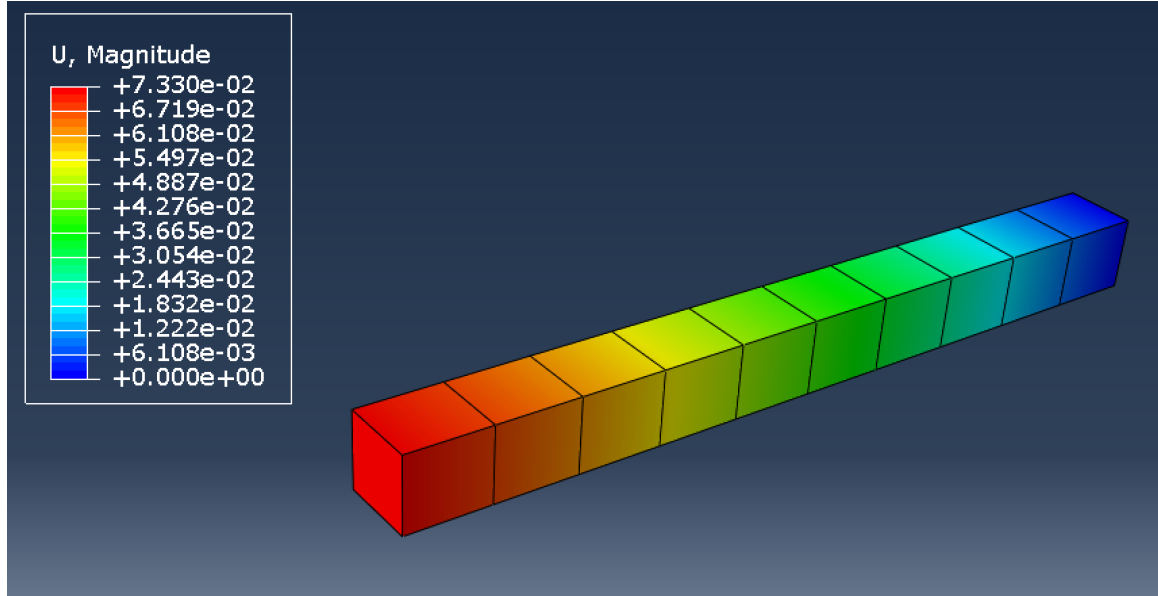


Práctica #2, Tema 1.

Modelo de elementos finitos para una barra elástica 1D



José M.^a Goicolea, Pedro Navas

GRUPO DE MECÁNICA COMPUTACIONAL, UPM

5 de febrero de 2020

Índice

1. Objetivos de la práctica	3
2. Base teórica y discretización con Elementos Finitos	3
2.1. Ecuaciones de gobierno	3
2.2. Condiciones de contorno y fuerzas volumétricas	3
2.3. Formulación fuerte y solución analítica	4
2.4. Formulación débil	5
2.5. Funciones de forma	6
2.6. Funciones de forma elementales	7
3. Simulación con MatLab / Octave	10
3.1. Código MatLab / Octave	11
4. Simulación con Abaqus	15
4.1. Módulo Part	15
4.2. Módulo Property	15
4.3. Módulo Assembly	18
4.4. Módulo Step	18

4.5. Módulo Load	19
4.6. Módulo Mesh	22
4.7. Módulo Job	24
4.8. Módulo Visualization	25
4.9. Comparativa con solución analítica en MatLab / Octave	28

1. Objetivos de la práctica

En esta práctica se ejemplifica la programación en lenguaje **Matlab** de un código de elementos finitos, capaz de resolver un problema elástico en una dimensión en el que se somete a cargas una barra estructural en 1D (este programa puede ser ejecutado también mediante el software libre **Octave**). Partimos de las siguientes premisas:

1. En las bases teóricas tenemos una serie de nociones en cuanto a la construcción de matrices de rigidez elementales, las cuales han de ser programadas por el alumnado.
2. A su vez, estas matrices elementales han de pasar a formar parte de un sistema global a través del ensamblado de la matriz de rigidez global, para la cual también se dispone de ciertas nociones teóricas.
3. Para la resolución del sistema de ecuaciones es necesario conocer las condiciones de contorno de carga y desplazamiento del problema. Se han de aplicar las mismas al sistema matricial que se ha desarrollado en lenguaje **Matlab**.

A continuación vamos a simular una barra similar a la resuelta en **Matlab** / **Octave** en 1D con la suite **Abaqus**. Esto nos ayudará a asimilar los conceptos básicos de dicha suite en 3D, tanto el preproceso, cálculo y postproceso, que nos ayudarán en las siguientes prácticas. De los resultados pretendemos la comparación con una solución analítica.

