría de Timoshenko.....	4
2.3.2 Teorías de un orden más alto.....	4
3. VIGAS COMPUESTAS LAMINADAS DE TIMOSHENKO.....	5
3.1 CINEMÁTICA DE LAS VIGAS PLANAS LAMINADAS.....	5
3.2 TENSIONES Y ESFUERZOS.....	7
3.3 MATRIZ CONSTITUTIVA GENERALIZADA.....	7
3.4 ACOPLAMIENTO MEMBRANA-FLEXIÓN.....	9
3.5 TENSIONES INICIALES Y DEFORMACIONES TÉRMICAS.....	10
3.6 DISTRIBUCIÓN DE TENSIONES Y DEFORMACIONES.....	11
3.7 COEFICIENTE DE FORMA O DISTORSIÓN K_z	12
3.8 PRINCIPIO DE LOS TRABAJOS VIRTUALES.....	14
3.9 ELEMENTO DE VIGA DE DOS NODOS DE TIMOSHENKO.....	15
3.10 EFECTO BLOQUEO EN VIGAS LAMINADAS COMPUESTAS.....	19
3.11 VECTOR DE FUERZAS NODALES	21
4. IMPLEMENTACIÓN NUMÉRICA.....	22
4.1 INTRODUCCIÓN A MATLAB Y GID.....	22
4.2 MAT-fem.....	23

4.3 PROGRAMACIÓN NUMÉRICA DE VIGAS COMPUESTAS DE TIMOSHENKO...	24
4.3.1 Programa de cálculo en Matlab.....	24
4.3.2 Interfaz implementada en GiD.....	57
4.3.3 Archivos de resultados.....	71
5. EJEMPLOS.....	72
5.1 VOLADIZO CON CARGA PUNTUAL EN UN EXTREMO.....	72
5.2 VIGA BIEMPOTRADA CON CARGA REPARTIDA.....	83
5.3 VIGA CON SECCIÓN EN DOBLE T.....	90
6. CONCLUSIONES.....	98
7. BIBLIOGRAFÍA.....	99
ANEJO 1. MUELLE DE LA ENERGÍA.....	100
1. INTRODUCCIÓN.....	100
2. ANTECEDENTES.....	100
3. REFUERZO CON MATERIALES COMPUESTOS.....	103
4. ANÁLISIS ESTRUCTURAL.....	104
ANEJO 2. EJEMPLOS COMPARANDO TCB Y 3D.....	120
ANEJO 3. RUTINAS DE PROGRAMACIÓN.....	139

LISTADO DE VARIABLES

VARIABLE	DIMENSIONES	DESCRIPCIÓN
D_mata	1 x 1	Componente axial matriz constitutiva
D_matab	1 x 1	Componente acoplamiento axial-flector matriz constitutiva
D_matb	1 x 1	Componente a flexión matriz constitutiva
D_mats	1 x 1	Componente cortante matriz constitutiva
Dsp_lyr	nfnod x 3 x (layers+1)	Matriz de desplazamiento de cada capa
ElemFor	1 x nnode	Vector de fuerza elemental
FreeNodes	1 x (nodof-fix)	Grados de libertad libres
K_axial	2 x 2	Contribución a axial de la matriz de rigidez elemental
K_bend	4 x 4	Contribución a flexión de la matriz de rigidez elemental
K_elem	6 x 6	Matriz elemental
K_flex	4 x 4	Contribución a flexión y cortante de la matriz de rigidez
K_shear	4 x 4	Contribución a cortante de la matriz de rigidez
ResStr_lyr	nelem x 3 x layers	Matriz de tensiones generalizadas según capas
S1	1 x 1	Variable auxiliar para calcular kz
S2	1 x 1	Variable auxiliar para calcular kz
S3	1 x 1	Variable auxiliar para calcular kz
StifMat	nndof x nndof	Matriz de rigidez global
Str_lyr	nfnod x 3 x 2·layers	Matriz de tensiones según capas
area	1	Área de la sección
area_layers	1 x layers	Área de cada capa
const	1	Constante para calcular las matrices de rigidez
coor_x	1 x 2	Coordenada horizontal de los nodos de cada elemento
coord_ZG	(layers+1) x 1	Coordenada de cada capa respecto el centro de gravedad de la sección
coord_z	(layers+1) x 1	Coordenada de cada capa respecto fibra inferior de la sección

coord_zg	1	Centro de gravedad de la sección
coord_zp	(layers+1) x 1	Coordenada de cada capa respecto al eje neutro de la sección
coordinates	npnod x 1	Matriz de coordenadas nodales en sistema global
d	1 x 1	Distancia entre eje neutro y centro de gravedad
denss	layers x 1	Densidad de cada capa
dofpe	nnode x dofpn	Número de grados de libertad por elemento
dofpn	3	Número de grados de libertad por nodo
elements	nelem x 2	Matriz de conectividades
eqnum	1 x dofpe	Número de ecuación para cada grado de libertad
fa	1 x 1	Componente carga axil en el vector de fuerzas elemental
file_name	1	Nombre del fichero de datos
fix	1 x fix	Vector de grados de libertad prescritos
fixnodes	fix x 3	Especificación del nodo y su grado de libertad prescrito
force	nndof x 1	Vector global de fuerzas
fz	1 x 1	Componente carga vertical en el vector de fuerzas elemental
ielem	1 x 1	Indica el elemento
ieqn	1 x 1	Indica el grado de libertad para aplicar condiciones de contorno
ii	1 x 1	Variable auxiliar para construir las matrices globales
itim	1 x 1	Apaga el cronómetro
kz	1 x 1	Coefficiente de forma
layers	1 x 1	Número de capas
len	1 x 1	Longitud de cada elemento
lnods	1 x 2	Número global del nodo de los nodos elementales
nelem	1 x 1	Número de elementos
nndof	1 x 1	Número total de grados de libertad
nnode	1 x 1	Número de nodos por elemento
npnod	1 x 1	Número total de nodos
pointload	núm.cargas:3 x 3	Cargas puntuales
poiss	layers x 1	Coefficiente de Poisson por

		capa
pp	layers x 1	Peso propio de cada capa
shear	layers x 1	Módulo de rigidez por capa
thickness	layers x 1	Espesor de cada capa
ttim	1 x 1	Contador de tiempo
u	nndof x 1	Vector nodal de desplazamientos
uniload	nelem x 2	Cargas distribuidas
width	layers x 1	Ancho de cada capa
young	layers x 1	Módulo de Young por capa
zg_layers	layers x 1	Centro de gravedad de cada capa

ÍNDICE DE FIGURAS, FOTOS Y TABLAS

ÍNDICE DE FIGURAS

Fig. 1 Parámetros que aparecen en las componentes de la matriz constitutiva de una sección rectangular de material compuesto	8
Fig. 2 Posición del eje neutro	10
Fig. 3 Distribución de tensiones normal y tangencial en el canto de una sección compuesta de 4 capas	11
Fig. 4 Esquema de funcionamiento de MAT-fem	24
Fig. 5 Variables descritas en el fichero de datos asociadas al material	26
Fig. 6 Variables descritas en el fichero de datos asociadas a la topología del problema	27
Fig.7 Variables descritas en el fichero de datos asociadas a las condiciones de contorno del problema	28
Fig.8, 9, 11, 12, 13, 14, 15, 16, 17, 18 y 19 Composite_Beam_Timoshenko_Reduced	29-36
Fig. 10 Timing	30
Fig. 20, 21, 22, 23, 24, 25, 26 y 27 Composite_Beam_Timoshenko	37-41
Fig. 28, 29, 30, 31, 32, 33, 34, 35, 36 y 37 Composite_Layers_Beam_Timoshenko	43-51
Fig. 38, 39, 40, 41, 42, 43, 44 y 45 ToGid_Composite_Beam_Timoshenko	51-56
Fig. 46 Restricción de desplazamientos	59
Fig. 47 Condiciones de cargas puntuales	59
Fig. 48 Condiciones de carga distribuida	60
Fig. 49 Información interna del problema referente a considerar el peso propio de la estructura	61
Fig. 50 Asignación de las propiedades del material según la capa	62
Fig. 51 Magnitudes y las unidades que pueden tomar	63
Fig. 52 Unidades del Sistema Internacional	64
Fig. 53 Símbolos en cuanto a condiciones de apoyo	65
Fig. 54 Representación de los símbolos de las cargas puntuales	67
Fig. 55 Símbolo para carga uniformemente distribuida	68
Fig. 56 Propiedades del material en MATLAB, GiD y en el fichero de datos	69
Fig. 57 Archivo de extensión tcl	70
Fig. 58 Esquema de procedimiento de MAT-fem	72
Fig. 59 Preproceso	73

Fig. 60 Deformada en el postproceso	74
Fig. 61 Convergencia resultados	75
Fig. 62 Desplazamiento horizontal.....	76
Fig. 63 Gráfica h(m)-u(m)	76
Fig. 64 Cortante.....	76
Fig. 65 Momento.....	77
Fig. 66 Tensión normal	77
Fig. 67 Tensión tangencial xz.....	77
Fig. 68 Materiales.....	78
Fig. 69 Condiciones de contorno.....	78
Fig. 70 Deformada de la viga.....	78
Fig. 71 Convergencia de resultados	79
Fig. 72 Desplazamiento horizontal.....	80
Fig. 73 Tensión normal	80
Fig. 74 Tensión tangencial (Tau_xz)	81
Fig. 75 Tiempo invertido según la teoría de Timoshenko	82
Fig.76 Gráfica tiempo invertido en el cálculo de viga en 3D	82
Fig. 77 Convergencia resultados	84
Fig. 78 Convergencia de resultado.....	85
Fig. 79 Deformada en el postproceso	86
Fig. 80 Deformada.....	86
Fig. 81 Desplazamiento horizontal TCB.....	87
Fig. 82 Desplazamientos horizontales 3-D	87
Fig. 83 Tensión normal.....	87
Fig. 84 Tensión normal.....	88
Fig. 85 Tensión tangencial xz.....	88
Fig. 86 Tensión tangencial xz.....	88
Fig. 87 Tensión tangencial según TCB y 3D	89
Fig. 88 Tiempo invertido en el cálculo según TCB.....	90
Fig. 89 Tiempo invertido en el cálculo de viga en 3D.....	90
Fig. 90 Convergencia de resultados de TCB	91

Fig. 91 Convergencia de resultados en 3D	92
Fig. 92 Desplazamiento vertical TCB	93
Fig. 93 Desplazamiento vertical 3D	93
Fig. 94 Desplazamiento horizontal TCB.....	94
Fig. 95 Desplazamiento horizontal 3D	94
Fig. 96 Tensión normal TCB.....	94
Fig. 97 Tensión normal 3D	95
Fig. 98 Tensión tangencial xz TCB	95
Fig. 99 Tensión tangencial xz 3D	96
Fig. 100 Cortante TCB.....	96
Fig. 101 Tiempo computacional según TCB y 3D	97
Fig. 102 Puente de la energía.....	100
Fig. 103 Características de la sección.....	101
Fig. 104 Sección tipo del puente de la energía.....	102
Fig. 105 Envolvente de momentos en la viga más solicitada	102
Fig. 106 Sección estructura muelle de la energía.....	104
Fig. 107 Viga en TCB.....	105
Fig. 108 Zoom de la viga donde se distinguen las capas	106
Fig. 109 Condiciones de contorno.....	107
Fig. 110 Tensiones en la viga.....	107
Fig. 111 Tensiones en las fibras extremas de la pieza teniendo en cuenta el pretensado.....	110
Fig. 112 Esquema de fuerzas para obtener la resistencia última de la sección	111
Fig. 113 Esquema momento resistente con la lámina adherida	116
Fig. 114 Momento resistente según los cables afectados y el refuerzo.....	118
Fig. 115 Condiciones de contorno de la viga.....	120
Fig. 116 Deformada de la viga.....	121
Fig. 117 Gráfica de convergencia	122
Fig. 118 Desplazamiento horizontal (mm)	122
Fig. 119 Esfuerzo cortante (N)	123
Fig. 120 Momento flector (KNm)	123
Fig. 121 Tensión normal (N/mm ²)	123

Fig. 122 Tensión tangencial xz (N/mm ²)	123
Fig. 123 Materiales de la viga.....	124
Fig. 124 Condiciones de contorno.....	124
Fig.125 Deformada (mm)	124
Fig 126 Convergencia de resultados	125
Fig. 127 Desplazamientos horizontales (mm)	126
Fig. 128 Tensión normal (N/mm ²)	126
Fig. 129 Tensión tangencial xz (N/mm ²)	126
Fig. 130 Gráfica tiempo invertido en el cálculo de TCB.....	127
Fig. 131 Gráfica tiempo invertido en el cálculo de viga en 3D	128
Fig. 132 Preproceso.....	129
Fig. 133 Deformada en el postproceso	130
Fig. 134 Desplazamiento vertical a lo largo del eje de la viga (mm)	130
Fig. 135 Desplazamiento horizontal en la sección x=L/4.....	131
Fig. 136 Tensión normal en la sección x=L/4.....	131
Fig. 137 Tensión tangencial en la sección x=L/4.....	132
Fig. 138 Desplazamiento horizontal (mm)	132
Fig. 139 Esfuerzo cortante (N)	133
Fig. 140 Momento (Nmm)	133
Fig. 141 Tensión normal (N/mm ²)	133
Fig. 142 Tensión tangencial xz (N/mm ²)	134
Fig. 143 Convergencia de resultados	134
Fig. 144 Viga con las 10 capas bien diferenciadas.....	135
Fig. 145 Desplazamiento vertical (mm)	136
Fig. 146 Desplazamiento horizontal (mm)	136
Fig. 147 Tensión normal (MPa)	137
Fig. 148 Tensión tangencial (MPa)	137

ÍNDICE DE FOTOS

Foto 1 Lámina de 50x1,2mm.....	119
Foto 2 Lámina de 80x1,2mm.....	119
Foto 3 Lámina de 100x1,2mm.....	119

ÍNDICE DE TABLAS

Tabla 1 Módulo de Young (E), densidad (ρ) y tensión última a tracción (σ_u^t) de algunas fibras y metales.....	3
Tabla 2 Módulo de Young (E), densidad (ρ), tensión última a tracción y compresión (σ_u^t , σ_u^c) de algunas fibras y metales	3
Tabla 3 Características de los materiales	73
Tabla 4 Tabla de resultados	74
Tabla 5 Solución obtenida con la malla de 2001 nodos	75
Tabla 6 Tabla de resultados	79
Tabla 7 Solución obtenida con la malla de 13160 nodos	80
Tabla 8 Comparación de resultados.....	81
Tabla 9 Datos de los materiales y dimensiones de la sección de la viga	83
Tabla 10 Valores de convergencia	84
Tabla 11 Valores de convergencia	85
Tabla 12 Valores obtenidas según TCB y 3D	86
Tabla 13 Características de la sección.....	91
Tabla 14 Comparación de resultados.....	93
Tabla 15 Tensión a tracción y compresión admisible según el tipo de hormigón.....	102
Tabla 16 Dimensiones de las láminas	103
Tabla 17 Propiedades de la lámina de fibra de carbono	103
Tabla 18 Características de la sección dividida en capas	105
Tabla 19 Refuerzo según el número de alambres afectado.....	113
Tabla 20 Características del problema en función del número de cables afectado.....	114
Tabla 21 Espesor de la capa de acero según el número de cables afectado	114
Tabla 22 Tensiones en función del número de cables afectado y del refuerzo que se utiliza	115
Tabla 23 Posición de la fibra neutra y momento resistente de la sección	116
Tabla 24 Momento resistente de la sección reforzada en función del número de alambres afectado.....	117

Tabla 25 Datos de los materiales y dimensiones de la sección de la viga	120
Tabla 26 Valores de convergencia	121
Tabla 27 Valores obtenidos según TCB	122
Tabla 28 Valores de convergencia	125
Tabla 29 Solución obtenida con la malla de 10284 nodos	126
Tabla 30 Comparación de resultados.....	127
Tabla 31 Características y dimensiones de la sección de la viga	129
Tabla 32 Valores obtenidos mediante la malla de 300 nodos	132
Tabla 33 Resultados obtenidos mediante la malla de 11959 nodos.....	135
Tabla 10 Comparación de resultados.....	138

1. INTRODUCCIÓN

Los materiales compuestos se han convertido en una de las tecnologías más prometedoras en el diseño de estructuras en los últimos años. Su aplicación ha ido creciendo durante la última década debido a las características tan ventajosas que proporciona el hecho de poder diseñar un material con propiedades mecánicas superiores a los materiales tradicionales.

Estos materiales poseen una elevada relación resistencia-peso y rigidez-peso, lo cual los hace especialmente adecuados para estructuras en las que el peso constituye una variable esencial en el proceso de diseño. Además, los materiales compuestos son resistentes a la corrosión y son térmicamente estables, lo cual permite su aplicación en la construcción de estructuras sometidas a la acción de ambientes agresivos, como partes de plataformas offshore, armaduras pasivas, armaduras activas y refuerzos de estructuras, entre otros. Las ventajas que suponen la gran resistencia y poco peso de estos materiales los han colocado, en algunos casos, por delante de materiales como el hormigón y el acero.

En lo que se refiere al refuerzo de estructuras, cabe mencionar que en los últimos años se ha producido un aumento en el uso de los materiales compuestos para este tipo de actuaciones ya que su bajo peso facilita su transporte, manejo y colocación y acorta los plazos de tiempo en obra, aspectos altamente valorados hoy en día en ingeniería civil.

A pesar de las numerosas investigaciones que avalan el buen funcionamiento de estos materiales, una de las dificultades que se encuentra para su utilización con fines estructurales es la incertidumbre que generan las uniones y juntas, ya que son los puntos débiles de la estructura. Además, hoy en día es necesaria la aparición de técnicas numéricas eficientes para modelar con éxito el comportamiento estructural de los materiales compuestos.

Haciendo uso de los elementos finitos, se ha conseguido desarrollar la teoría que analiza el comportamiento de las estructuras de materiales compuestos. Existen diferentes teorías dependiendo del tipo de estructura (viga, lámina o placa) y del grado de aproximación que se requiera. Sin embargo, es necesario implementar métodos computacionales que sean capaces de reproducir fielmente estas teorías.

La presente tesina se centra en estudiar el comportamiento de las vigas laminadas formadas por materiales compuestos y, en particular, se presentará una panorámica del cálculo de elementos finitos de este tipo de vigas discretizadas en elementos de dos nodos de Timoshenko. Una vez expuesta dicha teoría, se implementará numéricamente mediante el programa MAT-fem, el cual es la interacción entre MATLAB, que permite la programación y cálculo numérico, y GID, sobre el cual se manipulan geometrías y se visualizan resultados. Además, se presentaran varios ejemplos de vigas calculadas según la teoría de Timoshenko, comparando los resultados con el cálculo tridimensional de manera que podamos verificar la fiabilidad de la teoría y su implementación numérica en MAT-fem. Finalmente se concluye mostrando un caso real de refuerzo de vigas mediante materiales compuestos, que aporta una visión actual de la funcionalidad de este tipo de estructuras.

2. VIGAS PLANAS LAMINADAS

2.1 INTRODUCCIÓN

Las vigas compuestas están formadas por capas de materiales compuestos. Estas capas se denominan láminas, en las que el refuerzo puede ser mediante fibras largas paralelas (lámina unidireccional) o mediante un tejido que se obtendría entrelazando las fibras (lámina bidireccional). Las láminas se apilan unas sobre otras y existe continuidad de la matriz entre ellas en la dirección perpendicular al laminado. Es importante apuntar que cada lámina puede tener sus fibras con una orientación distinta a la del resto. Es esta posibilidad la que permite apilar las láminas con la dirección de sus fibras orientadas de manera que se realice un diseño óptimo del material compuesto.

2.2 PROPIEDADES DE LOS MATERIALES COMPUESTOS

En la tabla 1 se compara el módulo de Young, la densidad y la tensión última a tracción de rotura de algunas fibras con distintos metales. Se puede observar como la resistencia es mayor en las fibras que en los metales y, en cambio, la densidad es menor. Evidentemente, los metales pueden utilizarse directamente como material estructural, mientras que no es así con las fibras. Sin embargo, la elevada relación resistencia/densidad de las fibras justifica la aplicación estructural de estos materiales.

		E (GPa)	ρ (Kg/m ³)	σ_u^t (MPa)
FIBRAS DE MATERIALES COMPUESTOS	<i>Vidrio E</i>	72	2550	3400
	<i>Vidrio S</i>	86	2500	4800
	<i>Carbono</i>	190	1410	1700
	<i>Boro</i>	400	2570	3400
	<i>Grafito</i>	250	1410	1700
METALES	<i>Aluminio</i>	73	2700	620
	<i>Titanio</i>	115	4700	1900
	<i>Acero</i>	210	7700	4100

Tabla 1 Módulo de Young (E), densidad (ρ) y tensión última a tracción (σ_u^t) de algunas fibras y metales

En cuanto a la matriz del compuesto, los materiales poliméricos como el epoxi son los más comúnmente utilizados. Las propiedades mecánicas de este material son las siguientes:

	E (GPa)	ρ (Kg/m ³)	σ_u^t (MPa)	σ_u^c (MPa)
Epoxi	3.5	1300	50	140

Tabla 2 Módulo de Young (E), densidad (ρ), tensión última a tracción y compresión (σ_u^t , σ_u^c) de algunas fibras y metales

Resulta necesario homogeneizar las propiedades de los materiales de un compuesto de manera que tomemos un único valor para las diferentes características mecánicas del material. Una manera de aproximar estas propiedades, teniendo en cuenta la aportación de las fibras y la matriz es realizar el siguiente cálculo:

$$E_c = V_f E_f + (1 - V_f) E_m \quad , \quad \rho_c = V_f \rho_f + (1 - V_f) \rho_m$$

$$\sigma_{cu}^t = V_f \sigma_{fu}^t + (1 - V_f) \sigma_{mu}^t \quad , \quad \sigma_{cu}^c = \sigma_{mu}^c \quad (1)$$

donde E_c , ρ_c y σ_{cu}^t son el módulo de Young, la densidad y la tensión última a tracción del material compuesto; y V_f es el volumen en porcentaje de fibras del compuesto. Se observa que el valor de la tensión última a compresión del compuesto σ_{cu}^c se toma igual a la tensión última a compresión de la matriz σ_{mu}^c .

Es importante tener en cuenta que en un material compuesto las fibras son las que más contribuyen a soportar los esfuerzos a los que se encuentra sometida la estructura, algo

que resulta evidente si comparamos las propiedades de cada material de las tablas anteriores. Sin embargo, la matriz es esencial para el buen comportamiento del compuesto ya que mantiene todas las fibras unidas y proporciona una superficie adecuada al material. Además, consigue distribuir las tensiones sobre cada fibra.

2.3 TEORÍAS DE VIGAS COMPUESTAS LAMINADAS

2.3.1 Teoría de Euler-Bernoulli y teoría de Timoshenko

Haciendo referencia al cálculo a flexión de vigas formadas por un material homogéneo, se recuerda que las dos teorías de cálculo son la teoría clásica de Euler-Bernoulli y la de Timoshenko. La primera, basada en la ortogonalidad de las secciones transversales rectas con la fibra media, prescinde del efecto de la deformación por cortante y, por tanto, su utilización es válida solo para vigas esbeltas (relación longitud/canto mayor a 10). Sin embargo, la segunda (teoría de Timoshenko) permite la no ortogonalidad de las secciones transversales con la fibra media tras la deformación. Esto introduce de forma natural el efecto de la deformación por cortante y la posibilidad de estudiar tanto vigas esbeltas como gruesas.

En las vigas compuestas laminadas, el efecto de la deformación por cortante es relevante debido a la heterogeneidad del material en dirección transversal al eje de la viga. Por ello, la teoría de Timoshenko es particularmente apropiada para este tipo de estructuras. Como veremos en el capítulo siguiente, el único inconveniente que encontramos en los elementos de viga de Timoshenko es que se produce un efecto bloqueo de la solución al analizar vigas de gran esbeltez, lo que deriva en resultados erróneos. Sin embargo, este efecto se puede suprimir utilizando técnicas especiales, como la integración reducida de la matriz de rigidez a cortante.

2.3.2 Teorías de un orden más alto

Debido a que las propiedades del material en las vigas compuestas no son uniformes a lo largo del eje seccional de la viga, aparece un desplazamiento axial que se tiene que tener en cuenta y la teoría de Timoshenko lo contempla. Este desplazamiento es la principal diferencia con las teorías de vigas homogéneas. No obstante, la teoría de Timoshenko supone una variación lineal del campo de desplazamientos sobre toda la sección de la viga, sin tener en cuenta diferencias según las capa. De ahí que, en vigas muy heterogéneas, esta teoría tienda a

subestimar la tensión axial de la fibra superior e inferior de la sección y , también, presente errores en el desplazamiento horizontal de las interfaces entre capas.

Sin embargo, es posible realizar aproximaciones más precisas utilizando las llamadas *teorías de un orden más alto*. Dichas teorías admiten que los desplazamientos varían linealmente en el interior de las capas, aunque con diferente pendiente en cada una.

Éstas, se utilizan cuando el espesor de la viga es demasiado grande o existe una heterogeneidad en el material tan alta que la teoría de Timoshenko no consigue resultados demasiado fiables. Existen dos teorías de un orden más alto, la llamada teoría *layer-wise* y la teoría del *zigzag*. A pesar de que estas teorías proporcionan resultados más precisos, la teoría de Timoshenko describe suficientemente bien el comportamiento de las vigas compuestas que nos podemos encontrar en la mayoría de casos.

3. VIGAS COMPUESTAS LAMINADAS DE TIMOSHENKO

En los capítulos anteriores hemos hecho referencia a los materiales compuestos indicando sus propiedades, señalando que existen diferentes tipos de estructuras en las que es apropiado el uso de estos materiales haciendo especial hincapié en las vigas laminadas y, por último, hemos enunciado diferentes teorías para el cálculo de este tipo de estructuras mediante elementos finitos. De aquí en adelante, nos centraremos en las vigas compuestas de Timoshenko.

3.1 CINEMÁTICA DE LAS VIGAS PLANAS LAMINADAS

Consideraremos una viga de eje x recto en la que el plano xz es un plano principal de inercia. La viga tiene una longitud L y la sección transversal está formada por un apilamiento de capas de material compuesto con propiedades que dependen solo de la coordenada vertical z . Admitiremos que se cumplen las hipótesis de la teoría de vigas planas de Timoshenko:

- i. El desplazamiento lateral v es nulo.
- ii. Todos los puntos de una sección tienen el mismo desplazamiento vertical.
- iii. Las secciones normales permanecen planas después de la deformación, pero no necesariamente ortogonales al eje deformado de la viga.

Teniendo en cuenta las tres hipótesis anteriores, se puede expresar el campo de desplazamientos de una sección de la siguiente manera:

$$\begin{aligned} u(x, z) &= u_0(x) - z\theta(x) \\ w(x, z) &= w_0(x) \end{aligned} \quad (2)$$

donde u_0 y w_0 son el desplazamiento horizontal y vertical, respectivamente, del eje de la viga y θ es el giro de la sección (definido positivo en sentido antihorario).

Derivando el campo de desplazamientos se deduce la deformación axial y la deformación por cortante como:

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} = \frac{\partial u_0}{\partial x} - z \frac{\partial \theta}{\partial x} \\ \gamma_{xz} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \frac{\partial w_0}{\partial x} - \theta \end{aligned} \quad (3)$$

Se comprueba, como apuntábamos anteriormente, que la deformación por cortante γ_{xz} no es nula en la teoría de Timoshenko. Este aspecto tiene una importancia capital en el análisis de vigas multilaminadas, ya que la deformación por cortante tiene una gran influencia en este tipo de estructuras.

La ecuación (3) puede ser reescrita en forma matricial según:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \gamma_{xz} \end{Bmatrix} = \begin{bmatrix} 1 & -z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta \end{bmatrix}^T = \mathbf{S} \hat{\boldsymbol{\varepsilon}} \quad (4a)$$

con

$$\mathbf{S} = \begin{bmatrix} 1 & -z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \hat{\boldsymbol{\varepsilon}} = \begin{bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta \end{bmatrix}^T \quad (4b)$$

donde $\boldsymbol{\varepsilon}$ es el vector de deformaciones, $\hat{\boldsymbol{\varepsilon}}$ el vector de deformaciones generalizadas que contiene el alargamiento $(\frac{\partial u}{\partial x})$, la curvatura $(\frac{\partial \theta}{\partial x})$ y la cizalladura $(\frac{\partial w_0}{\partial x} - \theta)$. La matriz \mathbf{S} es la matriz de transformación que relaciona las deformaciones con los desplazamientos y depende de la coordenada vertical z .

3.2 TENSIONES Y ESFUERZOS

La tensión normal σ_x y tangencial τ_{xz} en cada punto de la sección se relacionan con las deformaciones correspondientes según las expresiones:

$$\sigma_x = E \varepsilon_x = E \left(\frac{\partial w_0}{\partial x} - z \frac{\partial \theta}{\partial x} \right) \quad (5a)$$

$$\tau_{xz} = G \gamma_{xz} = G \left(\frac{\partial w_0}{\partial x} - \theta \right) \quad (5b)$$

siendo $E=E(x,z)$ y $G=G(x,z)$ el módulo de Young y el módulo de rigidez, respectivamente, de los materiales compuestos de la viga.

Es posible reescribir las ecuaciones (5a) y (5b) en forma de matriz como:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \tau_{xz} \end{Bmatrix} = \begin{bmatrix} E & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \gamma_{xz} \end{Bmatrix} = \mathbf{D} \boldsymbol{\varepsilon} = \mathbf{D} \hat{\boldsymbol{\varepsilon}} \quad (6)$$

donde \mathbf{D} es la matriz constitutiva que relaciona tensiones y deformaciones en cada punto de la viga.

La matriz de tensiones generalizados o esfuerzos se puede obtener como:

$$\hat{\boldsymbol{\sigma}} = \begin{Bmatrix} N \\ M \\ Q \end{Bmatrix} = \iint_A \begin{Bmatrix} \sigma_x \\ -z \sigma_x \\ \tau_{xz} \end{Bmatrix} dA = \iint_A \mathbf{S}^T \boldsymbol{\sigma} dA \quad (7)$$

donde $\hat{\boldsymbol{\sigma}}$ es el vector de esfuerzos que contiene el esfuerzo axial N , el momento flector M y el esfuerzo cortante Q y \mathbf{S} es la matriz de transformación de la ecuación (4b).

3.3 MATRIZ CONSTITUTIVA GENERALIZADA

Sustituyendo la ecuación (6) en (7) se deduce que:

$$\hat{\boldsymbol{\sigma}} = \left(\iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA \right) \hat{\boldsymbol{\varepsilon}} = \hat{\mathbf{D}} \hat{\boldsymbol{\varepsilon}} \quad (8)$$

donde $\hat{\boldsymbol{\varepsilon}}$ es el vector de deformaciones generalizadas y $\hat{\mathbf{D}}$ es la matriz constitutiva generalizada. Las componentes de esta matriz se definen a continuación:

$$\widehat{\mathbf{D}} = \iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA = \begin{bmatrix} \widehat{D}_a & \widehat{D}_{ab} & 0 \\ \widehat{D}_{ab} & \widehat{D}_b & 0 \\ 0 & 0 & \widehat{D}_s \end{bmatrix} \quad (9)$$

con

$$\begin{aligned} \widehat{D}_a &= \iint_A E(x,z) dA & , & \quad \widehat{D}_{ab} = - \iint_A E(x,z) z dA \\ \widehat{D}_b &= \iint_A E(x,z) z^2 dA & , & \quad \widehat{D}_s = k_z D_s \quad \text{con} \quad D_s = \iint_A G(x,z) dz \end{aligned} \quad (10)$$

donde \widehat{D}_a es la rigidez axial, \widehat{D}_b es la rigidez de flexión, \widehat{D}_{ab} es la rigidez debida al acoplamiento axial-flexión, \widehat{D}_s es la rigidez de cortante y k_z es el denominado coeficiente de corrección del esfuerzo cortante que calcularemos más adelante.

En el caso de las vigas laminadas de materiales compuestos, el cálculo de las componentes de la matriz constitutiva se obtiene teniendo en cuenta la rigidez de cada capa y se calcula como sigue:

$$\begin{aligned} \widehat{D}_a &= \sum_{k=1}^{n_l} b_k h_k E_k \\ \widehat{D}_{ab} &= - \sum_{k=1}^{n_l} b_k h_k \bar{z}_k E_k \\ \widehat{D}_b &= \frac{1}{3} \sum_{k=1}^{n_l} (z_{k+1}^3 - z_k^3) b_k E_k \\ D_s &= \sum_k b_k h_k G_k \end{aligned}$$

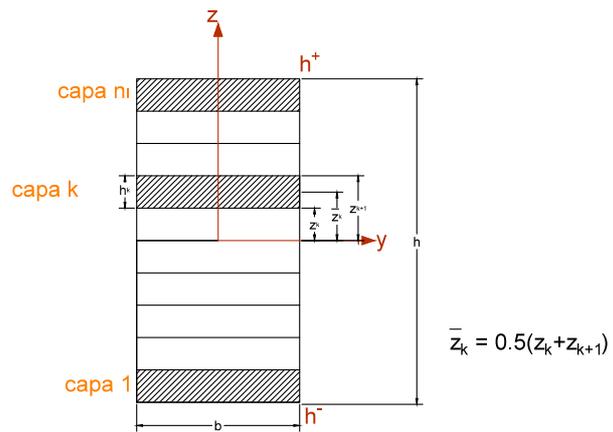


Fig. 1 Parámetros que aparecen en las componentes de la matriz constitutiva de una sección rectangular de material compuesto (11)

donde n_l es el número de capas, cada una de espesor h_k y ancho b_k , módulos de Young y rigidez E_k y G_k y \bar{z}_k es la coordenada vertical del punto medio de cada capa.

3.4 ACOPLAMIENTO MEMBRANA-FLEXIÓN

El acoplamiento membrana-flexión se manifiesta por el hecho de que un esfuerzo axial provoca curvaturas y un momento flector provoca alargamientos, de manera que aparece una componente en la matriz constitutiva (\tilde{D}_{ab}) que impide que ésta sea diagonal. Sin embargo, si el eje de la viga coincide con el eje neutro este acoplamiento desaparece y, por tanto $\tilde{D}_{ab} = 0$. La posición del eje neutro se puede obtener imponiendo esta condición.

Definimos la coordenada vertical relativa $z' = z - d$ donde d es la distancia vertical entre el eje de la viga (x) y el eje neutro. Si desplazamos el eje de la viga colocándolo en la posición del eje neutro, entonces:

$$\tilde{D}_{ab} = - \iint_A E z' dA = - \iint_A E(z - d) dA = 0 \quad (12)$$

Según la ecuación (11) y (12) podemos obtener la distancia entre el eje de la viga y el eje neutro:

$$d = - \frac{\tilde{D}_{ab}}{\tilde{D}_a} \quad (13)$$

Es recomendable asegurar que el eje de la viga x y el eje neutro coincidan, de manera que $\tilde{D}_{ab} = 0$ en todos los casos. De esta forma la matriz constitutiva es diagonal y facilita el cálculo ya que los efectos axial y flector están desacoplados. Para conseguirlo únicamente se tiene que desplazar el eje x una distancia d , como se observa en la figura 2. Al variar la posición del origen de coordenadas tendremos que cambiar z por z' en las ecuaciones descritas anteriormente.

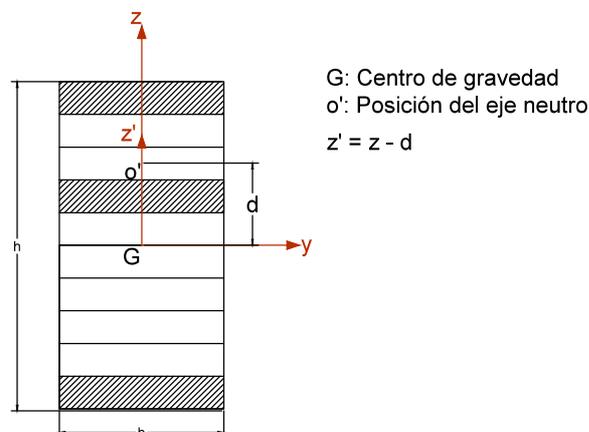


Fig. 2 Posición del eje neutro

Es importante recalcar que, al trasladar el eje de coordenadas, las expresiones \widehat{D}_a y \widehat{D}_s no varían ya que no dependen de la coordenada z , \widehat{D}_b tampoco varía ya que

$$\iint_A Ez'^2 dA = \iint_A Ez^2 dA \quad (14)$$

En cambio, el variar z por z' sí afecta a los desplazamientos horizontales u y al cálculo de la tensión normal.

$$\begin{aligned} u &= u_0 - z'\theta \\ \sigma_x &= E\left(\frac{\partial u_0}{\partial x} - z'\frac{\partial \theta}{\partial x}\right) \end{aligned} \quad (15)$$

El resultado obtenido para el desplazamiento vertical w , el giro θ y la tensión tangencial son independientes de la posición del origen de coordenadas.

3.5 TENSIONES INICIALES Y DEFORMACIONES TÉRMICAS

Cuando existe una deformación inicial axial debido a efectos térmicos (ε_x^o) y tensiones iniciales (σ_x^o, τ_{xz}^o) las ecuaciones que relacionan deformación y tensión quedan de la siguiente manera:

$$\begin{aligned} \sigma_x &= E(\varepsilon_x + \varepsilon_x^o) + \sigma_x^o \\ \tau_{xz} &= G \gamma_{xz} + \tau_{xz}^o \end{aligned} \quad (16)$$

donde $\varepsilon_x^o = \alpha \Delta T$, siendo α el coeficiente de dilatación térmico y ΔT el incremento de temperatura. Hay que recordar que la tensión tangencial inicial debido a la temperatura es nula.

La ecuación de las tensiones generalizadas se modifica como sigue:

$$\widehat{\sigma} = \widehat{D}\widehat{\varepsilon} + \widehat{\sigma}^o \quad (17)$$

donde $\widehat{\sigma}^o$ es el vector de esfuerzos iniciales dado por:

$$\hat{\mathbf{g}} = [N^0, M^0, Q^0]^T \quad (18)$$

con

$$N^0 = \iint_A [-E\varepsilon_x^0 + \sigma_x^0] dA \quad M^0 = \iint_A [E\varepsilon_x^0 + \sigma_x^0]z dA \quad Q^0 = \iint_A \tau_{xz}^0 dA \quad (19)$$

3.6 DISTRIBUCIÓN DE TENSIONES Y DEFORMACIONES

Es necesario detenerse un instante y estudiar el aspecto que toman la distribución de tensiones y deformaciones en el caso de una viga laminada compuesta.

En la figura 3 se observa como la tensión normal varía linealmente con pendientes distintas en las diferentes capas, con posibilidad de que haya saltos en la interface de éstas dependiendo de sus propiedades. Es decir, a lo largo del canto la distribución de tensiones σ_x es discontinua, debido a las diferentes propiedades del material de las capas. Sin embargo, las deformaciones ε_x no tiene en cuenta las capas y varían linealmente con el canto.

Mención aparte merecen las tensiones tangenciales, las cuales varían siguiendo una parábola distinta según la capa en la que se encuentren. Sin embargo, la teoría de Timoshenko realiza la aproximación de suponer tensión tangencial constante en cada capa. Observando la ecuación (5b), la tensión τ_{xz} se calcula multiplicando la deformación por cortante, que es constante, por el módulo de rigidez G, que depende de la capa en la que nos encontramos.

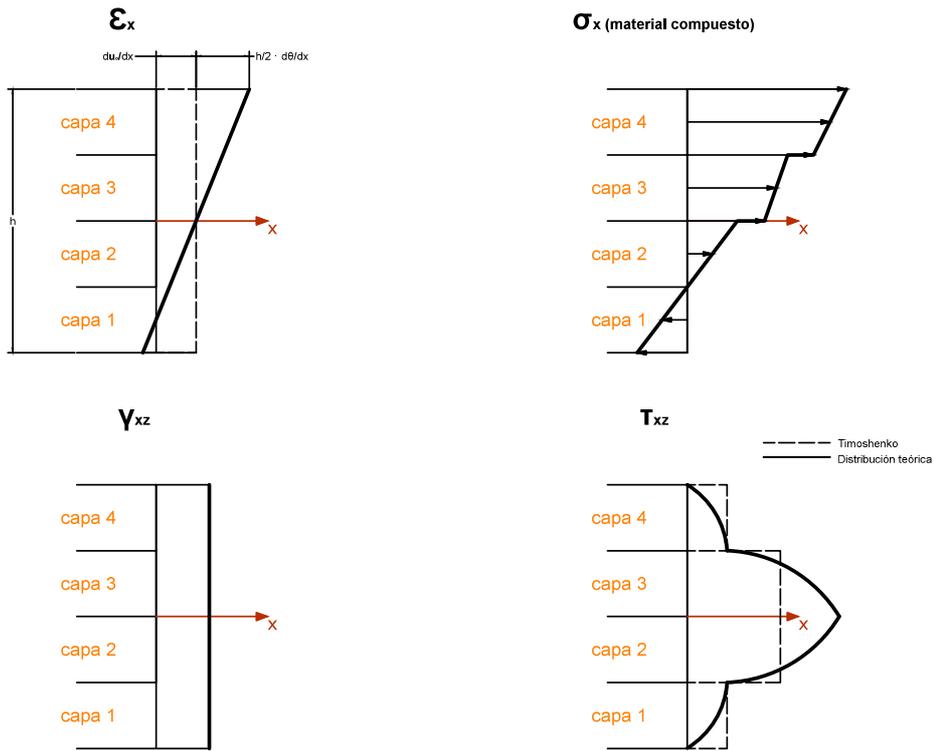


Fig. 3 Distribución de tensiones normal y tangencial en el canto de una sección compuesta de 4 capas

3.7 COEFICIENTE DE FORMA O DISTORSIÓN k_z

El coeficiente de distorsión k_z es el parámetro corrector del esfuerzo cortante. Su nombre se debe a que tiene en cuenta el efecto de que en realidad las secciones no se mantienen exactamente planas y tienen una distorsión longitudinal.

Suponiendo flexión cilíndrica alrededor del eje y , es decir $\tau_{xy} = 0$, el valor de k_z se calcula igualando la energía de deformación por cortante "exacta" (U_1) con la energía de deformación por cortante según la teoría de vigas de Timoshenko (U_2):

$$U_1 = \frac{1}{2} \iint_A \frac{\tau_{xz}^2}{G} dA \quad U_2 = \frac{1}{2} \frac{Q^2}{k_z D_s} \quad (20)$$

donde τ_{xz} es la tensión tangencial exacta. Igualando las expresiones (20) obtenemos el parámetro corrector k_z :

$$k_z = \frac{Q^2}{D_s} \left[\iint_A \frac{\tau_{xz}^2}{G} dA \right]^{-1} \quad (21)$$

Por tanto, si invertimos el orden de la ecuación (8) obtenemos que:

$$\begin{aligned} \frac{\partial u_0}{\partial x} &= \frac{1}{\tilde{D}} [\tilde{D}_b N - \tilde{D}_{ab} M] \\ \frac{\partial \theta}{\partial x} &= \frac{1}{\tilde{D}} [-\tilde{D}_{ab} N + \tilde{D}_a M] \end{aligned} \quad (22)$$

con

$$\tilde{D} = \tilde{D}_a \tilde{D}_b - \tilde{D}_{ab}^2 \quad (23)$$

Sustituyendo las ecuación (22) en (5a) se obtiene la tensión normal σ_x de cada capa en función de los esfuerzos N y M:

$$\sigma_x = \frac{E}{\tilde{D}} [\tilde{D}_b N - \tilde{D}_{ab} M - z(-\tilde{D}_{ab} N + \tilde{D}_a M)] \quad (24)$$

La distribución exacta de τ_{xz} que satisface las ecuaciones de equilibrio de la elasticidad es la siguiente:

$$\tau_{xz}(z) = - \int_{h^-}^z \frac{\partial \sigma_x}{\partial x} dz \quad (25)$$

Sustituyendo la ecuación (24) en la (25) y tomando $\frac{\partial N}{\partial x} = 0$ y $\frac{\partial M}{\partial x} = -Q$ se obtiene:

$$\tau_{xz}(z) = \frac{-Q}{\tilde{D}} F(z) \quad (26)$$

con

$$F(z) = \tilde{D}_\alpha S(z) + \tilde{D}_{\alpha b} \int_{h^-}^z zE(z)dz \quad , \quad S(z) = \int_{h^-}^z zE(z)dz \quad (27)$$

donde h^- es la coordenada de la fibra inferior de la sección de la viga y $S(z)$ es el momento estático del módulo de Young respecto a la coordenada z .

Por tanto, sustituyendo la ecuación (26) en la (21) se obtiene el coeficiente de forma:

$$k_z = \frac{\tilde{D}^2}{D_s} \left[\iint_A \frac{F^2(z)}{G(z)} dA \right]^{-1} \quad (28)$$

En el caso que el eje de la viga x coincida con el eje neutro, $F = \tilde{D}_\alpha S(z)$ y $\tilde{D} = \tilde{D}_\alpha \tilde{D}_b$ con lo que el coeficiente corrector del esfuerzo cortante es:

$$k_z = \frac{\tilde{D}_b^2}{D_s} \left[\iint_A \frac{S^2(z)}{G(z)} dA \right]^{-1} \quad (29)$$

3.8 PRINCIPIO DE LOS TRABAJOS VIRTUALES

Suponiendo únicamente cargas repartidas \mathbf{q} , la expresión del principio de trabajos virtuales (PTV) es:

$$\iiint_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_l \delta \mathbf{u}^T \mathbf{q} dx \quad (30)$$

donde $\delta \mathbf{u} = [\delta u_0, \delta w_0, \delta \theta]^T$ es el vector de desplazamientos virtuales, $\delta \boldsymbol{\varepsilon}$ y $\boldsymbol{\sigma}$ son el vector de deformaciones virtuales y el vector de tensiones, respectivamente, y $\mathbf{q} = [f_x, f_z, m]^T$ es el vector de fuerzas externas actuando sobre el eje de la viga debido a cargas distribuidas horizontales f_x , verticales f_z y a un momento distribuido m .