Los resultados obtenidos por los dos métodos numéricos explicados, **Matlab** / **Octave** y **Abaqus**, se pueden comparar con solución analítica que el propio alumno ha de obtener. Dicha comparación ha de ser llevada a cabo con las herramientas de dibujo de **Matlab** / **Octave**.

2. Base teórica y discretización con Elementos Finitos

2.1. Ecuaciones de gobierno

La ecuación que gobierna el fenómeno físico es la de una barra elástica cuyo desplazamiento se define por $u(x)$. Como se ha descrito, esta es una ecuación elíptica lineal en una dimensión, con la siguiente forma:

$$\frac{d(A\sigma)}{dx} + q(x) = \frac{d}{dx} \left(EA \frac{du}{dx} \right) + q(x) = 0 \quad (1)$$

donde E es el módulo de Young, un parámetro elástico del material que describe su rigidez. $q(x)$ representa las fuerzas distribuidas para este problema, por unidad de longitud x . La resolución numérica de este problema supone encontrar un $u(x)$, en el intervalo abierto $(0, L)$, tal que el sistema esté en equilibrio teniendo en cuenta unas condiciones de contorno definidas para el problema en cuestión.

2.2. Condiciones de contorno y fuerzas volumétricas

Las posibles condiciones de contorno que nos encontramos en el problema a resolver se resumen en la Fig. 1.

Sobre las condiciones de contorno, nos podemos encontrar de dos tipos:

- *Tipo Dirichlet o Esenciales*

Aquellas que se aplican en el campo en el que hemos definido la ecuación diferencial, en nuestro caso $u(x)$. En este caso son condiciones de contorno “en desplazamiento”.

$$u(0) = u_0$$

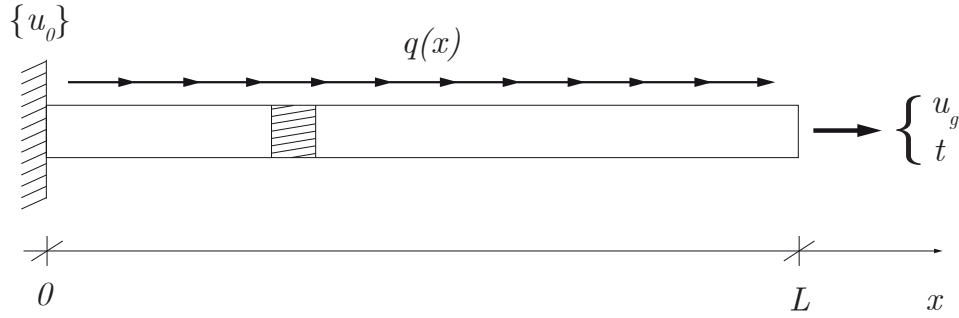


Figura 1: Definición del problema y sus condiciones de contorno.

$$u(L) = u_g$$

■ *Tipo Neumann o Naturales*

Aquellas que se aplican en la derivada espacial del campo en el que hemos definido la ecuación diferencial, que teniendo en cuenta la ecuación constitutiva, sería:

$$t = EA \left. \frac{du}{dx} \right|_{x=L}$$

Estas podrían considerarse como condiciones de contorno “en fuerzas”.

Por otro lado nos encontramos las *fuerzas volumétricas o distribuidas*, aquellas que se aplican a todo el dominio y que en la ecuación (1) están definidas por $q(x)$. Estas fuerzas pueden depender de la posición o bien ser constantes. Un ejemplo de este tipo de fuerzas en el problema mecánico son aquellas asociadas a la gravedad g , las cuales dependen de una sección $A(x)$ y una densidad que puede variar a lo largo del dominio, por lo que dicha fuerza también variaría con la posición:

$$q(x) = A(x)\rho(x)g$$

En el ejemplo a resolver en esta práctica, $q(x)$ será una función dada dependiente de la posición x , que representa la carga aplicada por unidad de longitud.

2.3. Formulación fuerte y solución analítica

La ecuación (1) se puede considerar como la formulación fuerte del problema, que se denomina así porque se obliga a que la ecuación en derivadas parciales se cumpla en cada punto del intervalo de interés, $(0, L)$.

Como vemos, dicha ecuación involucra una derivada segunda del campo $u(x)$, lo que implica que dicho campo ha de ser derivable dos veces con respecto a x .

A continuación se detalla la obtención de la solución analítica, para la cual supondremos que el área A y el módulo de Young E son constantes para toda la barra. Partimos de la integración de la formulación fuerte entre un punto $y \in (0, L)$ y L :

$$A \int_y^L \frac{d\sigma}{dx} dx = - \int_y^L q(x) dx \quad (2)$$

$$\sigma(L) - \sigma(y) = -\frac{1}{A} \int_y^L q(x) dx \quad (3)$$

Si empleamos la definición que nos aporta la ecuación constitutiva, $\sigma = E du/dx$:

$$E \frac{du(y)}{dy} = E \frac{du}{dx} \Big|_{x=L} + \frac{1}{A} \int_y^L q(x) dx \quad (4)$$

De nuevo, integramos entre 0 y un punto $z \in (0, L)$:

$$\int_0^z E \frac{du(y)}{dy} dy = \int_0^z \left(E \frac{du}{dx} \Big|_{x=L} + \frac{1}{A} \int_y^L q(x) dx \right) dy \quad (5)$$

Suponiendo conocida la derivada de u con respecto a x en el punto $x = L$, el resultado obtenido de la integral es:

$$Eu(z) - Eu(0) = E \frac{du}{dx} \Big|_{x=L} z + \frac{1}{A} \int_0^z \int_y^L q(x) dx dy \quad (6)$$

Por tanto, la solución de $u(z)$ resulta:

$$u(z) = u(0) + \frac{du}{dx} \Big|_{x=L} z + \frac{1}{EA} \int_0^z \int_y^L q(x) dx dy \quad (7)$$

La solución analítica de la ecuación (7) depende de dos constantes para las que son necesarias la aplicación de las dos condiciones de contorno que tenemos, $u(0)$ y $du/dx|_{x=L}$, condiciones de contorno esencial y natural respectivamente. Para el caso de no tener una condición de contorno natural en el extremo $x = L$ sino una condición en desplazamientos, el término $du/dx|_{x=L}$ se puede calcular imponiendo $z = L$, siendo que $u(L)$ es conocido.

Para el caso particular en que las condiciones de contorno sean: $u(0) = 0$ (extremo fijo) y $EAd u/dx|_L = P$ (carga P en $x = L$), y la fuerza distribuida sea una función lineal $q(x) = q_0 + rx$, la expresión de la solución analítica es:

$$\begin{aligned} u(x) &= \frac{1}{EA} \left[\left(P + q_0 L + r \frac{L^2}{2} \right) x - \left(q_0 \frac{x^2}{2} + r \frac{x^3}{6} \right) \right] \\ \sigma(x) &= E \frac{du}{dx} = \frac{1}{A} \left[P + q_0 (L - x) + \frac{r}{2} (L^2 - x^2) \right] \end{aligned} \quad (8)$$

Teniendo en cuenta las hipótesis realizadas, solo podremos obtener expresiones analíticas para valores sencillos de $E(x)$ y $q(x)$, abandonando la idea cuando estos valores se complican, por lo que hemos de buscar soluciones aproximadas. Una de las técnicas más empleadas es la del método de los elementos finitos, para el que necesitamos la forma débil del problema, que pasamos a describir en la siguiente sección.

2.4. Formulación débil

Esta formulación se consigue con una función de ponderación arbitraria $w(x)$, por la que se multiplica la ecuación de la formulación fuerte y se integra sobre todo el dominio:

$$\int_0^L w \frac{d}{dx} \left(EA \frac{du}{dx} \right) dx + \int_0^L w q dx = 0 \quad (9)$$

Integrando por partes el primer término:

$$\begin{aligned} \frac{d}{dx} \left[w \cdot EA \frac{du}{dx} \right] &= \frac{dw}{dx} \cdot EA \frac{du}{dx} + w \cdot \frac{d}{dx} \left(EA \frac{du}{dx} \right) \\ \left[w \cdot EA \frac{du}{dx} \right]_0^L - \int_0^L \frac{dw}{dx} \cdot EA \frac{du}{dx} dx &= \int_0^L w \cdot \frac{d}{dx} \left(EA \frac{du}{dx} \right) dx \end{aligned} \quad (10)$$

obtenemos el resultado siguiente sustituyendo en la ecuación (9):

$$\int_0^L \frac{dw}{dx} \cdot EA \frac{du}{dx} dx = \int_0^L wq dx + w(L)t_L - w(0)t_0 \quad (11)$$

donde t_L y t_0 son las condiciones de contorno naturales en los extremos, que físicamente corresponden a las fuerzas aplicadas en dichos extremos. Si en el extremo $x = 0$ se aplica una condición (esencial) de desplazamiento impuesto o fijo, en este caso a la función de ponderación se le exige también $w(0) = 0$, por lo que el último sumando de la ecuación (11) desaparece.

2.5. Funciones de forma

En el Método de los elementos finitos es necesario realizar la aproximación de la incógnita $u(x)$ por funciones de aproximación, también llamadas funciones de forma, a través de la expresión:

$$u(x) \approx u_h(x) = \sum_{B=1}^{N_{\text{nod}}} u_B N_B(x) \quad (12)$$

Se emplea un conjunto finito (N_{nod}) de funciones de interpolación $N_B(x)$. Esto conlleva un error que en principio disminuye cuanto mayor sea el número de nodos N_{nod} o el orden de las funciones de interpolación.