La integral del lado izquierdo de la ecuación (30) representa el trabajo virtual interno. Esta expresión la podemos expresar en términos de esfuerzos y deformaciones generalizadas de la siguiente manera. Haciendo uso de las ecuaciones (4) y (6), la ecuación (30) puede reescribirse:

$$\iiint_V \delta \varepsilon^T \boldsymbol{\sigma} dV = \int_l \delta \hat{\boldsymbol{\varepsilon}}^T \left[\iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA \right] \hat{\boldsymbol{\varepsilon}} dx = \int_l \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\mathbf{D}} \hat{\boldsymbol{\varepsilon}} dx = \int_l \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\sigma}} dx \quad (31)$$

Por tanto, el PTV puede expresarse:

$$\int_l \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\sigma}} dx = \int_l \delta \mathbf{u}^T \mathbf{q} dx \quad (32)$$

Hay que tener en cuenta que las derivadas que aparecen en el PTV son de primer orden. De este modo, puede utilizarse una aproximación de elementos finitos lagrangiana en la que únicamente se exige continuidad de los desplazamientos u_0 , w_0 y del giro θ .

3.9 ELEMENTO DE VIGA DE DOS NODOS DE TIMOSHENKO

Discretizamos la viga en elementos de dos nodos de longitud $l^{(e)}$ con una aproximación lineal de u_0 , w_0 y θ como

$$\mathbf{u} = \begin{Bmatrix} u_0 \\ w_0 \\ \theta \end{Bmatrix} = \sum_{i=1}^2 N_i(\xi) \mathbf{a}_i^{(e)} \quad \text{con} \quad \mathbf{a}_i^{(e)} = \begin{Bmatrix} u_0 \\ w_0 \\ \theta \end{Bmatrix}_i \quad (33)$$

donde u_{0i} , w_{0i} y θ_i son los valores nodales del desplazamiento axial, la flecha y el giro, respectivamente y $N_i(\xi)$ son funciones de forma lineales dadas por

$$N_i(\xi) = \frac{1}{2} (1 + \xi \xi_i) \quad i = 1, 2 \quad (34)$$

donde la coordenada normalizada ξ se define como $\xi = \frac{2(x-x_c)}{l^{(e)}}$, siendo x_c la coordenada del centro del elemento.

Sustituyendo la aproximación (33) en la expresión del vector de deformaciones generalizadas (4b):

$$\hat{\boldsymbol{\varepsilon}} = \begin{Bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta \end{Bmatrix} = \sum_{i=1}^2 \mathbf{B}_i \mathbf{a}_i^{(e)} = \mathbf{B} \mathbf{a}^{(e)} \quad (35)$$

con

$$\mathbf{a}^{(e)} = \begin{Bmatrix} a_1^{(e)} \\ a_2^{(e)} \end{Bmatrix} \quad \text{y} \quad \mathbf{B}_i = \begin{Bmatrix} B_{ai} \\ \dots \\ B_{bi} \\ \dots \\ B_{si} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial x} \\ 0 & \frac{\partial N_i}{\partial x} & -N_i \end{bmatrix} \quad (36)$$

donde \mathbf{B}_{ai} , \mathbf{B}_{bi} y \mathbf{B}_{si} son las matrices de tamaño 1x3 de deformaciones generalizadas de membrana, flexión y cortante, respectivamente.

Sustituyendo la ecuación constitutiva (17) en el PTV (ecuación 32) y utilizando la ecuación (35) se obtiene, tras simplificar los movimientos virtuales, la ecuación matricial de equilibrio como

$$\mathbf{K} \mathbf{a} = \mathbf{f} \quad (37)$$

donde \mathbf{a} es el vector de movimientos nodales, \mathbf{K} es la matriz de rigidez y \mathbf{f} es el vector de fuerzas nodales equivalentes de la viga. La matriz \mathbf{K} y el vector \mathbf{f} se forman ensamblando las contribuciones de cada elemento dadas por

$$\mathbf{K}_{ij}^{(e)} = \int_{l^{(e)}} \mathbf{B}_i^T \hat{\mathbf{D}} \mathbf{B}_j dx \quad (38)$$

$$\mathbf{f}_i^{(e)} = \begin{Bmatrix} f_{xi} \\ f_{zi} \\ m_i \end{Bmatrix} = \int_{l^{(e)}} N_i^{(e)} \mathbf{q} dx - \int_{l^{(e)}} \mathbf{B}_i^T \hat{\boldsymbol{\sigma}}^0 dx \quad i, j = 1, 2 \quad (39)$$

La segunda integral en la expresión (39) se refiere al efecto de la deformación y tensión iniciales.

Haciendo uso de las expresiones de $\tilde{\mathbf{D}}$ y \mathbf{B}_i , la matriz de rigidez puede calcularse teniendo en cuenta contribuciones distintas:

$$\mathbf{K}_{ij}^{(e)} = \mathbf{K}_{a_{ij}}^{(e)} + \mathbf{K}_{b_{ij}}^{(e)} + \mathbf{K}_{s_{ij}}^{(e)} + \mathbf{K}_{ab_{ij}}^{(e)} + [\mathbf{K}_{ab_{ij}}^{(e)}]^T \quad (40)$$

donde

$$\mathbf{K}_{r_{ij}}^{(e)} = \int_{I^{(e)}} \mathbf{B}_{r_i}^T \tilde{\mathbf{D}}_r \mathbf{B}_{r_j} dx \quad r = a, b, s \quad (41)$$

y

$$\mathbf{K}_{ab_{ij}}^{(e)} = \int_{I^{(e)}} \mathbf{B}_{a_i}^T \tilde{\mathbf{D}}_{ab} \mathbf{B}_{b_j} dx \quad (42)$$

En las expresiones anteriores los índices a,b,s y ab denotan respectivamente la contribución de la membrana, flexión, cortante y acoplamiento membrana-flexión a la matriz de rigidez del elemento de viga.

Como se ha demostrado en el apartado 3.4 de este capítulo, si el eje de la viga x coincide con el eje neutro, entonces $\tilde{\mathbf{D}}_{ab}$ es cero y en consecuencia $\mathbf{K}_{ab}^{(e)}$ también lo es. Esto permite que la matriz de rigidez elemental se pueda definir como:

$$\mathbf{K}^{(e)} = \begin{bmatrix} \mathbf{K}_a^{(e)} & 0 \\ 0 & \mathbf{K}_f^{(e)} \end{bmatrix} \quad (43)$$

con el vector de desplazamientos nodales ordenado de la siguiente manera:

$$\mathbf{a}^{(e)} = [u_{0_1}, u_{0_2}, w_{0_1}, \theta_1, w_{0_2}, \theta_2]^T \quad (44)$$

En la ecuación (43), $\mathbf{K}_a^{(e)}$ se refiere a la matriz de rigidez para un elemento de dos nodos cargado axialmente, mientras que $\mathbf{K}_f^{(e)}$ es la matriz de rigidez (bajo efectos de la flexión) de un elemento de dos nodos para vigas de Timoshenko e incorpora la contribución tanto a flexión como a cortante. Las dos matrices se muestran a continuación:

$$\mathbf{K}_a^{(\varepsilon)} = \frac{\widehat{D}_a}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (45)$$

$$\mathbf{K}_f^{(\varepsilon)} = \mathbf{K}_b^{(\varepsilon)} + \mathbf{K}_s^{(\varepsilon)} \quad (46)$$

con $\mathbf{K}_b^{(\varepsilon)}$ y $\mathbf{K}_s^{(\varepsilon)}$ son, respectivamente, las matrices de rigidez a flexión y a cortante descritas en la expresión (41). Sin embargo, estas matrices se pueden expresar en el sistema natural de coordenadas. Realizando el cambio de variable $dx = \frac{l^{(\varepsilon)}}{2} d\xi$, las matrices de la ecuación (46) se reescriben así:

$$\mathbf{K}_b^{(\varepsilon)} = \int_{-1}^{+1} \mathbf{B}_b^T (\widehat{D}_b) \mathbf{B}_b \frac{l^{(\varepsilon)}}{2} d\xi \quad , \quad \mathbf{K}_s^{(\varepsilon)} = \int_{-1}^{+1} \mathbf{B}_s^T (\widehat{D}_s) \mathbf{B}_s \frac{l^{(\varepsilon)}}{2} d\xi \quad (47)$$

Las integrales de la expresión (47) se calculan utilizando una cuadratura 1D de Gauss:

$$\mathbf{K}_r^{(\varepsilon)} = \sum_{p=1}^{n_p} (\mathbf{B}_r^T \widehat{D}_r \mathbf{B}_r)_p w_p \frac{l^{(\varepsilon)}}{2} \quad , \quad \text{con } r = b, s \quad (50)$$

donde n_p es el número de puntos de integración en el elemento viga y w_p son los pesos de las cuadraturas.

La matriz de rigidez a flexión $\mathbf{K}_b^{(\varepsilon)}$ se evalúa de manera exacta mediante una integración numérica de Gauss-Legendre utilizando un único punto de integración:

$$\mathbf{K}_b^{(\varepsilon)} = \left(\frac{\widehat{D}_b}{l} \right)^{(\varepsilon)} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (51)$$

Para el cálculo de la matriz de rigidez a cortante $\mathbf{K}_s^{(\varepsilon)}$ se necesitan dos puntos de integración de Gauss, de manera que la integración exacta proporciona el siguiente resultado:

$$K_s^{(\varepsilon)} = \left(\frac{\widetilde{D}_s}{l} \right)^{(\varepsilon)} \begin{bmatrix} 1 & \frac{l^{(\varepsilon)}}{2} & -1 & \frac{l^{(\varepsilon)}}{2} \\ \frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{3} & -\frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{6} \\ -1 & -\frac{l^{(\varepsilon)}}{2} & 1 & -\frac{l^{(\varepsilon)}}{2} \\ \frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{6} & -\frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{3} \end{bmatrix} \quad (52)$$

En el análisis de vigas esbeltas, esta matriz de rigidez a cortante (52) produce problemas en los resultados ya que presenta valores demasiado rígidos, es lo que se llama efecto bloqueo de la solución. En el siguiente capítulo veremos en detalle este fenómeno y describiremos la solución que se adopta para evitarlo.

3.10 EFECTO BLOQUEO EN VIGAS LAMINADAS COMPUESTAS

La rigidez a flexión de la viga, como hemos visto en la expresión (46), se divide en un término que representa la rigidez a cortante y otro que representa la rigidez a flexión. El valor relativo del primer término con respecto al segundo afecta a la solución del problema de vigas. Dicho valor puede cuantificarse con el coeficiente β , dado por la siguiente expresión:

$$\beta = \frac{12\widetilde{D}_b}{l^2\widetilde{D}_s} \quad (53)$$

Un valor pequeño de β indica que la influencia de la deformación a cortante es despreciable. Este parámetro depende de la geometría y de las propiedades mecánicas de la sección y es inversamente proporcional a la esbeltez de la viga (λ). De ahí que, cuando la esbeltez de la viga aumenta, el parámetro β se hace más pequeño y por tanto la deformación por cortante disminuye, tal y como cabría esperar en vigas esbeltas.

Vamos a estudiar la influencia de λ en la solución numérica. A continuación se presenta los resultados obtenidos en el cálculo de un voladizo bajo una carga puntual y sección homogénea. El análisis se realiza mediante la teoría de Euler-Bernoulli, que no tiene en cuenta la deformación por cortante, y la teoría de Timoshenko, en la que sí interviene. Los desplazamientos que se obtienen mediante el cálculo "exacto" de cada teoría son:

$$w^{EB} = \frac{l^3}{3EI_y} P \quad ; \quad w^T = \left(\frac{l}{GA^*} + \frac{l^3}{3EI_y} \right) P \quad (54)$$

donde el subíndice EB se refiere a Euler-Bernoulli y T a Timoshenko. No entraremos en detalles del cálculo de estos desplazamientos ya que no es objeto principal de este capítulo, sino que nos interesa la consecuencia que se deduce de estos resultados.

El efecto de la deformación por cortante es despreciable en vigas esbeltas, de manera que la solución calculada según la teoría de Timoshenko debería coincidir con la de Euler-Bernoulli al analizar este tipo de vigas. Veamos que sucede con el ratio que expresa la relación de flechas de cada teoría:

$$r_w = \frac{w^T}{w^{EB}} = \frac{3(4\lambda^2 + 3)}{4\lambda^2(\lambda^2 + 3)} \quad (55)$$

El ratio r_w debería tender a 1 a medida que la esbeltez incrementara, sin embargo, para vigas infinitamente esbeltas ($\lambda \rightarrow \infty$) este ratio tiende a 0. Este ejemplo intenta demostrar que, a medida que la esbeltez de una viga aumenta, la solución numérica se va rigidizando progresivamente. Este fenómeno se conoce como *efecto bloqueo de la solución* y nos indica que no es posible realizar el análisis de vigas esbeltas con la teoría de Timoshenko.

Sin embargo, hay varios procedimientos que consiguen eliminar este efecto bloqueo y, en consecuencia, permitir que la teoría de Timoshenko sea válida tanto para vigas esbeltas como para vigas de canto grueso. La opción más popular, y que utilizaremos, es la de reducir la influencia de la rigidez a cortante mediante integración reducida. Es decir, en lugar de calcular $K_s^{(e)}$ con dos puntos de integración, la calcularemos con un único punto. La integral $K_b^{(e)}$ sí que se calculará de manera exacta, ya que para ello era necesario sólo un punto de integración.

Por tanto, los elementos de viga compuesta de dos nodos de Timoshenko los calcularemos evaluando las integrales de las matrices de rigidez en un solo punto de integración de Gauss-Legendre. La expresión $K_b^{(e)}$ la hemos definido en la ecuación (51) y $K_s^{(e)}$ se describe a continuación:

$$K_s^{(\varepsilon)} = \left(\frac{\widehat{D}_s}{l} \right)^{(\varepsilon)} \begin{bmatrix} 1 & \frac{l^{(\varepsilon)}}{2} & -1 & \frac{l^{(\varepsilon)}}{2} \\ \frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{4} & -\frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{4} \\ -1 & -\frac{l^{(\varepsilon)}}{2} & 1 & -\frac{l^{(\varepsilon)}}{2} \\ \frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{4} & -\frac{l^{(\varepsilon)}}{2} & \frac{(l^{(\varepsilon)})^2}{4} \end{bmatrix} \quad (56)$$

3.11 VECTOR DE FUERZAS NODALES

Una vez hemos calculado la matriz de rigidez de la viga, en la que aparece la aportación de la rigidez a axil, a flexión y a cortante, podemos calcular el vector de fuerzas nodales como:

$$f^{(\varepsilon)} = \begin{Bmatrix} f_a^{(\varepsilon)} \\ f_f^{(\varepsilon)} \end{Bmatrix} \quad (57)$$

donde $f_a^{(\varepsilon)}$ es el vector de fuerzas nodales para cargas horizontales:

$$f_{a_i}^{(\varepsilon)} = \int_{l^{(\varepsilon)}} N_i n \, dx \quad , \quad i = 1,2 \quad (58)$$

donde n es una carga axil distribuida a lo largo del eje de la viga. Si realizamos el cálculo de la integral (58) en el sistema natural de coordenadas, entonces:

$$f_{a_i}^{(\varepsilon)} = \int_{-1}^{+1} N_i n \frac{l^{(\varepsilon)}}{2} \, d\varepsilon \quad , \quad i = 1,2 \quad (59)$$

Para una carga uniformemente distribuida en dirección axil:

$$f_{a_i}^{(\varepsilon)} = \frac{l^{(\varepsilon)}}{2} n \quad (60)$$

Por otra parte, $f_f^{(\varepsilon)}$ es el vector de fuerzas nodales a flexión y se describe mediante la siguiente expresión:

$$f_f^{(e)} = [f_{z1}, m_1, f_{z2}, m_2]^T \quad (61)$$

donde f_{zi} y m_i son las fuerzas verticales y momentos de cada nodo, que varían según la carga aplicada a la viga. Para una carga uniformemente repartida q :

$$f_f^{(e)} = \left[\frac{ql}{2}, \frac{ql^2}{12}, \frac{ql}{2}, -\frac{ql^2}{12} \right]^T \quad (62)$$

Finalmente, el vector de fuerzas nodales será una matriz 6x1 en la que aparecerá la influencia de las cargas horizontales $f_a^{(e)}$ y las cargas verticales $f_f^{(e)}$.

4. IMPLEMENTACIÓN NUMÉRICA

4.1 INTRODUCCIÓN A MATLAB Y GID

Una vez estudiada la teoría de Timoshenko para vigas compuestas, en el siguiente capítulo vamos a implementar dicha teoría numéricamente.

La importancia del método de los elementos finitos radica en la iteración de los cálculos sobre una gran cantidad de elementos, por tanto es necesario un lenguaje de programación que permita simplificar la codificación de algoritmos y que optimice los recursos del ordenador. MATLAB es un programa eficiente en este aspecto ya que permite hacer uso de toda la librería matricial implementada. Se trata de una herramienta diseñada para trabajar con matrices, facilitando todas las operaciones del álgebra matricial desde el punto de vista numérico y de almacenamiento de memoria.

Sin embargo, no es suficiente contar solo con un programa de cálculo óptimo para analizar estructuras con el método de los Elementos Finitos, sino que también es necesario contar con una interfaz adecuada que permita preparar los datos del problema, generar mallas acordes al tipo de problema a resolver y presentar los resultados de manera que la interpretación de éstos sea el principal objetivo. Aquí entra en juego el pre/postprocesador GID.

GID es una herramienta diseñada para tratar adecuadamente las entidades geométricas, realizar la asignación de las propiedades materiales y definir las condiciones de contorno del problema. Además, permite una rápida interpretación a través de la visualización de los resultados.

Después de enumerar las características de MATLAB y GID, es fácilmente interpretable que la complementariedad de estos dos programas es el camino perfecto para analizar numéricamente estructuras mediante elementos finitos. Esta interacción conjunta de GID y MATLAB se consigue a través de MAT-fem.

MAT-fem divide los trabajos de la siguiente manera. La tarea de manipular geometrías y discretizaciones la realiza GID, escribiendo el fichero de entrada de datos que servirá para que MATLAB sea capaz de calcular el problema. Finalmente, GID recoge los ficheros de datos para su visualización gráfica e interpretación. Lo veremos con mayor profundidad en los siguientes apartados.

4.2 MAT-FEM

MAT-fem es un programa de ejecución TOP-DOWN. El esquema que sigue el programa se muestra en la figura 4. La primera tarea es definir el fichero de datos, que aporta la información necesaria para calcular el problema en cuestión. Una vez se realiza la lectura de este fichero, se arma, para todos los elementos, la matriz de rigidez y el vector de fuerzas nodales, procediendo directamente al ensamblaje de estos. Se termina de aplicar las condiciones de carga al vector de fuerzas nodales antes de eliminar las ecuaciones correspondientes a los grados de libertad conocidos y poder resolver el sistema de ecuaciones. Conocidos los desplazamientos se calculan las tensiones y esfuerzos y se finaliza con la escritura de datos que permite visualizar los resultados en GID.

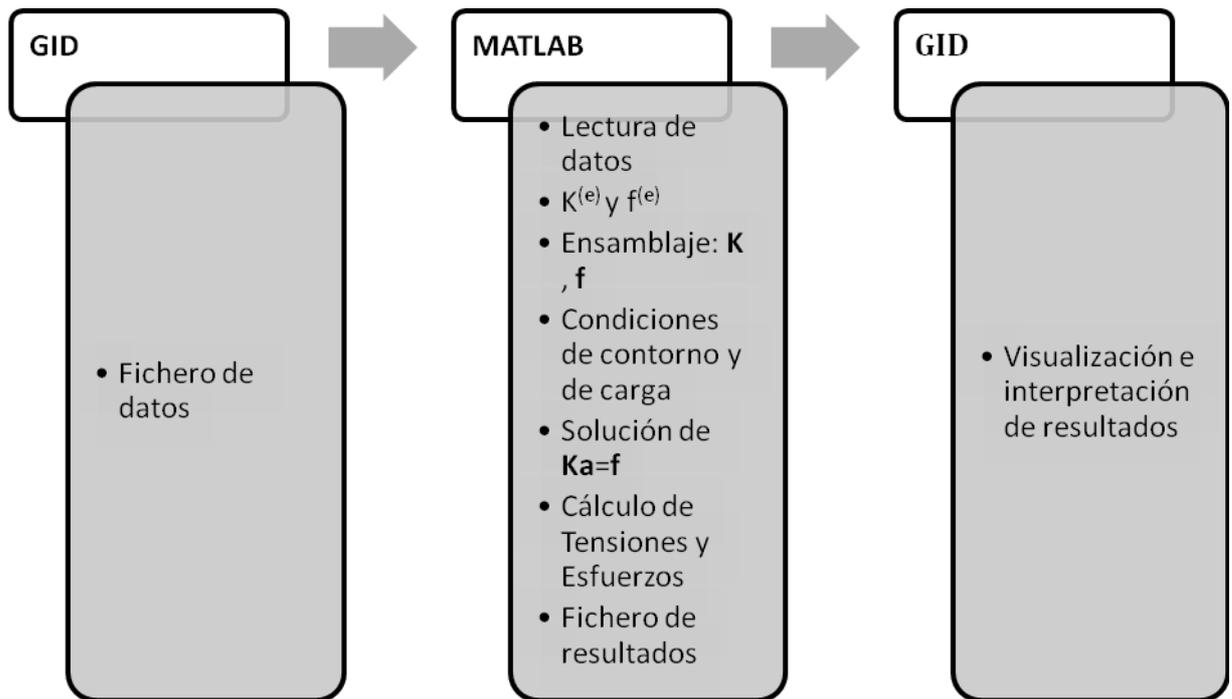


Fig. 4 Esquema de funcionamiento de MAT-fem

Una vez tenemos en mente el procedimiento que sigue MAT-fem para analizar estructuras por elementos finitos, estamos en disposición de describir el programa diseñado para el cálculo de vigas compuestas de Timoshenko.

4.3 PROGRAMACIÓN NUMÉRICA DE VIGAS COMPUESTAS DE TIMOSHENKO

4.3.1 Programa de cálculo en MATLAB

El programa de MATLAB que permite calcular las vigas compuestas de Timoshenko por elementos finitos se describe por medio de diferentes “scripts” (archivos externos de Matlab). Los algoritmos que se definen en cada uno de estos archivos son los que permiten obtener los resultados que describen el comportamiento estructural de este tipo de vigas. Estos archivos son los siguientes:

- a) Composite_Beam_Timoshenko_Reduced.m
- b) Composite_Layers_Beam_Timoshenko.m

- c) Timing.m
- d) ToGid_Composite_Beam_Timoshenko.m

Cada uno de estos archivos se muestra completamente en el anejo 1 de la presente tesina. Se ha creído oportuno nombrar las variables y anotar algunas explicaciones de la programación en lengua inglesa, ya que los diferentes paquetes que existen de cálculo de estructuras de MAT-fem están escritos en esta lengua.

Antes de ejecutar el programa en sí, es obvio que de alguna manera es necesario alimentarlo con la información correspondiente a las coordenadas nodales, la discretización en elementos finitos y las condiciones iniciales que describen el problema a resolver. El fichero que proporciona esta información es el denominado *fichero de datos*, que, como comentábamos con anterioridad, nos lo facilita el programa GID. Este fichero es una subrutina del programa de cálculo en MATLAB y, por tanto, es importante familiarizarse con las variables que utiliza.

Fichero de datos

Como se ha podido intuir, el fichero de entrada de datos utilizará la sintaxis de MATLAB y en él se definen directamente las variables utilizadas por el programa para describir el problema a resolver. Se distinguen tres grupos de variables: las asociadas al material, las que definen la topología del problema y las que definen las condiciones de contorno asociadas.

En la siguiente captura de pantalla del fichero de datos se distinguen las variables asociadas al material.

```

%=====
====
% MAT-fem_Beams 1.0 - MAT-fem is a learning tool for understanding
%                       the Finite Element Method with MATLAB and GiD
%=====
====
% PROBLEM TITLE = Untitled
%
%   Material Properties
%
layers = 3.00e+00;

young = [
    2.1000000000e+11 ;
    3.0000000000e+10 ;
    3.0000000000e+10 ];

poiss = [
    2.0000000000e-01 ;
    2.0000000000e-01 ;
    2.0000000000e-01 ];

denss = zeros(3,1) ;

thickness = [
    2.0000000000e+00 ;
    1.6000000000e+01 ;
    2.0000000000e+00 ];

width = [
    1.0000000000e+01 ;
    1.0000000000e+01 ;
    1.0000000000e+01 ];

```

Fig. 5 Variables descritas en el fichero de datos asociadas al material

Primero de todo se define el número de capas (*layers*) en que se compone la viga y se presentan las características de cada una de ellas: el módulo de Young (*young*), el coeficiente de Poisson (*poiss*), la densidad de cada material (*denss*) y, finalmente, el espesor y ancho de cada capa (*thickness* y *width*, respectivamente). Se hace notar que, a diferencia de las vigas de material homogéneo, las propiedades mecánicas son matrices en las que cada fila se refiere a una capa distinta.

Después de definir las propiedades y dimensiones de la viga, se define la topología del problema. Es decir, la información de la malla en que se discretiza la estructura. En la captura de pantalla siguiente se presentan los datos de una malla conformada por dos elementos.

```

% Coordinates
%
global coordinates

coordinates = [
    1.000000000e+01   ;
    5.000000000e+00   ;
    0.000000000e+00   ] ;

%
% Elements
%
global elements

elements = [
    3   ,   2   ;
    2   ,   1   ] ;

```

Fig. 6 Variables descritas en el fichero de datos asociadas a la topología del problema

En la anterior figura, se distinguen las coordenadas globales de los nodos en que se divide la viga (*coordinates*) y la numeración global de éstos, que conforman los elementos de dos nodos de Timoshenko y definen sus conectividades (*elements*). La variable *elements* tiene tantas filas como elementos tenga la malla en cuestión y siempre tendrá dos columnas, una por cada nodo del elemento. No está de más adelantar, aunque lo veremos con más detalle en el apartado 4.3.2, que la entrada de datos se realiza en una dimensión (coordenada x), en cambio, en el postproceso, los resultados se visualizan en dos dimensiones, pudiéndose así distinguir las capas de la estructura y el comportamiento estructural a través del espesor de la viga.

Finalmente se muestra el último grupo de variables, las que definen las condiciones de contorno del problema, como se muestra en la figura 7.

```

% Fixed Nodes
%
fixnodes = [
1 , 1 , 0.000000000e+00 ;
1 , 2 , 0.000000000e+00 ;
1 , 3 , 0.000000000e+00 ;
3 , 1 , 0.000000000e+00 ;
3 , 2 , 0.000000000e+00 ;
3 , 3 , 0.000000000e+00 ] ;
%
% Point loads
%
pointload = [
1 , 1 , 0.000000000e+00 ;
1 , 2 , -1.000000000e+01 ;
1 , 3 , 0.000000000e+00 ] ;
%
% Side loads
%
uniload = sparse ( 2 , 2 );
uniload ( 1 , 1 ) = 0.000000000e+00 ;
uniload ( 2 , 1 ) = 0.000000000e+00 ;
uniload ( 1 , 2 ) = -5.000000000e+00 ;
uniload ( 2 , 2 ) = -5.000000000e+00 ;

```

Fig.7 Variables descritas en el fichero de datos asociadas a las condiciones de contorno del problema

La variable *fixnodes* corresponde a los grados de libertad restringidos de acuerdo con el problema a resolver. Esta variable es una matriz donde el número de filas corresponde al número de grados de libertad prescrito, mientras que el número de columnas indican en el siguiente orden el nodo restringido, el grado de libertad vinculado (1 para desplazamiento horizontal-x, 2 para desplazamiento vertical-y y 3 para el giro) y el valor del grado de libertad conocido.

La siguiente variable que aparece es *pointload*, que se usa para definir las cargas puntuales. Se trata de una matriz donde el número de filas es el número de cargas definidas en el problema y el número de columnas corresponden al número de nodo sobre el cual actuará la carga, la dirección en que actúa (1 significa que la carga actúa en dirección x, 2 en dirección y y 3 que la carga es un momento puntual) y el valor de la carga. Finalmente, *uniload* contiene la información sobre las cargas uniformemente repartidas a lo largo de los elementos. Es una matriz donde el número de filas es el número de elementos con carga y las dos columnas se refieren al número de elemento y al grado de libertad de la carga (1 es carga horizontal y 2 carga vertical).

El fichero de datos tendrá la extensión.m a fin de que MATLAB pueda reconocerlo como un script válido. No hay que olvidar que este archivo no es otra cosa más que una asignación directa a las variables dentro del programa.

Como apuntábamos al inicio del capítulo, la programación en MATLAB se hace a partir de diferentes scripts. Vamos a describir a continuación el programa de cálculo para vigas compuestas de Timoshenko e iremos haciendo referencia a cada uno de estos archivos y explicaremos la funcionalidad que tiene cada uno de ellos dentro del programa numérico en su globalidad.

Composite Beam Timoshenko Reduced.m

Mat-fem comienza limpiando todas las variables de la memoria con la instrucción *clear*, seguidamente pregunta al usuario el nombre del fichero de datos que se va a utilizar y que se ha explicado en el apartado anterior. No se debe incluir la extensión .m cuando se escriba el nombre del fichero. En la figura 8 se muestran las primeras líneas del código del programa y que corresponden a la inicialización de variables así como a la puesta a punto del reloj del programa (*tic*), almacenando en la variable *ttim* el tiempo total de ejecución.

```

%% 2 Nodes Composite Beam using Timoshenko Theory
% Reduce the influence of the transverse shear stiffness
(thick and slender beams)
% Clear memory and variables

clear

file_name = input('Enter the file name :','s');

tic;                % Start clock
ttim = 0;           % Initialize time counter
eval (file_name);  % Read input file

```

Fig.8 Composite_Beam_Timoshenko_Reduced

A continuación se guardan las dimensiones básicas de una serie de variables que utilizará el programa más adelante. Se utilizan las matrices del fichero de datos para definir estas dimensiones. Es decir, el número de nodos (*npnod*) corresponde al número de filas de la matriz *coordinates*, el número de elementos (*nelem*) en que se divide la malla es el número de filas de la matriz *elements* y las columnas de esta matriz corresponden al número de nodos por

elemento (*nnode*). Además, se añade directamente el número de grados de libertad por nodo *dofpn* y realizando operaciones con estas variables encontramos el número de grados de libertad por elemento y el número de grados de libertad total (*dofpe* y *nndof*, respectivamente).

```
% Finds basics dimensions
npnod = size(coordinates,1);           % Number of nodes
nelem = size(elements,1);             % Number of elements
nnode = size(elements,2);             % Number of nodes por element
dofpn = 3;                             % Number of DOF per node
dofpe = nnode*dofpn;                  % Number of DOF per element
nndof = npnod*dofpn;                  % Number of total DOF

ttim = timing('Time needed to read the input file',ttim);
%Reporting time
```

Fig. 9 Composite_Beam_Timoshenko_Reduced

Se hace notar que la última orden de la figura anterior corresponde a la llamada que se hace a la función *timing*. A lo largo del programa se utiliza esta rutina para calcular el tiempo de ejecución entre dos puntos del código, de esta manera se puede observar cual son las partes del programa que requieren mayor esfuerzo computacional. En este caso, calcula el tiempo que necesita la computadora para leer el fichero de datos. En la figura que se presenta a continuación se observa el script donde se define la función *timing*:

```
function t = timing (text,time)

%% timing Evaluates and acumulate the used time.
%
% Parameters:
%
% Input, text : String to show.
%           time : Accumulate time at t-1
%
% Output, t   : Accumulate time at t

itim = toc;                               % Close previous tic
fprintf(1,[text,' %12.6f \n'],itim);      % Output time and text
t = itim + time;                           % Accumulate time in t

tic                                       % Open a new tic
```

Fig. 10 Timing

Este script tiene la función claramente marcada de contar el tiempo que se tarda entre dos puntos del código. Primero para el cronómetro con la orden *toc*, después almacena y

escribe el tiempo que ha pasado entre los dos puntos del código con *itim* y *fprintf* y, finalmente, guarda el tiempo acumulado en la variable *t* y vuelve a inicializar el cronómetro con la orden *tic*.

Volviendo al script anterior, el siguiente paso es definir las dimensiones de las matrices globales del problema, de manera que cuando se realicen las operaciones de ensamblaje, el programa tenga almacenadas las matrices de rigidez global (*StifMat*), el vector de fuerzas global (*force*), el vector de reacciones global (*reaction*) y el vector de desplazamientos global (*u*).

```
% Dimension the global matrices.

StifMat = sparse ( nn dof , nn dof ); % Create the global
                                         stiffness matrix
force    = sparse ( nn dof , 1 );      % Create the global force
                                         vector
reaction = sparse ( nn dof , 1 );      % Create the global reaction
                                         vector
u        = sparse ( nn dof , 1 );      % Nodal variables
```

Fig. 11 Composite_Beam_Timoshenko_Reduced

Ahora se definen las propiedades del material que serán necesarias a lo largo del programa y que no se han definido directamente en el fichero de datos, como el módulo de rigidez (*shear*) y el área total de la sección de la viga (*area*).

```
% Material properties (Constant over the domain)

shear = young ./ (2 * (1 + poiss));
area = thickness * width ;
```

Fig.12 Composite_Beam_Timoshenko_Reduced

Además, es necesario calcular la coordenada vertical de la fibra inferior y superior de cada capa (*coord_ZG*), definida respecto al centro de gravedad de la sección (*coord_zg*).

El esquema que se ha seguido para calcular *coord_ZG* es el que se observa en la imagen siguiente. Se calcula primero la coordenada vertical *z* de la fibra inferior y superior de

cada capa respecto a la fibra inferior de la sección (*coord_z*), posteriormente se obtiene el centro de gravedad de cada capa respecto al punto más bajo de la sección y, finalmente, se calcula el vector *coord_ZG* respecto el centro de gravedad global de la sección.

```

% Create coordinate z vector of points between each layer

coord_z = [zeros(1);thickness] ;

for j=2:layers+1
    coord_z(j,1) = coord_z(j,1)+coord_z(j-1,1); % Coordinate z
                                                    % respect
                                                    % the bottom of
                                                    % the section
end

for i=1:layers
    zg_layers(i)=(coord_z(i)+coord_z(i+1))*0.5; % Center of
                                                    % gravity of
                                                    % each layer
end

for i=1:layers
    area_layers(i)=thickness(i)*width(i);      % Area of each
                                                    % layer
end

coord_zg = (area_layers*zg_layers')/area;      % Center of
                                                    % gravity of
                                                    % the section

coord_ZG = coord_z-(coord_zg*ones(layers+1,1)); % Coordinate z
                                                    % respect the
                                                    % center of
                                                    % gravity of
                                                    % the section
    
```

Fig. 13 Composite_Beam_Timoshenko_Reduced

La obtención de las coordenadas verticales z anteriores es necesaria para el cálculo de las componentes de la matriz constitutiva generalizada, que se expone en la siguiente figura.

```

% Components of the generalized constitutive matrix

% Axial stiffness
D_mata = 0;
for i=1:layers
    D_mata = D_mata + width(i) * thickness (i) * young(i);
end

% Bending stiffness
D_matb = 0 ;
for i=1:layers
    D_matb = D_matb + (1/3)*((coord_ZG(i+1))^3-
(coord_ZG(i))^3)*width(i)*young(i);
end

% Coupling axial-bending stiffness
D_matab = 0;
for i = 1:layers
    D_matab = D_matab+(-
1/2)*(coord_ZG(i+1)+coord_ZG(i))*width(i)*thickness(i)*young(i);
end

% Shear stiffness
D_mats=0;
for i=1:layers
    D_mats = D_mats +
thickness(i)*width(i)*(young(i)/(2*(1+poiss(i))));
end

```

Fig. 14 Composite_Beam_Timoshenko_Reduced

Es necesario calcular el parámetro de forma (k_z) para obtener la correcta rigidez a cortante (D_{mats}). Haciendo referencia al capítulo 3.7, donde se expone la teoría para el cálculo de este parámetro k_z , recordamos que

$$k_z = \frac{\hat{D}_b^2}{D_s} \left[\iint_A \frac{S^2(z)}{G(z)} dA \right]^{-1} \quad (63)$$

En la siguiente figura se adjunta el algoritmo utilizado para su obtención. Se hace notar que las variables $S1$, $S2$ y $S3$ corresponden al momento estático de Young respecto a la fibra de coordenada z , estas variables son necesarias para calcular la integral que aparece en (63).

```

% Compute the shear correction parameter for bending around the y
axis
% (kz)

S1 = 0;
S3 = 0;
for i = 1 : layers
    if i > 1
        S1 = S1 + young(i-1)* (coord_ZG(i)^2 - coord_ZG(i-1)^2);
    end
    S2 = 1/20*young(i)^2*(coord_ZG(i+1)^5-coord_ZG(i)^5)+...
        1/6*(S1-young(i)*coord_ZG(i)^2) * young(i) *
        (coord_ZG(i+1)^3- coord_ZG(i)^3)+...
        1/4*(S1-young(i)*coord_ZG(i)^2)^2 * (coord_ZG(i+1)-
        coord_ZG(i));
    S3 = S3 + width(i)/shear(i)*S2 ;
end

kz = D_matb^2/D_mats/S3;

% New shear stiffness

D_mats = kz*D_mats;

```

Fig. 15 Composite_Beam_Timoshenko_Reduced

Tal y como vimos en el capítulo en el que se explicaba la teoría de vigas compuestas de Timoshenko, es muy práctico referenciar las coordenadas verticales z respecto al eje neutro de la sección, y esto se hace desplazando una distancia d el centro de gravedad. A continuación se muestra el cálculo de esta distancia y el nuevo vector de coordenadas verticales de las fibras superior e inferior de cada capa calculado respecto al eje neutro ($coord_zp$).

```

% Vertical distance between the global axis x and the neutral
axis.

d = - D_matab / D_mata ;

% Create coordinate z vector of points between each layer respect
% neutral axis

coord_zp = coord_ZG-(d*ones(layers+1,1));

```

Fig. 16 Composite_Beam_Timoshenko_Reduced

Ahora es posible obtener la rigidez de acoplamiento axil-flector respecto al eje neutro, que, evidentemente, será nulo.

```

%   New coupling axial-bending stiffness

D_matab=0;

    for i = 1:layers

D_matab =D_matab + (-
1/2)*(coord_zp(i+1)+coord_zp(i))*width(i)*thickness(i)*young(i);

    end

ttim = timing('Time needed to set initial values',ttim); %Reporting
time

```

Fig. 17 Composite_Beam_Timoshenko_Reduced

Se observa en la figura anterior que vuelve a llamarse a la función `timing`, esta vez para calcular el tiempo necesario para almacenar las variables iniciales. Es decir, se mide el tiempo que ha pasado desde el último `ttim`.

Una vez definidas todas las variables que describen las propiedades mecánicas del problema, es momento de realizar un ciclo elemental donde se calcule y ensamble las matrices de rigidez y el vector de cargas nodales equivalente para cada elemento.

En la siguiente captura de pantalla se observa que la primera orden dentro del ciclo es definir la longitud de cada elemento (len), que se consigue a partir de las variables del fichero de datos que nos proporciona las coordenadas x de cada nodo. En el vector `lnods` se almacenan las conectividades nodales del elemento a tratar y en la matriz `coor_x` se guardan las coordenadas de dichos nodos.

```

% Element cycle

for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode);

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem. X
coordinate

    len = coor_x(2) - coor_x(1); % x_j > x_i

```

Fig. 18 Composite_Beam_Timoshenko_Reduced

El siguiente paso es definir las matrices de rigidez elementales.

```

const = D_matb/len;

K_bend = [ 0 , 0 , 0 , 0 ;
           0 , 1 , 0 , -1 ;
           0 , 0 , 0 , 0 ;
           0 , -1 , 0 , 1 ];

K_bend = K_bend * const;

const = D_mats/len;

K_shear = [ 1 , len/2 , -1 , len/2 ;
            len/2 , len^2/4 , -len/2 , len^2/4 ;
            -1 , -len/2 , 1 , -len/2 ;
            len/2 , len^2/4 , -len/2 , len^2/4 ];

K_shear = K_shear * const;

K_flex = K_bend + K_shear;

const = D_mata/len;

K_axial = [ 1 , -1 ;
            -1 , 1 ];

K_axial = K_axial * const ;

```

Fig. 19 Composite_Beam_Timoshenko_Reduced

En el cuadro anterior se observa que se ha calculado la matriz elemental de igual manera como se explicó en el capítulo 3.9 de teoría. Es importante apuntar que la forma de ordenar las componentes de la matriz de rigidez está relacionada con el orden de las

componentes del vector de desplazamientos. Según la expresión (44) $\mathbf{a}^{(e)} = [u_{0_1}, u_{0_2}, w_{0_1}, \theta_1, w_{0_2}, \theta_2]^T$. Sin embargo, con el objetivo de presentar los resultados de una forma más intuitiva y sencilla se cambiará el orden de los desplazamientos de manera que primero aparezcan los tres grados de libertad del nodo 1 y a continuación los tres del segundo nodo. Es decir, intercambiando la componente dos y tres y, posteriormente, la tres por la cuatro, el nuevo vector de desplazamientos será

$$\mathbf{a}^{(e)} = [u_{0_1}, w_{0_1}, \theta_1, u_{0_2}, w_{0_2}, \theta_2]^T \quad (64)$$

de manera que la matriz de rigidez elemental también tendrá que intercambiar la fila dos por la tres y ésta por la cuatro. El mismo intercambio se realiza con las columnas. Estas operaciones algebraicas se tienen en cuenta en el programa según el siguiente cuadro:

```
K_elem(:, [2 3]) = K_elem(:, [3 2]);
K_elem(:, [3 4]) = K_elem(:, [4 3]);

K_elem([2,3], :) = K_elem([3 2], :);
K_elem([3,4], :) = K_elem([4 3], :);
```

Fig. 20 Composite_Beam_Timoshenko

Una vez definida la matriz de rigidez elemental es momento de describir el vector de fuerzas nodales de la viga.

```
pp = thickness.*width.*denss;

fa = uniload(ielem,1) * len/2 ;
fz = (-sum(pp) + uniload(ielem,2)) * len/2;

ElemFor = [ fa, fz , 0, fa, fz, 0]; %elemental force vector.
```

Fig. 21 Composite_Beam_Timoshenko

En el cuadro anterior se distingue, por un lado, el vector de fuerzas nodales axial fa , que se calcula llamando a la primera columna de la variable *uniload*, que corresponde a la carga horizontal distribuida; y, por otro lado, aparece el vector de fuerzas nodales a flexión fz en el que intervienen la carga vertical distribuida, segunda columna de la variable *uniload*, y el peso

propio de la viga *pp*. Finalmente, se forma el vector de fuerzas elementales (*ElemFor*) teniendo en cuenta el orden correcto, primero los tres grados de libertad del nodo 1 y después los del nodo 2.

Antes de realizar el ensamblaje se define el vector *eqnum* que contiene el número de ecuaciones en numeración global correspondiente a cada una de las ecuaciones de la matriz de rigidez elemental. Una vez se conoce la posición donde se ubica cada componente elemental dentro de la matriz global, se realiza el ensamblaje obteniendo la matriz de rigidez global (*StifMat*) y el vector de fuerzas nodales global (*force*) de la viga.

El ensamblaje se realiza mediante dos ciclos, el primero permite ensamblar el vector de cargas nodales equivalente y dentro del segundo se calcula la matriz de rigidez término a término.

```

% Finds the equation number list for the i-th element
for i=1:nnode
    ii = (i-1)*dofpn;
    for j =1:dofpn
        eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build the equation
number list
    end
end

% Assemble the force vector and the stiffness matrix
for i = 1 : dofpe
    ipos = eqnum(i);
    force (ipos) = force(ipos) + ElemFor(i);
    for j = 1 : dofpe
        jpos = eqnum(j);
        StifMat (ipos,jpos) = StifMat (ipos,jpos) + K_elem(i,j);
    end
end

end % End element cycle

ttim = timing('Time to assemble the global system',ttim);
%Reporting time
    
```

Fig. 22 Composite_Beam_Timoshenko

En el cuadro anterior se observa como al ensamblar las matrices finaliza el ciclo y, posteriormente, se calcula el tiempo que tarda el programa en ensamblar las matrices globales

del sistema.

Llegados a este punto, es necesario añadir las cargas puntuales al vector de fuerzas y aplicar las condiciones de contorno propias del problema a resolver. Se observa en la siguiente imagen que vuelve a aparecer la rutina `timing` calculando el tiempo que se tarda en añadir las cargas puntuales.

```
% Add point loads conditions to the force vector

for i = 1 : size(pointload,1)
    ieqn = (pointload(i,1)-1)*dofpn+pointload(i,2);    % Finds eq.
                                                       number
    force(ieqn) = force(ieqn) + pointload(i,3);      % add the
                                                       force
end

ttim = timing('Time for apply side and point load',ttim);
%Reporting time
```

Fig. 23 Composite_Beam_Timoshenko

Las cargas puntuales se añaden en el vector de fuerzas nodales, teniendo en cuenta si se trata de una carga horizontal, vertical o un momento. Es decir, es necesario definir un ciclo con el número de cargas contenidas en la variable `pointload` y encontrar para cada una de ellas el número de ecuación asociado (`ieqn`) necesario para agregar el valor de la carga al vector `force`.

```
% Applies the Dirichlet conditions and adjust the right hand side.

for i = 1 : size(fixnodes,1)
    ieqn = (fixnodes(i,1)-1)*dofpn+fixnodes(i,2) ;    % Finds eq.
                                                       number
    u (ieqn) = fixnodes(i,3);                        % and store
                                                       the solution
                                                       in u and mark
    fix(i) = ieqn ;                                  % the eq as a
                                                       fix value
end

force = force - StifMat * u ;    % adjust the rhs with the
                                known values
```

Fig. 24 Composite_Beam_Timoshenko

En la figura 24 se observa el ciclo sobre los grados de libertad prescritos, definidos como el número de filas de la variable *fixnodes*, y como sus valores son asignados al vector de movimientos *u*. Además, se define un vector *fix* que contiene el número de las ecuaciones de aquellos grados de libertad que se encuentran restringidos.

Finalmente se actualiza el vector *force* de cargas nodales equivalentes multiplicando la matriz de rigidez global por el vector de movimientos que, en este momento, solo contiene los valores prescritos.

Ahora ya estamos en disposición de calcular los movimientos no prescritos, esto se lleva a cabo utilizando el vector *FreeNodes* que contiene la lista de aquellas ecuaciones que hay que resolver.

```
% Compute the solution by solving StifMat * u = force for the
% remaining unknown values of u.

FreeNodes = setdiff ( 1:nndof, fix ); % Finds the free node list

u(FreeNodes) = StifMat(FreeNodes,FreeNodes) \ force(FreeNodes);

ttim = timing('Time to solve the stiffness matrix',ttim);
%Reporting time
```

Fig. 25 Composite_Beam_Timoshenko

El vector *FreeNodes* es utilizado como un índice de los grados de libertad desconocidos y nos permite escribir de manera sencilla la solución del sistema de ecuaciones, obteniendo así el vector de desplazamientos no prescritos. Además, vuelve a aparecer la función *timing* que, en este caso, calcula el tiempo computacional que necesita el programa para resolver el sistema de ecuaciones.

Los desplazamientos en las vigas compuestas, como vimos en el capítulo donde se explicó la teoría de este tipo de estructuras, dependen de la coordenada vertical *z*. Además, las tensiones y esfuerzos también varían a lo largo del espesor en función de la capa en la que nos encontramos. Ésta dependencia de las capas de la viga es el punto en el que radica la principal diferencia con las vigas homogéneas.

Por tanto, es necesario crear un nuevo script con la función que calcule los desplazamientos (*Dsp_lyr*), tensiones (*Str_lyr*) y esfuerzos (*ResStr_lyr*) en función de la capa. A

ésta nueva subrutina, a la que se le ha dado el nombre de *Composite_Layers_Beam_Timoshenko*, se le llama desde el script actual como se observa en la siguiente captura de pantalla:

```
% Compute the layer's results

[Dsp_lyr Str_lyr ResStr_lyr] =
Composite_Layers_Beam_Timoshenko(layers, young, shear, poiss, kz, width, t
thickness, coord_ZG, coord_zp, u);

ttim = timing('Time to solve the layer results', ttim);
%Reporting time
```

Fig. 26 Composite_Beam_Timoshenko

En el siguiente apartado se describe en detalle esta función, que calcula los resultados según las capas de la viga.

Finalmente, se vuelve a llamar a una nueva subrutina donde se procede a volcar los valores de desplazamientos, tensiones y esfuerzos a los ficheros de postproceso, que GiD utilizará para representar gráficamente los resultados. Este nuevo script se denomina *ToGiD_Composite_Beam_Timoshenko* y se comentará más adelante. En la siguiente imagen aparece la llamada que se le hace a esta subrutina desde el archivo actual.

```
% Graphic representation

ToGiD_Composite_Beam_Timoshenko(file_name, layers, coord_ZG, u, Dsp_lyr,
Str_lyr, ResStr_lyr);

ttim = timing('Time used to write the solution', ttim);
%Reporting time
itim = toc; % Close last tic

fprintf(1, '\n Total running time %12.6f \n', ttim);
%Reporting final time
```

Fig. 27 Composite_Beam_Timoshenko

En las dos capturas de pantalla anteriores se observa que se vuelve a calcular el tiempo entre dos puntos del código mediante la subrutina *timing*, en el primer caso se obtiene el tiempo que tarda el programa en calcular los resultados en función de las capas y, en el

segundo, calcula el tiempo necesario para escribir la solución. En éste último caso, además, se para el cronómetro y se presenta el tiempo consumido en cada una de las fases descritas con anterioridad y el tiempo total que se ha necesitado para calcular el programa.

Composite Layers Beam Timoshenko.m

En esta subrutina, como comentábamos con anterioridad, se calcula los desplazamientos, tensiones y esfuerzos a lo largo del espesor de la viga. El cálculo de estas variables es necesario ya que la presentación de resultados se visualizará en dos dimensiones, con el objetivo de observar gráficamente el comportamiento estructural de cada capa.

Los desplazamientos se evalúan, en dirección horizontal, en los nodos de la malla en que se discretiza la viga y, en dirección vertical, se calcula en los puntos superior e inferior de cada capa. Esto es los puntos de interface entre capas añadiendo el punto inferior y superior de la sección de la viga.

Tanto en el cálculo de las tensiones como de los esfuerzos aparece la matriz de deformaciones B, la cual se calcula en los puntos de integración, por lo que la evaluación de las tensiones y esfuerzos se refieren a dichos puntos. La diferencia se encuentra en que en el caso de las tensiones se transfieren los resultados a los nodos y en cambio en los esfuerzos se evalúan en los puntos de Gauss. En dirección transversal, las tensiones se calculan en los puntos superior e inferior de cada capa mientras que los esfuerzos se evalúan en el punto intermedio de cada una, ya que según la teoría de Timoshenko para vigas compuestas, los esfuerzos son constantes en cada una de las capas.

En las siguientes imágenes se observa la subrutina que calcula los desplazamientos, tensiones y deformaciones..

```

function [Dsp_lyr Str_lyr ResStr_lyr] =
Composite_Layers_Beam_Timoshenko(layers, young, shear, poiss, kz, width,
thickness, coord_ZG, coord_zp, u)

global coordinates;
global elements;

nfnod = size(coordinates,1);           % Number of nodes
nelem = size(elements,1);             % Number of elements
nnode = size(elements,2);            % Number of nodes por element
dofpn = 3;                            % Number of DOF per node
dofpe = nnode*dofpn;                 % Number of DOF per element
nndof = nfnod*dofpn;                 % Number of total DO
eqnum = zeros(dofpe);                % Equation number list
Dsp_lyr = zeros(nfnod,3, layers+1);   % Create array for
                                     displacements
Str_lyr = zeros(nfnod,3,2*layers);    % Create array for stresses
ResStr_lyr = zeros(nelem,4, layers);  % Create array for resultant
                                     stresses

```

Fig. 28 Composite_Layers_Beam_Timoshenko

En la figura anterior se observa que la rutina *Composite_Layers_Beam_Timoshenko* se inicia indicando los parámetros de entrada, que son el número de capas de la viga (*layers*), el módulo de Young (*Young*), el módulo de rigidez (*shear*), el coeficiente de Poisson (*poiss*), el parámetro de forma (*kz*), el ancho y espesor de cada capa (*width* y *thickness*), el vector de coordenadas vertical, medido respecto al centroide de la sección, de la fibra inferior y superior de cada capa (*coord_ZG*), este vector respecto al eje neutro (*coord_zp*) y, por último, los desplazamientos nodales (*u*). Adicionalmente se utilizarán las coordenadas nodales *coordinates* y las conectividades *elements* que han sido definidas como variables globales.

Después se indican las variables necesarias para programar la rutina. Algunas de ellas son las mismas que se utilizaban en el script principal (figura 9). La variable *eqnum* vuelve a ser el vector reservado para indicar el número de ecuaciones en numeración global y, además, se crean tres variables nuevas que son:

- ✓ *Dsp_lyr* = zeros(*nfnod*,3,*layers*+1): son las matrices de desplazamientos donde *layers*+1 se refiere al número de matrices, *nfnod* es el número de filas de cada matriz y 3 es el número de columnas (u_o, w_o, θ). El número de matrices corresponde con el número de puntos en sección transversal donde calcularemos los desplazamientos.
- ✓ *Str_lyr* = zeros(*nfnod*,3,2**layers*): matriz de tensiones donde cada una de las

componentes tienen el mismo significado que las anteriormente comentadas. Se hace notar que ahora el número de matrices es el doble que el número de capas ya que se evalúan las tensiones en la fibra inferior y superior de cada una de ellas. Además, cada una de las columnas son σ_x, τ_{xz} y la tercera es necesaria para obtener las tensiones suavizadas en los nodos.

- ✓ ResStr_lyr = zeros(nelem,3,layers): matriz de esfuerzos N, Q y M_x . Los esfuerzos en dirección transversal se evalúan en el punto medio de cada capa, con lo que el número de matrices reservadas será *layers* y, en dirección horizontal, se evalúan en el punto medio de cada elemento, es decir, *nelem* será el número de filas de cada matriz.

El orden en que se ha descrito cada una de las tres variables anteriores será el que se seguirá en la rutina para calcularlos. Empecemos, por tanto, por obtener los desplazamientos.

```

%% Layer Displacements
% Finds the equation number list for the i-th node

dof_1=zeros(npnod,1); %dof_1 belong to the axial displacement
dof_2=zeros(npnod,1); %dof_2 belong to the vertical displacement
dof_3=zeros(npnod,1); %dof_3 belong to the rotation

for node = 1:npnod
    dof_1(node)=(node-1)*dofpn + 1;
    dof_2(node)=(node-1)*dofpn + 2;
    dof_3(node)=(node-1)*dofpn + 3;
end

% Displacement field

for lyr = 1:layers+1
    Dsp_lyr(:,1,lyr) = u(dof_1) - coord_zp(lyr)*u(dof_3);
    Dsp_lyr(:,2,lyr) = u(dof_2);
    Dsp_lyr(:,3,lyr) = u(dof_3);
end

```

Fig. 29 Composite_Layers_Beam_Timoshenko

La forma de proceder tal y como podemos observar en la figura 29 es, en primer lugar, definir los grados de libertad dof_1, dof_2 y dof_3 correspondientes a cada desplazamiento u_o, w_o y θ respectivamente. Estas matrices tendrán tantas filas como número de nodos tenga la malla en que se discretiza la viga. Después, se indica la ubicación que tienen en el vector de desplazamientos global dependiendo del número de nodo.

Finalmente se calculan las matrices de desplazamientos teniendo en cuenta los

distintos materiales en dirección transversal. Se obtiene según la teoría de vigas compuestas de Timoshenko, descrita en la expresión (2). Se hace notar que la coordenada vertical está referenciada al eje neutro de la viga, de manera que no haya acoplamiento axil-flector tal y como comentábamos en capítulos anteriores.

Una vez obtenemos el campo de desplazamientos de la viga se procede a calcular las tensiones y los esfuerzos. Para ello es necesario definir las componentes de la matriz constitutiva que intervienen en el cálculo de tensiones y de esfuerzos.

En el caso de las tensiones, recordamos que según la teoría de vigas compuestas de Timoshenko:

$$\sigma_x = E \varepsilon_x = E \cdot \left(\frac{\partial u_0}{\partial x} - z' \frac{\partial \theta}{\partial x} \right) = E \frac{\partial u_0}{\partial x} - E z' \frac{\partial \theta}{\partial x} \quad (65)$$

$$\tau_{xz} = G \gamma_{xz} = G \left(\frac{\partial w_0}{\partial x} - \theta \right) \quad (66)$$

donde E y G son el módulo de Young y de rigidez, respectivamente, z' es la coordenada vertical respecto al eje neutro, y u₀, w₀ y θ son el desplazamiento horizontal, vertical y el giro que se producen en la estructura. Además, ε_x, γ_{xz}, σ_x y τ_{xz} son la deformación normal y tangencial y sus respectivas tensiones.

$$\hat{\varepsilon} = \begin{Bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta \end{Bmatrix} = \mathbf{B}_i \boldsymbol{\alpha}_i^{(e)} = \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_b \\ \mathbf{B}_s \end{bmatrix}_i \cdot \begin{bmatrix} u_0 \\ w_0 \\ \theta \end{bmatrix}_i ; \quad i = 1,2 \quad (67)$$

donde **B_a**, **B_b**, **B_s** son la componente axil, flexión y cortante de la matriz de deformaciones. El subíndice *i* se refiere al número de nodo.

Sustituyendo la expresión (67) en las ecuaciones (65) y (66), se obtiene que las tensiones en cada nodo son:

$$\sigma_x = E \mathbf{B}_a \mathbf{u}^T - E z' \mathbf{B}_b \mathbf{u}^T$$

$$\tau_{xz} = G \mathbf{B}_s \mathbf{u}^T \quad (68)$$

En la expresión (68) se distinguen los parámetros propios del material (\cdot) que afectan al cálculo de las tensiones y es lo siguiente que se programa:

```

%% Layer Stresses and Resultant Stresses

Da = zeros(2*layers,1);
Db = zeros(2*layers,1);
Ds = zeros(2*layers,1);

for i=1:layers
    Da(2*(i-1)+1) = young(i);
    Da(2*(i-1)+2) = young(i);
    Db(2*(i-1)+1) = -young(i)*coord_zp(i);
    Db(2*(i-1)+2) = -young(i)*coord_zp(i+1);
    Ds(2*(i-1)+1) = kz*shear(i);
    Ds(2*(i-1)+2) = kz*shear(i);
end
    
```

Fig. 30 Composite_Layers_Beam_Timoshenko

Se observa que el cálculo de cada parámetro se realiza en la fibra inferior y superior de cada capa, debido a que es en esos puntos donde se evalúan las tensiones, por lo que se obtiene dos veces por capa las características del material.

Por lo que se refiere a las tensiones generalizadas (esfuerzos), según la expresión (8) $\hat{\sigma} = \tilde{D}\hat{\epsilon}$ y (67) se demuestra que:

$$N = \tilde{D}_a B_a u^T \quad ; \quad Q = \tilde{D}_b B_b u^T \quad ; \quad M = \tilde{D}_s B_s u^T \quad (69)$$

con lo que

```

Da_lyr = zeros(layers,1);
Db_lyr = zeros(layers,1);
Dab_lyr = zeros(layers,1);
Ds_lyr = zeros(layers,1);

for i=1:layers
    Da_lyr(i) = width(i) * thickness (i) * young(i);
    Db_lyr(i) = (1/3)*((coord_ZG(i+1))^3-
(coord_ZG(i))^3)*width(i)*young(i);
    Ds_lyr(i)= thickness(i)*width(i)*(young(i)/(2*(1+poiss(i))));
end

Ds_lyr=kz*Ds_lyr;

```

Fig. 31 Composite_Layers_Beam_Timoshenko

Hay que destacar dos cosas de la imagen anterior, por un lado y a diferencia de las tensiones, los esfuerzos se evalúan una vez en cada capa ya que los esfuerzos se obtienen en el centro de éstas. Por otro lado, en el cálculo de la componente a flexión `Db_lyr` aparece el vector de coordenadas `z` respecto al centro de gravedad y no respecto al eje neutro, se recuerda que este parámetro da el mismo resultado calculado respecto a un eje u otro.

Una vez hemos definido estos parámetros iniciales se realiza el ciclo elemental que permite obtener las tensiones y esfuerzos de la viga.

```

% Element cycle.
for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode);

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem. X
                                                    coordinate

    % Finds the equation number list for the i-th element

    for i=1:nnode
        ii = (i-1)*dofpn;
        for j =1:dofpn
            eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build the equation
                                                    number list
        end
    end
end

```

Fig. 32 Composite_Layers_Beam_Timoshenko

En el inicio del ciclo se realizan las mismas operaciones que hacíamos en la figura 18, de manera que llamamos a las conectividades entre elementos y a las coordenadas de los nodos. También se construye el vector de posición que indica el número de las ecuaciones.

A continuación, recuperamos el vector desplazamientos y se calcula la longitud de los elementos.

```
% Recover the nodal displacements for the i-th element  
  
u_elem(1:dofpe)=u(eqnum(1:dofpe));  
  
len = coor_x(2) - coor_x(1); % x_j > x_i
```

Fig. 33 Composite_Layers_Beam_Timoshenko

En la siguiente figura se indica que el punto de Gauss se evalúa en el 0.0 y se presentan las componentes de la matriz de deformaciones.

```
% One gauss point for stress evaluation  
  
gaus0 = 0.0; % One Gauss point for stresses evaluation  
  
bmat_a=[ -1/len, 0, 0, 1/len, 0, 0 ];  
  
bmat_b=[0, 0, -1/len, 0, 0, 1/len];  
  
bmat_s=[0, -1/len, -1/2, 0, 1/len, -1/2];
```

Fig. 34 Composite_Layers_Beam_Timoshenko

donde $bmat_a = B_a$, $b_matb = B_b$ y $b_mats = B_s$, componentes de la matriz de deformaciones. Para entender la rutina, que se muestra en la figura anterior, que define las componentes de la matriz de deformaciones tenemos que hacer referencia a la siguiente expresión donde se describe el valor de esta matriz:

$$\mathbf{B} = \begin{bmatrix} \frac{(-1)^i}{l} & 0 & 0 \\ 0 & 0 & \frac{(-1)^i}{l} \\ 0 & \frac{(-1)^i}{l} & -\frac{1}{2} \end{bmatrix} ; i = 1,2 \quad (70)$$

donde l es la longitud del elemento y el parámetro i se refiere al número de nodo. La primera fila corresponde a la componente axial (B_a), la segunda a la influencia de la flexión (B_b) y la última a la componente a cortante (B_s). Se hace notar, pues, que las matrices de la figura 34 tienen dimensiones 1x6 debido a que las primeras tres componentes se refieren al nodo 1 del elemento y las últimas tres al nodo 2.

Siguiendo las ecuaciones de la expresión (68) se programa la rutina que calcula las tensiones:

```
Str1_g0 = (Da*bmat_a + Db*bmat_b)*transpose(u_elem);
Str2_g0 = (Ds*bmat_s)*transpose(u_elem);

Str_lyr(lnods(1),1,:) = Str_lyr(lnods(1),1,:)+
permute(Str1_g0,[2 3 1]);

Str_lyr(lnods(2),1,:) = Str_lyr(lnods(2),1,:)+
permute(Str1_g0,[2 3 1]);

Str_lyr(lnods(1),2,:) = Str_lyr(lnods(1),2,:)+
permute(Str2_g0,[2 3 1]);

Str_lyr(lnods(2),2,:) = Str_lyr(lnods(2),2,:)+
permute(Str2_g0,[2 3 1]);

Str_lyr(lnods(1),3,:) = Str_lyr(lnods(1),3,:)+1;
Str_lyr(lnods(2),3,:) = Str_lyr(lnods(2),3,:)+1;
```

Fig. 35 Composite_Layers_Beam_Timoshenko

En la imagen anterior, la tensión normal σ_x es la variable Str1_g0 y la tensión tangencial τ_{xz} es la denominada Str2_g0. Estas tensiones se evalúan en los puntos de Gauss, en el centro del elemento, sin embargo se realiza una translación hacia los nodos. Es importante observar la necesidad de realizar un cambio en las dimensiones de la matriz Str1_g0 y Str2_g0 para encajar sus componentes, que están dispuestas en una sola matriz, en las diferentes matrices de la variable Str_lyr. Esto último se realiza mediante la herramienta

permute (que cambia el orden en que están dispuestas las filas, columnas y el número de matrices).

Se sigue el mismo proceso para obtener las tensiones generalizadas, esta vez siguiendo la expresión (69):

```
ResStr1_g0 = (Da_lyr*bmat_a) *transpose(u_elem);  
ResStr2_g0 = (Db_lyr*bmat_b)*transpose(u_elem);  
ResStr3_g0 = (Ds_lyr*bmat_s)*transpose(u_elem);  
  
ResStr_lyr(ielem,1,:)=permute(ResStr1_g0,[2 3 1]);  
ResStr_lyr(ielem,2,:)=permute(ResStr2_g0,[2 3 1]);  
ResStr_lyr(ielem,3,:)=permute(ResStr3_g0,[2 3 1]);  
  
end % End element cycle
```

Fig. 36 Composite_Layers_Beam_Timoshenko

En este caso los esfuerzos N, Q y M corresponden a las variables ResStr1_g0, ResStr2_g0 y ResStr3_g0 respectivamente. Como hacíamos anteriormente, se realiza el cambio de dimensiones para formar las matrices de esfuerzos ResStr_lyr. Sin embargo, se hace notar que no se realiza ninguna translación hacia los nodos ya que los esfuerzos se evalúan en los puntos de Gauss directamente. Una vez calculadas las tensiones y esfuerzos de toda la viga se cierra el ciclo elemental.

Finalmente, después de haber calculado las tensiones y los esfuerzos en todos los elementos es necesario realizar una suavización nodal. Debido a que los nodos interiores pertenecen a varios elementos al mismo tiempo, nos encontramos con que hay dos tensiones calculadas para un mismo nodo. La forma de encontrar un único valor es realizar este suavizado nodal a través de calcular la media de estas tensiones. Por ello se construye la acumulación de las tensiones en la tercera columna de la variable Str_lyr, como se observa en la figura 35, para posteriormente realizar la media de los valores que confluyen en el nodo en cuestión, tal y como vemos en la siguiente imagen:

```

for i = 1 : npnod
    Str_lyr(i,1,:) = Str_lyr(i,1,:)./Str_lyr(i,3,:);
    Str_lyr(i,2,:) = Str_lyr(i,2,:)./Str_lyr(i,3,:);
end

```

Fig. 37 Composite_Layers_Beam_Timoshenko

ToGid_Composite_Beam_Timoshenko

Finalmente procedemos a explicar el último script necesario para la programación de vigas compuestas en MATLAB. Como decíamos con anterioridad, una vez tenemos calculados los desplazamientos nodales, las tensiones y los esfuerzos se procede a volcar estos valores a los ficheros de postproceso de los que se alimentará GiD para graficar los resultados. Esto se realiza mediante la subrutina *ToGid_Composite_Beam_Timoshenko*, cuya llamada desde el script principal se mostraba en la figura 27.

Esta subrutina empieza como la anterior, indicando los parámetros de entrada: el fichero de datos (*file_name*), el número de capas (*layers*), la coordenada vertical respecto al centro de gravedad (*coord_ZG*), el vector de desplazamientos nodal (*u*), la matriz de desplazamientos según las capas (*Dsp_lyr*), la matriz de tensiones y la de esfuerzos (*Str_lyr* y *ResStr_lyr*).

```

function
ToGid_Composite_Beam_Timoshenko(file_name, layers, coord_ZG, u, Dsp_lyr,
Str_lyr, ResStr_lyr)
%
% Parameters:
%
% Input, file_name : GiD File name
% layers : number of layers
% coord_ZG : coordinate-z respect the gravity center
% u : Nodal displacements
% Dsp_lyr : array for displacements
% Str_lyr : array for stresses
% ResStr_lyr: array for resultant stresses
%
% Output, none
%

```

Fig. 38 ToGid_Composite_Beam_Timoshenko

Después se hace referencia a las variables que se usarán, a las coordenadas globales de la malla y a las conectividades entre elementos. Además, como la visualización de resultados será en dos dimensiones, los elementos de dos nodos de viga se convertirán en elementos de cuatro nodos, cuadriláteros, con lo que se definen este tipo de elementos (*qeletyp*). Por último, se indican los dos archivos que se obtendrán al correr el programa, el archivo donde se indica las características de la malla (*msh_file*) y el archivo de resultados (*res_file*).

```
global coordinates;
global elements;

nelem = size(elements,1);           % Number of elements
npnod = size(coordinates,1);       % Number of nodes

qeletyp = 'Quadrilateral';

msh_file = strcat(file_name, '.flavia.msh');
res_file = strcat(file_name, '.flavia.res');
```

Fig. 39 ToGid_Composite_Beam_Timoshenko

Se tiene que formar la nueva malla de elementos cuadriláteros a partir de las coordenadas de los nodos en una dimensión. Antes de esto se describen nuevas variables que serán necesarias para definir la nueva malla: *qnelem* es el número de elementos cuadriláteros, *qnode* es el número de nodos por elementos, es decir 4, y por último se define *qnpnode* como el número total de nodos.

```

%% Mesh with quadrilateral elements (2D beam)
qnelem = nelem*layers;           % Number of quadrilateral elements
qnode = 4;                       % Number of nodes per
                                % quadrilateral elements
qnpnod = npnod*layers*2;        % Number of nodes

% Coordinates matrix of quadrilateral elements (2D)

qcoordinates = zeros(qnpnod,2);
for i=1:layers
    for j=1:npnod
        qcoordinates((2*i-2)*npnod+j,:)=[coordinates(j)
coord_ZG(i)];
        qcoordinates((2*i-1)*npnod+j,:)=[coordinates(j)
coord_ZG(i+1)];
    end
end

```

Fig.40 ToGid_Composite_Beam_Timoshenko

Una vez se obtienen las coordenadas de los nodos de la estructura en 2D, se crean las matrices de elementos y las conectividades entre ellos.

```

% Elements matrix of quadrilateral elements (2D)

qelements=zeros(qnelem,qnode+1);
for i = 1:layers
    for j = 1 : nelem
        qelements((i-1)*nelem+j,:) = [((2*i-2)*npnod+
elements(j,:))...
((2*i-1)*npnod + elements(j,end:-1:1))  i];
    end
end
% Conectivity matrix by layer (2D)

Nod_lyr = zeros(npnod, layers, 2);
Elem_lyr = zeros(nelem, layers);
for i = 1 : layers
    for j = 1 : npnod
        Nod_lyr(j,i,1) = (2*i-2)*npnod + j;
        Nod_lyr(j,i,2) = (2*i-1)*npnod + j;
    end
    for n = 1 : nelem
        Elem_lyr(n,i) = (i-1)*nelem + n;
    end
end

```

Fig.41 ToGid_Composite_Beam_Timoshenko

La variable *qelements* describe una matriz donde las filas indican el elemento al que se hace referencia y las columnas los nodos que lo forman, en la última columna se indica a que

capa pertenece. En cuanto a las variables *Nod_lyr* y *Elem_lyr* son las matrices cuyas componentes se refieren a la numeración de los nodos, en el caso de *Nod_lyr*, que servirán de ubicación para los resultados de desplazamientos y tensiones y, en el caso de *Elem_lyr* se muestra la numeración de los elementos, que servirán para posicionar los esfuerzos.

El siguiente paso es generar el archivo con las características de la malla (.msh) y la representación de ésta en dos dimensiones.

```

%%Generates the mesh file

fid = fopen(msh_file,'w');
fprintf(fid, '### \n');
fprintf(fid, '# MAT_FEM Compiste Laminated TBT \n');
fprintf(fid, '# \n');

% Generates the mesh in 2D

fprintf(fid, 'Group "Thickness_Analysis_2D" \n');
fprintf(fid, 'MESH "2D_beam" dimension %3.0f  Elemtyp %s  Nnode
%2.0f \n \n', 2, qeletyp, qnnode);
fprintf(fid, 'coordinates \n');
for i = 1 : qnnode
    fprintf(fid, '%6.0f %12.5d %12.5d
\n', i, qcoordinates(i,1), qcoordinates(i,2));
end
fprintf(fid, 'end coordinates \n \n');
fprintf(fid, 'elements \n');
for i = 1 : qelem
    fprintf(fid, '%6.0f %6.0f %6.0f  %6.0f %6.0f %3.0f
\n', i, qelements(i,:));
end
fprintf(fid, 'end elements \n \n');
fprintf(fid, 'end group \n \n');

status = fclose(fid);
    
```

Fig.42 ToGid_Composite_Beam_Timoshenko

El fichero anterior muestra, en primer lugar, un listado de todos los nodos de la malla con sus coordenadas *x* y *z*. En segundo lugar se representa las conectividades de los nodos formando los elementos cuadriláteros, indicando el número de elemento y la capa a la que pertenece.

A continuación, se genera el archivo de resultados (.res) y se representan todos ellos.

```

%% Generates the result file

fid = fopen(res_file,'w');
fprintf(fid,'Gid Post Results File 1.0 \n');
fprintf(fid,'### \n');
fprintf(fid,'# MAT_FEM Composite Laminated TBT \n');
fprintf(fid,'# \n');
fprintf(fid,'GaussPoints "GP_quad" Elemtyp %s "2D_beam"
\n',qeletyp);
fprintf(fid,'Number of Gauss Points: 1 \n');
fprintf(fid,'Natural Coordinates: Internal \n');
fprintf(fid,'end gausspionts \n');
fprintf(fid,'# \n');

```

Fig.43 ToGid_Composite_Beam_Timoshenko

```

%% Thickness results along each layer

fprintf(fid,'OnGroup "Thickness_Analysis_2D" \n');
fprintf(fid,'Result "Displacements" "Load Analysis" 1 Vector
OnNodes \n');
fprintf(fid,'ComponentNames "X-Displ", "Y-Displ", "Z-Displ" \n');
fprintf(fid,'Values \n');
for i=1:layers
    for j = 1 : npnod
        fprintf(fid,'%6.0i %13.5d %13.5d 0.0
\n',Nod_lyr(j,i,1),full(Dsp_lyr(j,1,i)),full(Dsp_lyr(j,2,i)));
        fprintf(fid,'%6.0i %13.5d %13.5d 0.0
\n',Nod_lyr(j,i,2),full(Dsp_lyr(j,1,i+1)),full(Dsp_lyr(j,2,i+1)));
    end
end
fprintf(fid,'End Values \n');
fprintf(fid,'# \n');

fprintf(fid,'Result "Rotation" "Load Analysis" 1 Vector
OnNodes \n');
fprintf(fid,'ComponentNames "X-Rot", "Y-Rot", "Z-Rot" \n');
fprintf(fid,'Values \n');
for i=1:layers
    for j = 1 : npnod
        fprintf(fid,'%6.0i 0.0 0.0 %13.5d
\n',Nod_lyr(j,i,1),full(Dsp_lyr(j,3,i)));
        fprintf(fid,'%6.0i 0.0 0.0 %13.5d
\n',Nod_lyr(j,i,2),full(Dsp_lyr(j,3,i+1)));
    end
end
fprintf(fid,'End Values \n');
fprintf(fid,'# \n');

```

Fig.44 ToGid_Composite_Beam_Timoshenko

La presentación de resultados se organiza indicando primero los desplazamientos y después las tensiones y esfuerzos. Se puede observar en la imagen anterior como la manera de proceder siempre es la misma, primero se indica cuál es el resultado a presentar, se inicia un ciclo por cada una de las capas de la viga y se representan los valores de la matriz `Dsp_lyr` en el nodo de la matriz que le corresponde `Nod_lyr`, teniendo en cuenta que cada columna se refiere a un desplazamiento o al giro.

```
fprintf(fid, 'Result "Stresses" "Load Analysis" 1 Vector OnNodes
\n');
  fprintf(fid, 'ComponentNames "Sigma_x", "Tau_xz", "zero" \n');
  fprintf(fid, 'Values \n');
  for i=1:layers
    for j = 1 : npnod
      fprintf(fid, '%6.0i %12.5d %12.5d 0.0
\n', Nod_lyr(j,i,1), Str_lyr(j,1, (i-1)*2+1), Str_lyr(j,2, (i-1)*2+1));
      fprintf(fid, '%6.0i %12.5d %12.5d 0.0
\n', Nod_lyr(j,i,2), Str_lyr(j,1, (i-1)*2+2), Str_lyr(j,2, (i-1)*2+2));
    end
  end
  fprintf(fid, 'End Values \n');
  fprintf(fid, '# \n');

  fprintf(fid, 'Result "Resultant_Stresses" "Load Analysis" 1 Vector
OnGaussPoints "GP_quad" \n');
  fprintf(fid, 'ComponentNames "Nx", "Qy", "Mz" \n');
  fprintf(fid, 'Values \n');
  for i=1:layers
    for j = 1 : nelem
      fprintf(fid, '%6.0i %12.5d %12.5d %12.5d
\n', Elem_lyr(j,i), ResStr_lyr(j,1,i), ResStr_lyr(j,3,i),
ResStr_lyr(j,2,i));
    end
  end
  fprintf(fid, 'End Values \n');
  fprintf(fid, 'End ongroup \n');
  fprintf(fid, '# \n');

  status = fclose(fid);
```

Fig.45 ToGid_Composite_Beam_Timoshenko

Después de los desplazamientos se representan tensiones y esfuerzos. El esquema es el mismo, únicamente se tiene en cuenta que los esfuerzos se evalúan en los elementos, no en los nodos como todas las variables anteriores, con lo que los valores `ResStr_lyr` se posicionan en las componentes de la matriz `Elem_lyr`.

4.3.2 Interfaz implementada en GiD

Una vez se ha programado en MATLAB las rutinas necesarias para el cálculo de vigas compuestas de Timoshenko, es momento de implementar la interfaz necesaria dentro de GiD que sea capaz de modelar este tipo de vigas y podamos definir materiales, condiciones de contorno y de más características propias de las vigas compuestas.

En el presente apartado se va a describir diferentes ficheros que son necesarios para crear el nuevo módulo necesario para el análisis de vigas compuestas. Esta colección de ficheros forma el denominado **Problem Type**. Esta carpeta debe tener la extensión .gid y debe guardarse en la carpeta de Problem Types dentro del directorio donde se guardó el programa GiD, en nuestro caso hemos llamado a esta carpeta MAT-fem_Composite_Beams.gid.

Estos ficheros generan las condiciones y las propiedades del material, así como la información del problema que será transferida a la malla en cuestión, al mismo tiempo proporcionan la posibilidad de implementar los iconos y símbolos que representan algunas de las condiciones del problema.

Los diferentes ficheros tienen el mismo nombre que el Problem Type y varía su extensión dependiendo de la función que realice:

- | | | |
|------|------------------------------------|---|
| i. | <i>MAT-fem_Composite_Beams.cnd</i> | Define las condiciones del problema en cuestión. |
| ii. | <i>MAT-fem_Composite_Beams.prb</i> | Información interna del módulo. |
| iii. | <i>MAT-fem_Composite_Beams.mat</i> | Define las propiedades de los materiales. |
| iv. | <i>MAT-fem_Composite_Beams.uni</i> | Sistema de unidades. |
| v. | <i>MAT-fem_Composite_Beams.sim</i> | Símbolos de las distintas condiciones. |
| vi. | <i>MAT-fem_Composite_Beams.bas</i> | Información sobre el fichero de entrada de datos. |
| vii. | <i>MAT-fem_Composite_Beams.tcl</i> | Archivos con extensión tcl. |

Además, también hay una serie de ficheros .geo e imágenes .gif que corresponden a los distintos iconos utilizados por el módulo que aparecerán en el programa GiD para asignar las propiedades del material, condiciones de carga y de contorno, etc.

De aquí en adelante vamos a explicar cada uno de los ficheros anteriores haciendo referencia tanto a la programación numérica como a la visualización en GiD.. En el anejo 3 se adjunta toda la programación al detalle de todos los ficheros, sin embargo se mostrarán, en las explicaciones siguientes, una parte de las rutinas que sirvan para entender las ventanas gráficas que aparecen en el programa GiD.

MAT-fem Composite Beams.cnd

Este fichero contiene toda la información sobre las condiciones de contorno del problema, las cuales tienen que ser aplicadas a distintas entidades. Es importante tener en cuenta sobre qué tipo de entidades (puntos, líneas, superficies o volúmenes) se van a aplicar las condiciones y a qué tipo de entidades de la malla se transferirán (nodos o elementos). La información que se incluye en este fichero son las condiciones de apoyo y las condiciones de carga, que vamos a describir a continuación.

La forma de proceder del fichero .cnd, que se observa en la figura 46, es la siguiente. Se inicia especificando el tipo de condición que se quiere representar, después se define sobre qué entidades se aplican estas condiciones y a qué entidad se referirán en la malla posterior. A continuación se indica el grado de libertad a restringir, en el caso de condiciones de apoyo, y el valor que toma éste. En el caso de la condición de carga se indica la dirección y sentido de la fuerza y su valor. Para finalizar cada condición la rutina termina con un END CONDITION. En los siguientes ficheros también se podrá comprobar como después de cada rutina se finaliza con un END seguido del nombre que corresponda dependiendo del archivo.

En la siguiente imagen se presenta una parte de la programación de la rutina referida a las condiciones de apoyo acompañada de la ventana gráfica que aparecerá en el programa GiD.

BOOK: Displacement_Constraints
 NUMBER: 1 CONDITION: Point_Constraints
 CONDTYPE: over points
 CONDMESHTYPE: over nodes
 QUESTION:Horizontal_Constraint:#CB#(1,0)
 VALUE:1
 DEPENDENCIES: (0,SET,Displacement-X,0.0m)(#DEFAULT#,RESTORE,Displacement X,0.0m)
 QUESTION:Displacement-X#UNITS#
 VALUE:0.000m

Fig. 46 Restricción de desplazamientos

Es importante comentar que la orden que aparece en la figura anterior #CB#(1,0), indica que esta condición se representará mediante un “Combo Box”, lo cual significa que aparecerá la opción de restringir un grado de libertad, que podrá ser elegida (1) o no (0).

En cuanto a las condiciones de carga puntual o repartida:

NUMBER: 2 CONDITION: Point_Load
 CONDTYPE: over points
 CONDMESHTYPE: over nodes
 QUESTION: Horizontal-Force:#UNITS#
 VALUE: 0.0N
 QUESTION:Vertical-Force:#UNITS#
 VALUE: 0.0N
 QUESTION: Moment:#UNITS#
 VALUE: 0.0N*m
 END CONDITION

Fig. 47 Condiciones de cargas puntuales

La orden #UNITS# permite variar las unidades de la variable, el fichero donde se definen estas unidades lo veremos más adelante.

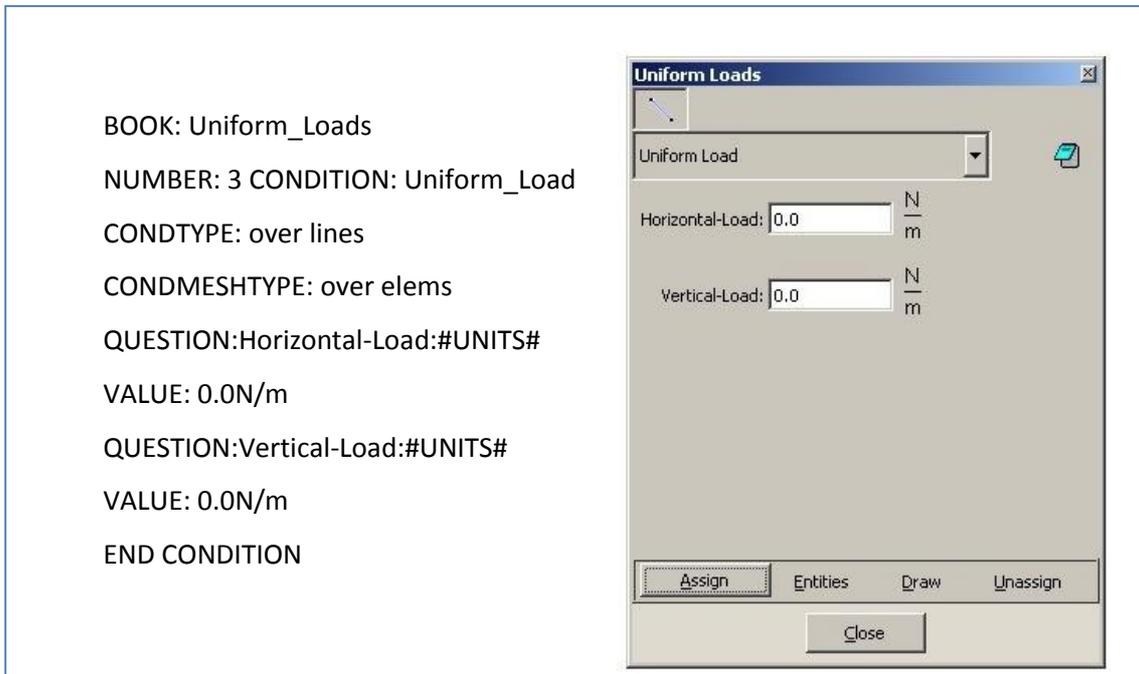


Fig. 48 Condiciones de carga distribuida

MAT-fem Composite Beams.prb

En este fichero de extensión .prb se incluye la información general del problema. Lo más importante es definir si se tiene en cuenta el peso propio de la viga o no, esto se hace creando un “Combo Box” como se hacía para el caso de los desplazamientos restringidos.

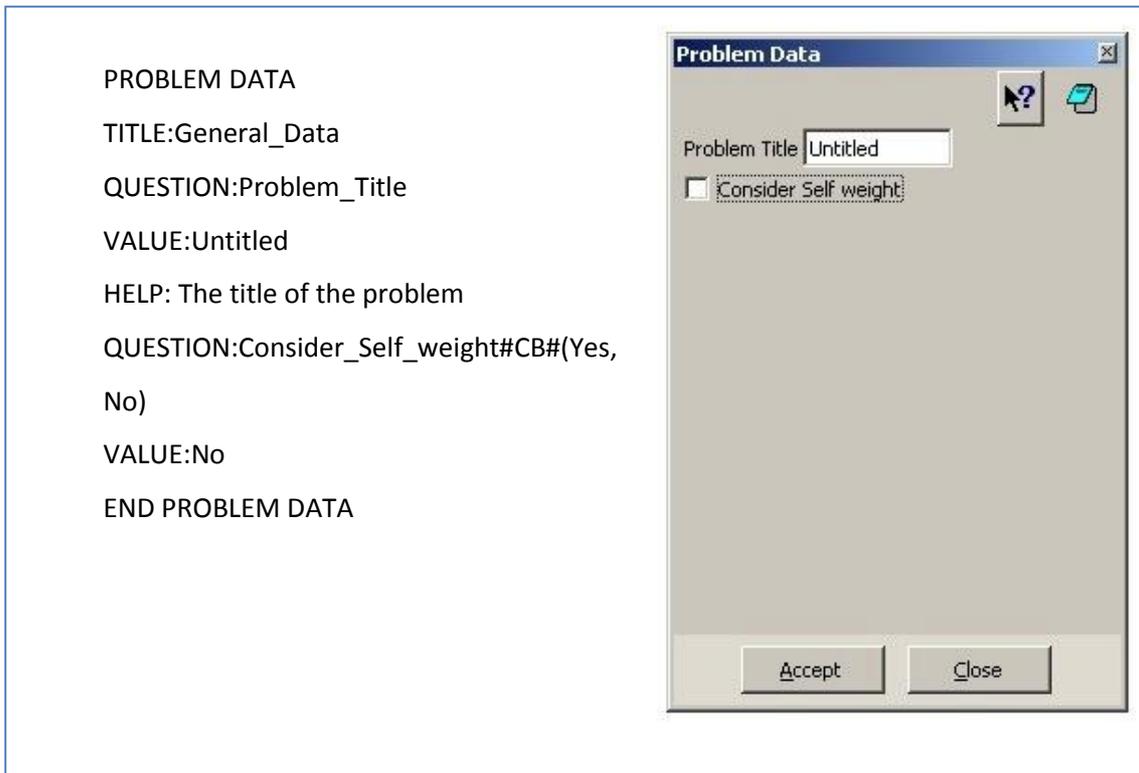


Fig. 49 Información interna del problema referente a considerar el peso propio de la estructura

MAT-fem Composite Beams.mat

Este archivo incluye la definición de los distintos materiales y sus propiedades. Hay programado un material inicial compuesto por tres capas con unas características definidas, sin embargo se puede modelar las características del material y el número de capas en función del problema a resolver.

En la siguiente figura se muestra la manera de presentar este fichero y viene acompañado por la ventana gráfica que aparecerá en el preproceso de GiD para definir los materiales.

```
NUMBER: 1 MATERIAL: Composite_Section
QUESTION: LAYER
VALUE: 3
QUESTION: YOUNG(LAYER,_____E__N/m^2____)
VALUE: ## 6 1 2.1e11 2 3.0e10 3 3.0e10
QUESTION: POISSON(LAYER,_____poiss_____)
VALUE: ## 6 1 0.2 2 0.2 3 0.2
QUESTION: SELF-WEIGHT(LAYER,_____poiss__N/m^3)
VALUE: ## 6 1 78000 2 25000 3 78000
QUESTION: THICKNESS(LAYER,_____h__m_____)
VALUE: ## 6 1 2 2 16 3 2
QUESTION: WIDTH(LAYER,_____b__m_____)
VALUE: ## 6 1 10 2 10 3 10
END MATERIAL
```

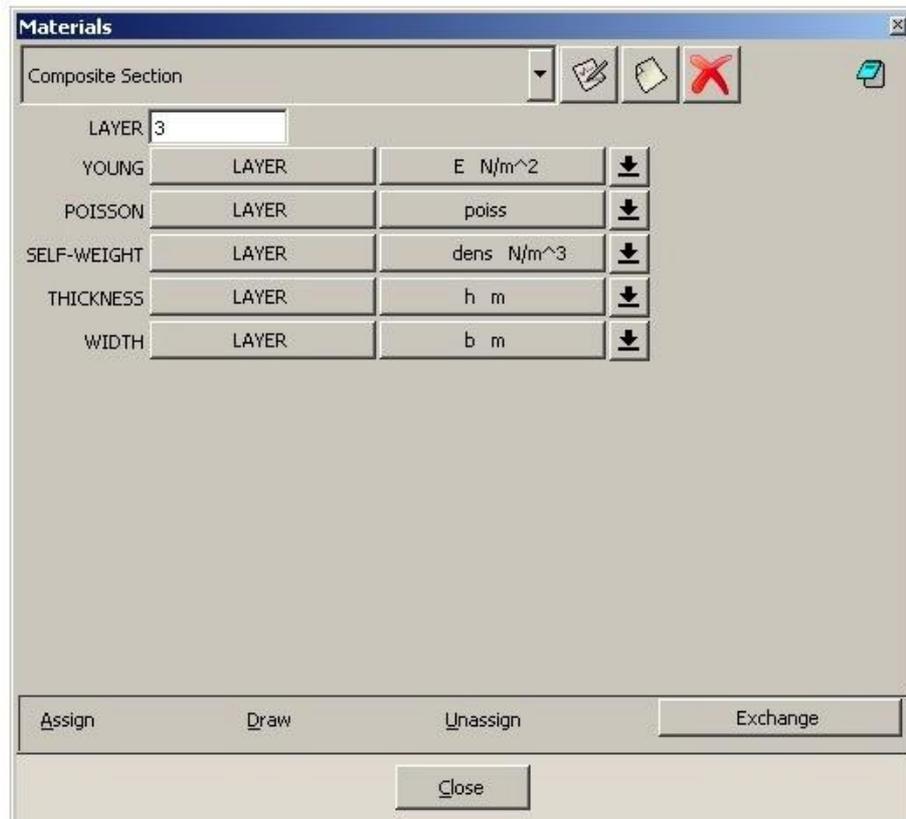


Fig. 50 Asignación de las propiedades del material según la capa

Se observa que se define la sección compuesta en función del número de capas (LAYER), el módulo de Young (YOUNG), el coeficiente de Poisson (POISS), la densidad (SELF-WEIGHT), el espesor (THICKNESS) y el ancho (WIDTH) de cada capa. Estas variables se presentan en forma de matriz donde las filas son el número de capas y las columnas se refieren a la capa en cuestión y al valor que toma la variable. Se puede aumentar el número de filas dependiendo del problema a resolver.

MAT-fem Composite Beams.uni

En este archivo se muestra una tabla con diferentes magnitudes y sus unidades de referencia. Además, se presenta una serie de unidades y el factor de conversión entre la unidad y la unidad de referencia.

```
BEGIN TABLE
LOAD : N {reference}, 1e-1 kgf, 1e-4 ton, 1e-3 kN, 1e-6 MN, 0.2248090247 lbf
MOMENT : N*m {reference}, 1e-1 kgf*m, 1e-4 ton*m, 1e-3 kN*m, 1e-6 MN*m,
8.850749004 lbf*in
UNIFORM_LOAD : N/m {reference}, 1e-1 kgf/m, 1e-4 ton/m, 1e-3 kN/m, 1e-6 MN/m,
0.571015e-2 lbf/in
AREA : m^2 {reference}, 1e+4 cm^2, 1e+6 mm^2, 1550 in^2
INERTIA : m^4 {reference}, 1e+8 cm^4, 1e+12 mm^4, 2402509.610 in^4
SELF_WEIGHT : N/m^3 {reference}, 1e-1 kgf/m^3, 1e-4 ton/m^3, 1e-3 kN/m^3, 1e-6
MN/m^3, 0.3683959876e-5 lbf/in^3
END
```

Fig. 51 Magnitudes y las unidades que pueden tomar

```
BEGIN SYSTEM(INTERNATIONAL)
LENGTH : m
ANGLE : rad
UNIFORM_LOAD : N/m
PRESSURE : N/m^2
LOAD : N
MOMENT : N*m
SELF_WEIGHT : N/m^3
AREA : m^2
INERTIA : m^4
END

USER DEFINED: DISABLED
```

Fig. 52 Unidades del Sistema Internacional

En la imagen anterior se observa como además se presenta el sistema de unidades que se utiliza, en este caso el Sistema Internacional. La última orden USER DEFINED: DISABLED significa que no se le permite al usuario crear su propio sistema de unidades.