El Método de Galerkin, que es el método más empleado y, por tanto, el que vamos a utilizar, emplea para las funciones de ponderación $w(x)$ la misma interpolación que para la función incógnita $u(x)$:

$$w(x) \approx w_h(x) = \sum_{A=1}^{N_{\text{nod}}} w_A N_A(x) \quad (13)$$

La interpolación de las derivadas que aparecen en la forma débil será

$$\frac{du_h}{dx} = \sum_{B=1}^{N_{\text{nod}}} u_B \frac{dN_B}{dx}; \quad \frac{dw_h}{dx} = \sum_{A=1}^{N_{\text{nod}}} w_A \frac{dN_A}{dx} \quad (14)$$

Sustituyendo en las integrales de la forma débil (Eq. (11)), obtenemos:

$$\int_0^L \left(\sum_{A=1}^{N_{\text{nod}}} w_A \frac{dN_A}{dx} \right) EA \left(\sum_{B=1}^{N_{\text{nod}}} u_B \frac{dN_B}{dx} \right) dx = \sum_{A,B=1}^{N_{\text{nod}}} w_A \left[\int_0^L \frac{dN_A}{dx} EA \frac{dN_B}{dx} dx \right] u_B \quad (15)$$

donde

$$\int_0^L \frac{dN_A}{dx} EA \frac{dN_B}{dx} dx = K_{AB}$$

es decir, la denominada matriz de rigidez, $[\mathbf{K}]$. Análogamente, para las acciones aplicadas o fuentes se obtienen los términos del vector de fuerzas:

$$\int_0^L \left(\sum_{A=1}^{N_{\text{nod}}} w_A N_A \right) q dx = \sum_{A=1}^{N_{\text{nod}}} w_A \underbrace{\left[\int_0^L N_A q dx \right]}_{f_A^{\text{vol}}} \quad (16)$$

$$\left[w EA \frac{du}{dx} \right]_0^L = w(L)t_L - w(0)t_0 = \sum_{A=1}^{N_{\text{nod}}} w_A f_A^{\text{ext}} \quad (17)$$

De las ecuaciones (15), (16) y (17) resulta el siguiente sistema de ecuaciones algebraicas lineales:

$$\sum_{A,B=1}^{N_{\text{nod}}} w_A K_{AB} u_B = \sum_{A=1}^{N_{\text{nod}}} w_A (f_A^{\text{vol}} + f_A^{\text{ext}}) = \sum_{A=1}^{N_{\text{nod}}} w_A f_A \quad (18)$$

donde las fuerzas aplicadas se obtienen como suma de las volumétricas (interiores distribuidas) y las exteriores en el contorno

$$f_A = f_A^{\text{vol}} + f_A^{\text{ext}}$$

Y teniendo en cuenta que $w(x)$ son arbitrarias, w_A también lo serán, por lo que se obtiene la ecuación matricial:

$$\sum_{B=1}^{N_{\text{nod}}} K_{AB} u_B = f_A \quad \Leftrightarrow \quad [\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{f}\} \quad (19)$$

2.6. Funciones de forma elementales

En la práctica el cálculo de las integrales y la expresión de las funciones de interpolación se hacen elemento a elemento. Esto facilita sobremanera el cálculo que se hace con el mismo algoritmo en cada uno de los elementos, para luego ensamblar las matrices globales. Las funciones de interpolación tienen soporte compacto, es decir son cero fuera del subdominio Ω^e correspondiente al elemento (e) en cuestión. Se establece una numeración y coordenadas locales en cada elemento, que tienen su correspondencia con las globales.

En el caso a resolver 1D podemos suponer funciones de forma lineales de dos nodos como los de la figura 2.

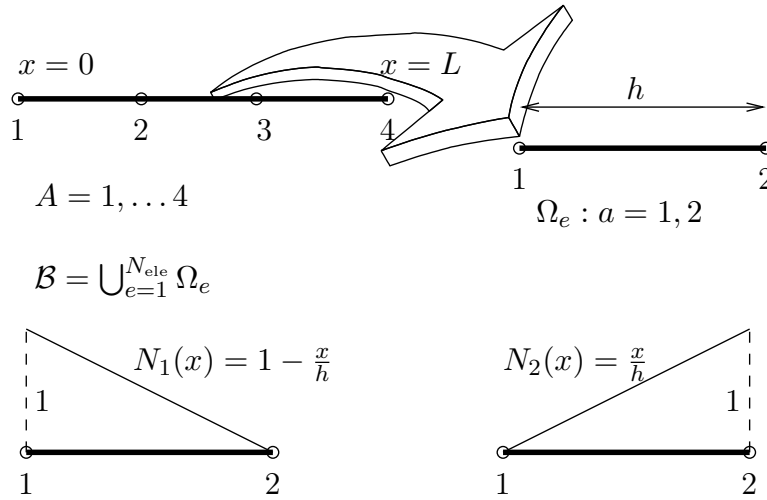


Figura 2: Esquema de las funciones de forma lineales del elemento 2 de una discretización 1D de una barra con 3 elementos