MAT-fem Composite Beams.sim

El fichero con extensión .sim comprende distintos símbolos que serán de utilidad a la hora de representar algunas condiciones en el estado de preproceso de GiD.

En la siguiente imagen se presenta la rutina referente a las condiciones de apoyo en los puntos.

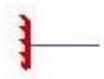
cond Point_Constraints	
4	
global	
cond(int,1) && cond(int,3) && cond(int,5)	
1	
0	
0	
pc_emp.geo	
global	
cond(int,1) && cond(int,3)	
1	
0	
0	
ep_apoiol.geo	
global	
cond(int,1) cond(int,3)	
cond(int,3)	
cond(int,1)*(-1)	
0	
ep_apoiro-rodillo.geo	
global	
cond(int,5)	
1	
0	
0	
pc_sim.geo	

Fig. 53 Símbolos en cuanto a condiciones de apoyo

La rutina se inicia con el nombre de la condición a la que se hace referencia, en el caso de la imagen anterior es la condición de apoyo puntual (Point_Constraints). Después se escribe el número de símbolos que se definirán para esa condición, cuatro en este caso. Posteriormente se indica que la orientación de los símbolos es respecto a los ejes globales ya que se tratan de condiciones sobre puntos. Luego se define qué condiciones se tienen que cumplir y para ello se hace referencia a las condiciones que se describen en el fichero .cnd. La

manera de definir el tipo de condición es escribiendo $cond(type,i)$, donde $type$ se refiere a si la variable es un número real (*real*) o entero (*int*) y la i es el número del campo para esta condición en particular. La orden $\&\&$ quiere decir que las dos o más condiciones se tienen que cumplir a la vez, en cambio la orden $||$ significa que es suficiente con que se cumpla una de las condiciones. En lo que se refiere a los símbolos para representar la restricción puntual de desplazamientos, el número 1 se refiere a desplazamiento horizontal, el 3 al desplazamiento vertical y la rotación se define mediante el número 5. Una vez se definen las condiciones se indica el archivo de extensión *.geo* que representará el símbolo. Además se tiene que precisar la orientación de éste teniendo en cuenta que el origen corresponde al punto (0,0,0) y el vector de referencia es el (1,0,0).

Es importante recordar que la asignación de materiales y condiciones de contorno, que se realiza en el programa GiD durante el preproceso, se hace sobre la viga en una dimensión. Es por eso que las condiciones de apoyo se realizarán sobre puntos, que serán nodos en la malla, y las propiedades del material se asignarán a líneas, que se transforman en elementos cuando se discretice la viga.

A continuación se presenta la rutina para condiciones de carga puntual y carga distribuida. El proceso es el mismo que el anteriormente descrito, únicamente se hace notar que los símbolos de la carga distribuida están orientados respecto los ejes globales ya que las vigas siempre estarán representadas paralelas al eje x global, sino tendrían que referirse a los ejes locales según la orientación de las líneas respecto a la horizontal.

Los números a los que hacen referencia las condiciones se refieren a carga horizontal (1), carga vertical (2) y momento (3).

cond Point_Load	
4	
global	
cond(real,1) && cond(real,2) && cond(real,3)	
cond(real,2)	
cond(real,1)*(-1)	
0	
pl_fl_m.geo	
global	
cond(real,1) && cond(real,3) cond(real,2) && cond(real,3)	
cond(real,2)	
cond(real,1)*(-1)	
0	
pl_fl_m.geo	
global	
1	
cond(real,1)	
cond(real,2)	
0	
ep_fletxa.geo	
global	
cond(real,3)	
cond(real,3)	
0	
0	
pl_mom.geo	

Fig. 54 Representación de los símbolos de las cargas puntuales

```
cond Uniform_Load
1
global
cond(real,1) && cond(real,2) || cond(real,1) || cond(real,2)
cond(real,2)
cond(real,1)*(-1)
0
ep_flechas.geo
```



Fig. 55 Símbolo para carga uniformemente distribuida

MAT-fem Composite Beams.bas

Una vez se han asignado las condiciones de contorno, las propiedades de los materiales y se ha generado la malla, es necesario producir un archivo de entrada de datos que sea procesado por el programa de cálculo MATLAB para analizar la viga compuesta. El fichero de extensión .bas sirve de nexo entre el programa GiD y el programa de cálculo MATLAB de manera que describe la estructura y el formato del fichero de entrada para que la información obtenida en GiD pueda ser procesada por MATLAB.

Este archivo recoge la información asignada en GiD y la escribe de forma que la estructura final tenga el formato del fichero de datos, que veíamos en la figura 5.

```

NUMBER: 1 MATERIAL: Composite_Section
QUESTION: LAYER
VALUE: 3
QUESTION: YOUNG(LAYER,_____E___N/m^2_____)
VALUE: #N# 6 1 2.1e11 2 3.0e10 3 3.0e10

```

```

% Material Properties
%
*loop materials
*format " layers = %10.2e; "
*MatProp(1);
*set var L=MatProp(1,int)
*set var N=MatProp(2,int)
young = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(2,*operation(i+1));
*end for

```

```

% Material Properties
%
layers = 3.00e+00;

young = [
Fig. 2.100000000e+11 ;
56 3.000000000e+10 ;
Prop 3.000000000e+10 ];
iada

```

des del material en MATLAB, GiD y en el fichero de datos

En las imágenes anteriores se visualiza como

el archivo de extensión .bas capta la información de la rutina programada en el archivo de extensión .mat, para posteriormente presentarla en el fichero de datos. Se muestra únicamente las primeras líneas de la rutina ya que se cree suficiente para entender la forma de proceder de este archivo, en el anejo 3 se puede visualizar toda la programación.

Las propiedades de los materiales se extraen del archivo .mat mediante la orden *MatProp(i,type)* con *i* el número del campo de la propiedad y *type* indicando si es un valor real o un entero, y se van ordenando de manera que nos aparezcan como vemos en la figura 56. El número de capas se refiere a número entero y el valor obtenido del archivo .mat también lo es, así que simplemente hay que hacer referencia a esta propiedad. Sin embargo, para el módulo de Young buscamos ordenarlo en forma de matriz de manera que cada fila se refiera a cada una de las capas de la viga, por tanto, en este caso tenemos que indicar, en el archivo de extensión .mat, los valores del módulo de Young de cada capa y escribirlos en el formato correcto.

Siguiendo el mismo procedimiento se escribe en el formato correcto las propiedades de materiales que faltan, las coordenadas y elementos de la malla, los nodos con movimientos preescritos, las cargas puntuales y las cargas repartidas.

MAT-fem Composite Beams.tcl

Son archivos adicionales al programa GiD escritos en lenguaje de programación Tcl/Tk. Permite crear nuevas ventanas y funcionalidades que añadir al programa. Entre otras funcionalidades, el archivo con extensión .tcl se encarga de escribir la rutina que despliega el logo y la función del programa.

Este archivo crea los iconos, que aparecen en el programa GiD, de asignación de condiciones de contorno, materiales, información interna del problema y la generación de la malla.

```
proc MyBitmaps { dir { type "DEFAULT INSIDELEFT" } } {  
  global MyBitmapsNames MyBitmapsCommands MyBitmapsHelp MAT-  
  fem_Composite_Beams  
  
  set MyBitmapsNames(0) "fix.gif pload.gif uload.gif material.gif \  
    units.gif mesh.gif write.gif"  
  set MyBitmapsCommands(0) [list [list -np- GidOpenConditions  
  "Displacement_Constraints" \  
    [list -np- GidOpenConditions "Point_Loads" \  
    [list -np- GidOpenConditions "Uniform_Loads" \  
    [list -np- GidOpenMaterials] \  
    [list -np- GidOpenProblemData] \  
    "Meshing generate" \  
    "File WriteCalcFile" ]
```

Fig. 57 Archivo de extensión tcl

A continuación se indican los iconos a los que se hace referencia en el archivo de extensión .tcl:



Icono para identificar puntos con movimientos prescritos. Al presionar en este icono nos aparece la ventana de la figura 46.



Este icono sirve para asignar cargas puntuales. Clicando encima de él aparece la ventana que vimos en la figura 47.



Presionando sobre este icono se abre la ventana de la figura 48 que nos permite asignar las cargas distribuidas.

 Este icono nos abre una ventana donde asignar las propiedades de los materiales de la viga compuesta. Esta ventana la vimos en la figura 50.

 Si se presiona en este icono nos aparece la ventana de la información interna del problema, la que vimos en la figura 49.

 Mediante este icono se forma la malla.

 Este es el archivo que permite escribir el fichero de entrada de datos para MATLAB. Es necesario añadir la extensión .m al nombre del fichero.

4.3.3 Archivos de resultados

En los apartados anteriores se ha descrito el programa de cálculo en MATLAB que permite analizar vigas compuestas de Timoshenko, evidentemente son necesarios los datos iniciales del problema a resolver de manera que, a través de GiD, proporcionamos un fichero de datos a MATLAB para que calcule el problema. Una vez calculado, MATLAB devuelve dos ficheros con la información de la malla (archivo con extensión .flavia.msh) y la información de los resultados (fichero con extensión .flavia.res) que serán visualizados con GiD. Este procedimiento se representa mediante el siguiente esquema:

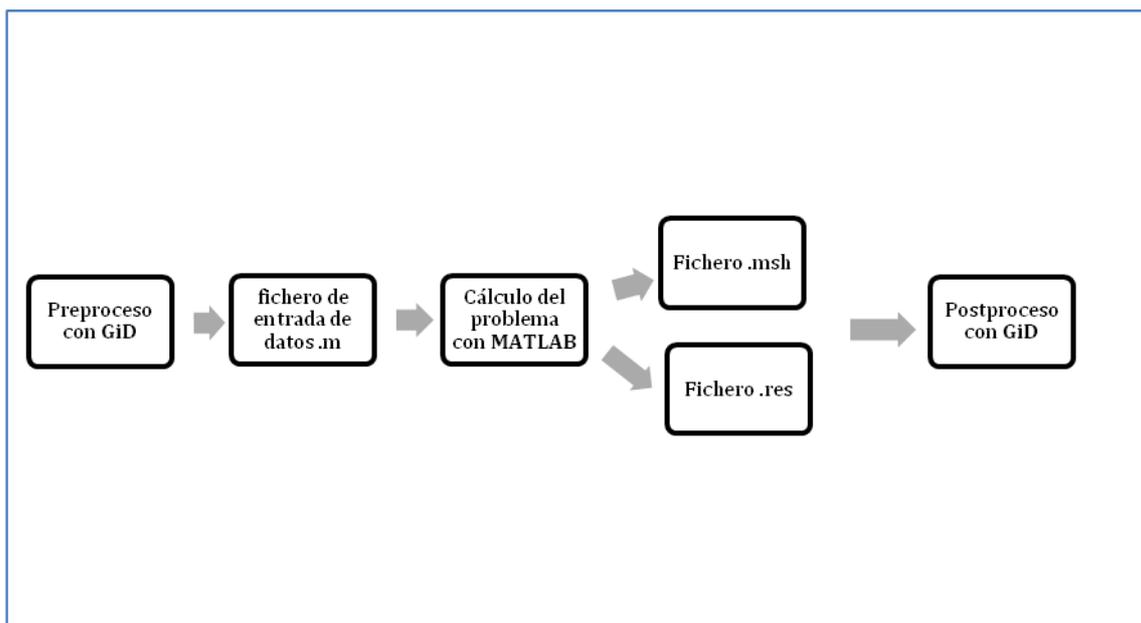


Fig. 58 Esquema de procedimiento de MAT-fem

La forma en que se presentan los resultados gráficamente en GiD lo veremos en el siguiente capítulo mediante varios ejemplos.

5. EJEMPLOS

En este capítulo vamos a comprobar la funcionalidad del programa y su eficiencia computacional frente al método tridimensional para calcular vigas compuestas.

Vamos a ver una serie de ejemplos donde se analiza el mismo problema mediante el programa de MAT-fem para vigas compuestas de Timoshenko y mediante el cálculo en tres dimensiones, de manera que podamos verificar que los resultados obtenidos por el programa expuesto en esta tesina son suficientemente fiables si los comparamos con la solución 3-D.

5.1 VOLADIZO CON CARGA PUNTUAL EN UN EXTREMO

Este primer problema estudia uno de los casos más sencillos en el análisis de estructuras. Se trata de una viga de longitud L empotrada en un extremo y libre en el otro, cargada por una fuerza puntual P en el extremo libre. A continuación se muestran las características geométricas de la viga y las propiedades de los materiales que la componen:

- ✓ $L = 10\text{m}$
- ✓ $P = 100\text{KN}$
- ✓ Sección rectangular formada por tres capas cuyas propiedades se detallan en la siguiente tabla:

CAPA	MÓDULO DE YOUNG(N/m ²)	COEFICIENTE DE POISSON	DENSIDAD (N/m ³)	ESPESOR (m)	ANCHO (m)
1	$2.1 \cdot 10^{11}$	0.3	0	0.25	0.5
2	$3.0 \cdot 10^{10}$	0.2	0	0.5	0.5
3	$2.1 \cdot 10^{11}$	0.3	0	0.25	0.5

Tabla 3 Características de los materiales

A continuación se van a exponer, por separado, los resultados obtenidos al analizar esta viga mediante la teoría de Timoshenko y, posteriormente, se presentarán los resultados

que se deducen de analizar la viga como sólido tridimensional. Finalmente se realiza una comparación de resultados y se comentan las conclusiones que se derivan de ésta.

I. SOLUCIÓN TEORÍA DE VIGAS COMPUESTAS DE TIMOSHENKO (TCB)

Como se ha visto en los capítulos anteriores, en la fase de preproceso, en que se asignan las propiedades del material y las condiciones geométricas y de contorno de la viga, se trabaja con la estructura en 1D asignando las características a puntos y líneas, dependiendo del caso. Por tanto, la viga se presenta como una línea donde sus extremos se representan mediante dos puntos. En nuestro caso, en el primer punto se restringen los tres desplazamientos posibles, ya que tenemos un empotramiento, y en el segundo punto se aplica una carga puntual. Las propiedades geométricas de la viga y las características de los materiales que la forman se asignan a la línea que representa la estructura.

El siguiente paso es formar la malla discretizando la viga en elementos unidimensionales de dos nodos cada uno.



Fig. 59 Preproceso

Una vez se construye la malla se crea el fichero de datos que alimentará al programa de cálculo en MATLAB, descrito en el capítulo 4.3.1. Después, éste nos proporcionará los dos archivos necesarios para visualizar los resultados en GiD, de extensión .msh y .res.

En la siguiente figura se presenta la solución obtenida por el programa. Como se puede observar y como comentamos anteriormente, la estructura en el postproceso se representa en dos dimensiones. De esta manera se visualiza fácilmente las diferentes capas que componen la viga y se puede estudiar el comportamiento estructural de ésta tanto en dirección horizontal como en vertical.

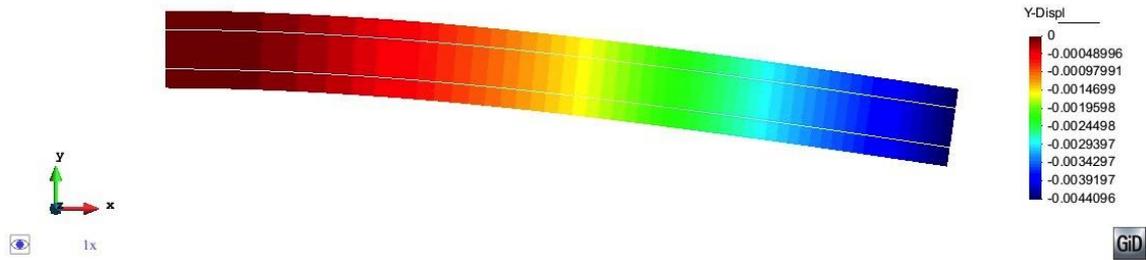


Fig. 60 Deformada en el postproceso

Convergencia de resultados

En la gráfica 61 se observa cómo el desplazamiento vertical del extremo libre va variando en función del número de nodos. Se distingue fácilmente cómo a medida que la malla se densifica, el valor converge hacia la solución óptima. La malla que proporciona unos resultados lo suficientemente fiables es la de 2001 nodos. También se adjunta la tabla con los valores obtenidos.

Nodos	Log(nodos)	Desplazamiento vertical
		(m)
11	1,04	-0,0043989
21	1,32	-0,0044069
51	1,71	-0,0044092
201	2,30	-0,0044096
2001	3,30	-0,0044096

Tabla 4 Tabla de resultados

Para poder observar de una manera más detallada la curva de convergencia se realiza la gráfica en escala logarítmica.

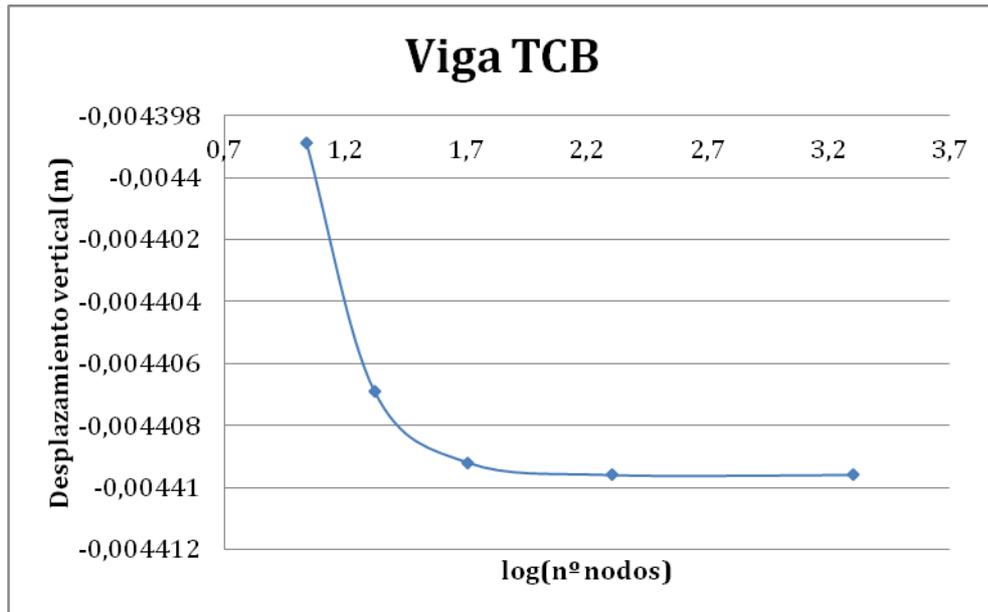


Fig. 61 Convergencia resultados

Solución obtenida

A continuación se muestra los resultados obtenidos mediante la malla de 2001 nodos y, posteriormente, se adjuntan las imágenes más representativas del comportamiento estructural de la viga obtenidas con Gid. Los valores que se muestran corresponden a la fibra inferior de la primera capa del extremo libre de la viga.

Desp_x (m)	Desp_y (m)	Giro	Qy (N)	Mz (Nm)	Sigma_x (N/m ²)	Tau_xz (N/m ²)
-0.00032	-0.0044096	-0.00064	-43299	-122.5	-3360	-3.4639·10 ⁵

Tabla 5 Solución obtenida con la malla de 2001 nodos

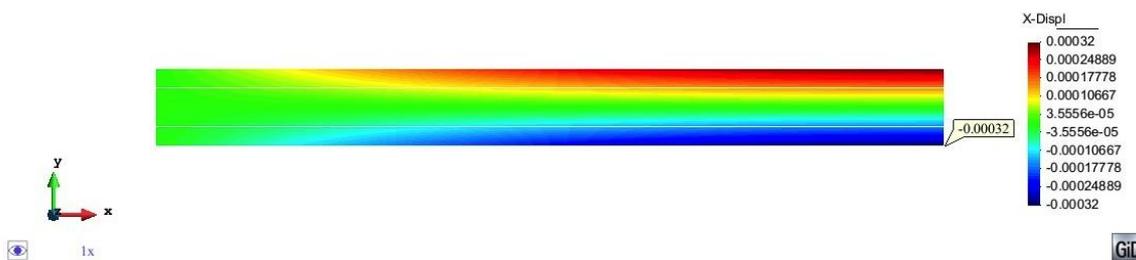


Fig. 62 Desplazamiento horizontal

Evidentemente los desplazamientos horizontales de la viga se producirán en gran medida en las fibras extremas de la sección y serán mayores a medida que nos acercamos al extremo libre de la viga. En la siguiente gráfica se muestra que la teoría de Timoshenko supone una distribución lineal del desplazamiento horizontal en toda la sección de la viga, sin tener en cuenta distintas pendientes según la capa.

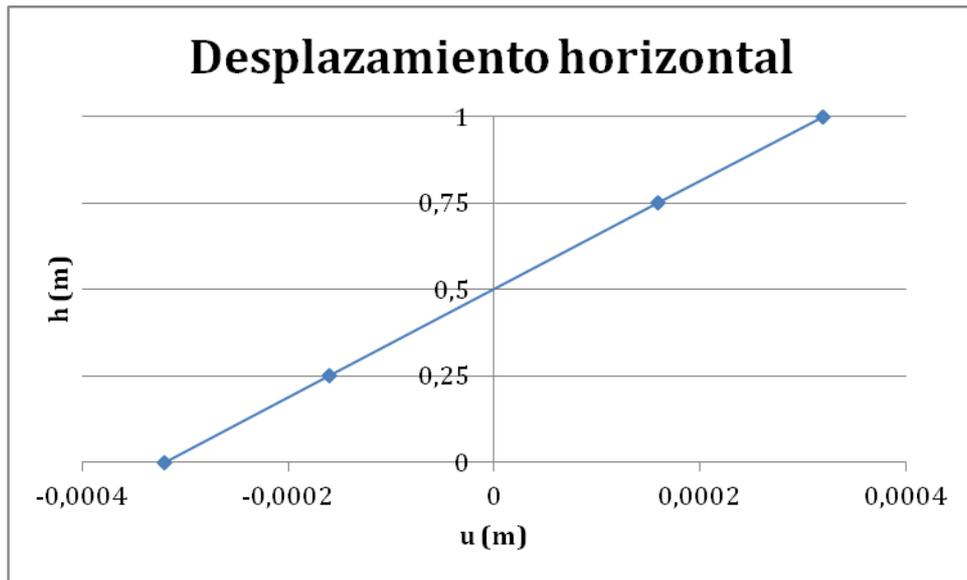


Fig. 63 Gráfica h(m)-u(m)

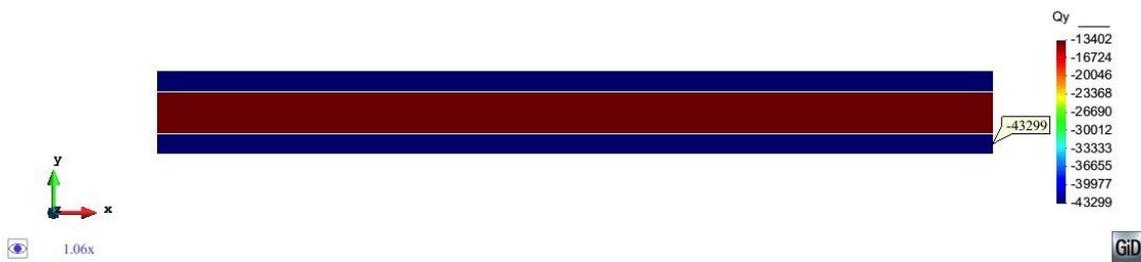


Fig. 64 Cortante

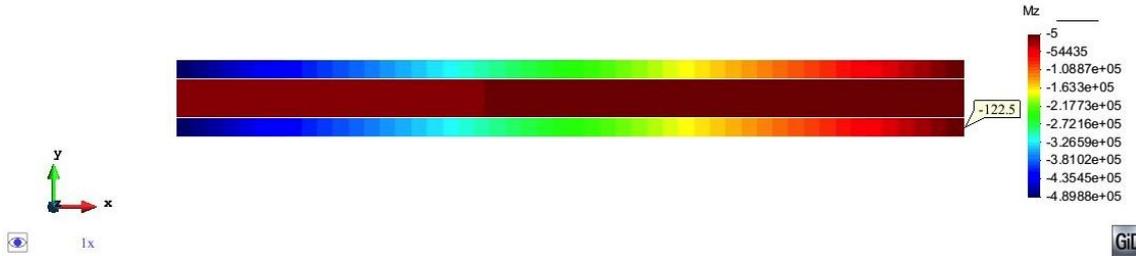


Fig. 65 Momento

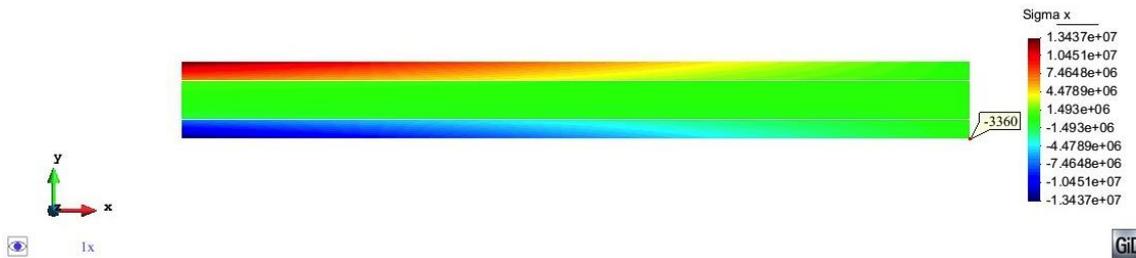


Fig. 66 Tensión normal

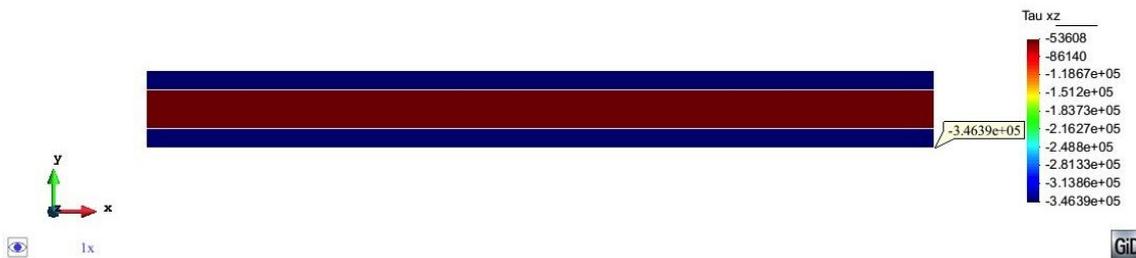


Fig. 67 Tensión tangencial xz

En esta última figura, se distingue realmente bien que la distribución de tensiones tangencial a lo largo de la sección de la viga toma valores uniformes en cada capa.

II. SOLUCIÓN SEGÚN TEORÍA DE SÓLIDO 3D

Con el objetivo de obtener el resultado más real posible, sin fenómenos locales producidos por la carga puntual, se sustituye esta fuerza de 100KN por una carga uniformemente repartida en la sección de la viga cuyo valor es:

$$p = \frac{P}{A} = \frac{100000N}{0.5 \text{ m}^2} = \frac{200000N}{\text{m}^2} \quad (71)$$

donde A es el área de la sección.

La malla en que se discretiza la viga se compone de elementos tetraédricos de cuatro nodos cada uno. Tanto la geometría inicial cómo los resultados se muestran en tres dimensiones.

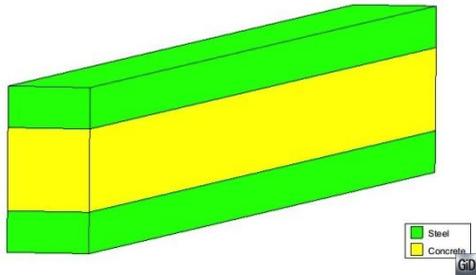


Fig. 68 Materiales

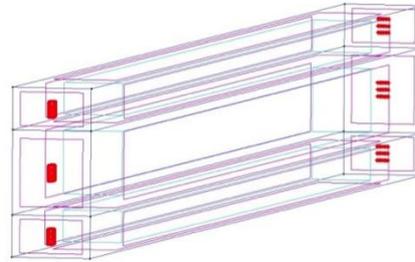


Fig. 69 Condiciones de contorno

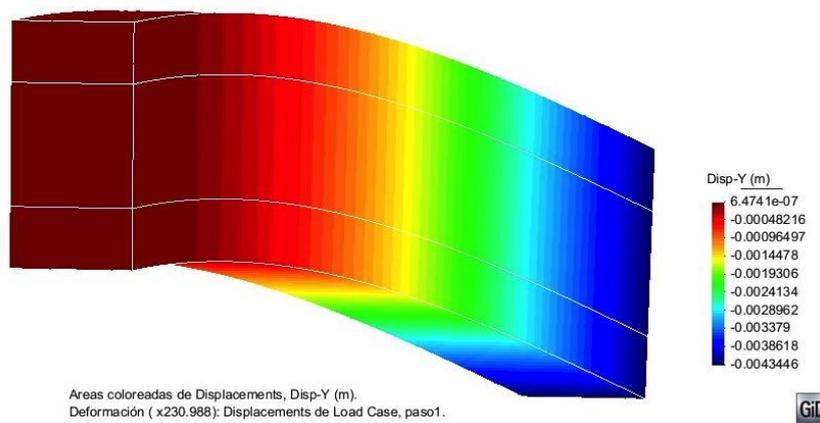


Fig. 70 Deformada de la viga

Convergencia de resultados

En la siguiente gráfica se vuelve a comprobar, ahora para el caso tridimensional, cómo la solución converge hacia un valor óptimo. Se realiza el estudio de la evolución que experimenta el desplazamiento vertical del extremo libre a medida que aumenta el número de nodos de la malla. Se presenta la tabla con los valores obtenidos teniendo en cuenta que la malla de 13160 es una muy buena aproximación de la realidad.

Nodos	Log	Desplazamiento vertical
-------	-----	-------------------------

	(nodos)	(m)
159	2,201	-0,0034846
1544	3,188	-0,0041095
3163	3,500	-0,0042727
5946	3,774	-0,0043087
7510	3,875	-0,0043202
10176	4,007	-0,0043323
11599	4,064	-0,0043409
13160	4,119	-0,0043446

Tabla 6 Tabla de resultados

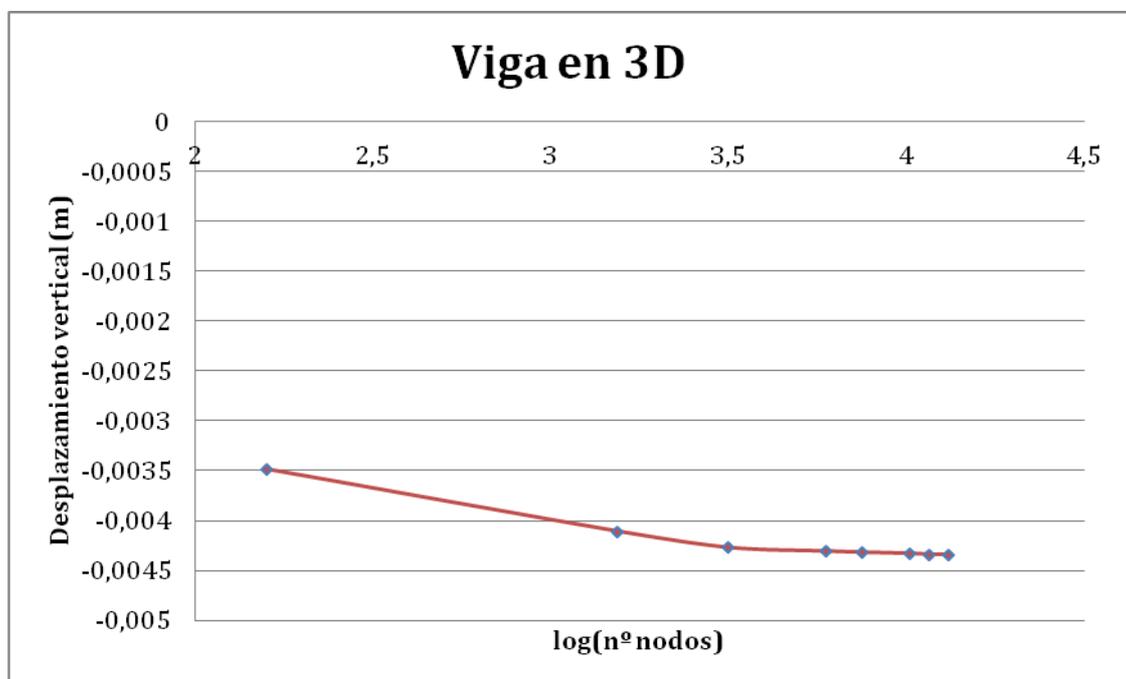


Fig. 71 Convergencia de resultados

Solución obtenida

A continuación se muestran los resultados obtenidos mediante la malla de 13160 nodos. De igual manera que hacíamos en el caso anterior, se muestran los resultados en la fibra inferior de la primera capa del extremo libre de la viga.

Desp_x	Desp_y	Sigma_x	Tau_xz
--------	--------	---------	--------

(m)	(m)	(N/m ²)	(N/m ²)
0.0003176	-0.0043446	-83623	1.124·10 ⁵

Tabla 7 Solución obtenida con la malla de 13160 nodos

Las siguientes imágenes muestran el comportamiento estructural de la viga en tres dimensiones a causa de una carga puntual en su extremo libre, la cual es equivalente a una carga uniformemente repartida en la sección de la viga.

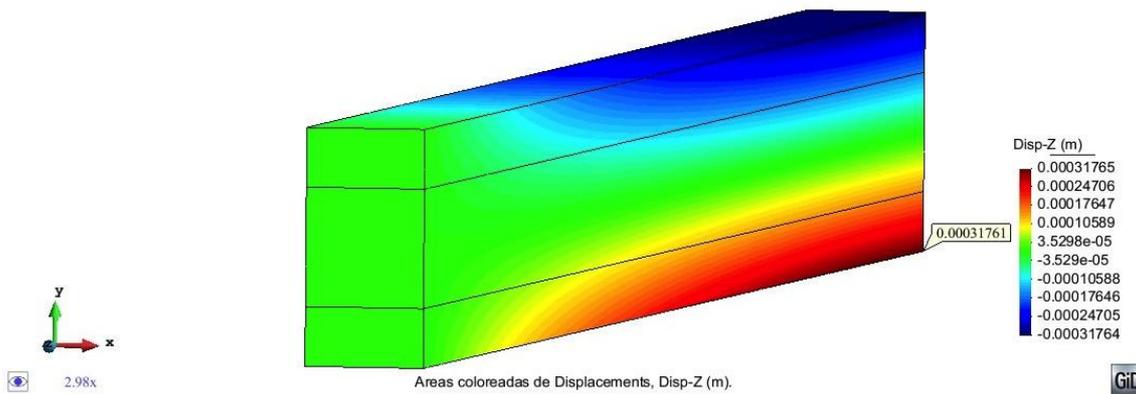


Fig. 72 Desplazamiento horizontal

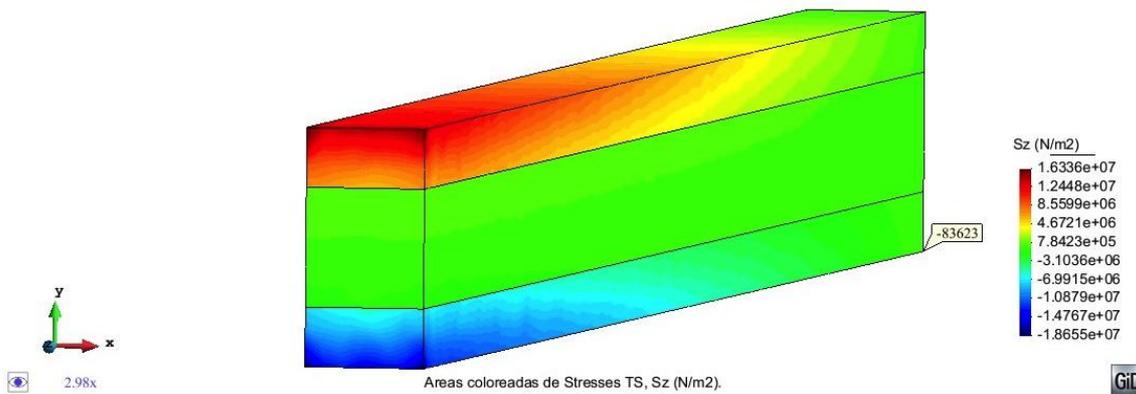


Fig. 73 Tensión normal

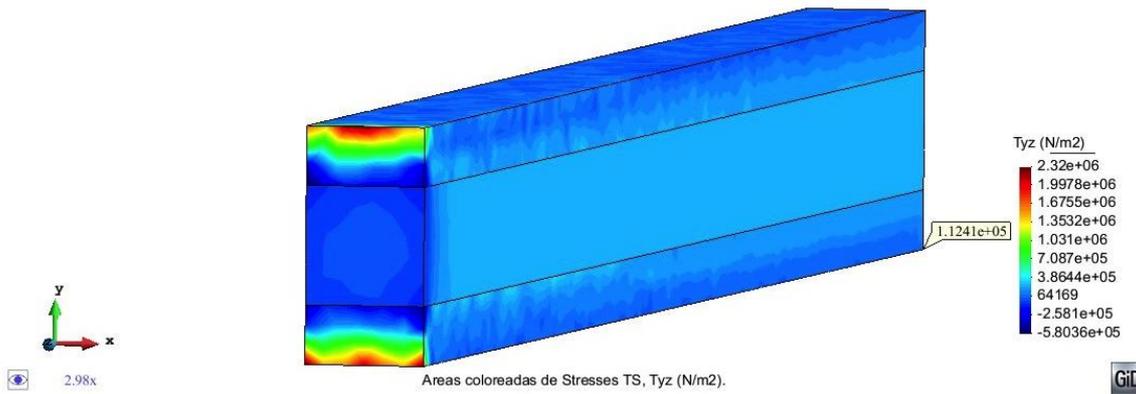


Fig. 64 Tensión tangencial (Tau_xz)

A continuación se comparan los resultados en valor absoluto, ya que los ejes en cada método siguen sentidos distintos, obtenidos de la misma viga según los dos métodos, a la vez que se muestra la diferencia entre los valores en porcentaje. Este último valor se ha calculado respecto la solución en tres dimensiones:

$$Dif(\%) = \left| \frac{X_{3D} - X_{TCB}}{X_{3D}} \right| \cdot 100 \quad (72)$$

dónde:

- ✓ X_{3D} es el valor obtenido mediante la teoría de sólido 3D.
- ✓ X_{TCB} es el valor obtenido mediante la teoría de viga compuesta de Timoshenko.

	Desp_x (m)	Desp_y (m)	Sigma_x (N/m ²)	Tau_xz (N/m ²)
Sólido 3D	0.0003176	0.0043446	83623	$1.124 \cdot 10^5$
TCB	0.00032	0.0044096	3360	$3.4639 \cdot 10^5$
Diferencia (%)	0.756	1.496	95.982	208.176

Tabla 8 Comparación de resultados

Se puede observar cómo los desplazamientos según las dos teorías dan aproximadamente el mismo valor. En cambio, hay más diferencia en las tensiones debido a que son valores más elevados y pueden sufrir errores numéricos en el cálculo. Igualmente, se observa en las imágenes que el comportamiento tensional en la viga es el mismo y los valores son del mismo orden de magnitud.

III. TIEMPO INVERTIDO EN EL CÁLCULO

A continuación se presentan dos gráficas en las que se estudia el tiempo computacional necesario para calcular el mismo problema siguiendo cada uno de los métodos descritos.

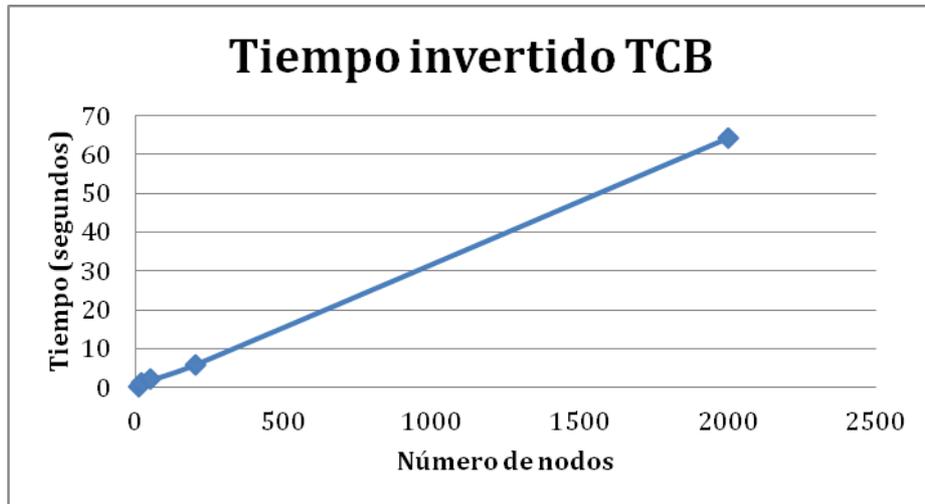


Fig. 75 Tiempo invertido según la teoría de Timoshenko

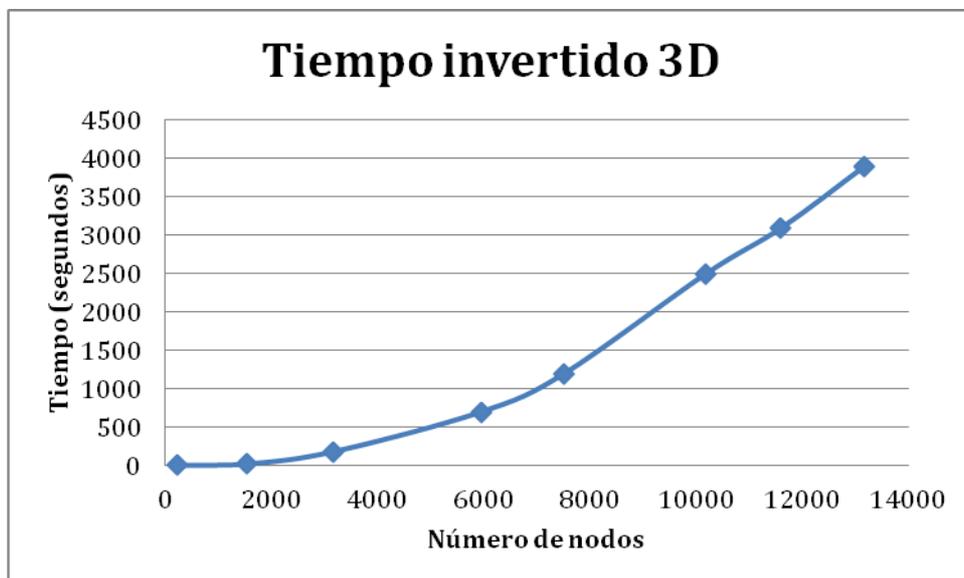


Fig.76 Gráfica tiempo invertido en el cálculo de viga en 3D

Según estas graficas se puede concluir sin temor a equivocarnos que el problema calculado en MAT-fem consume un tiempo computacional considerablemente menor, del

I. ESTUDIO DE CONVERGENCIA

Teoría de Timoshenko para vigas compuestas

El estudio de convergencia se realiza midiendo el desplazamiento vertical en el centro de la viga en función del número de nodos. El valor óptimo se encuentra, en este caso, para la malla de 1501 nodos. Se muestra la gráfica de convergencia en escala logarítmica y también se adjunta la tabla con los valores obtenidos.

Nodos	log(nodos)	Desplazamiento vertical (mm)
21	1,32	-10,768
61	1,78	-10,856
151	2,18	-10,865
241	2,38	-10,866
601	2,78	-10,867
1501	3,18	-10,867

Tabla 10 Valores de convergencia

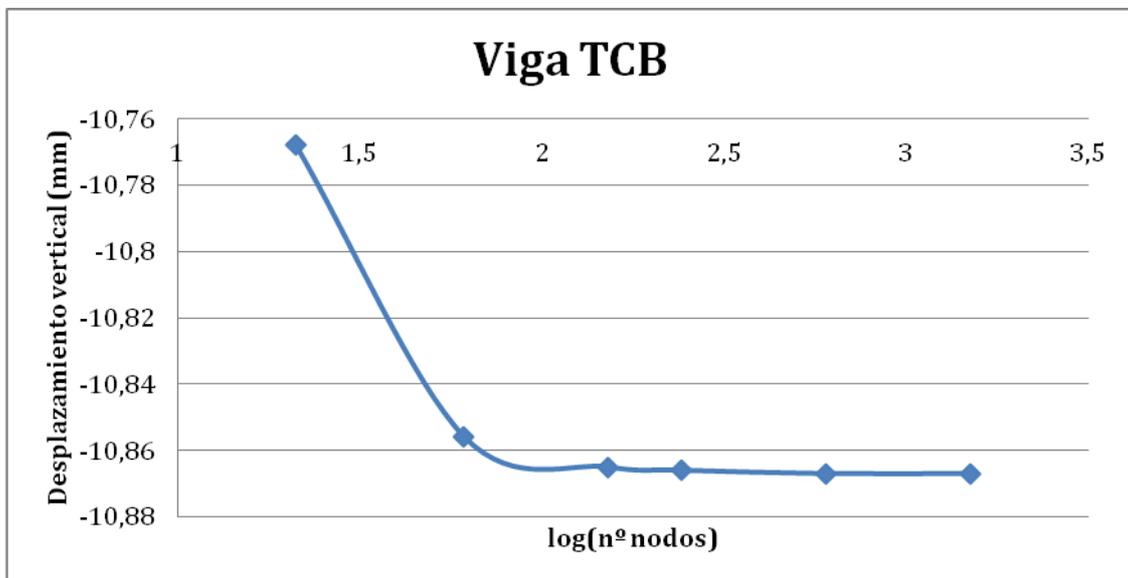


Fig. 77 Convergencia resultados

Teoría tridimensional

En cuanto a la convergencia de resultados en el análisis tridimensional los resultados obtenidos son:

Nodos	Log(nodos)	Desplazamiento vertical (mm)
14815	4,17	-10,663
23347	4,37	-10,701
37169	4,57	-10,73
73863	4,87	-10,755
167453	5,22	-10,774
225420	5,35	-10,779
293212	5,47	-10,781

Tabla 11 Valores de convergencia

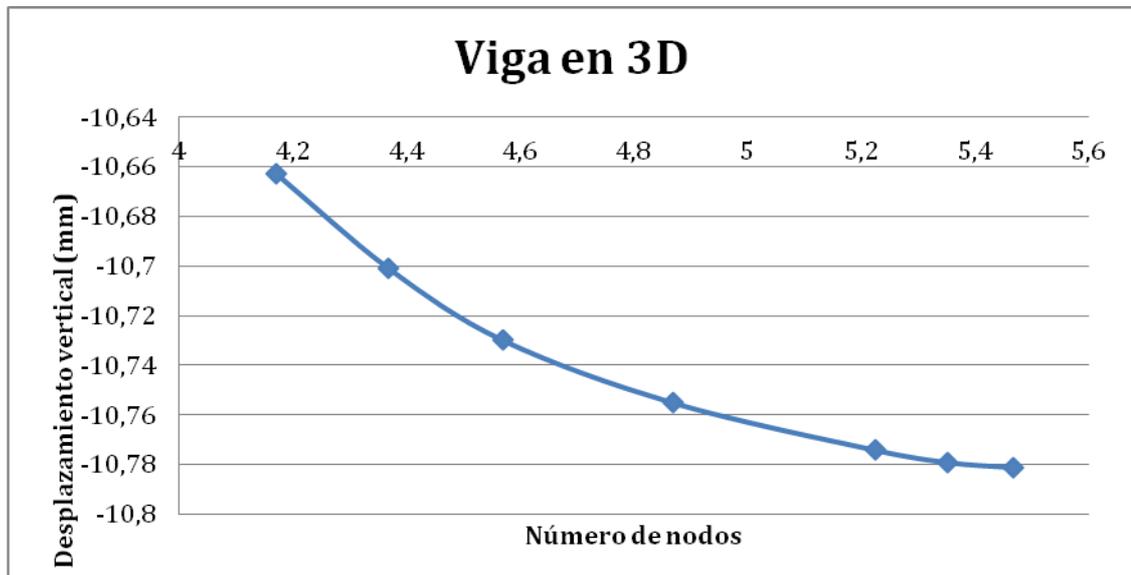


Fig. 78 Convergencia de resultado

II. PRESENTACIÓN DE RESULTADOS

A continuación se presentan los resultados más representativos que permiten comparar las soluciones según Timoshenko con la viga en tres dimensiones. De la misma manera que hacíamos en el ejemplo anterior se cuantifica la diferencia de los resultados en porcentaje. En este caso se ha elegido como punto de referencia para comparar resultados, la última fibra del punto central de la viga.

	Desp_x (mm)	Desp_y (mm)	Sigma_x (N/mm ²)	Tau_xz (N/mm ²)
Sólido 3D	0	-10.781	12045	0
TCB	0	-10.867	12118	0
Diferencia (%)	0	0.80	0.61	0

Tabla 12 Valores obtenidas según TCB y 3D

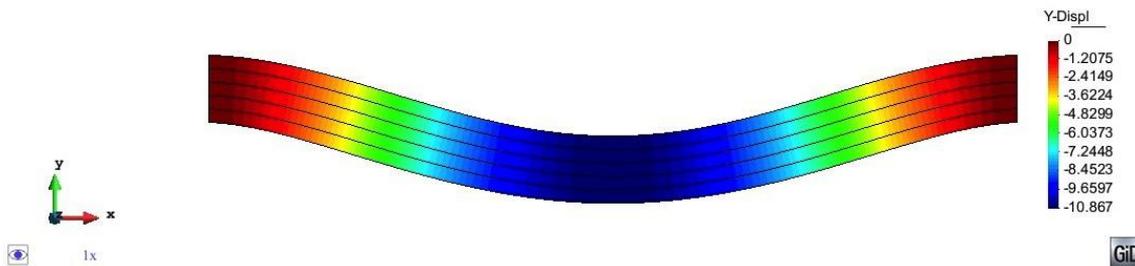


Fig. 79 Deformada en el postproceso

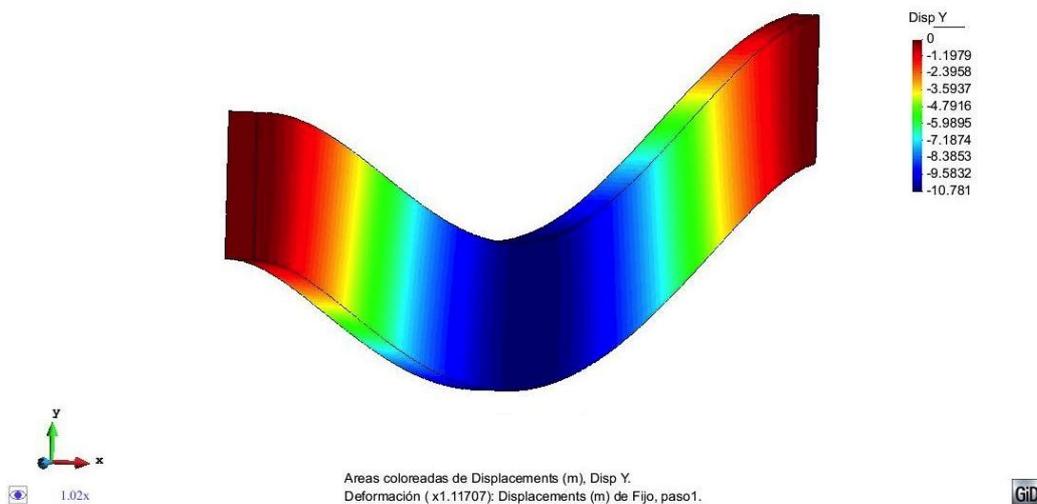


Fig. 80 Deformada

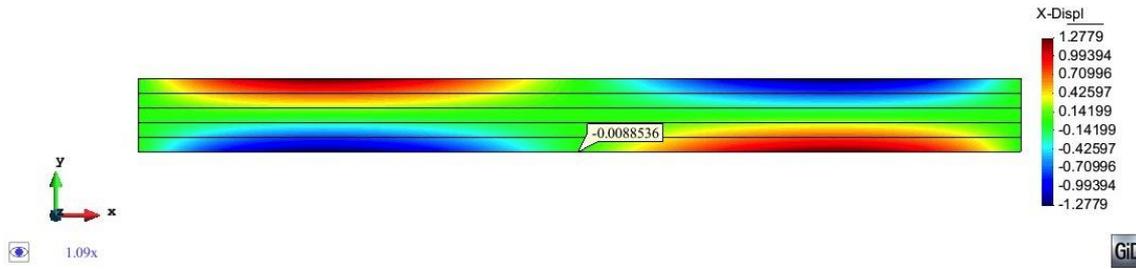


Fig. 81 Desplazamiento horizontal TCB

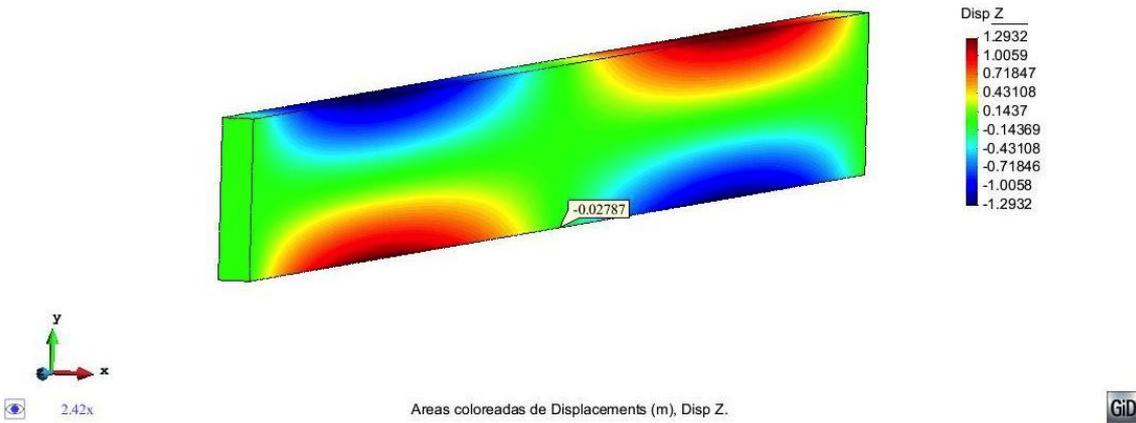


Fig. 82 Desplazamientos horizontales 3-D

El comportamiento del desplazamiento horizontal indica claramente que las zonas con mayor y menor movimiento son idénticas. El signo varía ya que el eje de la viga tiene sentido contrario en cada caso. Además, se puede concluir que los valores obtenidos en el centro de la viga serán tan próximos a cero que se pueden despreciar.

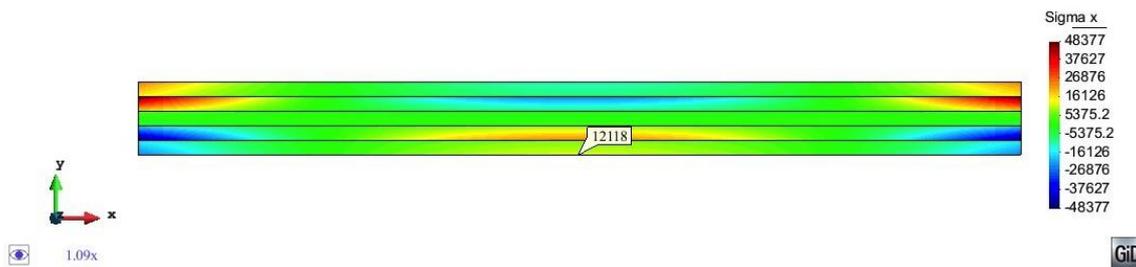


Fig. 83 Tensión normal

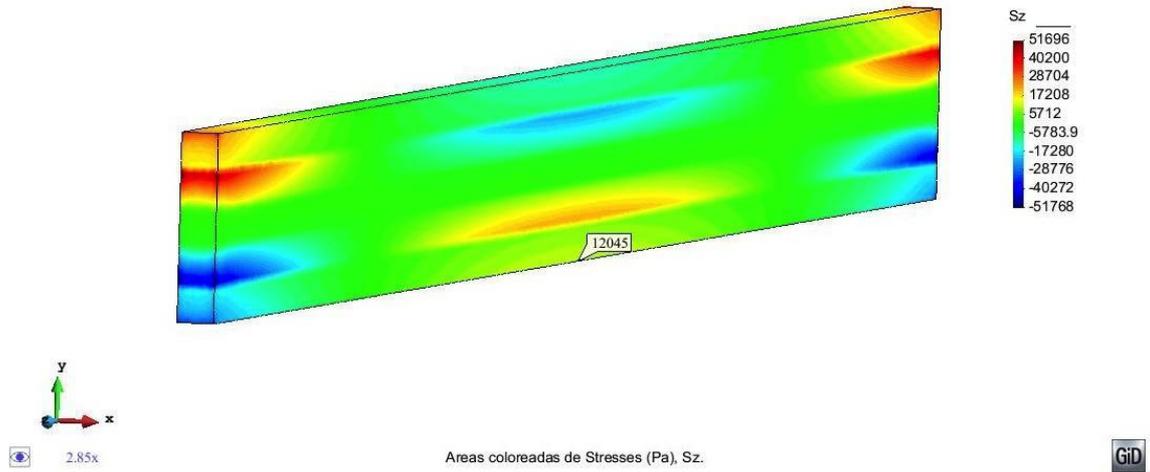


Fig. 84 Tensión normal

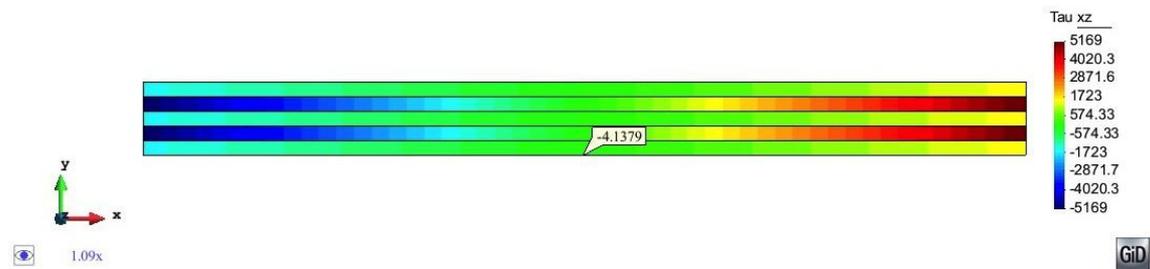


Fig. 85 Tensión tangencial xz

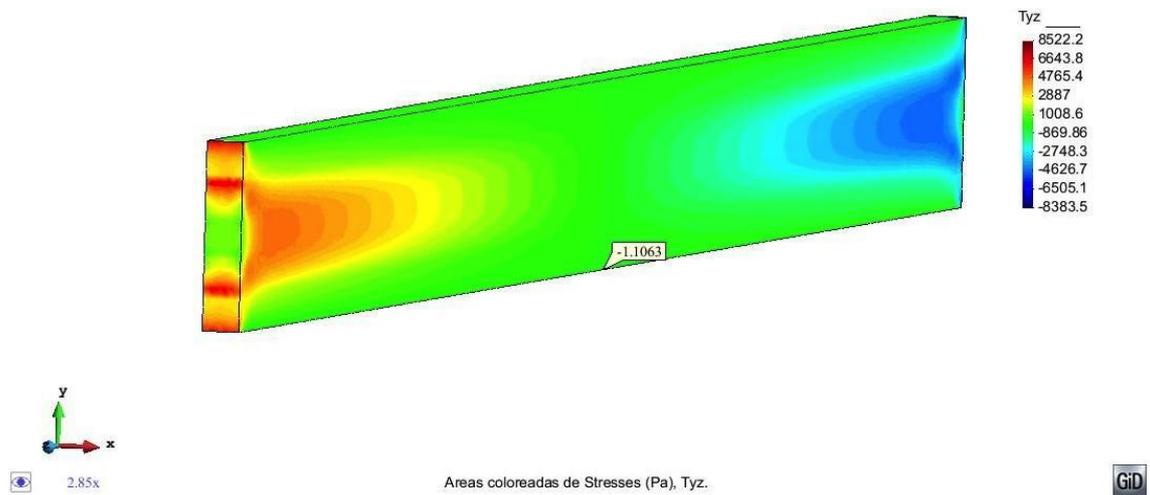


Fig. 86 Tensión tangencial xz

De la misma manera que sucedía en el estudio del desplazamiento axial, las tensiones tangenciales se pueden suponer 0 en el punto de referencia. Esto no es nada descabellado si nos fijamos en las órdenes de magnitud en que se mueve esta tensión.

Se compara a continuación estas tensiones en la sección extrema de la viga de las dos teorías, de manera que se muestre la distribución real que tendrían las tensiones y la aproximación que realiza Timoshenko suponiéndolas uniformes en cada capa.

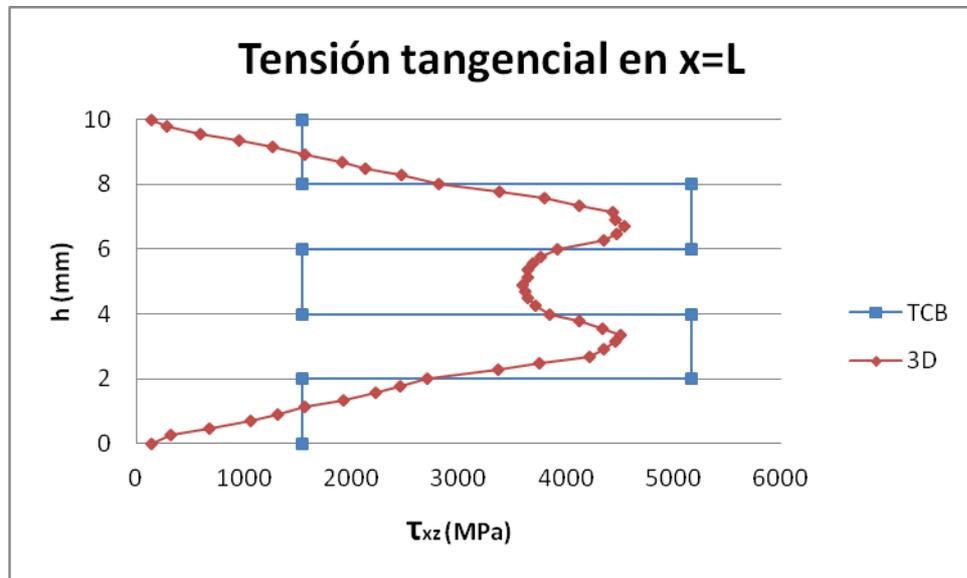


Fig. 87 Tensión tangencial según TCB y 3D

Esta gráfica representa fielmente la aproximación que supone Timoshenko en lo que se refiere a las tensiones tangenciales, tomando valores uniformes en las capas. En cambio, teóricamente estos valores no serían uniformes sino que tendrían una variación parabólica distinta según la capa, que se muestra según la teoría tridimensional.

Observando las distintas gráficas anteriormente expuestas se distingue claramente que el comportamiento estructural de la viga es el mismo en ambos métodos. Además, el programa según MAT-fem aporta unos resultados muy parecidos a los obtenidos analizando la viga en tres dimensiones. A continuación se vuelve a hacer referencia a la diferencia que existe en el tiempo computacional necesario en cada caso, claramente el método más rápido, y sin perder precisión, es el de Timoshenko.

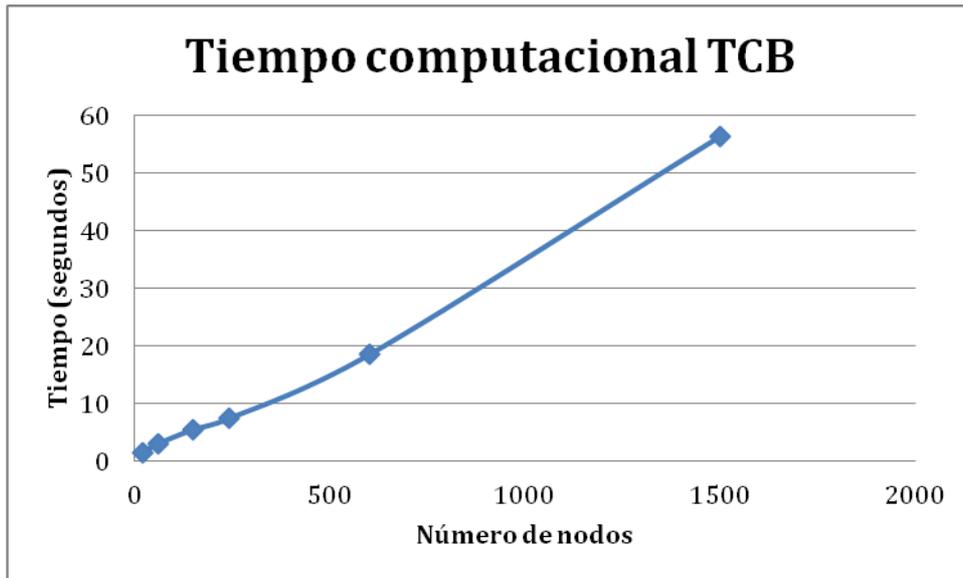


Fig. 88 Tiempo invertido en el cálculo según TCB

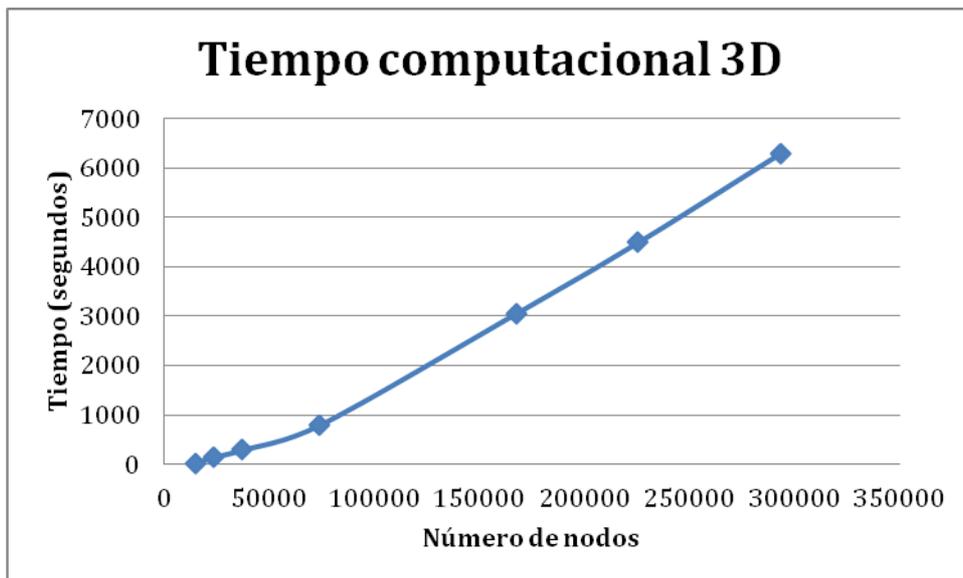


Fig. 89 Tiempo invertido en el cálculo de viga en 3D

5.3 VIGA CON SECCIÓN EN DOBLE T

Se ha podido comprobar en los ejemplos anteriores que el programa calculado funciona perfectamente en el caso de que la sección de la viga sea rectangular, en el presente ejemplo vamos a estudiar una viga con sección en doble T para comprobar que sucede cuando la sección tiene capas de distintos anchos. Esto nos servirá como introducción al ejemplo real del anejo 1, en el que se estudia un caso real de refuerzo con materiales compuestos sobre una viga con sección en doble T.

El presente ejemplo es un voladizo de longitud 1m sometida a una carga en el extremo libre de 100KN. La viga está compuesta totalmente de acero y tiene las características de una viga con sección IPE200. A pesar de que la estructura está compuesta por el mismo material, tendremos que dividir la sección en tres capas que representen las dos alas y el alma de la viga.

CAPA	YOUNG(N/mm ²)	POISSON	DENSIDAD (N/m ³)	ESPESOR (mm)	ANCHO (mm)
1	$2.1 \cdot 10^5$	0.3	0	8.5	100
2	$2.1 \cdot 10^5$	0.3	0	183	5.6
3	$2.1 \cdot 10^5$	0.3	0	8.5	100

Tabla 13 Características de la sección

I. ESTUDIO DE CONVERGENCIA

La convergencia de resultados se ha realizado observando la evolución a medida que se aumenta el número de nodos del desplazamiento vertical en el extremo libre de la viga. Se observa en las siguientes graficas en escala logarítmica como las mallas que aportan unos resultados suficientemente buenos son, para la teoría de Timoshenko, 5001 nodos y, en el análisis tridimensional, 12105.

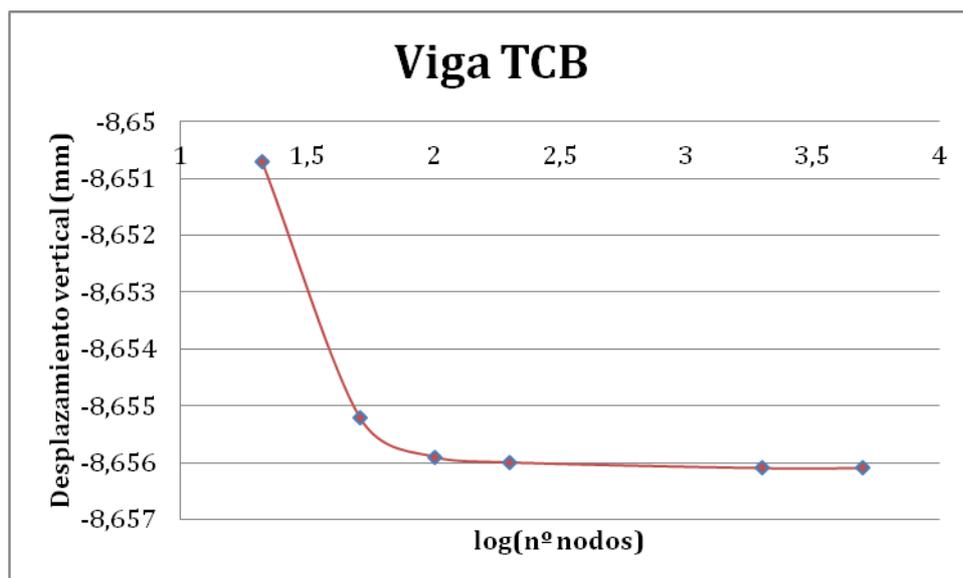


Fig. 90 Convergencia de resultados de TCB

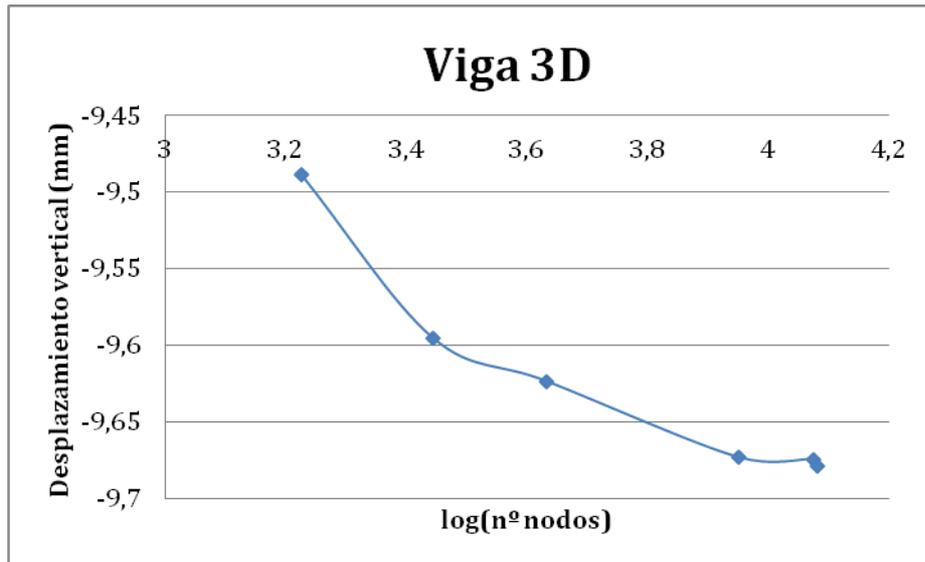


Fig. 91 Convergencia de resultados en 3D

II. PRESENTACIÓN DE RESULTADOS

En este apartado se muestran los resultados obtenidos calculando la estructura siguiendo la teoría de Timoshenko para vigas compuestas y analizándola como sólido 3-D. De esta manera comprobaremos si el programa calculado es óptimo también cuando la viga no es de sección rectangular.

En el análisis tridimensional, para evitar problemas locales y conseguir resultados óptimos, se tendrá en cuenta la carga puntual como una fuerza uniformemente distribuida en el alma de la pieza.

$$p = \frac{P}{A} = \frac{100000N}{1024.8mm^2} = \frac{97.58N}{mm^2} \quad (72)$$

dónde

- A: área de la sección
- P: carga puntual
- p: carga equivalente uniformemente repartida

Las gráficas obtenidas en GiD y que permiten comparar el comportamiento estructural de las vigas dependiendo del método en que se calcula son las que se muestran a continuación. Se presenta también, como hemos venido haciendo en los casos anteriores, una tabla con los valores de desplazamientos y tensiones de un punto de referencia, en este caso

se ha tomado la fibra inferior del extremo libre de la viga. Hay que apuntar, sin embargo, que el valor de la tensión tangencial en el caso tridimensional se ha tomado en el alma de la viga, ya que la contribución de las alas en cuanto al esfuerzo cortante es despreciable.

	Desp_x (mm)	Desp_y (mm)	Sigma_x (N/mm2)	Tau_xz (N/mm2)
Sólido 3D	1.2901	-9.6703	0	31.549
TCB	-1.2901	-8.6561	0	-36.7
Diferencia (%)	0	10.49	0	16.32

Tabla 14 Comparación de resultados

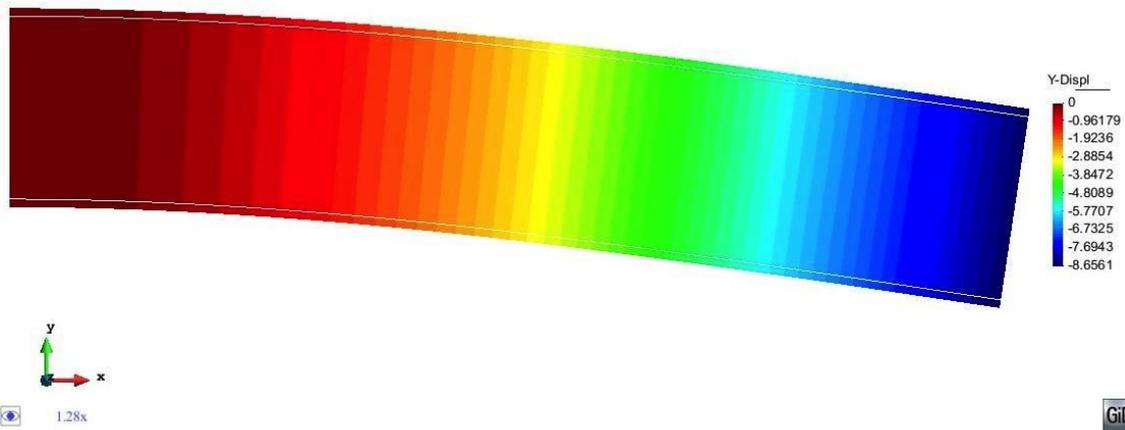


Fig. 92 Desplazamiento vertical TCB

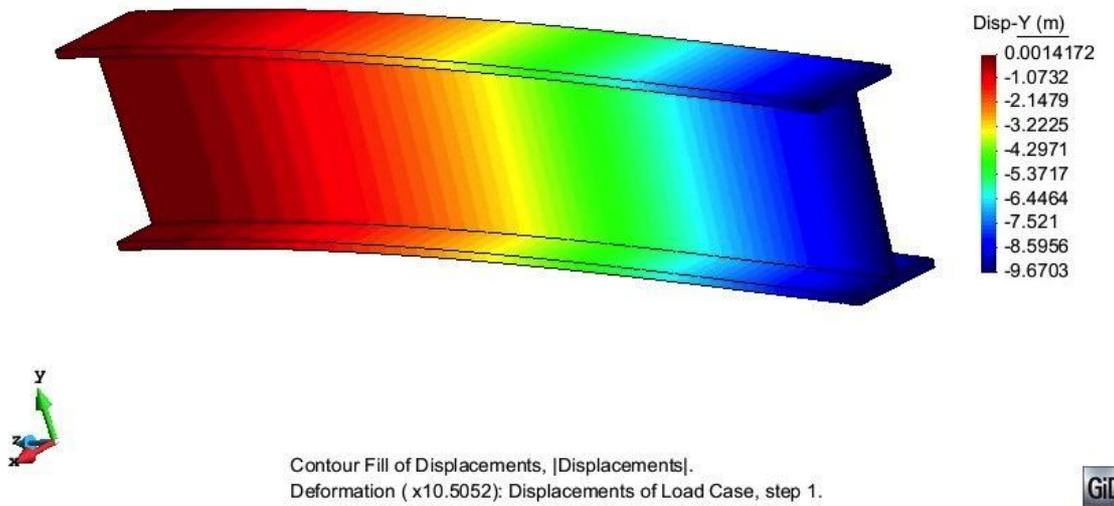


Fig. 93 Desplazamiento vertical 3D

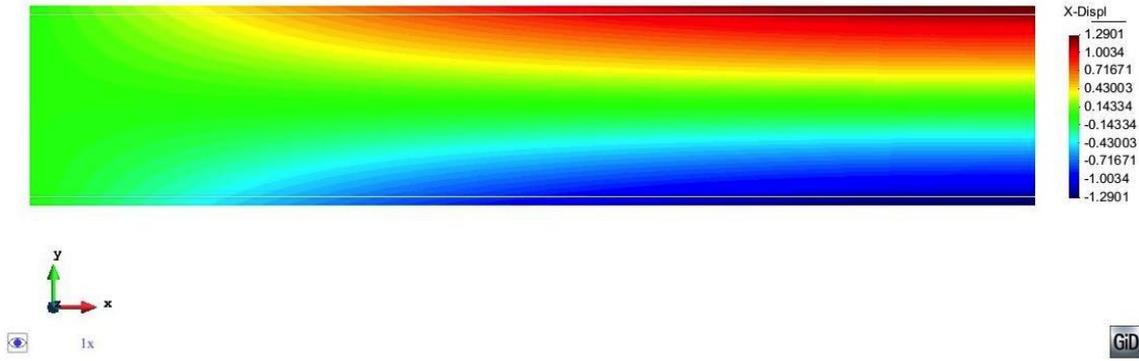
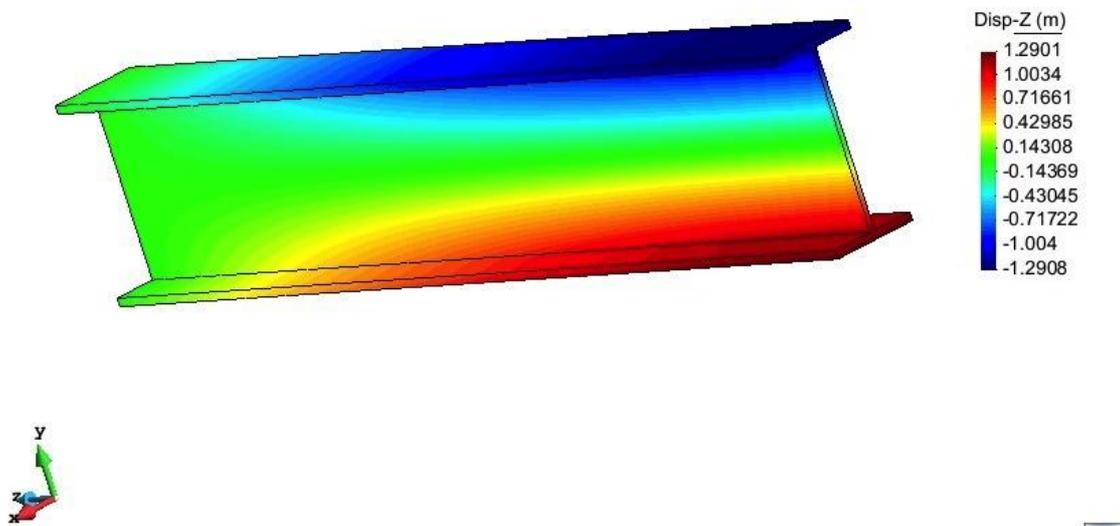


Fig. 94 Desplazamiento horizontal TCB



Contour Fill of Displacements, Disp-Z (m).

Fig. 95 Desplazamiento horizontal 3D

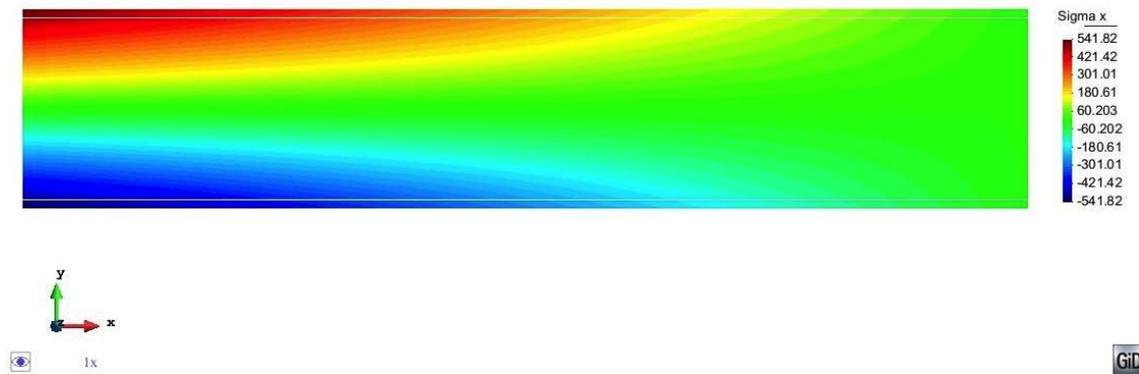


Fig. 96 Tensión normal TCB

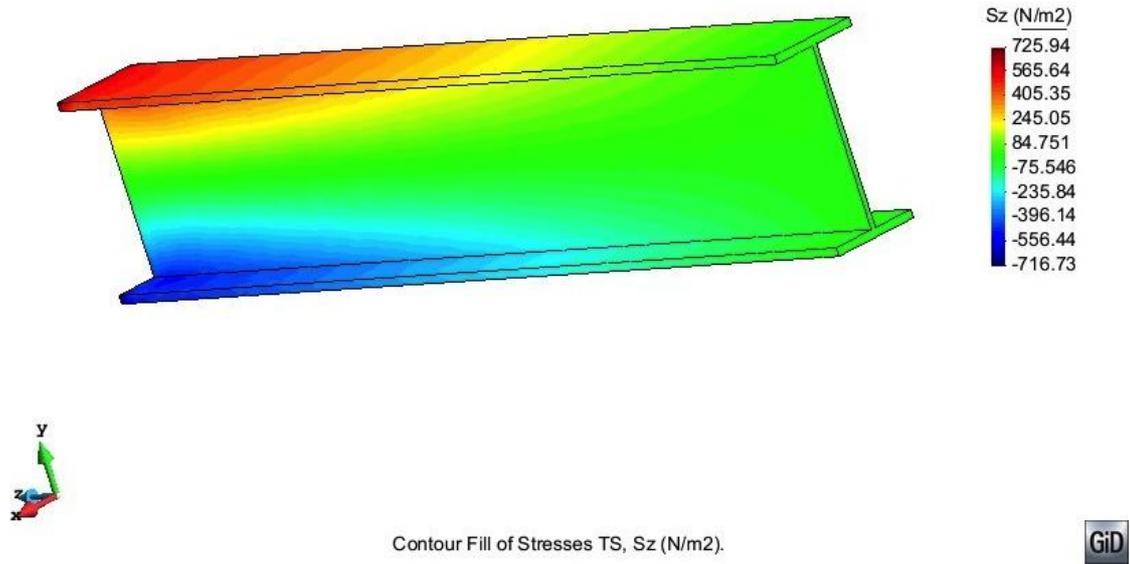


Fig. 97 Tensión normal 3D



Fig. 98 Tensión tangencial xz TCB

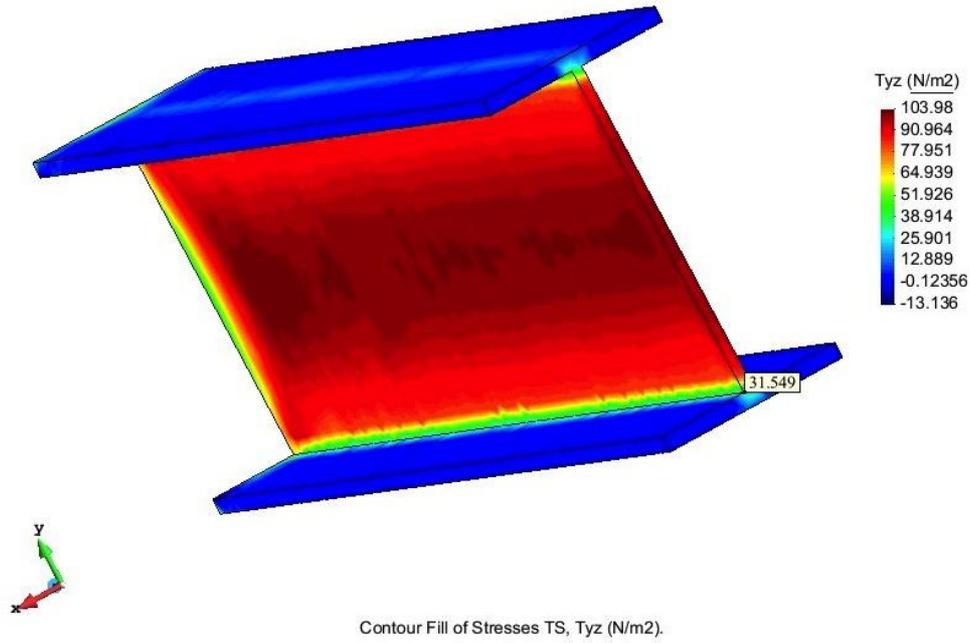


Fig. 99 Tensión tangencial xz 3D

La tensión tangencial toma un valor constante en toda la viga, en la teoría de Timoshenko, debido a que las almas no contribuyen a absorber esta tensión. Es por esto que, como se había comentado anteriormente, el valor en la pieza 3D se obtiene en la fibra inferior del alma de la viga.

Finalmente, a modo de observar que el comportamiento en las vigas con sección en doble T, calculadas según Timoshenko, obedece al mismo patrón que en secciones rectangulares, se aporta información gráfica del cortante en la viga. Se distingue que, como cabía esperar, el cortante es uniforme en cada capa.



Fig. 100 Cortante TCB

A continuación se muestra el tiempo que necesita la computadora para calcular la estructura dependiendo de la teoría escogida.

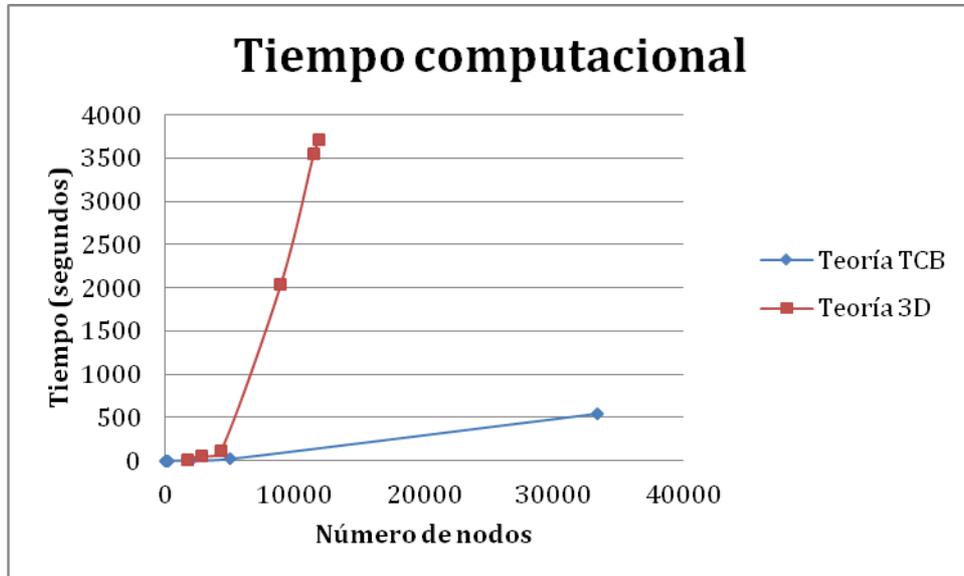


Fig. 101 Tiempo computacional según TCB y 3D

Se vuelve a poner de manifiesto la potencia del programa en MAT-fem y como el tiempo computacional necesario es bastante menor que el necesario si calculamos la estructura como sólido 3-D.

En el anejo 2 se presentan algunos ejemplos más. Teniendo en cuenta la diversidad de vigas que se han estudiado, de distintas secciones con más o menos capas de varios materiales, con condiciones de carga y de contorno diferentes, se pone de manifiesto que la teoría de Timoshenko para vigas compuestas representa fielmente el comportamiento de estas estructuras. Además, se comprueba también que la programación en MAT-fem se ha realizado correctamente.

6. CONCLUSIONES

El presente estudio se iniciaba explicando las aplicaciones tan positivas que tienen los materiales compuestos en el ámbito de la Ingeniería Civil, entre otros. Sin embargo, también apuntábamos la necesidad de mejorar las herramientas numéricas para calcular estructuras formadas por estos materiales.

A lo largo de la tesina, nos hemos centrado en las estructuras tipo viga formadas por distintas láminas y, en particular, se ha expuesto la teoría por elementos finitos según Timoshenko. Se ha implementado dicha teoría numéricamente mediante MAT-fem de manera que fuera posible comprobar su funcionalidad y eficiencia.

Hemos realizado distintos ejemplos y se ha demostrado que el programa describe fielmente el comportamiento real de este tipo de estructuras. Se ha llegado a esta conclusión a partir de comparar los resultados obtenidos mediante el programa de MAT-fem con el cálculo como sólido 3D de estas estructuras. Además, no solo se han obtenido los mismos resultados en cuanto a distribución de tensiones y desplazamientos sino que se ha consumido menos tiempo computacional.

Finalmente se ha presentado un ejemplo real, en el anejo 1, en el que se engloba todos los aspectos en los que se hace referencia en esta tesina. Por un lado, se corrobora la aplicabilidad de los materiales compuestos en la actualidad, sobretodo en el refuerzo de estructuras, debido a la alta relación resistencia/peso de estos materiales. Desde el punto de vista estructural, para el análisis de este caso se ha utilizado el programa de MAT-fem demostrando su eficiencia y utilidad.

7. BIBLIOGRAFÍA

- [1] E. Oñate, Analysis with the Finite Element Method. Linear Statics. Vol. 2 Beams, plates and shells. Editor Springer - CIMNE 2013.
- [2] E. Oñate, Cálculo de Estructuras por el Método de Elementos Finitos. Análisis estático lineal. Segunda edición, septiembre 1995.
- [3] S. Oller, Análisis y cálculo de estructuras de materiales compuestos. CIMNE, 2002.
- [4] Johnson, R.P., Composite structures of steel and concrete. 3rd edition, Oxford, 2004.
- [5] GiD customization manual. <http://gid.cimne.upc.es/support/manuals>
- [6] F. Zarate, E.Oñate, MAT-fem manual. CIMNE 2006.

ANEJO 1: MUELLE DE LA ENERGÍA

1. INTRODUCCIÓN

En este apartado se estudia un caso real de refuerzo de vigas mediante el uso de materiales compuestos. El problema en cuestión pretende reflejar globalmente las ideas expuestas en la presente tesina y, además, aportar un ejemplo actual y real del uso de este tipo de materiales en Ingeniería Civil.

Los datos necesarios para la realización del siguiente estudio han sido facilitados por Acciona Infraestructuras, la empresa encargada de redactar el informe de diseño del refuerzo para la reparación de la estructura objeto de estudio.

2. ANTECEDENTES

Este ejemplo está basado en el proyecto de urbanización del muelle de L'Energia en Barcelona. Se trata de un puente recto de dos vanos de 15+15 metros de longitud aproximada.