Teniendo en cuenta los valores de N_1 y N_2 que aparecen en la figura 2, se realizan las integrales de cada uno de los términos; por ejemplo K_{11}^e es:

$$K_{11}^e = \int_0^h \frac{dN_1}{dx} EA \frac{dN_1}{dx} dx = \frac{EA}{h^2} h = \frac{EA}{h} \quad (20)$$

y análogamente los demás:

$$K_{22}^e = \frac{EA}{h}; \quad K_{12}^e = K_{21}^e = -\frac{EA}{h} \quad (21)$$

resultando

$$[\mathbf{K}^e] = \frac{EA}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (22)$$

Similarmente el vector de fuerzas volumétricas del elemento resulta

$$\{\mathbf{f}^{\text{vol},e}\} = \frac{h}{6} \begin{Bmatrix} 2q_1 + q_2 \\ q_1 + 2q_2 \end{Bmatrix} \quad (23)$$

Una vez formadas las matrices elementales se convierten a numeración global,

$$\begin{aligned} K_{ab}^e &\rightarrow [\hat{\mathbf{K}}^e] \\ f_a^e &\rightarrow \{\hat{\mathbf{f}}^e\} \end{aligned}$$

ensamblándose finalmente estas matrices locales dentro de las matrices globales del sistema. Para el ejemplo de la figura 2, puesto que todos los elementos son iguales, las matrices elementales de rigidez son todas iguales entre sí, siendo su valor el de la matriz de la ecuación (22).

Para poder realizar el ensamblaje hay que tener en cuenta el lugar que van a ocupar en la matriz de rigidez global las distintas componentes de las matrices de rigidez locales. Si observamos la figura 2 vemos que el elemento 1 y el elemento 2 comparten el nodo número 2, por eso la posición (2,2) de la matriz de rigidez global será compartida por las matrices de rigidez locales de los elementos 1 y 2. La forma que tendrá esta matriz de rigidez global será:

$$K^{global} = \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & 0 & 0 \\ K_{21}^{(1)} & K_{22}^{(1)} + K_{11}^{(2)} & K_{12}^{(2)} & 0 \\ 0 & K_{21}^{(2)} & K_{22}^{(2)} + K_{11}^{(3)} & K_{12}^{(3)} \\ 0 & 0 & K_{21}^{(3)} & K_{22}^{(3)} \end{bmatrix} \quad (24)$$

De este ensamblaje, teniendo en cuenta los valores de las matrices de rigidez locales, obtenemos la matriz de rigidez global:

$$[\mathbf{K}] = \frac{EA}{h} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Las matrices elementales y global de fuerzas volumétricas son

$$\{\mathbf{f}^{\text{vol},e}\} = \frac{h}{6} \begin{Bmatrix} 2q_1 + q_2 \\ q_1 + 2q_2 \end{Bmatrix} \Rightarrow \{\mathbf{f}^{\text{vol}}\} = \frac{h}{6} \begin{Bmatrix} 2q_1^{(1)} + q_2^{(1)} \\ q_1^{(1)} + 2q_2^{(1)} + 2q_1^{(2)} + q_2^{(2)} \\ q_1^{(2)} + 2q_2^{(2)} + 2q_1^{(3)} + q_2^{(3)} \\ q_1^{(3)} + 2q_2^{(3)} \end{Bmatrix}$$

Si considerasemos una restricción del movimiento en el extremo $x = 0$ y una carga q_L aplicada en el extremo $x = L$, la ecuación matricial que resulta de aplicar el método de los Elementos Finitos es:

$$[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{f}^{\text{vol}}\} + \{\mathbf{f}^{\text{ext}}\}$$

$$\frac{EA}{h} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{Bmatrix} u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \frac{h}{6} \begin{Bmatrix} 2q_1^{(1)} + q_2^{(1)} \\ q_1^{(1)} + 2q_2^{(1)} + 2q_1^{(2)} + q_2^{(2)} \\ q_1^{(2)} + 2q_2^{(2)} + 2q_1^{(3)} + q_2^{(3)} \\ q_1^{(3)} + 2q_2^{(3)} \end{Bmatrix} + \begin{Bmatrix} R_0 \\ 0 \\ 0 \\ P \end{Bmatrix} \quad (25)$$

Al vector de fuerzas volumétricas se le suma el de externas, en este caso una posible carga en P. Por otro lado, en el extremo $x = 0$ existe una reacción para que el desplazamiento sea igual al impuesto, $u(0) = 0$. Dicha reacción se calcula *a posteriori*, una vez calculados los desplazamientos $\{\mathbf{u}\}$.

Una vez calculados los desplazamientos por la inversión de la matriz de rigidez, ya reducida por la aplicación de las condiciones de contorno, cabe la posibilidad de verificar el equilibrio, que se puede obtener como resultado de la suma de las fuerzas internas del cuerpo, que, en el caso del equilibrio, deben ser 0, siendo estas fuerzas internas calculadas como:

$$\{\mathbf{f}^{\text{int}}\} = [\mathbf{K}]\{\mathbf{u}\}$$

Por otro lado, la reacción en este caso se podría calcular multiplicando la fila eliminada de la matriz $[\mathbf{K}]$ por el vector de desplazamientos y restándole las fuerzas aplicadas en dicho nodo, que, al tratarse de un contorno tipo Dirichlet, solo podrán ser fuerzas volumétricas:

$$R_0 = K_{1j}u_j - f_1^{\text{vol}}$$

3. Simulación con MatLab / Octave

Se desea calcular la respuesta de una barra elástica unidimensional, de longitud $L = 10\text{mm}$ y sección uniforme $A = 1\text{mm}^2$. El material es elástico lineal, con módulo de Young $E = 1000\text{MPa}$. El extremo izquierdo ($x = 0$) está fijo mientras que sobre el derecho ($x = L$) actúa una fuerza axial de valor $P = 5\text{N}$, como se indica en la figura 3. Se podrá suponer que las deformaciones son pequeñas.

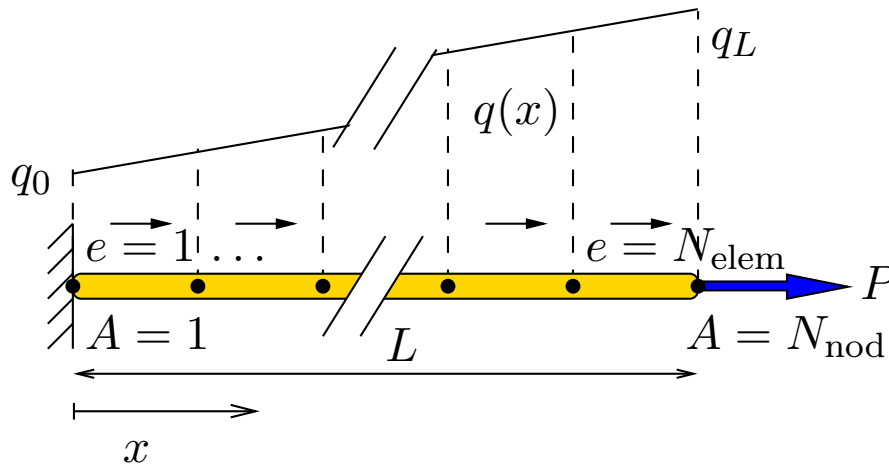


Figura 3: Modelo de fibra elástica 1D

A partir de los dígitos de las centenas (c), decenas (d) y unidades (u) del número de matrícula de cada estudiante, $N_{\text{mat}} = mcd u$, se tomarán los siguientes parámetros para el modelo:

- Número de elementos $N_{\text{elem}} = 11 - c$
- La carga distribuida $q(x)$, fuerza longitudinal por unidad de longitud, será lineal entre los extremos $x = 0$ y $x = L$, con los valores en los extremos $q_0 = d/10 \text{ N/mm}$, $q_L = q_0 + u/10 \text{ N/mm}$.

Los pasos a seguir son:

1. Planteamiento del problema elástico y condiciones de contorno.
2. Expresiones numéricas de las matrices de rigidez y de fuerzas del modelo completo.
3. Aplicación de condiciones de contorno y resolución.
4. Solución para los desplazamientos $u(x)$ y tensiones $\sigma(x)$ en cada punto, dibujando la gráfica en función de x y comparando con la solución analítica. (Nota: las tensiones se calculan a partir de los desplazamientos como $\sigma = E\varepsilon = E du/dx$, y empleando la aproximación en un elemento finito, $\sigma^h = E u_a d N_a / dx = E (u_2 - u_1) / h$.)

3.1. Código MatLab / Octave

La organización del código en **Matlab** / **Octave** es libre, pero, puesto que es la primera aproximación que el alumno hace a un código de elementos finitos, el seguir los esquemas propuestos a continuación puede resultar muy interesante para tener unas pautas.

■ Preproceso

A continuación se describe el preproceso, donde material, geometría, condiciones de contorno y mallado son definidas. Se debe también, según el tipo de problema y los grados de libertad que tenga el problema, definir el número de grados de libertad totales.