El paso tiene 9,10 metros de ancho y está resuelto mediante un tablero formado por 17 vigas prefabricadas de 0,72 m de canto junto con una losa armada de 0,25 m de espesor. El tablero descansa sobre estribos en sus extremos y sobre un soporte corrido central.

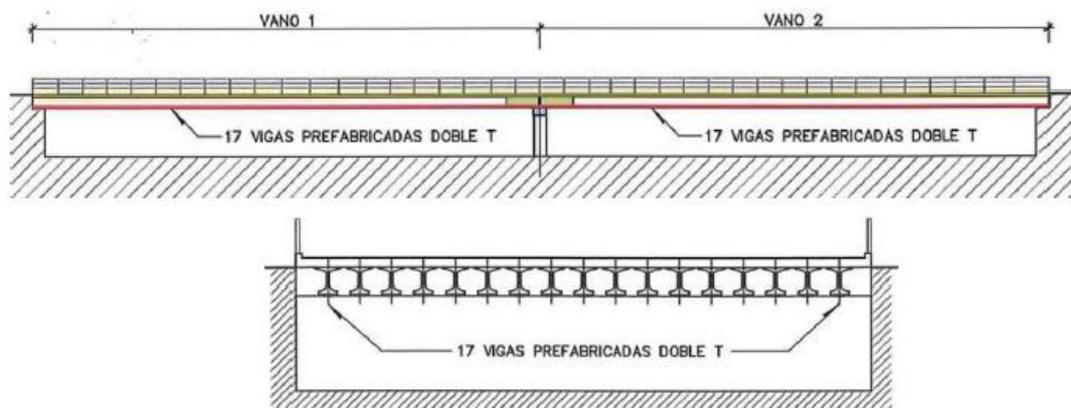


Fig. 102 Puente de la energía

Cada viga prefabricada se encuentra pretensada mediante 84 cables $2\phi 2,5$, en la siguiente figura se describe las características geométricas de la pieza:

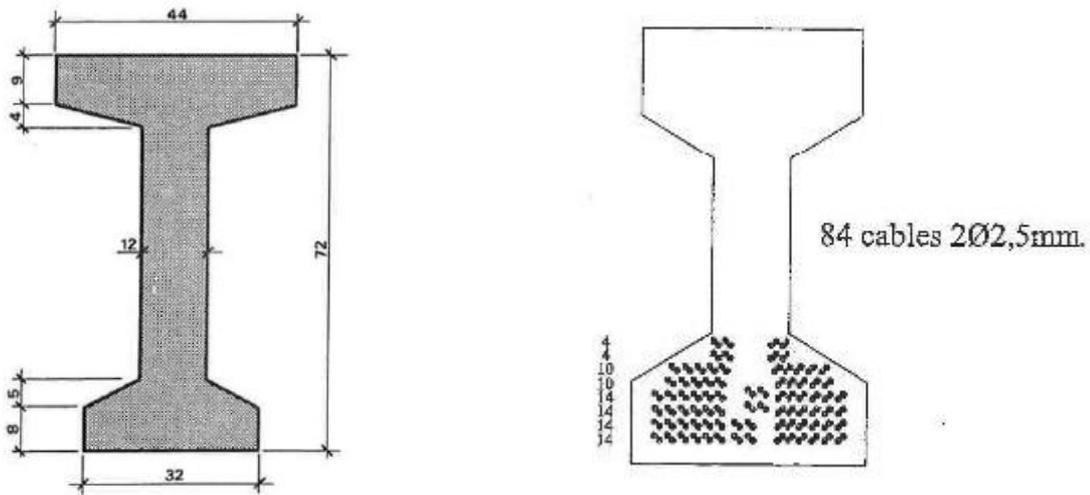


Fig. 103 Características de la sección

Cada cordón tiene un área de $4,91 \text{ mm}^2$ con lo que el área de pretensado es:

$$A_p = 84 \cdot 2 \cdot 4,91 = 824,88 \text{ mm}^2 \quad (73)$$

Los materiales empleados en el cálculo de la estructura han sido los siguientes:

- ✓ Hormigón in situ: HA-25 ($f_{ck} = 25 \text{ N/mm}^2$)
- ✓ Hormigón en vigas pretensadas: HA-35 ($f_{ck} = 35 \text{ N/mm}^2$)
- ✓ Acero pretensado: Y 1860 S ($f_{yk} = 1670 \text{ N/mm}^2$)

En cuanto a los coeficientes adoptados son, según el informe de cálculo:

- ✓ $\gamma_c = 1,76$ (Hormigón)
- ✓ $\gamma_s = 1,15$ (Acero)
- ✓ $\gamma_{fc} = 1,2$ (Lámina)

Por otro lado, necesitaremos las tensiones admisibles a compresión y tracción del hormigón:

$$\sigma_{c,adm} = 0,6 \cdot f_{ck}$$

$$\sigma_{T,adm} = 0,30 \cdot f_{ck}^{2/3} \quad (74)$$

	$\sigma_{c.adm}$ (MPa)	$\sigma_{T.adm}$ (MPa)
HA-25	15	2,56
HA-35	21	3,21

Tabla 15 Tensión a tracción y compresión admisible según el tipo de hormigón

Se consideran las siguientes hipótesis de carga teniendo en cuenta la sección tipo mostrada a continuación:

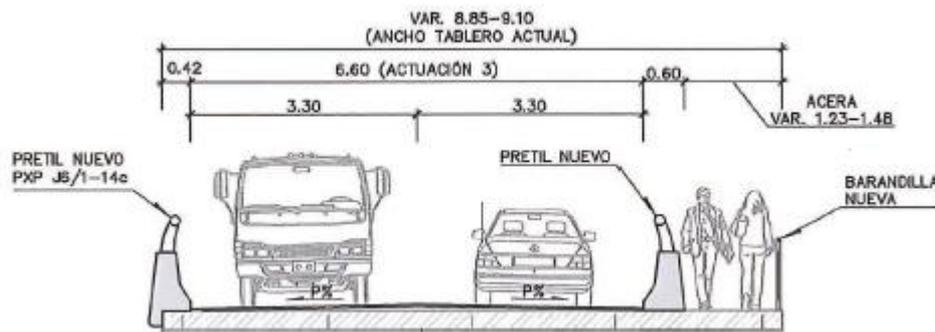


Fig. 104 Sección tipo del puente de la energía

- ✓ Peso específico del hormigón armado o pretensado: 25KN/m³
- ✓ Peso específico de la mezcla bituminosa: 23KN/m³
- ✓ Pretil de hormigón: 10,35 KN/m
- ✓ Solado zona peatonal: 23KN/m³
- ✓ Sobrecarga repartida debida al tráfico: 4KN/m²
- ✓ Sobrecarga puntual debida al tráfico: 600KN

Se conoce que la viga más solicitada se encuentra bajo el pretil de borde, cuya ley de momento flector es:

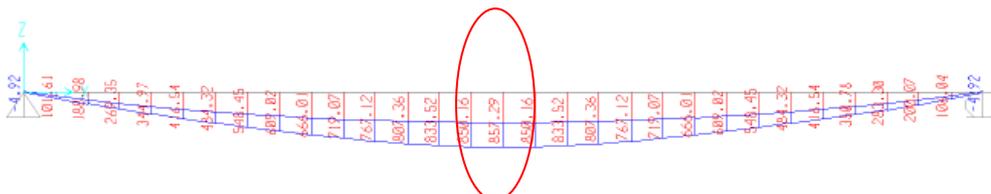


Fig. 105 Envolvente de momentos en la viga más solicitada

- Momento solicitante máximo = 857,29 KN/m

Las vigas que componen el puente de la energía deben reforzarse debido a la posibilidad de que se hayan cortado algunos de los cables de armado de la estructura, se desconoce el número exacto de cables cortados por eso se harán distintas hipótesis.

3. REFUERZO CON MATERIALES COMPUESTOS

El refuerzo se plantea con objeto de compensar la eventual pérdida de armado y obtener un momento flector resistente superior al de la sección original. El refuerzo de las vigas se realizará mediante adhesión de materiales compuestos armados a base de fibra de carbono. La geometría de las láminas que se utilizarán y los datos técnicos de ésta son:

Ancho (mm)	Espesor(mm)
50	1.2
80	1.2
100	1.2

Tabla 16 Dimensiones de las láminas

Característica	Unidades	Valor
Densidad	g/cm ³	1,6
Volumen de fibra	%	70
Módulo de elasticidad medio	GPa	170
Resistencia a tracción media:	MPa	3100
Elongación a rotura media	%	1,9

Tabla 17 Propiedades de la lámina de fibra de carbono

4. ANÁLISIS ESTRUCTURAL

Hipótesis en el análisis de la estructura

El análisis se va a centrar en la viga más solicitada teniendo en cuenta la proporción de losa que afecta a ésta. El pretensado se va a modelar como una capa de acero rectangular de las mismas características y con el mismo centroide y la misma área que la totalidad del pretensado original.

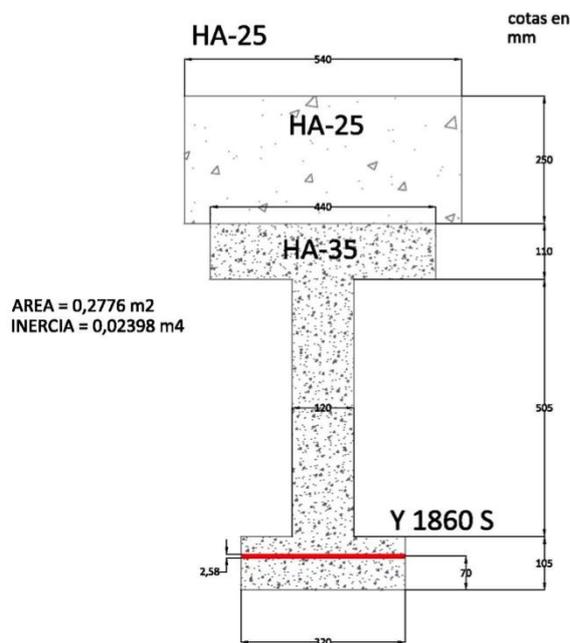


Fig. 106 Sección estructura muelle de la energía

Para calcular la viga con la teoría de Timoshenko para vigas compuestas tenemos que separar la sección de la pieza en distintas capas. Se observa en la figura anterior cómo se han eliminado los retranqueos en la viga tomando un valor medio del espesor, de manera que tanto las alas como el alma tengan sección rectangular. A continuación se describen las propiedades de cada capa:

CAPA	YOUNG(N/mm ²)	POISSON	DENSIDAD (N/m ³)	ESPEJOR (mm)	ANCHO (mm)
1	29778,88	0,2	0	33,71	320
2	200000	0,3	0	2,58	320
3	29778,88	0,2	0	68,61	320
4	29778,88	0,2	0	505	120
5	29778,88	0,2	0	110	440
6	27264,04	0,2	0	250	540

Tabla 18 Características de la sección dividida en capas

En la tabla anterior se observa como se ha tomado densidad nula ya que el peso propio de la estructura ya está incluido en la carga exterior.

Finalmente tenemos la sección de la estructura dividida en capas de manera que estamos en disposición de analizarla como viga compuesta de Timoshenko.

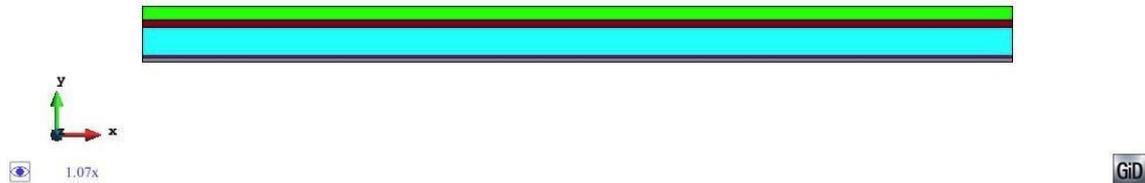


Fig. 107 Viga en TCB

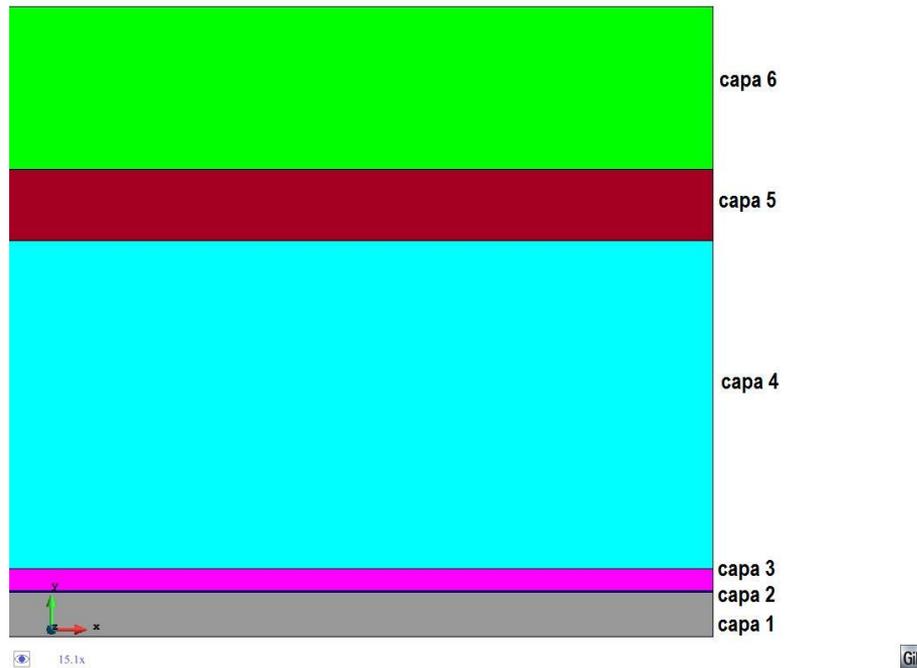


Fig. 108 Zoom de la viga donde se distinguen las capas

Fuerza de pretensado

Cómo se desconoce la fuerza a la que se tesaron los cables, vamos a seguir el siguiente proceso con tal de obtener dicha fuerza. Vamos a cargar la viga con una carga uniformemente repartida que produzca un momento máximo de 857,29 KNm en la sección central de la estructura, ya que es éste el esfuerzo debido a las cargas externas a la que se ve sometida la viga. A continuación, se estudiarán las tensiones en las fibras extremas de la estructura y se observará si superan los valores admisibles. Si los superan, se calculará la fuerza de pretensado necesaria para conseguir que esta tensión sea adecuada.

La carga uniformemente repartida que produce un momento de 857,29KNm en el centro de la viga es:

$$M = \frac{ql^2}{8} \quad (75)$$

$$q = \frac{8M}{l^2} = \frac{8 \cdot 857,29}{15^2} = 30,48 \text{ KNm} \quad (76)$$

Cargamos la viga compuesta por las distintas capas, incluido el acero que simula el pretensado, pero con los cables sin tesar.

Se ha discretizado la viga mediante una malla de 1876 nodos, los cuales se consideran suficientes para obtener resultados óptimos.

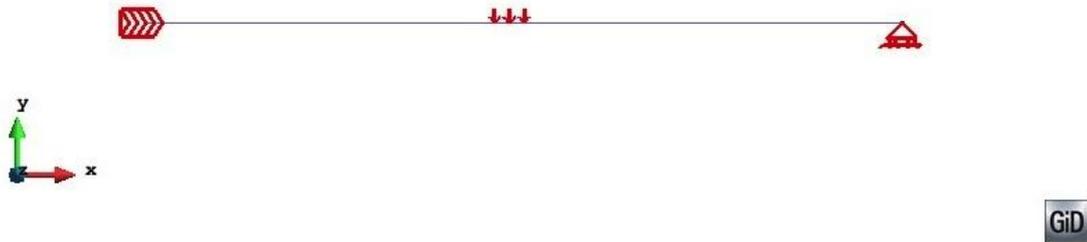


Fig. 109 Condiciones de contorno

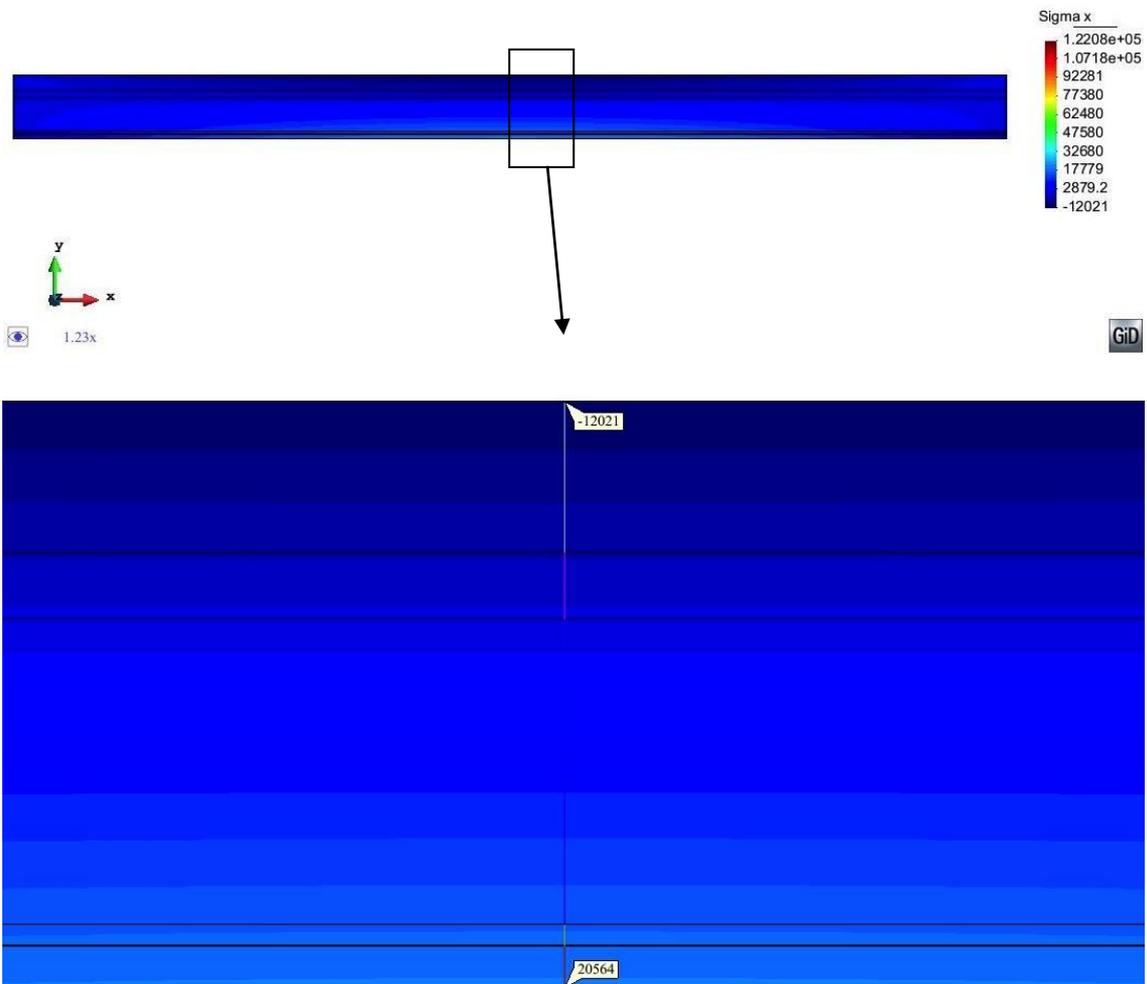


Fig. 110 Tensiones en la viga

En la imagen anterior las tensiones a tracción son positivas y las de compresión negativas. Se observa que las tensiones máximas tanto a compresión como a tracción en la sección más solicitada superan los valores admisibles. Hay que tener en cuenta que la tensión a compresión se produce en el hormigón HA-25 y la de tracción en el HA-35.

$$\begin{aligned}\sigma_{\text{comp}} &= 20,56 \text{ MPa} > 15 \text{ MPa} \\ \sigma_{\text{tracc}} &= 12,02 \text{ MPa} > 3,21 \text{ MPa}\end{aligned}$$

Hemos llegado a la conclusión que ya preveíamos, se necesita pretensar la sección para disminuir las tensiones en las fibras extremas. Dimensionaremos la fuerza de pretensado para la condición más restrictiva, es decir, que la tensión en la fibra inferior de la sección sea 3,21 MPa.

$$\sigma = \frac{M}{I} \cdot (z - z_G) \quad (77)$$

dónde σ es la tensión normal, M el momento producido, I la inercia de la sección, z es la coordenada vertical de la fibra y z_G es la coordenada vertical del centro de gravedad.

Teniendo en cuenta que el momento que solicita la sección es el momento debido a las cargas más el momento debido al pretensado la ecuación quedaría de la siguiente manera:

$$3210 = \frac{857,29 - M_p}{0,02398} \cdot 0,610 \quad (78)$$

dónde las unidades que se han utilizado son KN y m. No hay que olvidar que la inercia de la sección se ha calculado despreciando el acero, realmente tendríamos que tenerlo en cuenta haciendo uso del coeficiente de equivalencia $n = \frac{E_p}{E_h}$. Sin embargo, realizando esta simplificación no nos desviamos en exceso de la solución exacta y nos facilita la comparación con el modelo numérico.

Realizando el cálculo anterior el momento de pretensado necesario es:

$$M_p = 731,10 \text{ KNm} \quad (79)$$

Por tanto, la fuerza de pretensado se puede obtener dividiendo este momento entre la excentricidad e :

$$P = \frac{M_p}{e} = \frac{731,10}{0,610 - 0,07} = 1353,89 \text{ KN} \quad (80)$$

Vamos a verificar que esta fuerza no es superior al límite elástico del acero de pretensar Y1860S.

$$f_{yp} = 1670 \text{ MPa}$$

$$\sigma_p = \frac{P}{A_p} = \frac{1353,89 \cdot 10^3}{824,88} = 1641,32 \text{ MPa} < 1670 \text{ MPa} \quad (81)$$

Para conocer la fuerza que aporta cada cable (F_{p_a}) se divide la fuerza de pretensado P entre el número de cables total:

$$F_{p_a} = \frac{1353,89}{168} = 8,06 \text{ KN} \quad (82)$$

Ahora volvemos a calcular la viga teniendo en cuenta el momento de pretensado en la viga, en la siguiente imagen se observa como las tensiones extremas se han reducido de forma que ahora sí son menores a las admisibles:

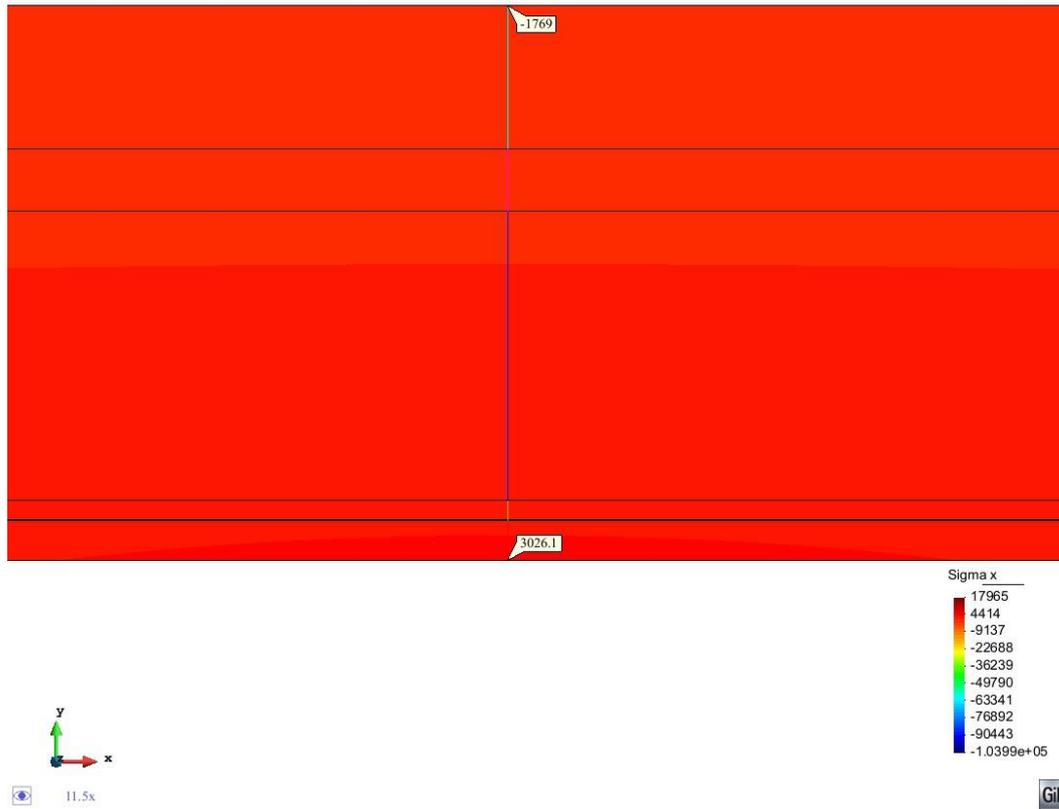


Fig. 111 Tensiones en las fibras extremas de la pieza teniendo en cuenta el pretensado

Por tanto, la viga original con todos los cables funcionando adecuadamente estaría sometida a unas tensiones a compresión y a tracción menor a las admisibles. Además, es posible que el proyecto de la estructura se diseñara teniendo en cuenta un pretensado parcial, con lo que la tensión de tracción podría llegar a ser mayor sin causar problemas en la estructura. Supondremos un grado de pretensado parcial $k=3$, de manera que las tensiones de tracción podrían llegar al valor de 9,61 MPa sin necesidad de crear problemas en el hormigón.

Análisis de la sección original

Vistas las tensiones producidas debido a las cargas exteriores y al pretensado, es necesario observar cual es la resistencia última de la sección original de la viga, es decir con todos los cables trabajando correctamente y a tensión última.

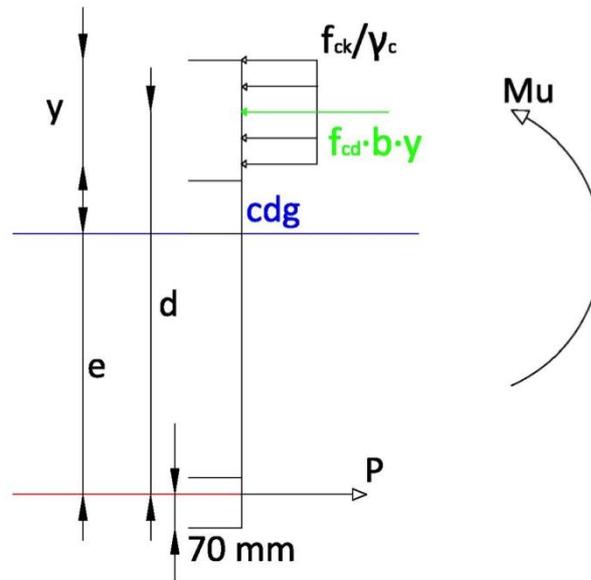


Fig. 112 Esquema de fuerzas para obtener la resistencia última de la sección

dónde:

- ✓ f_{ck} es la resistencia a compresión del hormigón HA-25: 25N/mm²
- ✓ γ_c es el coeficiente de seguridad del hormigón: 1,76
- ✓ $f_{cd} = f_{ck}/\gamma_c$
- ✓ b es el ancho de la losa: 540mm
- ✓ y es la profundidad de la fibra neutra
- ✓ d es la distancia entre el centroide del acero pretensado y la fuerza de compresión.
- ✓ e es la excentricidad: 520mm
- ✓ P es la fuerza última del acero:

$$P = \frac{f_{yp}}{\gamma_s} A_p \quad (83)$$

con f_{yp} la tensión última del acero, 1670 N/mm², γ_s el coeficiente de seguridad del acero, 1,15 y A_p el área de acero, 824,88mm².

Haciendo equilibrio de fuerzas horizontales encontramos la posición de la fibra neutra:

$$\sum F_h = 0$$

$$A_p f_{yp} = b f_{cd} y$$

$$y = \frac{A_p f_{yp} / \gamma_s}{b f_{cd}} \quad (84)$$

$$y = \frac{824,88 \cdot 1670 / 1.15}{540 \cdot 25 / 1.76} = 156,17 \text{ mm}$$

Haciendo equilibrio de momentos encontramos el momento resistente de la sección, lo haremos respecto el centro del bloque de compresiones:

$$M_{\text{resistente}} = P \cdot \left(970 - 70 - \frac{y}{2} \right) \quad (85)$$

donde 970 representa el canto total de la sección en mm y 70 la distancia entre la fibra inferior de la sección y el centroide del área de acero, también en mm. Con lo que:

$$M_{\text{resistente}} = \frac{1670}{1,15} 824,88 \cdot \left(970 - 70 - \frac{156,17}{2} \right) \cdot 10^{-6} = 984,55 \text{ KNm}$$

Como criterio de diseño sobre las vigas se ha planteado que todas ellas superen aproximadamente el mismo esfuerzo resistente. El esfuerzo resistente mínimo a resistir ha sido fijado en 900KNm. Teniendo en cuenta que el momento último solicitante en la viga es 857,29 KNm, aplicando este criterio de diseño contamos con un coeficiente de seguridad adicional. Por tanto, se observa claramente como la sección original cumplía ampliamente con este requisito.

Análisis del refuerzo de la estructura

El problema se ha producido al detectar que alguno de los cables está defectuoso. Como se desconoce el número exacto de éstos, la empresa encargada de reforzar la estructura realizó los siguientes casos, reforzando tal y como se describe a continuación:

Número de alambres afectado	Refuerzo necesario
6	No se considera refuerzo
12	1 lámina 50x1.2mm
16	1 lámina 80x1.2mm
25	2 láminas 100x1.2mm
36	3 láminas 100x1.2mm

Tabla 19 Refuerzo según el número de alambres afectado

El procedimiento que seguiremos para estudiar los distintos casos será el siguiente. Primero realizaremos un análisis de las tensiones en el hormigón haciendo uso del programa de MAT-fem para vigas compuestas de Timoshenko y, posteriormente, estudiaremos la repercusión que tiene, en el momento resistente último de la sección, el hecho de colocar o no el refuerzo.

Hay que tener en cuenta que las características de la sección varían al disminuir el número de cables pretensados. Evidentemente se reducirá tanto el área de pretensado como el momento producido por éste. Sin embargo, se considera constante la distancia desde la fibra inferior de la sección al centroide del pretensado, 70mm, al igual que la excentricidad de éste, $e=520\text{mm}$.

La nueva área de pretensado se obtiene reduciendo el número de cables que trabajan correctamente, es decir:

$$A_p = (168 - n) \cdot 4,91\text{mm}^2 \quad (86)$$

dónde 168 son el número de cables del pretensado original con una superficie de $4,91\text{mm}^2$ cada uno, y n es el número de cables afectados. Como conocemos la aportación a la fuerza total de pretensado de cada cable, según la expresión (82) es de $8,06\text{KN}$, se obtiene la fuerza de pretensado (P) y el momento que produce (M_p) de la siguiente manera:

$$P = (168 - n) \cdot 8,06 \quad (87)$$

$$M_p = P \cdot e \quad (88)$$

En la siguiente tabla se presentan los nuevos valores:

Alambres afectados	Área de Pretensado (mm²)	Momento de pretensado (KNm)
6	795,42	705,09
12	765,96	678,97
16	746,32	661,56
25	702,13	622,39
36	648,12	574,52

Tabla 20 Características del problema en función del número de cables afectado

Ahora ya podemos introducir los nuevos datos en GiD y realizar los cálculos necesarios para obtener las nuevas tensiones en el hormigón. Es importante apuntar que el área de pretensado se ha tenido en cuenta como una capa rectangular, como ya hicimos anteriormente. Al variar el área de acero también varía el espesor de ésta capa ya que se mantiene constante el ancho de 320mm. Los nuevos espesores según los cables afectados se presentan a continuación:

Alambres afectados	Espesor de la capa de acero (mm)
6	2,49
12	2,39
16	2,33
25	2,19
36	2.03

Tabla 21 Espesor de la capa de acero según el número de cables afectado

Vamos a analizar las tensiones en la sección más solicitada y en la fibra de hormigón más traccionada y comprimida. Se hace notar que el número de capas que se asignarán en el preproceso de GiD aumentará ya que se tiene que añadir las láminas del refuerzo. En el caso de que sea necesario refuerzo, mostraremos las tensiones antes y después de colocar el material compuesto, de manera que veamos cual es la aportación de éste. Los valores obtenidos se presentan a continuación:

Alambres afectados	Refuerzo	Tensión a tracción (MPa)	Tensión a compresión (MPa)
6	Sin lámina	3,66	2,136
12	Sin lámina	4,3	2,51
	1 lám. 50x1.2mm	4,27	2,49
16	Sin lámina	4,73	2,69
	1 lám. 80x1.2mm	4,68	2,74
25	Sin lámina	5,70	3,31
	2 lám. 100x1.2mm	5,55	3,27
36	Sin lámina	6,89	3,99
	3 lám 100x1.2mm	6,63	3,92

Tabla 22 Tensiones en función del número de cables afectado y del refuerzo que se utiliza

Se observa en la tabla anterior que el refuerzo no aporta una disminución importante en las tensiones. Sin embargo, el aumento de tensión que se produce debido a la reducción de acero de pretensar no es suficientemente grande como para producir problemas en el hormigón. Como vimos anteriormente, la tensión a tracción puede llegar hasta los 9,6 MPa, teniendo en cuenta pretensado parcial, y las tensiones a compresión podrían llegar hasta los 15MPa.

Sin embargo, la adhesión del refuerzo en la parte inferior de la viga nos aportará resistencia estructural y es éste el objetivo principal de esta actuación en el puente de la energía.

Siguiendo las expresiones (83, 84, 85) se calcula el momento resistente de la sección dependiendo del número de alambres afectados. En la siguiente tabla se muestran los resultados:

Alambres afectados	Posición fibra neutra y (mm)	Momento resistente (KNm)
6	150,59	952,61
12	145,01	920,43
16	141,29	898,84
25	132,93	849,88
36	122,70	789,32

Tabla 23 Posición de la fibra neutra y momento resistente de la sección

Se observa en la tabla anterior como el momento resistente disminuye a medida que aumenta el número de cables afectados. Tal y como veíamos con anterioridad, el momento resistente mínimo se ha fijado en 900 KNm. Se justifica, por tanto, la necesidad de realizar refuerzos sobre la viga. No será necesario reforzar cuando sean 6 los cables que no trabajen pero a medida que aumenta este número la cantidad de material compuesto debe aumentar, tal y como se preveía por la empresa encargada del diseño (tabla 19).

Para obtener el momento resistente último de la sección con el refuerzo, se realizará el mismo procedimiento que seguíamos en la sección original pero teniendo en cuenta la fibra de material compuesto adherida en la parte inferior de la viga. Por tanto, el esquema será el siguiente:

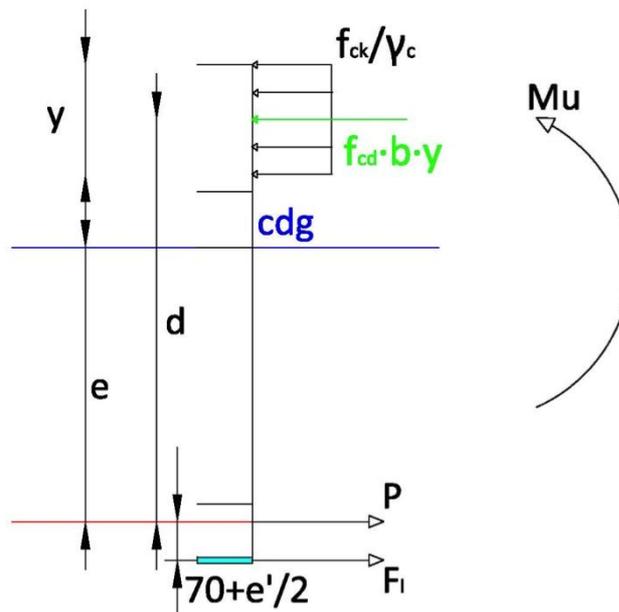


Fig. 113 Esquema momento resistente con la lámina adherida

donde e' es el espesor de la lámina o láminas del refuerzo y F_I es la fuerza última del material compuesto de fibra de carbono, que se obtiene según la expresión siguiente:

$$F_I = \frac{f_{fc,t}}{\gamma_{fc}} A_I \quad (89)$$

con $f_{fc,t}$ la resistencia a tracción de la lámina, 3100 MPa, γ_{fc} coeficiente de seguridad del compuesto, 1,2, y A_I es el área del refuerzo utilizado. Realizando equilibrio de fuerzas y de momentos se obtiene la posición de la fibra neutra y el momento resistente de la sección.

$$y = \frac{\left(\frac{A_p f_{yp}}{\gamma_s}\right) + \left(\frac{A_I f_{fc,t}}{\gamma_{fc}}\right)}{bf_{cd}} = \frac{F_T}{bf_{cd}} \quad (90)$$

donde F_T representa la fuerza total a tracción, sumando la contribución del acero y de la lámina.

$$M_{resistente} = F_T \left(900 - \frac{y}{2} - \frac{e'}{2} \right) \quad (91)$$

A continuación se muestran los valores obtenidos:

Alambres afectados	A_p (mm ²)	A_I (mm ²)	e' (mm)	y (mm)	Momento resistente (KNm)
12	765,96	60	1,2	165,22	1035,12
16	746,32	96	1,2	173,63	1082,19
25	702,13	240	2,4	213,76	1298,45
36	648,12	360	3,6	243,95	1452,46

Tabla 24 Momento resistente de la sección reforzada en función del número de alambres afectado

Con el objetivo de comparar y observar los resultados obtenidos después de reforzar la estructura, se adjunta una gráfica que muestra el momento resistente de la pieza dependiendo del número de alambres afectados. Esta relación se calcula con la viga sin reforzar y reforzada, de manera que se visualiza claramente la eficiencia del material compuesto:

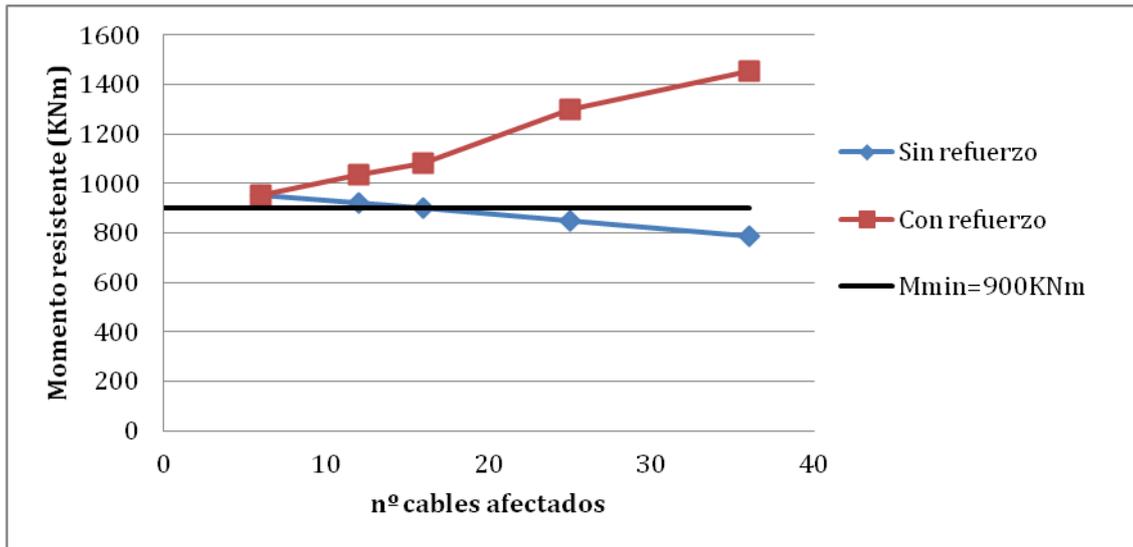


Fig. 114 Momento resistente según los cables afectados y el refuerzo

Se pueden obtener varias conclusiones de la gráfica anterior. Por un lado, se justifica claramente la necesidad de reforzar la viga a partir de 12 cables afectados, ya que cuando hay 6 cables defectuosos el momento resistido es superior al mínimo, fijado en 900 KNm. Cuando son 12 los cables el momento es sensiblemente superior al mínimo con lo que reforzarlo nos proporciona un mayor coeficiente de seguridad. Por otro lado, a medida que aumenta el número de cables, se debe proporcionar un mayor refuerzo ya que la incerteza en cuanto a los cables defectuosos causa peores consecuencias en un rango alto de alambres afectados que cuando nos movemos en número pequeños.

Con este ejemplo se ha observado claramente la eficiencia de los materiales compuestos, ya que una lámina de dimensiones relativamente pequeñas respecto a las dimensiones de la viga proporciona una resistencia adicional a la estructura realmente apreciable. La facilidad de montaje en obra de estos refuerzos hace que estas actuaciones sean muy competitivas respecto a otros modos de reforzar. Además, se ha podido comprobar la utilidad del programa de MAT-fem para vigas compuestas de Timoshenko en un caso real.

A continuación se presentan varias fotografías donde se visualiza el refuerzo de las vigas del puente de la energía.



Foto 1 Lámina de 50x1,2mm



Foto 2 Láminas de 80x1,2mm



Foto 3 Láminas de 100x1,2mm

En las fotografías anteriores se observa las diferentes láminas adheridas en la parte inferior de la viga con una pintura blanca, dependiendo de la necesidad de cada viga el refuerzo se realiza mediante láminas de 50, 80 o 100mm.

ANEJO 2: EJEMPLOS COMPARANDO TCB Y 3D

En el presente anejo se muestran dos ejemplos más, distintos a los que se han visto en el capítulo 5 de este estudio, a modo de cerciorar la eficiencia del programa calculado en MAT-fem para vías compuestas.

I. VIGA BIEMPOTRADA CON CARGA PUNTUAL EN EL CENTRO

La manera de proceder en la presentación de resultados será la misma que seguimos en el capítulo 5. Es decir, se calculará la estructura mediante la teoría de Timoshenko, y posteriormente se compararán resultados con el cálculo tridimensional. El ejemplo objeto de estudio es una viga biempotrada de longitud L con una carga puntual (P) en el centro de ésta. Los datos necesarios para el cálculo son los siguientes:

- ✓ $L = 120\text{mm}$; $h=10\text{mm}$; $b=2\text{mm}$ dónde h y b son el canto y ancho de la viga
- ✓ $P = 1\text{N}$
- ✓ Características de la sección:

CAPA	YOUNG(N/mm2)	POISSON	DENSIDAD (N/m3)	ESPEJOR (mm)	ANCHO (mm)
1	2.19·10 ⁵	0.25	0	2	2
2	7.3·10 ⁵	0.25	0	2	2
3	2.19·10 ⁵	0.25	0	2	2
4	7.3·10 ⁵	0.25	0	2	2
5	2.19·10 ⁵	0.25	0	2	2

Tabla 25 Datos de los materiales y dimensiones de la sección de la viga

SOLUCIÓN MEDIANTE TIMOSHENKO COMPOSITE BEAMS

La viga se divide en elementos unidimensionales de dos nodos cada uno, sin embargo la solución se muestra en 2D mediante elementos cuadriláteros de cuatro nodos cada uno.



Fig. 115 Condiciones de contorno de la viga

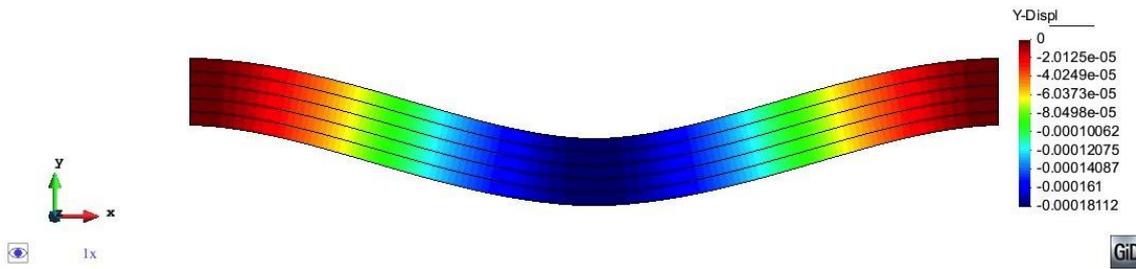


Fig. 116 Deformada de la viga

Convergencia de resultados

En la siguiente gráfica se observa cómo el desplazamiento vertical en el centro de la viga va variando en función del número de nodos. Se observa cómo a medida que la malla es más densa, el valor converge hacia la solución óptima. La gráfica de convergencia se presenta en escala logarítmica y viene acompañada de la tabla con los valores numéricos.

nodos	log(nodos)	desplazamiento vertical
		(mm)
25	1,40	-0,00017997
81	1,91	-0,00018102
121	2,08	-0,00018107
601	2,78	-0,00018112
1201	3,08	-0,00018112

Tabla 26 Valores de convergencia

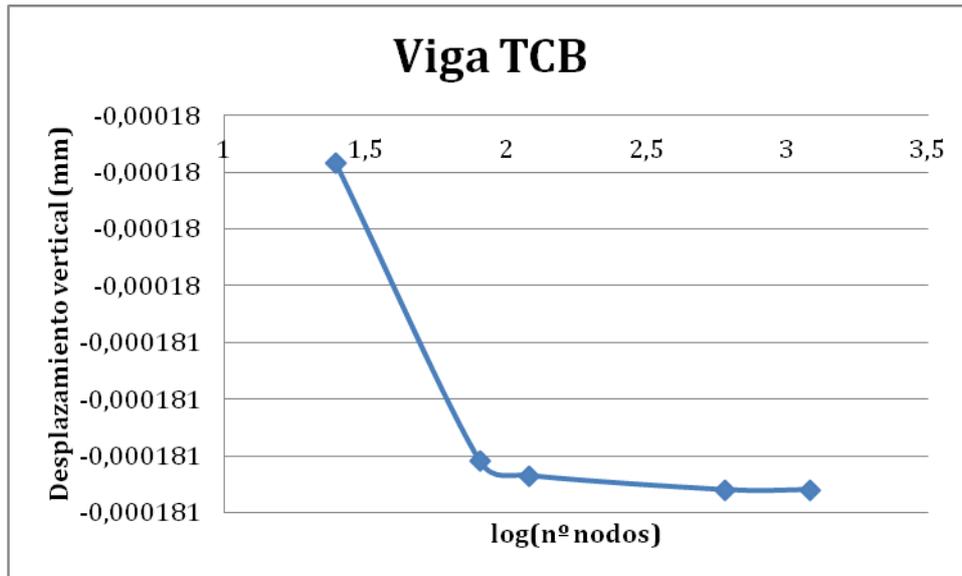


Fig. 117 Gráfica de convergencia

Presentación de la solución obtenida

A continuación se muestran los resultados obtenidos mediante la malla de 1201 nodos. Se muestra una tabla con los resultados obtenidos en la fibra inferior de la primera capa, en el centro de la viga. Además, se adjuntan las soluciones más representativas mediante gráficos:

Desp_x (mm)	Desp_y (mm)	Giro	Qy (N)	Mz (Nmm)	Sigma_x (N/mm ²)	Tau_xz (N/mm ²)
0	-0.0001811	0	0.051724	3.9521	0.30246	0

Tabla 27 Valores obtenidos según TCB

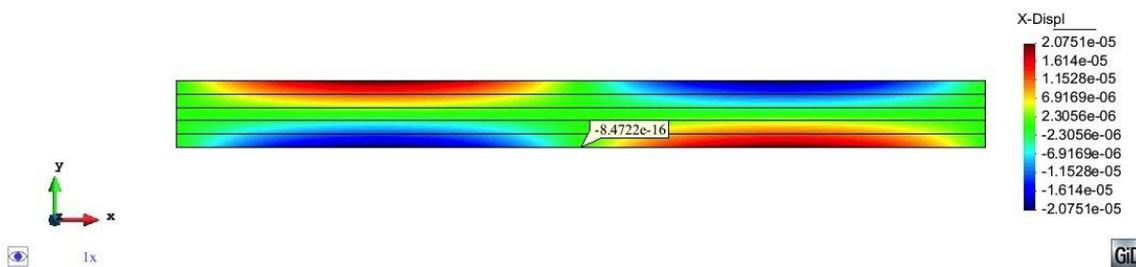


Fig. 118 Desplazamiento horizontal (mm)

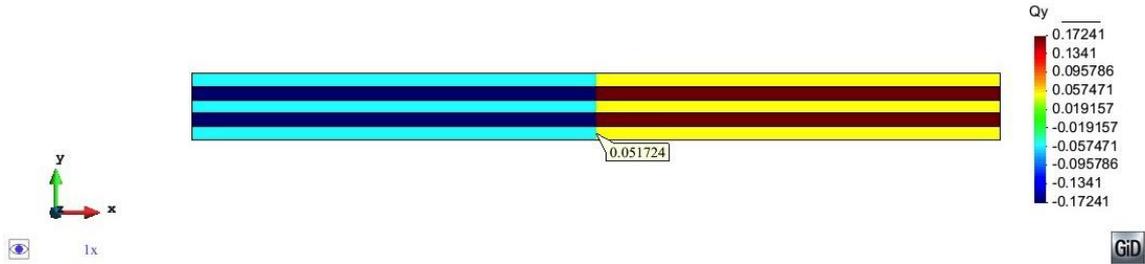


Fig. 119 Esfuerzo cortante (N)

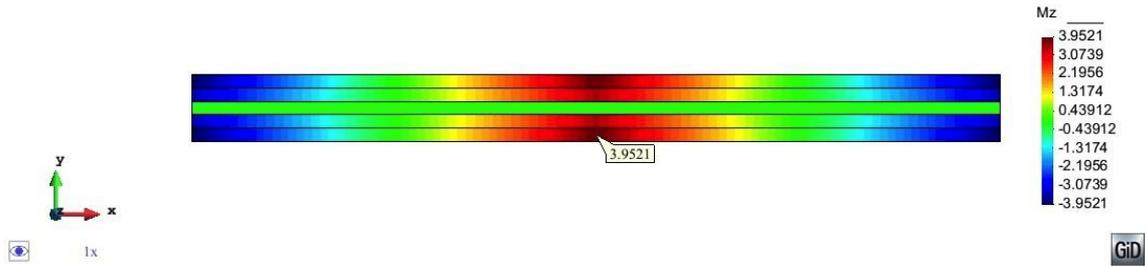


Fig. 120 Momento flector (KNm)

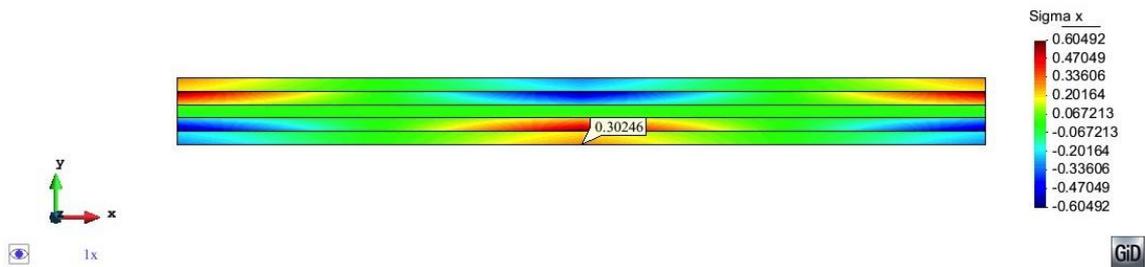


Fig. 121 Tensión normal (N/mm^2)

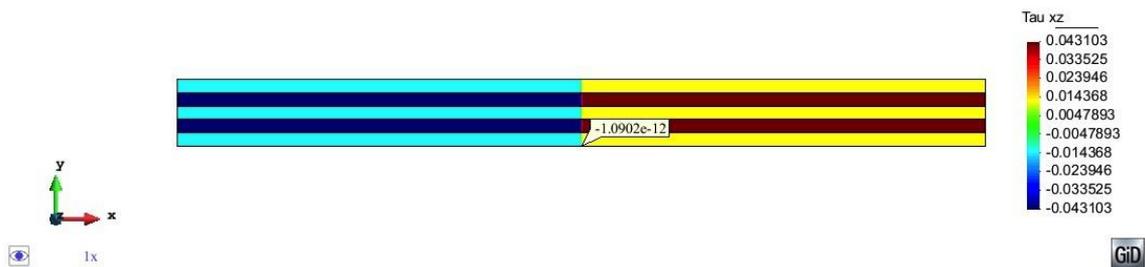


Fig. 122 Tensión tangencial xz (N/mm^2)

SOLUCIÓN MEDIANTE SÓLIDO 3D

La viga se divide en elementos tetraédricos de cuatro nodos cada uno. Tanto la geometría inicial cómo los resultados se muestran en tres

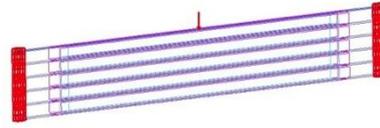
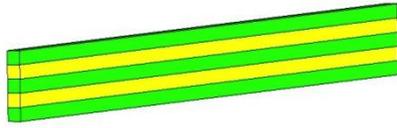
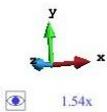
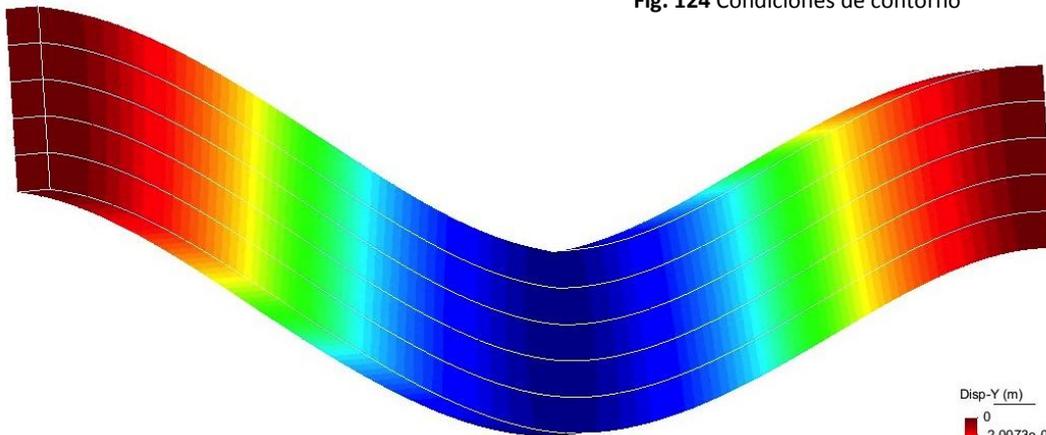


Fig. 123 Materiales de la viga



Fig. 124 Condiciones de contorno

dimensiones.



Areas coloreadas de Displacements, Disp-Y (m).
Deformación (x66663.2): Displacements de Load Case, paso1.

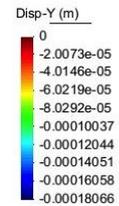


Fig.125 Deformada (mm)

Convergencia de resultados

Se presenta a continuación la tabla con los valores de convergencia y la gráfica que representa la tendencia hacia la solución óptima.

nodos	log(nodos)	desplazamiento vertical
-------	------------	-------------------------

		(mm)
531	2,73	-0,00014381
989	2,99	-0,00015688
2850	3,45	-0,00017566
5160	3,71	-0,00017939
7202	3,86	-0,00017956
8935	3,95	-0,00018003
10284	4,01	-0,0001807

Tabla 28 Valores de convergencia

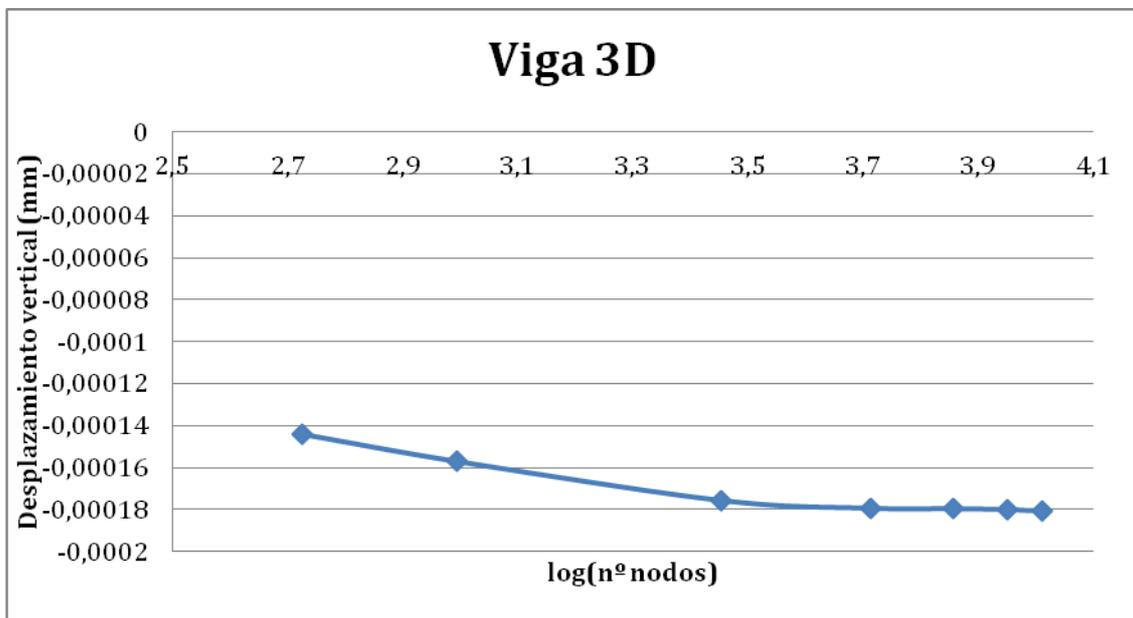


Fig 126 Convergencia de resultados

Solución obtenida

A continuación se muestra los resultados obtenidos mediante la malla de 10284 nodos. De igual manera que hacíamos en el caso anterior, se muestran los resultados de la fibra inferior de la primera capa del centro de la viga.

Desp_x	Desp_y	Sigma_x	Tau_xz
--------	--------	---------	--------

(m)	(m)	(N/m ²)	(N/m ²)
0	-0.0001807	0.2789	-0.0010096

Tabla 29 Solución obtenida con la malla de 10284 nodos

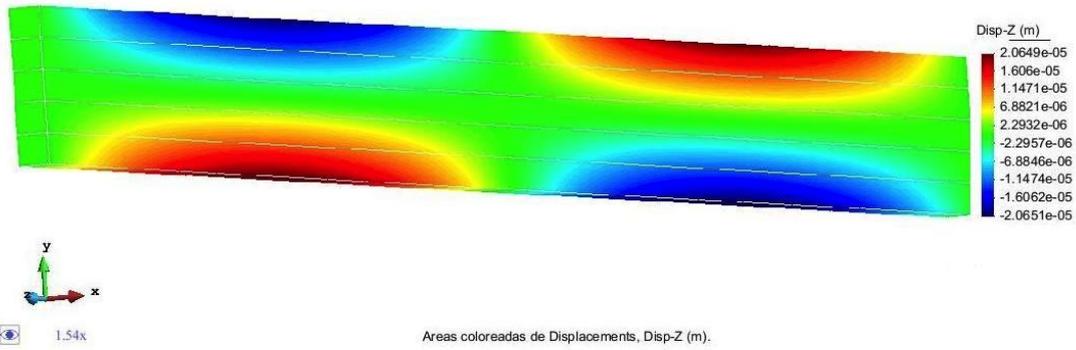


Fig. 127 Desplazamientos horizontales (mm)

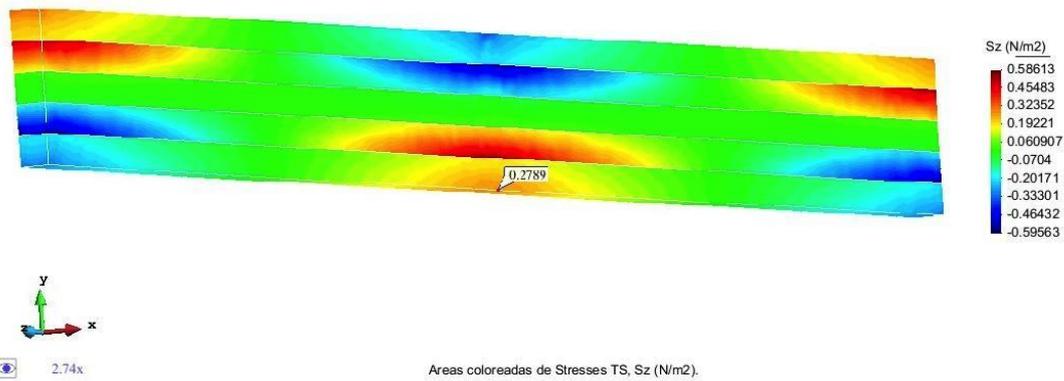


Fig. 128 Tensión normal (N/mm²)

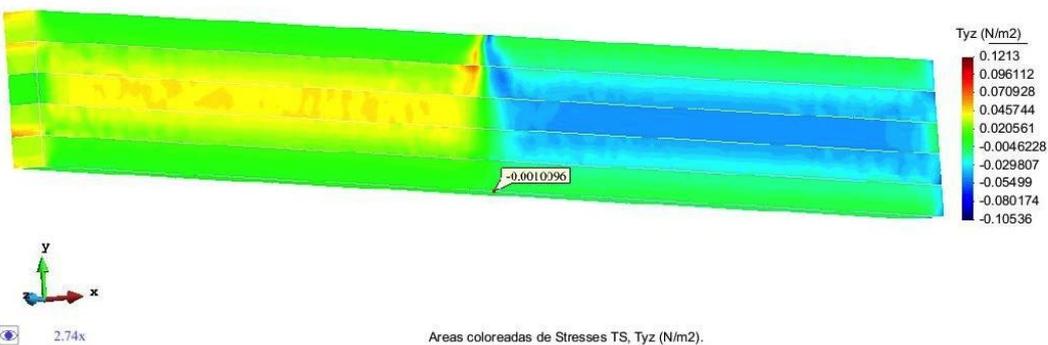


Fig. 129 Tensión tangencial xz (N/mm²)

A continuación se comparan los resultados obtenidos de la misma viga calculada mediante teoría de sólido 3D y mediante teoría de viga laminada compuesta de Timoshenko. Para poder confirmar que la teoría de Timoshenko es válida en el ejercicio propuesto, se ha calculado la diferencia en los valores obtenidos tomando como referencia la solución en tres dimensiones.

	Desp_x (mm)	Desp_y (mm)	Sigma_x (N/mm ²)	Tau_xz (N/mm ²)
Sólido 3D	0	-0.0001807	0.2789	0
TCB	0	-0.0001811	0.30246	0
Diferencia (%)	0	0.22	8.45	0

Tabla 30 Comparación de resultados

Tiempo invertido en el cálculo

A continuación se presenta las gráficas donde se distingue el tiempo necesario para calcular el problema en cuestión. Al igual que pasaba en los ejemplos anteriores, el programa de MAT-fem es más eficiente computacionalmente ya que presenta los mismos resultados en un tiempo mucho menor.

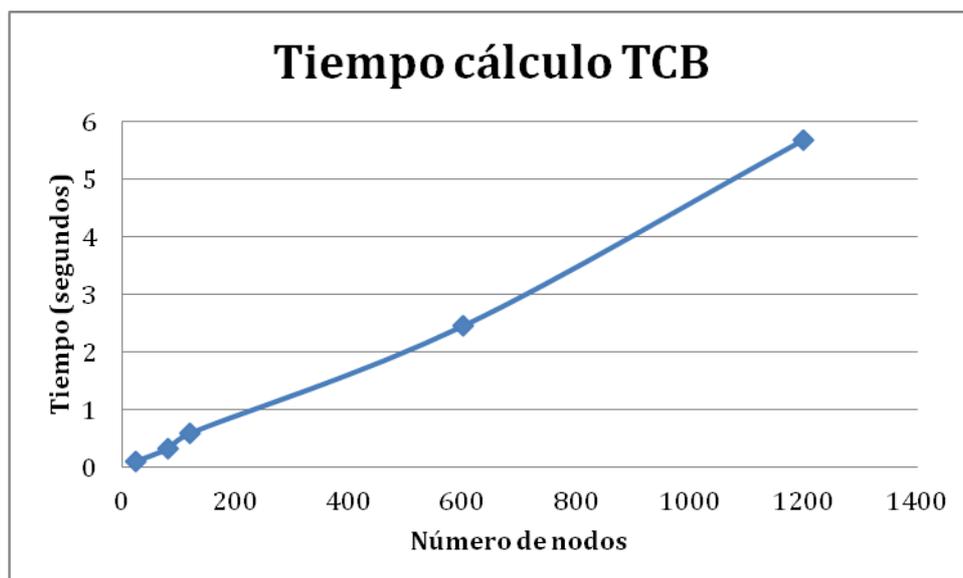


Fig. 130 Gráfica tiempo invertido en el cálculo de TCB

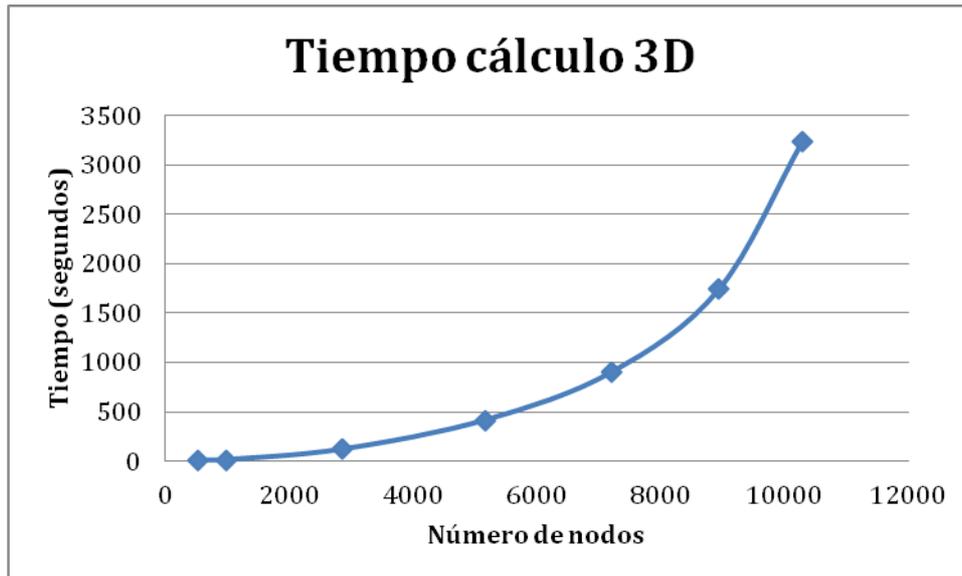


Fig. 131 Gráfica tiempo invertido en el cálculo de viga en 3D

II. VIGA BIEMPOTRADA CON CARGA REPARTIDA

El siguiente ejercicio trata de estudiar el cálculo de una viga biempotrada de longitud L con carga repartida P , de canto h y ancho b . Este ejercicio aparece en el libro *"Analysis with the Finite Element Method. Linear Statics"* de Eugenio Oñate, con lo que el objetivo de este caso es obtener los resultados según la teoría de Timoshenko con el programa de MAT-fem y poder comparar los resultados.

- ✓ $L = 100\text{mm}$; $h=5\text{mm}$; $b=1\text{mm}$
- ✓ $P = 1\text{KN/m}$
- ✓ Características de los materiales compuestos:

CAPA	YOUNG(N/mm ²)	POISSON	DENSIDAD (N/m ³)	ESPEJOR (mm)	ANCHO (mm)
1	5.3·10 ⁵	0.25	0	0.5	1
2	2.19·10 ⁵	0.25	0	0.6	1
3	0.82·10 ⁵	0.25	0	0.5	1
4	0.0073·10 ⁵	0.259	0	0.4	1
5	5.3·10 ⁵	0.25	0	0.7	1
6	0.0073·10 ⁵	0.259	0	0.1	1
7	7.30·10 ⁵	0.25	0	0.4	1
8	0.82·10 ⁵	0.25	0	0.5	1
9	2.19·10 ⁵	0.25	0	0.3	1
10	7.30·10 ⁵	0.25	0	1	1

Tabla 31 Características y dimensiones de la sección de la viga

Resultados obtenidos

La viga se divide en elementos unidimensionales de dos nodos cada uno, sin embargo la solución se muestra en 2D mediante elementos cuadriláteros de cuatro nodos cada uno.

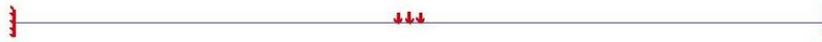


Fig. 132 Preproceso

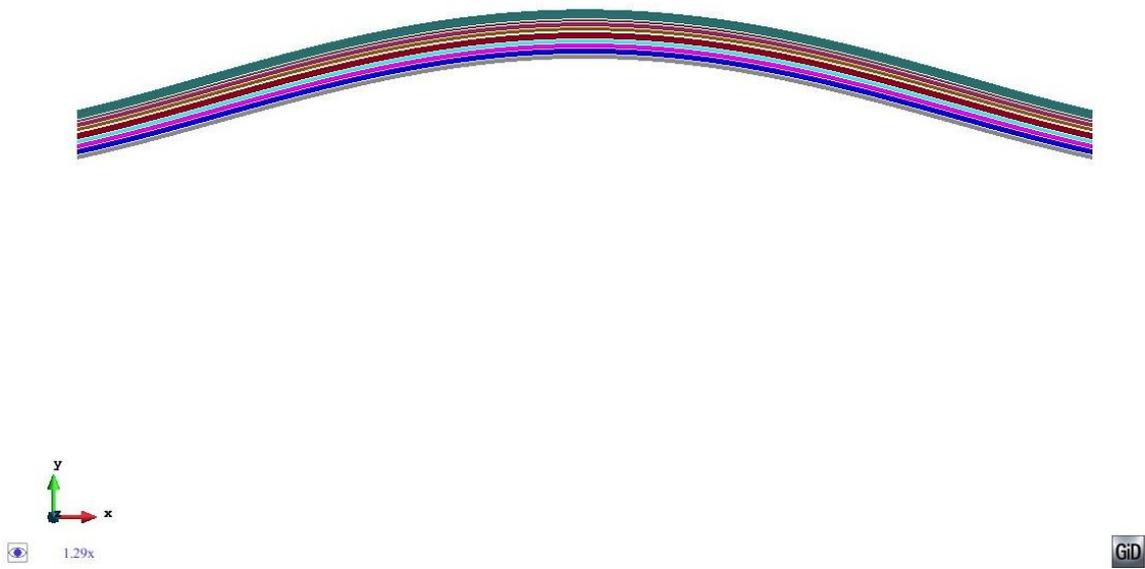


Fig. 133 Deformada en el postproceso

A continuación se muestran las gráficas de resultados que se pueden comparar con los resultados obtenidos en el libro. Los resultados son los obtenidos a partir de una malla de 300 nodos.

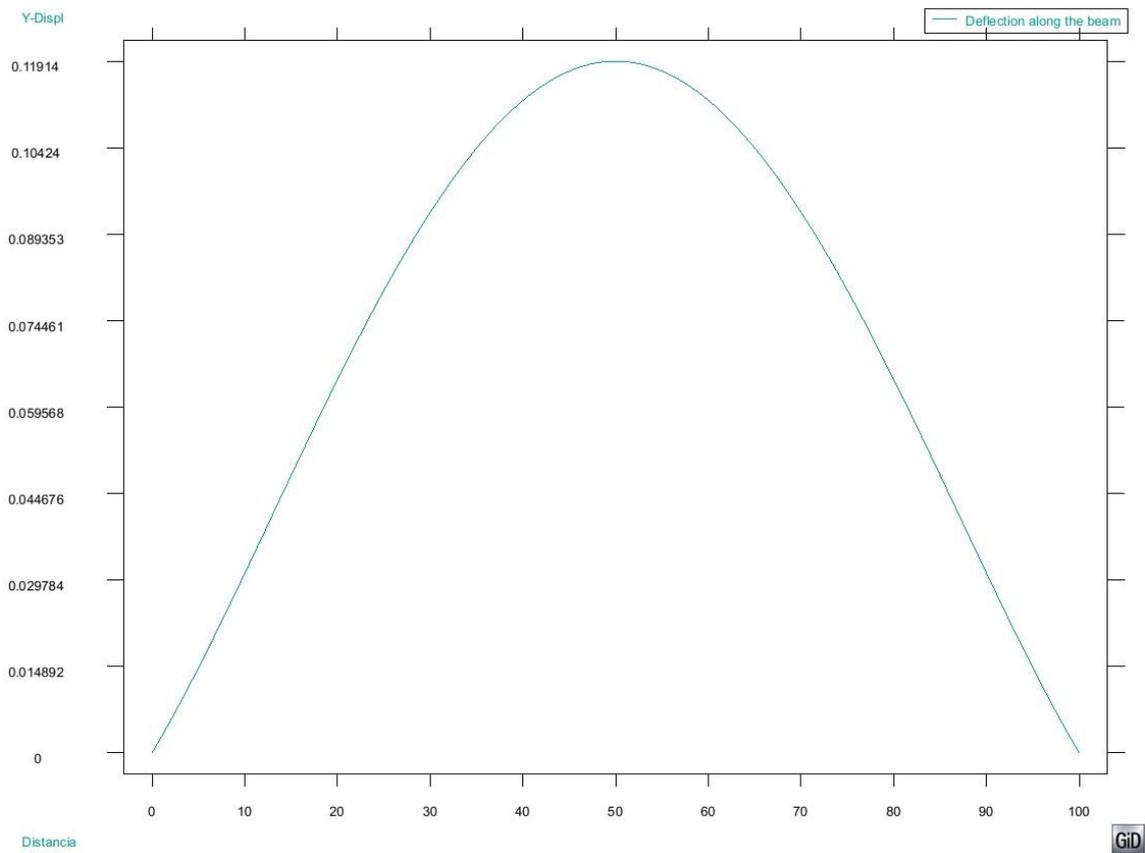


Fig. 134 Desplazamiento vertical a lo largo del eje de la viga (mm)

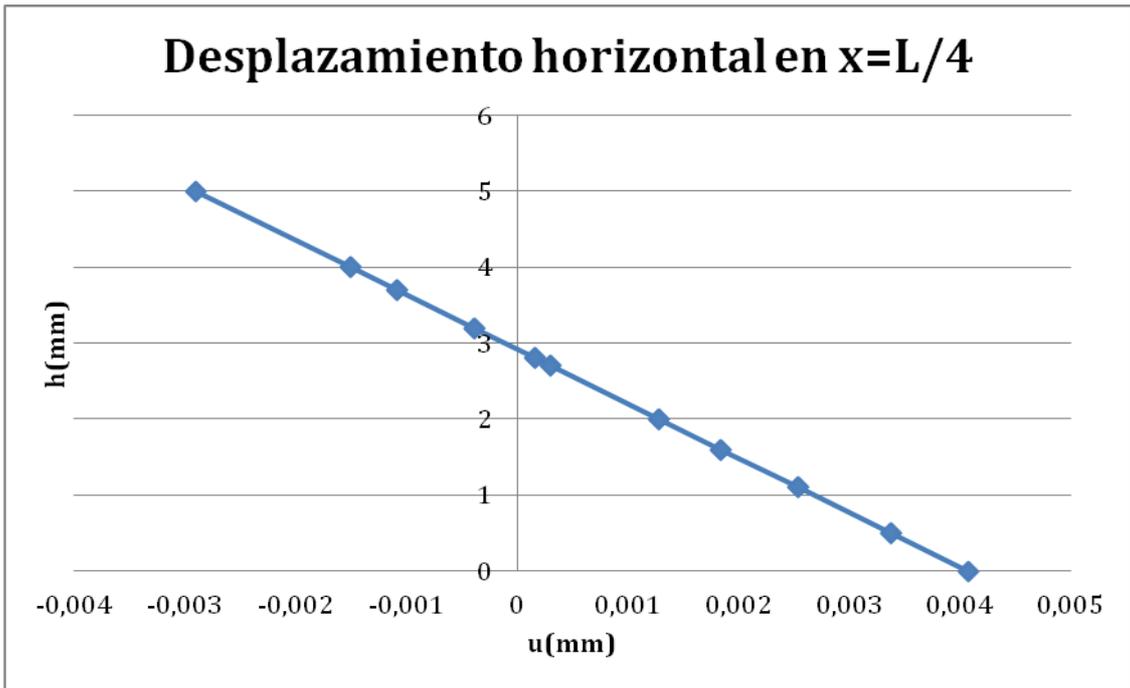


Fig. 135 Desplazamiento horizontal en la sección $x=L/4$

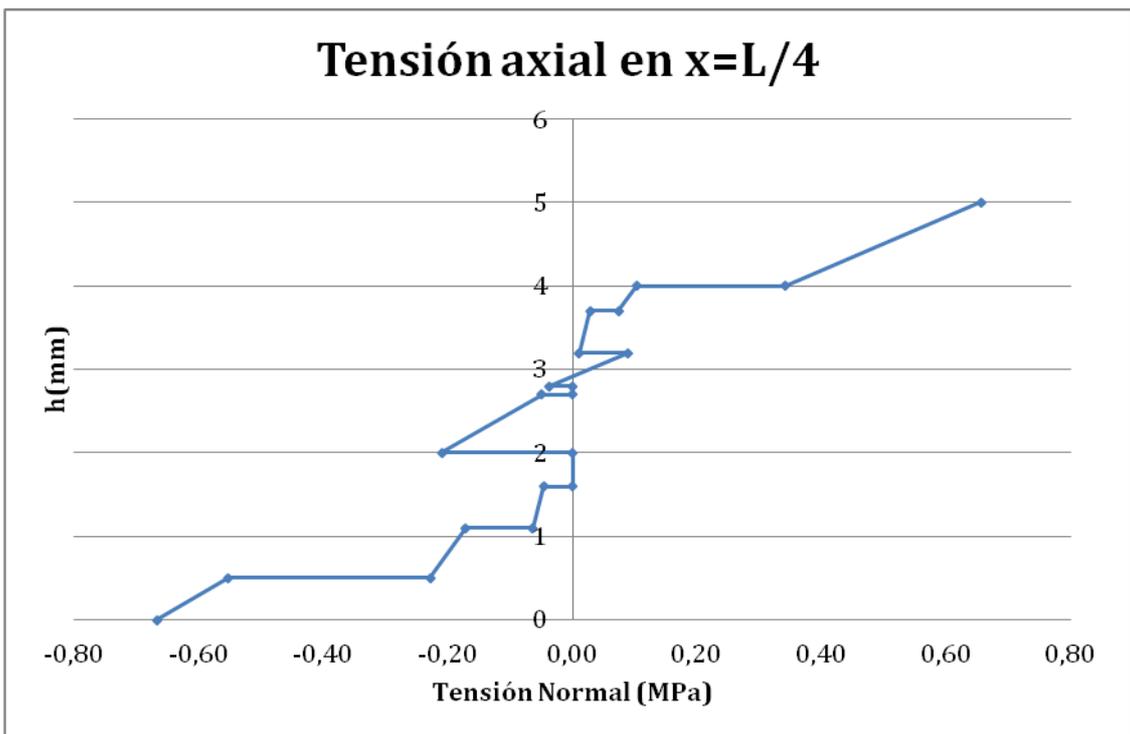


Fig. 136 Tensión normal en la sección $x=L/4$

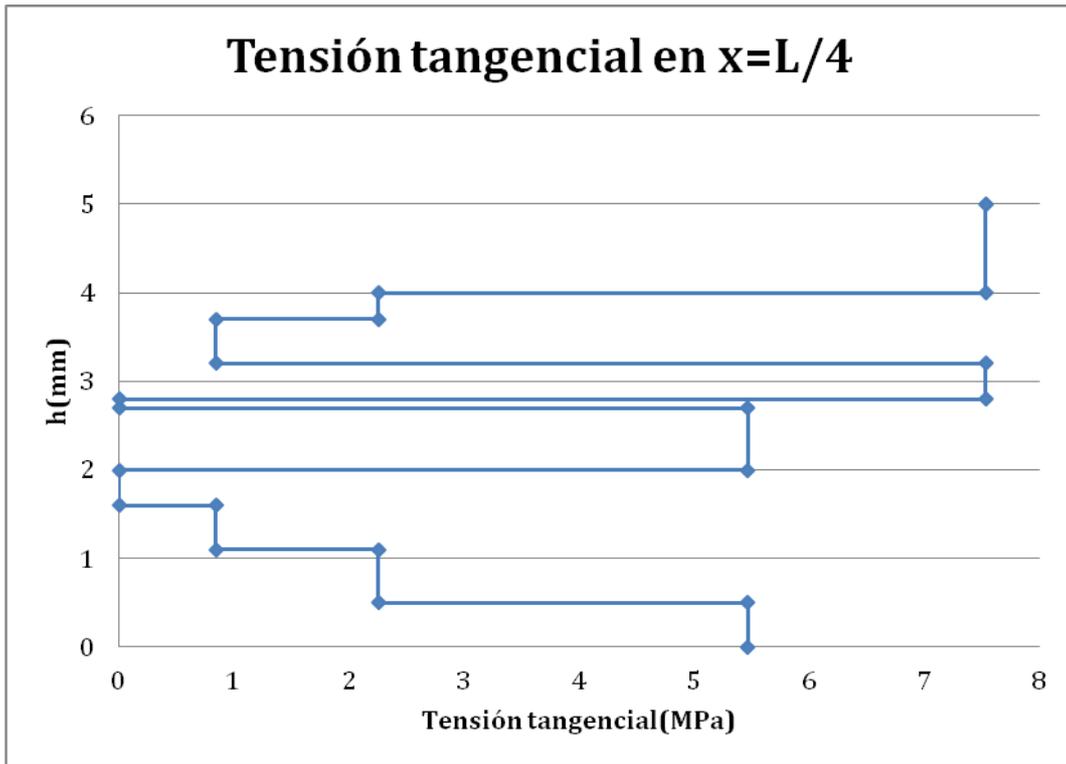


Fig. 137 Tensión tangencial en la sección x=L/4

La siguiente tabla muestra los resultados en la fibra inferior de la primera capa, en el centro de la viga, en una malla de 300 nodos:

Desp_x (mm)	Desp_y (mm)	Giro	Qy (N)	Mz (Nmm)	Sigma_x (N/mm ²)	Tau_xz (N/mm ²)
0	0.11913	0	0	-111.69	-128.25	0

Tabla 32 Valores obtenidos mediante la malla de 300 nodos

A continuación se muestran algunas imágenes del programa GiD donde se observa el comportamiento estructural en cuanto a desplazamientos y tensiones se refiere:

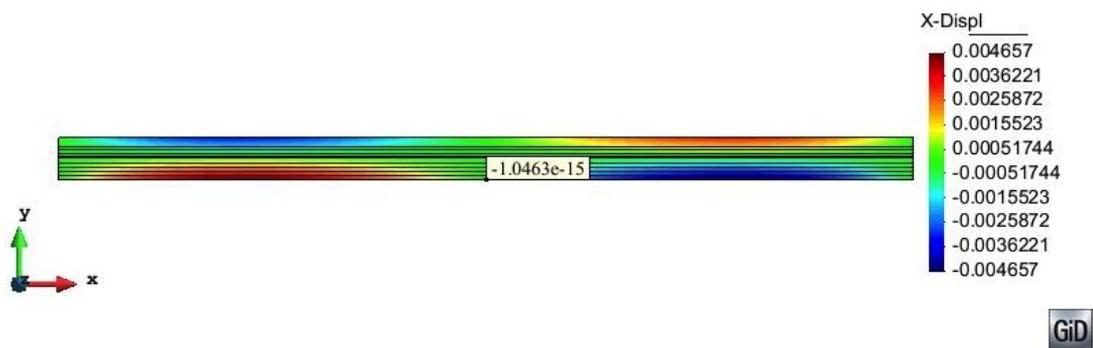


Fig. 138 Desplazamiento horizontal (mm)

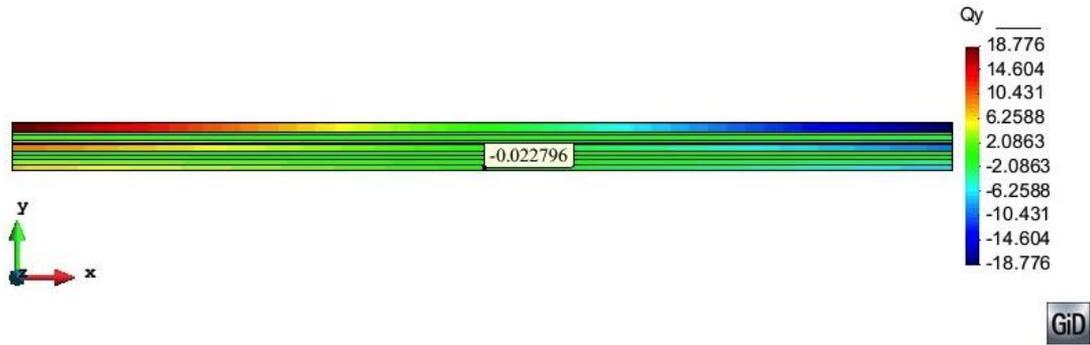


Fig. 139 Esfuerzo cortante (N)

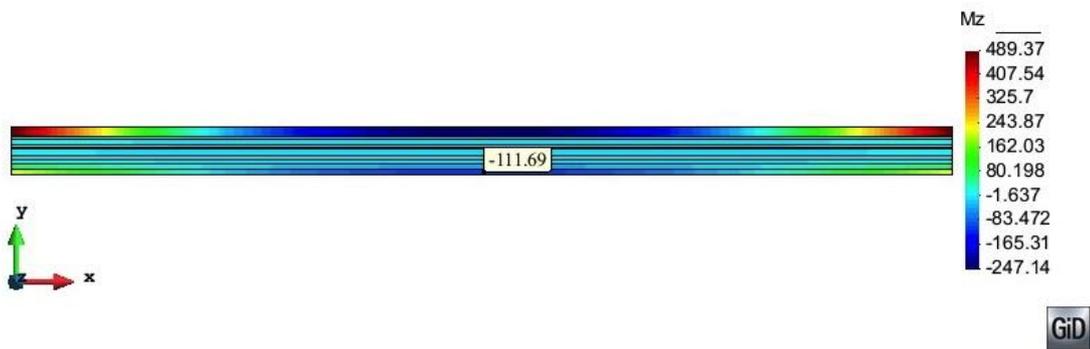


Fig. 140 Momento (Nmm)

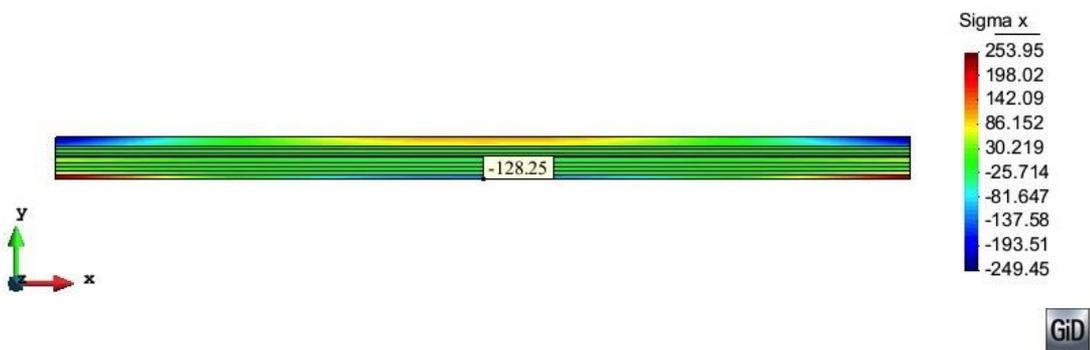


Fig. 141 Tensión normal (N/mm^2)

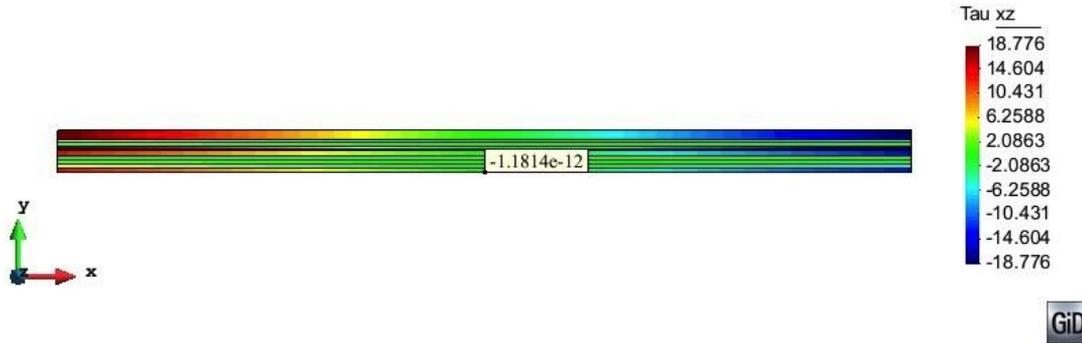


Fig. 142 Tensión tangencial xz (N/mm²)

Los resultados obtenidos coinciden en los que aparecen en el libro de Eugenio Oñate, sin embargo, no está de más calcular la viga mediante teoría tridimensional y cerciorar que los resultados obtenidos son correctos.

Solución tridimensional

A continuación se muestra la gráfica que demuestra que la solución converge correctamente.

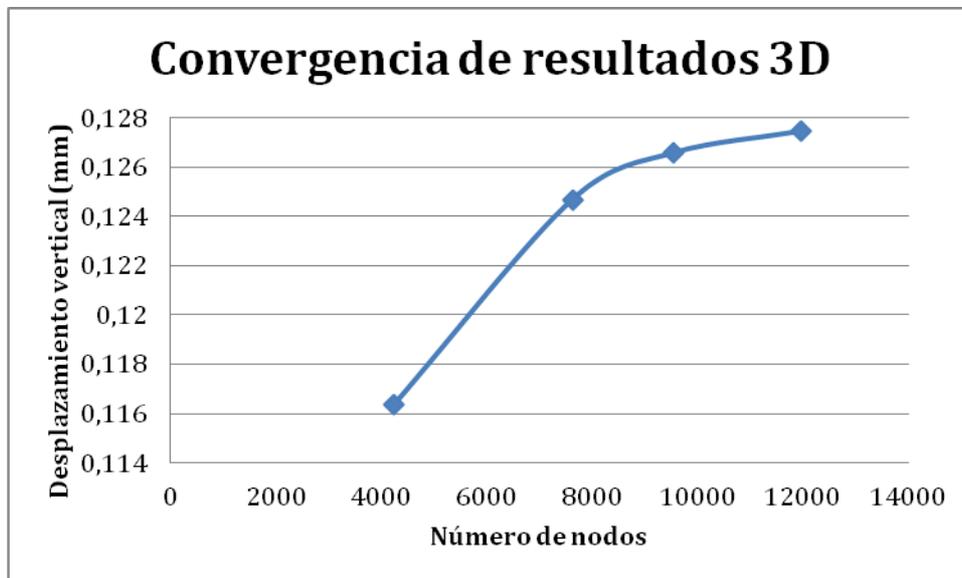


Fig. 143 Convergencia de resultados

En la siguiente tabla se muestran los resultados obtenidos mediante una malla de 11959 nodos. Los valores indicados en la tabla corresponden a los valores en el centro de la viga, en la primera capa y en la fibra inferior.