Código 1: Esquema Matlab (a)

```
% Definición de parámetros particulares de cada alumno:
Nmat=1124                                % número de matrícula del alumno
m=floor(Nmat/1000)                       % cifra de millares
c=floor((Nmat-m*1000)/100)               % cifra de centenas
d=floor((Nmat-m*1000-c*100)/10)          % cifra de decenas
u=Nmat-m*1000-c*100-d*10                 % cifra de unidades

% A: PREPROCESO
%-----

% 1. Geometría
A=1;                                     % Area
L=10;                                    % Longitud de la barra

% 2. Material
E=1000;                                  % Módulo elástico

% 3. Condiciones de contorno en carga
P=5;                                     % dato de fuerza aplicada en x=L
q0=d/10;                                 % Fuerza volumetrica en x=0
qf=q0+u/10;                             % Fuerza volumétrica en x=L

% 4. Malla
Nele=11-c;                              % número de elementos
Nnod=Nele+1;                            % número de nodos
q=linspace(q0,qf,Nnod);
h=L/Nele;                                % tamaño de cada elemento

% 5. Grados de Libertad del problema
% Dim = GDL totales = grados de libertad nodales x n° nodos
gdl=1;
Dim=Nnod*gdl;
```

■ Construcción de matrices elementales y globales

La primera parte del *solver* trata de construir, a partir de las matrices elementales, y tras el ensamblado, las matrices globales de rigidez y carga. La matriz de rigidez elemental es igual para todos los elementos, al tratarse de elementos del mismo tamaño y propiedades constantes. En cambio, la de fuerza volumétrica, en este caso varía, por ser una carga que varía a lo largo de la barra. Finalmente, en el último nodo, se ha de aplicar la carga externa de 5 N.

Código 2: Esquema Matlab (b)

```

% B: CONSTRUCCIÓN de MATRICES y VECTORES globales
%-----
% 0. Inicialización
K=zeros(Dim);
F=zeros(Dim,1);
Fext=zeros(Dim,1);

% 1. Matriz de rigidez elemental (común a todos los elementos)
k=(E*A/h)*[1,-1;-1,1];

% 2. Ensamblaje de la matriz global
for i=1:Nele
    K(i:i+1,i:i+1)=K(i:i+1,i:i+1)+k;           % ensambla rigidez
end

% 3. Ensamblaje del vector de fuerzas volumetricas global
for i=1:Nele
    % vector elemental de cargas distribuidas
    fvol=(h/6)*[2*q(i)+q(i+1);q(i)+2*q(i+1)];
    % ensambla cargas
    F(i:i+1)=F(i:i+1)+fvol;
end

% 4. Suma del vector de fuerzas externas
Fext(Nnod)=Fext(Nnod)+P;
F=F+Fext;           % suma cargas en contorno y cargas distribuidas

```

■ Aplicación de condiciones de contorno y resolución

Para el núcleo central del *solver* es necesario aplicar las condiciones de contorno, reduciendo el sistema matricial y, finalmente, invirtiendo esta matriz reducida. Se comprueba el equilibrio a posteriori.

Código 3: Esquema Matlab (c)

```

% C: APLICACIÓN de las CONDICIONES de CONTORNO y RESOLUCIÓN
%-----

% 1. Reducción de la matriz de rigidez
Kg=K(2:Nnod,2:Nnod);

% 2. Reducción del vector de fuerzas
Fg=F(2:Nnod);

% 3. Resuelve sistema reducido para desplazamientos
ug=Kg\Fg;

% 4. Desplazamientos y reacciones. Equilibrio: Sum fuerzas = 0
un=[0;ug];
fint=K*un;
r_0=fint(1)-F(1);
Eq=sum(fint);
if abs(Eq)>1e-10
    Disp('Equilibrio no alcanzado');
end

```

■ Postproceso

Dentro del postproceso nos interesa calcular las deformaciones y a través de ellas las tensiones a nivel elemental. Una vez calculada la solución analítica tanto de tensiones como de desplazamientos (ecuaciones (8)), podemos realizar una comparativa con nuestros resultados, a nivel nodal de los desplazamientos y a nivel elemental de las tensiones.