Desp_x (m)	Desp_y (m)	Sigma_x (N/m ²)	Tau_xz (N/m ²)
0	0.12744	-114.47	0.0266

Tabla 33 Resultados obtenidos mediante la malla de 11959 nodos

En lo que sigue, se muestran las imágenes que describen el comportamiento estructural de la viga en cuestión:

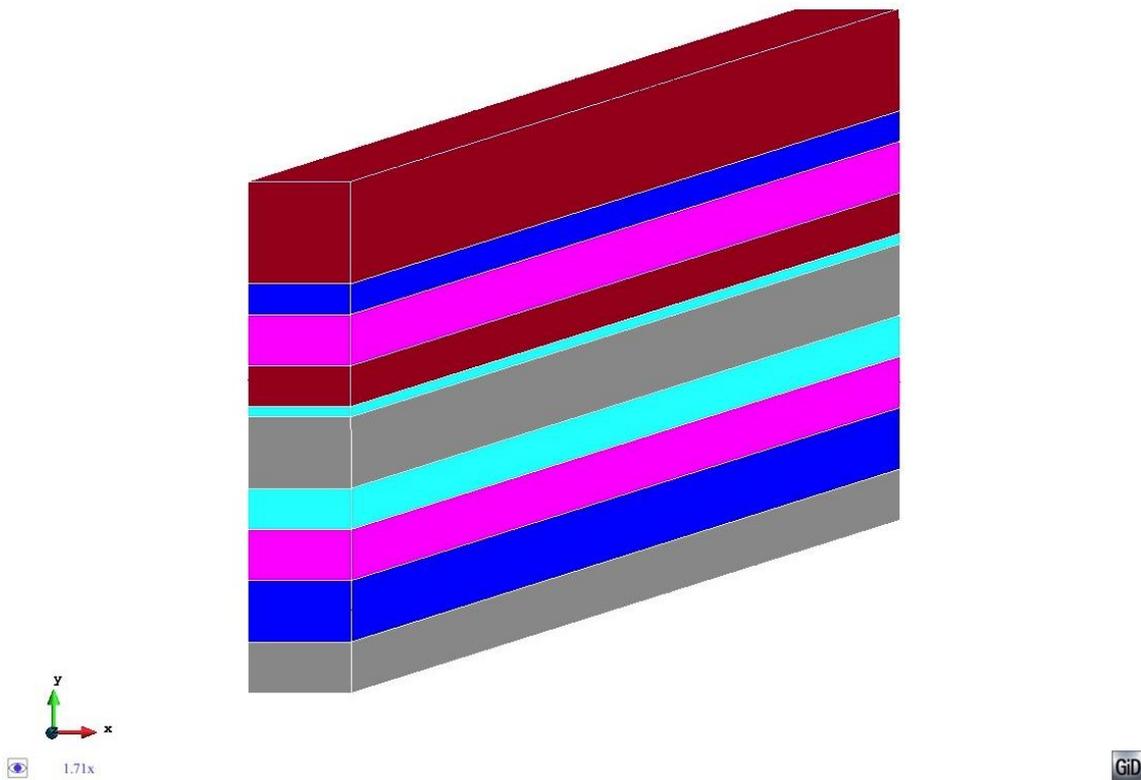
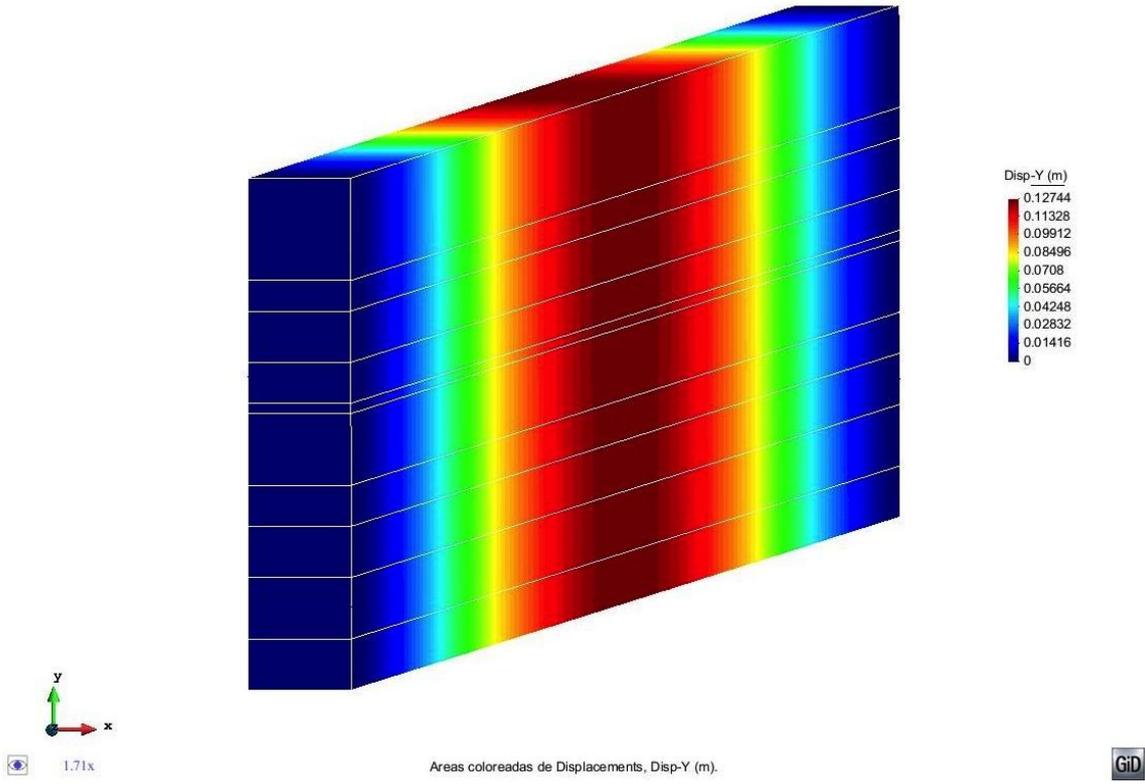
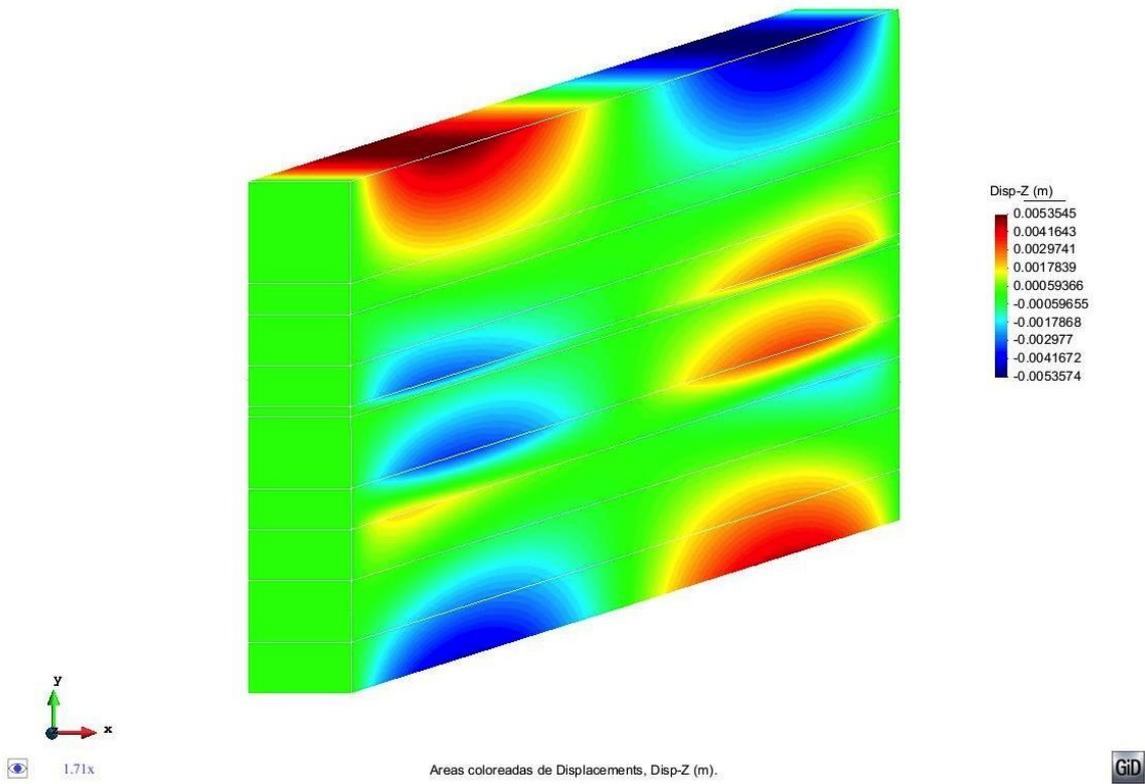


Fig. 144 Viga con las 10 capas bien diferenciadas



Areas coloreadas de Displacements, Disp-Y (m).

Fig. 145 Desplazamiento vertical (mm)



Areas coloreadas de Displacements, Disp-Z (m).

Fig. 146 Desplazamiento horizontal (mm)

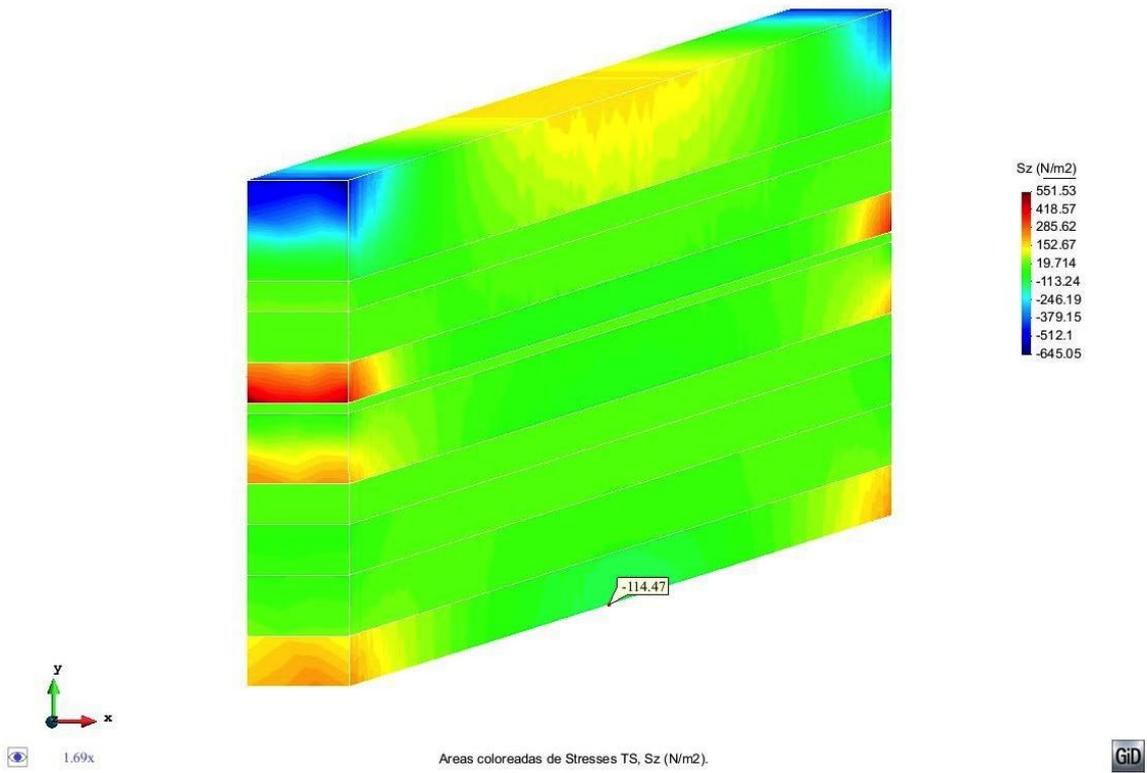


Fig. 147 Tensión normal (MPa)

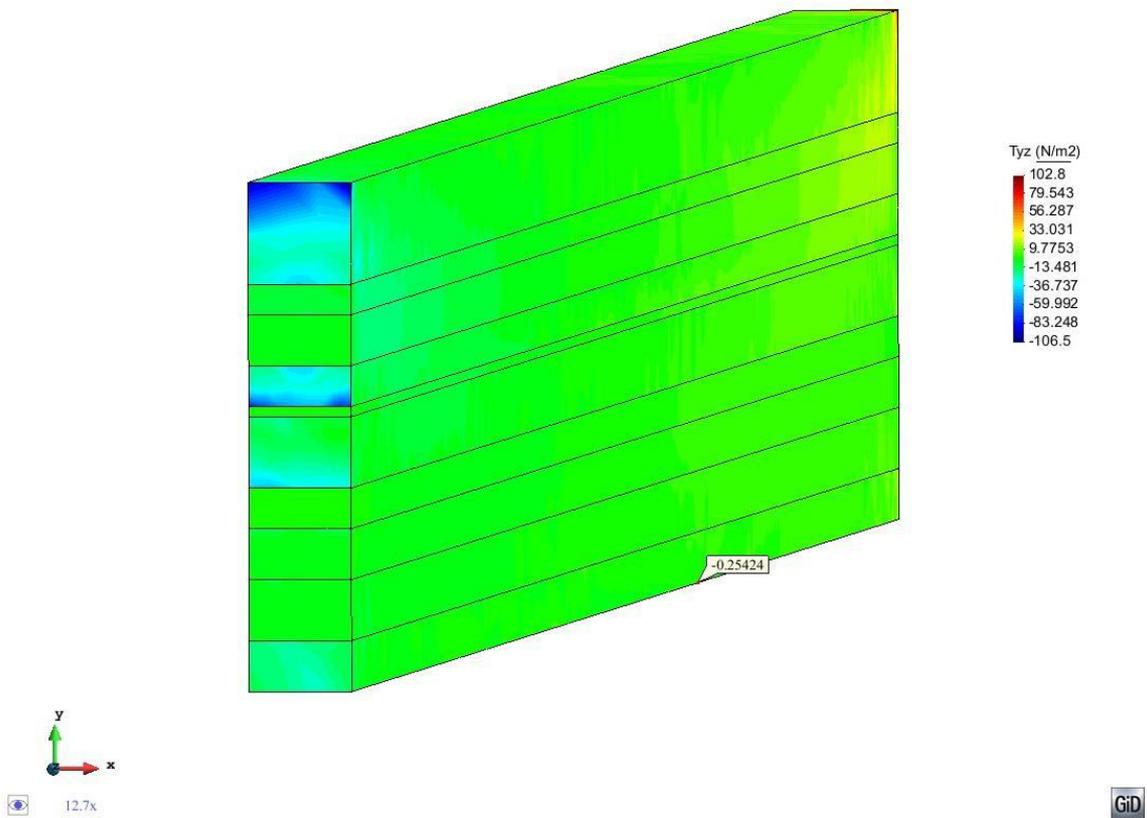


Fig. 148 Tensión tangencial (MPa)

Se observa como el comportamiento de la viga calculada en forma de sólido 3D es el mismo que el que obtuvimos mediante el método de Timoshenko para estructuras compuestas. Además, se observa en la siguiente tabla como los valores obtenidos son muy parecidos.

	Desp_x (mm)	Desp_y (mm)	Sigma_x (N/mm ²)	Tau_xz (N/mm ²)
Sólido 3D	0	0.12744	-114.47	-0.25424
TCB	0	0.11913	-128.25	0.04
Diferencia (%)	0	6.52	12.04	84

Tabla 34 Comparación de resultados

En el caso de las tensiones tangenciales, los valores obtenidos están lo suficiente cerca del cero como para suponer que la tensión en ese punto se anula. Si comparamos los valores con las órdenes de magnitud que aparecen en las tensiones se observa claramente que no es una aproximación descabellada.

Se puede concluir, por tanto, que el programa en MAT-fem representa fielmente el comportamiento estructural de la viga.

ANEJO 3: RUTINAS DE PROGRAMACIÓN

En el presente anejo se adjunta las rutinas de programación tanto en MATLAB como en GiD necesarias para calcular vigas compuestas según la teoría de Timoshenko.

I. PROGRAMACIÓN EN MATLAB*Composite Beam Timoshenko Reduced.m*

```

%% 2 Nodes Composite Beam using Timoshenko Theory
% Reduce the influence of the transverse shear stiffness (thick and
slender
% beams)
% Clear memory and variables.
clear

% The variables are read as a MAT-fem subroutine
% layers      =   Number of layers
% young       =   Young Modulus
%             =   matrix size: layers x 1
% poiss       =   Poission Ratio
%             =   matrix size: layers x 1
% denss      =   Material density
%             =   matrix size: layers x 1
% thickness   =   thickness of every layer
%             =   matrix size: layers x 1
% width      =   width of every layer
%             =   matrix size: layers x 1
% coordinates = [x] matrix size: nnode x ndime (1)
% elements    = [ inode, jnode ] element conectivity matrix
%             = matrix size: nelem x nnode; nnode = 2
% fixnodes   = [ node number, dof, fixed value] matrix with
%             = Dirichlet restrictions, were dof=1 for
horizontal
%             = displacement, dof=2 for vertical displacement
%             = and dof=3 for vertical derivative.
% pointload  = [ node number, dof, load value] matrix with
%             = nodal loads, were dof=1 for horizontal load,
%             = dof=2 for vertical load and dof=3 for moment.
% uniload    = [ uniform load ] sparse matrix size: nelem x 2

file_name = input('Enter the file name :','s');

tic;                % Start clock
ttim = 0;          % Initialize time counter
eval (file_name);  % Read input file

% Finds basics dimensions
npsod  = size(coordinates,1);      % Number of nodes
nelem  = size(elements,1);        % Number of elements
nnode  = size(elements,2);        % Number of nodes por element

```

```

dofpn = 3; % Number of DOF per node
dofpe = nnode*dofpn; % Number of DOF per element
nndof = npnod*dofpn; % Number of total DOF

ttim = timing('Time needed to read the input file',ttim);
%Reporting time

% Dimension the global matrices.

StifMat = sparse ( nndof , nndof ); % Create the global stiffness
matrix
force = sparse ( nndof , 1 ); % Create the global force
vector
reaction = sparse ( nndof , 1 ); % Create the global reaction
vector
u = sparse (nndof, 1); % Nodal variables

% Material properties (Constant over the domain)

shear = young./(2*(1+poiss));
area = thickness'*width ;

% Create coordinate z vector of points between each layer

coord_z = [zeros(1);thickness] ;
for j=2:layers+1
    coord_z(j,1) = coord_z(j,1)+coord_z(j-1,1); % Coordinate z
respect the % the bottom of
end % section

for i=1:layers
    zg_layers(i)=(coord_z(i)+coord_z(i+1))*0.5; % Center of
gravity of % each layer
end

for i=1:layers
    area_layers(i)=thickness(i)*width(i); % Area of each
layer
end

coord_zg = (area_layers*zg_layers')/area; % Center of gravity
of the % section

coord_ZG = coord_z-(coord_zg*ones(layers+1,1)); % Coordinate z
respect the % center of gravity
of the % section

% Components of the generalized constitutive matrix

% Axial stiffness
D_mata = 0;
for i=1:layers
    D_mata = D_mata + width(i) * thickness (i) * young(i);
end

```

```

% Bending stiffness
D_matb = 0 ;
for i=1:layers
    D_matb = D_matb + (1/3)*((coord_ZG(i+1))^3-
(coord_ZG(i))^3)*width(i)*young(i);
end

% Coupling axial-bending stiffness
D_matab = 0;
for i = 1:layers
    D_matab = D_matab+(-
1/2)*(coord_ZG(i+1)+coord_ZG(i))*width(i)*thickness(i)*young(i);
end

% Shear stiffness
D_mats=0;
for i=1:layers
    D_mats = D_mats +
thickness(i)*width(i)*(young(i)/(2*(1+poiss(i))));
end

% Compute the shear correction parameter for bending around the y
axis
% (kz)

S1 = 0;
S3 = 0;
for i = 1 : layers
    if i > 1
        S1 = S1 + young(i-1)* (coord_ZG(i)^2 - coord_ZG(i-1)^2);
    end
    S2 = 1/20*young(i)^2*(coord_ZG(i+1)^5-coord_ZG(i)^5)+...
        1/6*(S1-young(i)*coord_ZG(i)^2) * young(i) *
(coord_ZG(i+1)^3-coord_ZG(i)^3)+...
        1/4*(S1-young(i)*coord_ZG(i)^2)^2 * (coord_ZG(i+1)-
coord_ZG(i));
    S3 = S3 + width(i)/shear(i)*S2 ;
end
kz = D_matb^2/D_mats/S3;

% New shear stiffness

D_mats = kz*D_mats;

% Vertical distance between the global axis x and the neutral
axis.

d = - D_matab / D_mata ;

% Create coordinate z vector of points between each layer respect
% neutral axis

coord_zp = coord_ZG-(d*ones(layers+1,1));

% New coupling axial-bending stiffness
D_matab=0;
for i = 1:layers
    D_matab =D_matab + (-
1/2)*(coord_zp(i+1)+coord_zp(i))*width(i)*thickness(i)*young(i);
end

```

```

end

ttim = timing('Time needed to set initial values',ttim); %Reporting
time

% Element cycle

for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode);

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem. X
coordinate

    len = coor_x(2) - coor_x(1); % x_j > x_i

    const = D_matb/len;

    K_bend = [ 0 , 0 , 0 , 0 ;
              0 , 1 , 0 , -1 ;
              0 , 0 , 0 , 0 ;
              0 , -1 , 0 , 1 ];

    K_bend = K_bend * const;

    const = D_mats/len;

    K_shear = [ 1 , len/2 , -1 , len/2 ;
               len/2 , len^2/4 , -len/2 , len^2/4 ;
               -1 , -len/2 , 1 , -len/2 ;
               len/2 , len^2/4 , -len/2 , len^2/4 ];

    K_shear = K_shear * const;

    K_flex = K_bend + K_shear;

    const = D_mata/len;

    K_axial = [ 1 , -1 ;
               -1 , 1 ];

    K_axial = K_axial * const ;

    K_elem = [K_axial , zeros(2,4);
              zeros(4,2) , K_flex ];

    K_elem(:, [2 3]) = K_elem(:, [3 2]);
    K_elem(:, [3 4]) = K_elem(:, [4 3]);

    K_elem([2,3],:)=K_elem([3 2],:);
    K_elem([3,4],:)=K_elem([4 3],:);

    pp = thickness.*width.*denss;

    fa = uniload(ielem,1) * len/2 ;
    fz = (-sum(pp) + uniload(ielem,2)) * len/2;

```

```

ElemFor = [ fa, fz , 0, fa, fz, 0]; %elemental force vector.

% Finds the equation number list for the i-th element
for i=1:nnode
    ii = (i-1)*dofpn;
    for j =1:dofpn
        eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build the equation
number list
    end
end

% Assemble the force vector and the stiffness matrix
for i = 1 : dofpe
    ipos = eqnum(i);
    force (ipos) = force(ipos) + ElemFor(i);
    for j = 1 : dofpe
        jpos = eqnum(j);
        StifMat (ipos,jpos) = StifMat (ipos,jpos) + K_elem(i,j);
    end
end

end % End element cycle

ttim = timing('Time to assemble the global system',ttim);
%Reporting time

% Add point loads conditions to the force vector

for i = 1 : size(pointload,1)
    ieqn = (pointload(i,1)-1)*dofpn+pointload(i,2); % Finds eq.
number
    force(ieqn) = force(ieqn) + pointload(i,3); % add the
force
end

ttim = timing('Time for apply side and point load',ttim);
%Reporting time

% Applies the Dirichlet conditions and adjust the right hand side.

for i = 1 : size(fixnodes,1)
    ieqn = (fixnodes(i,1)-1)*dofpn+fixnodes(i,2) ; % Finds eq.
number
    u (ieqn) = fixnodes(i,3); % and store the solution
in u
    fix(i) = ieqn ; % and mark the eq as a fix
value
end

force = force - StifMat * u ; % adjust the rhs with the
known values

% Compute the solution by solving StifMat * u = force for the
% remaining unknown values of u.

FreeNodes = setdiff ( 1:nndof, fix ); % Finds the free node list
and

```

```

u(FreeNodes) = StifMat(FreeNodes,FreeNodes) \ force(FreeNodes);

ttim = timing('Time to solve the stiffness matrix',ttim);
%Reporting time

% Compute the layer's results

[Dsp_lyr Str_lyr ResStr_lyr] =
Composite_Layers_Beam_Timoshenko(layers,young, shear,poiss,kz,width,t
hickness,coord_ZG,coord_zp,u);

ttim = timing('Time to solve the layer results',ttim);
%Reporting time

% Graphic representation.

ToGiD_Composite_Beam_Timoshenko(file_name,layers,coord_ZG,u,Dsp_lyr,
Str_lyr,ResStr_lyr);

ttim = timing('Time used to write the solution',ttim);
%Reporting time
itim = toc; %Close
last tic
fprintf(1,'\n Total running time %12.6f \n',ttim); %Reporting
final time

```

Composite Layer Beam Timoshenko.m

```

function [Dsp_lyr Str_lyr ResStr_lyr] =
Composite_Layers_Beam_Timoshenko(layers,young, shear,poiss,kz,width,t
hickness,coord_ZG,coord_zp,u)

%Evaluates the layer's values in order to see the results in a 2D
graphic.
%Evaluates the displacements (Dsp_lyr) at the nodes and at the
points between layers.
%Evaluates the stresses (Str_lyr), for each layer, at the gauss
point and smooth the values
%to the nodes
%Evaluates the resultant stresses (ResStr_lyr) on Gauss points for
each layer.

global coordinates;
global elements;

npnod = size(coordinates,1); % Number of nodes
nelem = size(elements,1); % Number of elements
nnode = size(elements,2); % Number of nodes por element
dofpn = 3; % Number of DOF per node
dofpe = nnode*dofpn; % Number of DOF per element

```

```

nndof = npnod*dofpn; % Number of total DO
eqnum = zeros(dofpe); % Equation number list
Dsp_lyr = zeros(npnod,3, layers+1); % Create array for
displacements
Str_lyr = zeros(npnod,3,2*layers); % Create array for stresses
ResStr_lyr = zeros(nelem,3, layers); % Create array for resultant
stresses

%% Layer Displacements
% Finds the equation number list for the i-th node

dof_1=zeros(npnod,1); %dof_1 belong to the axial displacement
dof_2=zeros(npnod,1); %dof_2 belong to the vertical displacement
dof_3=zeros(npnod,1); %dof_3 belong to the rotation

for node = 1:npnod
    dof_1(node)=(node-1)*dofpn + 1;
    dof_2(node)=(node-1)*dofpn + 2;
    dof_3(node)=(node-1)*dofpn + 3;
end

% Displacement field

for lyr = 1:layers+1
    Dsp_lyr(:,1,lyr) = u(dof_1) - coord_zp(lyr)*u(dof_3);
    Dsp_lyr(:,2,lyr) = u(dof_2);
    Dsp_lyr(:,3,lyr) = u(dof_3);
end

%% Layer Stresses and Resultant Stresses

Da = zeros(2*layers,1);
Db = zeros(2*layers,1);
Ds = zeros(2*layers,1);

for i=1:layers
    Da(2*(i-1)+1) = young(i);
    Da(2*(i-1)+2) = young(i);
    Db(2*(i-1)+1) = -young(i)*coord_zp(i);
    Db(2*(i-1)+2) = -young(i)*coord_zp(i+1);
    Ds(2*(i-1)+1) = kz*shear(i);
    Ds(2*(i-1)+2) = kz*shear(i);
end

Da_lyr = zeros(layers,1);
Db_lyr = zeros(layers,1);
Dab_lyr = zeros(layers,1);
Ds_lyr = zeros(layers,1);

for i=1:layers
    Da_lyr(i) = width(i) * thickness (i) * young(i);
    Db_lyr(i) = (1/3)*((coord_ZG(i+1))^3-
(coord_ZG(i))^3)*width(i)*young(i);
    Ds_lyr(i)= thickness(i)*width(i)*(young(i)/(2*(1+poiss(i))));
end

Ds_lyr=kz*Ds_lyr;

```

```

% Element cycle.
for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode); %vector nodos del
elemento

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem. X
coordinate

    % Finds the equation number list for the i-th element

    for i=1:nnode
        ii = (i-1)*dofpn;
        for j =1:dofpn
            eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build the equation
number list
        end
    end

    % Recover the nodal displacements for the i-th element

    u_elem(1:dofpe)=u(eqnum(1:dofpe));

    len = coor_x(2) - coor_x(1); % x_j > x_i

    % One gauss point for stress evaluation

    gaus0 = 0.0; % One Gauss point for stresses evaluation

    bmat_a=[ -1/len, 0, 0, 1/len, 0, 0 ];
    bmat_b=[0, 0, -1/len, 0, 0, 1/len];
    bmat_s=[0, -1/len, -1/2, 0, 1/len, -1/2];

    Str1_g0 = (Da*bmat_a + Db*bmat_b)*transpose(u_elem);
    Str2_g0 = (Ds*bmat_s)*transpose(u_elem);

    Str_lyr(lnods(1),1,:) = Str_lyr(lnods(1),1,:)+permute(Str1_g0,[2
3 1]);
    Str_lyr(lnods(2),1,:) = Str_lyr(lnods(2),1,:)+permute(Str1_g0,[2
3 1]);
    Str_lyr(lnods(1),2,:) = Str_lyr(lnods(1),2,:)+permute(Str2_g0,[2
3 1]);
    Str_lyr(lnods(2),2,:) = Str_lyr(lnods(2),2,:)+permute(Str2_g0,[2
3 1]);
    Str_lyr(lnods(1),3,:) = Str_lyr(lnods(1),3,:)+1;
    Str_lyr(lnods(2),3,:) = Str_lyr(lnods(2),3,:)+1;

    ResStr1_g0 = (Da_lyr*bmat_a) *transpose(u_elem);
    ResStr2_g0 = (Db_lyr*bmat_b) *transpose(u_elem);
    ResStr3_g0 = (Ds_lyr*bmat_s) *transpose(u_elem);

```

```

ResStr_lyr(ielem,1,:)=permute(ResStr1_g0,[2 3 1]);
ResStr_lyr(ielem,2,:)=permute(ResStr2_g0,[2 3 1]);
ResStr_lyr(ielem,3,:)=permute(ResStr3_g0,[2 3 1]);

end % End element cycle

for i = 1 : npnod
    Str_lyr(i,1,:) = Str_lyr(i,1,:)./Str_lyr(i,3,:);
    Str_lyr(i,2,:) = Str_lyr(i,2,:)./Str_lyr(i,3,:);
end

```

ToGiD Composite Beam Timoshenko.m

```

function
ToGiD_Composite_Beam_Timoshenko(file_name, layers, coord_ZG, u, Dsp_lyr,
Str_lyr, ResStr_lyr)
%
% Parameters:
%
% Input, file_name : GiD File name
%         u         : Nodal variables
%         reaction  : Nodal reactions
%         Str       : Gauss Point Stresses
%
% Output, none
%
global coordinates;
global elements;

nelem = size(elements,1);           % Number of elements
npnod = size(coordinates,1);       % Number of nodes

qeletyp = 'Quadrilateral';

msh_file = strcat(file_name, '.flavia.msh');
res_file = strcat(file_name, '.flavia.res');

%% Mesh with quadrilateral elements (2D beam)
qnelem = nelem*layers;             % Number of quadrilateral elements
qnode = 4;                          % Number of nodes per
quadrilateral elements.
qnpnod = npnod*layers*2;           % Number of nodes

% Coordinates matrix of quadrilateral elements (2D)
qcoordinates = zeros(qnpnod,2);
for i=1:layers
    for j=1:npnod

```

```

        qcoordinates((2*i-2)*npnod+j,:)=[coordinates(j)
coord_ZG(i)];
        qcoordinates((2*i-1)*npnod+j,:)=[coordinates(j)
coord_ZG(i+1)];
    end
end

% Elements matrix of quadrilateral elements (2D)

qelements=zeros(qnelem,qnode+1);
for i = 1:layers
    for j = 1 : nelem
        qelements((i-1)*nelem+j,:) = [((2*i-2)*npnod+
elements(j,:))...
((2*i-1)*npnod + elements(j,end:-1:1)) i];
    end
end

% Conectivity matrix by layer (2D)

Nod_lyr = zeros(npnod,layers,2);
Elem_lyr = zeros(nelem,layers);
for i = 1 : layers
    for j = 1 : npnod
        Nod_lyr(j,i,1) = (2*i-2)*npnod + j;
        Nod_lyr(j,i,2) = (2*i-1)*npnod + j;
    end
    for n = 1 : nelem
        Elem_lyr(n,i) = (i-1)*nelem + n;
    end
end

%%Generates the mesh file

fid = fopen(msh_file,'w');
fprintf(fid,'### \n');
fprintf(fid,'# MAT_FEM Compiste Laminated TBT \n');
fprintf(fid,'# \n');

% Generates the mesh in 2D
fprintf(fid,'Group "Thickness_Analysis_2D" \n');
fprintf(fid,'MESH "2D_beam" dimension %3.0f Elemtyp %s Nnode
%2.0f \n \n',2,qeletyp,qnode);
fprintf(fid,'coordinates \n');
for i = 1 : qnpnod
    fprintf(fid,'%6.0f %12.5d %12.5d
\n',i,qcoordinates(i,1),qcoordinates(i,2));
end
fprintf(fid,'end coordinates \n \n');
fprintf(fid,'elements \n');
for i = 1 : qnelem
    fprintf(fid,'%6.0f %6.0f %6.0f %6.0f %6.0f %3.0f
\n',i,qelements(i,:));
end
fprintf(fid,'end elements \n \n');
fprintf(fid,'end group \n \n');

status = fclose(fid);

```

```

%% Generates the result file

fid = fopen(res_file,'w');
fprintf(fid,'Gid Post Results File 1.0 \n');
fprintf(fid,'### \n');
fprintf(fid,'# MAT_FEM Composite Laminated TBT \n');
fprintf(fid,'# \n');
fprintf(fid,'GaussPoints "GP_quad" Elemtyp %s "2D_beam"
\n',qeletyp);
fprintf(fid,'Number of Gauss Points: 1 \n');
fprintf(fid,'Natural Coordinates: Internal \n');
fprintf(fid,'end gausspionts \n');
fprintf(fid,'# \n');

%% Thickness results along each layer

fprintf(fid,'OnGroup "Thickness_Analysis_2D" \n');
fprintf(fid,'Result "Displacements" "Load Analysis" 1 Vector
OnNodes \n');
fprintf(fid,'ComponentNames "X-Displ", "Y-Displ", "Z-Displ" \n');
fprintf(fid,'Values \n');
for i=1:layers
    for j = 1 : npnod
        fprintf(fid,'%6.0i %13.5d %13.5d 0.0
\n',Nod_lyr(j,i,1),full(Dsp_lyr(j,1,i)),full(Dsp_lyr(j,2,i)));
        fprintf(fid,'%6.0i %13.5d %13.5d 0.0
\n',Nod_lyr(j,i,2),full(Dsp_lyr(j,1,i+1)),full(Dsp_lyr(j,2,i+1)));
    end
end
fprintf(fid,'End Values \n');
fprintf(fid,'# \n');

fprintf(fid,'Result "Rotation" "Load Analysis" 1 Vector OnNodes
\n');
fprintf(fid,'ComponentNames "X-Rot", "Y-Rot", "Z-Rot" \n');
fprintf(fid,'Values \n');
for i=1:layers
    for j = 1 : npnod
        fprintf(fid,'%6.0i 0.0 0.0 %13.5d
\n',Nod_lyr(j,i,1),full(Dsp_lyr(j,3,i)));
        fprintf(fid,'%6.0i 0.0 0.0 %13.5d
\n',Nod_lyr(j,i,2),full(Dsp_lyr(j,3,i+1)));
    end
end
fprintf(fid,'End Values \n');
fprintf(fid,'# \n');

fprintf(fid,'Result "Stresses" "Load Analysis" 1 Vector OnNodes
\n');
fprintf(fid,'ComponentNames "Sigma_x", "Tau_xz", "zero" \n');
fprintf(fid,'Values \n');
for i=1:layers
    for j = 1 : npnod
        fprintf(fid,'%6.0i %12.5d %12.5d 0.0
\n',Nod_lyr(j,i,1),Str_lyr(j,1,(i-1)*2+1),Str_lyr(j,2,(i-1)*2+1));
        fprintf(fid,'%6.0i %12.5d %12.5d 0.0
\n',Nod_lyr(j,i,2),Str_lyr(j,1,(i-1)*2+2),Str_lyr(j,2,(i-1)*2+2));
    end
end

```

```

end
fprintf(fid, 'End Values \n');
fprintf(fid, '# \n');

fprintf(fid, 'Result "Resultant_Stresses" "Load Analysis" 1 Vector
OnGaussPoints "GP_quad" \n');
fprintf(fid, 'ComponentNames "Nx", "Qy", "Mz" \n');
fprintf(fid, 'Values \n');
for i=1:layers
    for j = 1 : nelem
        fprintf(fid, '%6.0i %12.5d %12.5d %12.5d
\n', Elem_lyr(j,i), ResStr_lyr(j,1,i), ResStr_lyr(j,3,i), ResStr_lyr(j,2
,i));
    end
end
fprintf(fid, 'End Values \n');
fprintf(fid, 'End ongroup \n');
fprintf(fid, '# \n');

status = fclose(fid);

```

timing.m

```

function t = timing (text,time)

%% timing Evaluates and acumulate the used time.
%
% Parameters:
%
% Input, text : String to show.
%           time : Accumulate time at t-1
%
% Output, t    : Accumulate time at t

itim = toc; % Close previous tic
fprintf(1,[text, ' %12.6f \n'],itim); % Output time and text
t = itim + time; % Accumulate time in t

tic % Open an new tic

```

II. PROGRAMACIÓN EN GID

MAT-fem Composite Beams.cnd

```

BOOK: Displacement_Constraints
NUMBER: 1 CONDITION: Point_Constraints
CONDTYPE: over points
CONDMESHTYPE: over nodes
QUESTION:Horizontal_Constraint:#CB#(1,0)
VALUE:1
DEPENDENCIES: (0,SET,Displacement-X,0.0m)(#DEFAULT#,RESTORE,Displacement-X,0.0m)
QUESTION:Displacement-X#UNITS#
VALUE:0.000m
QUESTION:Vertical_Constraint:#CB#(1,0)
VALUE: 1
DEPENDENCIES: (0,SET,Displacement-Y,0.0m)(#DEFAULT#,RESTORE,Displacement-Y,0.0m)
QUESTION:Displacement-Y#UNITS#
VALUE:0.000m
QUESTION:Rotation_Constraint:#CB#(1,0)
VALUE: 1
DEPENDENCIES: (0,SET,Displacement-R,0.0rad)(#DEFAULT#,RESTORE,Displacement-
R,0.0rad)
QUESTION:Displacement-R#UNITS#
VALUE:0.000rad
END CONDITION
BOOK: Point_Loads
NUMBER: 2 CONDITION: Point_Load
CONDTYPE: over points
CONDMESHTYPE: over nodes
QUESTION: Horizontal-Force:#UNITS#
VALUE: 0.0N
QUESTION:Vertical-Force:#UNITS#
VALUE: 0.0N
QUESTION: Moment:#UNITS#
VALUE: 0.0N*m
END CONDITION

```

```

BOOK: Uniform_Loads
NUMBER: 3 CONDITION: Uniform_Load
CONDTYPE: over lines
CONDMESHTYPE: over elems
QUESTION:Horizontal-Load:#UNITS#
VALUE: 0.0N/m
QUESTION:Vertical-Load:#UNITS#
VALUE: 0.0N/m
END CONDITION

```

MAT-fem Composite Beams.prb

```
PROBLEM DATA
TITLE:General_Data
QUESTION:Problem_Title
VALUE:Untitled
HELP: The title of the problem
QUESTION:Consider_Self_weight#CB#(Yes,No)
VALUE:No
END PROBLEM DATA
```

MAT-fem Composite Beams.mat

```
NUMBER: 1 MATERIAL: Composite_Section
QUESTION: LAYER
VALUE: 3
QUESTION: YOUNG(LAYER,___YOUNG__N/m^2___)
VALUE: #N# 6 1 2.1e11 2 3.0e10 3 3.0e10
QUESTION: POISSON(LAYER,___POISSON___)
VALUE: #N# 6 1 0.2 2 0.2 3 0.2
QUESTION: SELF-WEIGHT(LAYER,___SELF-WEIGHT__N/m^3___)
VALUE: #N# 6 1 78000 2 25000 3 78000
QUESTION: THICKNESS(LAYER,___THICKNESS__m___)
VALUE: #N# 6 1 2 2 16 3 2
QUESTION: WIDTH(LAYER,___WIDTH__m___)
VALUE: #N# 6 1 10 2 10 3 10
END MATERIAL
```

MAT-fem Composite Beams.uni

```
BEGIN TABLE
LOAD : N {reference}, 1e-1 kgf, 1e-4 ton, 1e-3 kN, 1e-6 MN, 0.2248090247 lbf
MOMENT : N*m {reference}, 1e-1 kgf*m, 1e-4 ton*m, 1e-3 kN*m, 1e-6 MN*m,
8.850749004 lbf*in
UNIFORM_LOAD : N/m {reference}, 1e-1 kgf/m, 1e-4 ton/m, 1e-3 kN/m, 1e-6 MN/m,
0.571015e-2 lbf/in
AREA : m^2 {reference}, 1e+4 cm^2, 1e+6 mm^2, 1550 in^2
INERTIA : m^4 {reference}, 1e+8 cm^4, 1e+12 mm^4, 2402509.610 in^4
SELF_WEIGHT : N/m^3 {reference}, 1e-1 kgf/m^3, 1e-4 ton/m^3, 1e-3 kN/m^3, 1e-6
MN/m^3, 0.3683959876e-5 lbf/in^3
END
```

```

BEGIN SYSTEM(INTERNATIONAL)
LENGTH : m
ANGLE : rad
UNIFORM_LOAD : N/m
PRESSURE : N/m^2
LOAD : N
MOMENT : N*m
SELF_WEIGHT : N/m^3
AREA : m^2
INERTIA : m^4
END

USER DEFINED: DISABLED

```

MAT-fem Composite Beams.sim

```

cond Point_Constraints
4
global
cond(int,1) && cond(int,3) && cond(int,5)
1
0
0
pc_emp.geo
global
cond(int,1) && cond(int,3)
1
0
0
ep_apoiol.geo
global
cond(int,1) || cond(int,3)
cond(int,3)
cond(int,1)*(-1)
0
ep_apoio-rodillo.geo
global
cond(int,5)
1
0
0
pc_sim.geo
cond Point_Load
4
global
cond(real,1) && cond(real,2) && cond(real,3)
cond(real,2)

```

```
cond(real,1)*(-1)
0
pl_fl_m.geo
global
cond(real,1) && cond(real,3) || cond(real,2) && cond(real,3)
cond(real,2)
cond(real,1)*(-1)
0
pl_fl_m.geo
global
1
cond(real,1)
cond(real,2)
0
ep_fletxa.geo
global
cond(real,3)
cond(real,3)
0
0
pl_mom.geo
cond Uniform_Load
1
global
cond(real,1) && cond(real,2) || cond(real,1) || cond(real,2)
cond(real,2)
cond(real,1)*(-1)
0
ep_flechas.geo
```

MAT-fem Composite Beams.bas

```
*loop elems
*if(elemsmat==0)
*messagebox - An element without any material was found, some possibilities are that
you forget to assign a material to the problem or there is a repited entity over another -
Process finished with error
*endif
*end
*loop nodes
*if(NodesCoord(3)==0)
*else
*messagebox - This is a beams problem so z coordinate must be =0 for all nodes -
Process finished with error
*endif
*if(NodesCoord(2)==0)
```

```

*else
*messagebox - This is a beams problem so y coordinate must be =0 for all nodes -
Process finished with error
*endif
*end
*set var MATERIAL=0
*loop materials
*set var MATERIAL=Operation(MATERIAL(int)+1)
*end
*if(MATERIAL>1)
*messagebox - This program only allows one material
*endif
%=====
==
% MAT-fem_Beams 1.0 - MAT-fem is a learning tool for understanding
%           the Finite Element Method with MATLAB and GiD
%=====
==
% PROBLEM TITLE = *GenData(1)
%
% Material Properties
%
*loop materials
*format " layers = %10.2e; "
*MatProp(1);
*set var L=MatProp(1,int)
*set var N=MatProp(2,int)
young = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(2,*operation(i+1));
*end for
*format " %17.9e "
*MatProp(2,*operation(N))];
poiss = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(3,*operation(i+1));
*end for
*format " %17.9e "
*MatProp(3,*operation(N))];
*if(strcmp(GenData(2),"No")==0)
*format "%1i"
denss = zeros(*operation(L),1) ;
*else
denss = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(4,*operation(i+1));
*end for
*format " %17.9e "
*MatProp(4,*operation(N))];

```

```

*endif
thickness = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(5,*operation(i+1));
*end for
*format " %17.9e "
*MatProp(5,*operation(N))];
width = [
*for(i=1;i<=N-2;i=i+2)
*format " %17.9e "
*MatProp(6,*operation(i+1));
*end for
*format " %17.9e "
*MatProp(6,*operation(N))];
*end materials
%
% Coordinates
%
global coordinates
coordinates = [
*loop nodes
*format " %17.9e "
*if(NodesNum == npoin)
*NodesCoord(1) ] ;
*else
*NodesCoord(1) ;
*endif
*end nodes
%
% Elements
%
global elements
elements = [
*loop elems
*format " %6i %6i %6i %6i "
*if(nnode == 2)
*if(ElemsNum == nelem)
*ElemsConec(1) , *ElemsConec(2) ] ;
*else
*ElemsConec(1) , *ElemsConec(2) ;
*endif
*else
*if(ElemsNum == nelem)
*ElemsConec(1) , *ElemsConec(2) , *ElemsConec(3) ] ;
*else
*ElemsConec(1) , *ElemsConec(2) , *ElemsConec(3) ;
*endif
*endif
*end elems
%
% Fixed Nodes

```

```

%
fixnodes = [
*Set Cond Point_Constraints *nodes *or(1,int) *or(3,int) *or(5,int)
*Set var nod = 0
*loop nodes *OnlyInCond
*if(Cond(1,Int) == 1)
*Set var nod = nod + 1
*endif
*if(Cond(3,Int) == 1)
*Set var nod = nod + 1
*endif
*if(Cond(5,Int) == 1)
*Set var nod = nod + 1
*endif
*end nodes
*Set var nod2 = 1
*loop nodes *OnlyInCond
*if(Cond(1,Int) == 1)
*format " %6i %17.9e "
*if(nod==nod2)
*NodesNum() , 1 , *Cond(2) ] ;
*else
*NodesNum() , 1 , *Cond(2) ;
*endif
*Set var nod2 = nod2 + 1
*endif
*if(Cond(3,Int) == 1)
*format " %6i %17.9e "
*if(nod==nod2)
*NodesNum() , 2 , *Cond(4) ] ;
*else
*NodesNum() , 2 , *Cond(4) ;
*endif
*Set var nod2 = nod2 + 1
*endif
*if(Cond(5,Int) == 1)
*format " %6i %17.9e "
*if(nod==nod2)
*NodesNum() , 3 , *Cond(6) ] ;
*else
*NodesNum() , 3 , *Cond(6) ;
*endif
*Set var nod2 = nod2 + 1
*endif
*end nodes
%
% Point loads
%
*Set Cond Point_Load *nodes
*if((CondNumEntities(int)>0))
*Set var nod = 0
pointload = [

```

```
*loop nodes *OnlyInCond
*Set var nod = nod + 1
*format " %6i %17.9e "
*NodesNum , 1 , *cond(1,real) ;
*format " %6i %17.9e "
*NodesNum , 2 , *cond(2,real) ;
*format " %6i %17.9e "
*if(CondNumEntities(int)==nod)
*NodesNum , 3 , *cond(3,real) ] ;
*else
*NodesNum , 3 , *cond(3,real) ;
*endif
*end nodes
*else
pointload = [ ] ;
*endif
%
% Side loads
%
*Set Cond Uniform_Load *elems
uniload = sparse ( *nelem , 2);
*if(CondNumEntities(int)>0)
*loop elems *OnlyInCond *and(1,int)
*format " %6i %17.9e "
uniload (*ElemsNum,1) = *cond(1) ;
*end elems
*loop elems *OnlyInCond *and(2,int)
*format " %6i %17.9e "
uniload (*ElemsNum,2) = *cond(2) ;
*end elems
*endif
```