Código 4: Esquema Matlab (d)

```
% D: POSTPROCESO
%-----

% 1. Deformaciones y Tensiones
epsilon=zeros(Nele,1);
for i=1:Nele
    epsilon(i)=(un(i+1)-un(i))/h;
end
sigma=E*epsilon;

% 2. Solución analítica
dx=20;
xc=linspace(0,L,dx);
dq=qf-q0;
r=dq/L;
uc=1/(E*A)*((P+q0*L+1/2*r*L^2)*xc-(1/2*q0*xc.^2+1/6*r*xc.^3));
sigmac=1/A*(P+q0*(L-xc)+1/2*r*(L^2-xc.^2));

% 3. Gráfico

% 3.1. Dibuja la solución del desplazamiento en x
figure
xn=linspace(0,L,Nele+1);
plot(xn,un,'-+',xc,uc,'-', 'LineWidth',2, 'MarkerSize',10);
% hold
title(['Ejercicio de fibra elástica 1D MEF, Nmat=' num2str(Nmat)]);
legend({'Solución MEF', 'Solución analítica'}, 'Location', 'Northwest');
xlabel('Coordenada x (mm)');
ylabel('Desplazamiento u (mm)');

% 3.2. Dibuja la solución de la tensión en x
figure
xe=linspace(h/2,Nele*h-h/2,Nele);
plot(xe,sigma,'o',xc,sigmac,'-', 'LineWidth',2, 'MarkerSize',10);
hold;
xs=linspace(0,L,Nele+1);
stairs(xs,[sigma;sigma(Nele)], 'LineWidth',2);
title(['Ejercicio de fibra elástica 1D MEF, Nmat=' num2str(Nmat)]);
legend({'Solución MEF', 'Solución analítica'}, 'Location', 'Northeast');
xlabel('Coordenada x (mm)');
ylabel('Tensión \sigma (N/mm^2)');

pause;
```

A continuación, en la figura 4, podemos ver la solución obtenida con el código que hemos desarrollado en **MatLab / Octave** y la solución analítica a partir de las ecuaciones (8). Como vemos, la solución de los desplazamientos ha sido obtenida en los nodos, por lo que a la localización de cada nodo se corresponderá un valor del desplazamiento. En cambio, las tensiones son obtenidas en los puntos de integración de los elementos, en este caso un punto por elemento en el medio del mismo. Por ello debemos dibujar el valor de las tensiones en la localización de los puntos de integración. Además hemos añadido la función *stairs* para resaltar que el valor de las tensiones es constante en cada elemento.

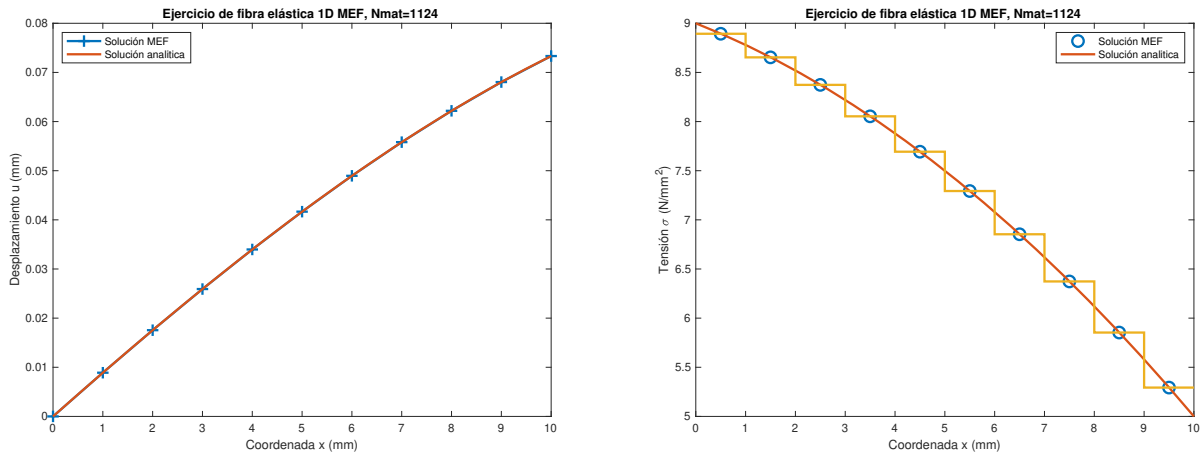


Figura 4: Comparación entre la solución obtenida con MatLab y la solución analítica.

4. Simulación con Abaqus

El modelo a realizar es similar al explicado en la sección 3, solo que en este caso empleamos el programa **Abaqus**. Aunque el modelo es unidimensional, decidimos emplear elementos 3D para ayudar a visualizar la geometría de la barra e introducir este tipo de modelado. Seguiremos los pasos similares a la práctica anterior, solo que en este caso vamos a realizar nuestro primer modelo 3D.

4.1. Módulo Part

En primer lugar, se ejecuta *Abaqus CAE* para crear un modelo nuevo. Se entra en el módulo **part**, activando el icono de crear una nueva parte, que se define como 3D, deformable, solid por extrusión (figura 5a). Una vez definido el problema, creamos un rectángulo (figura 5b) en las esquinas (-0.5,-0.5) y (0.5,0.5), por lo que tendremos un area de 1 mm^2 .

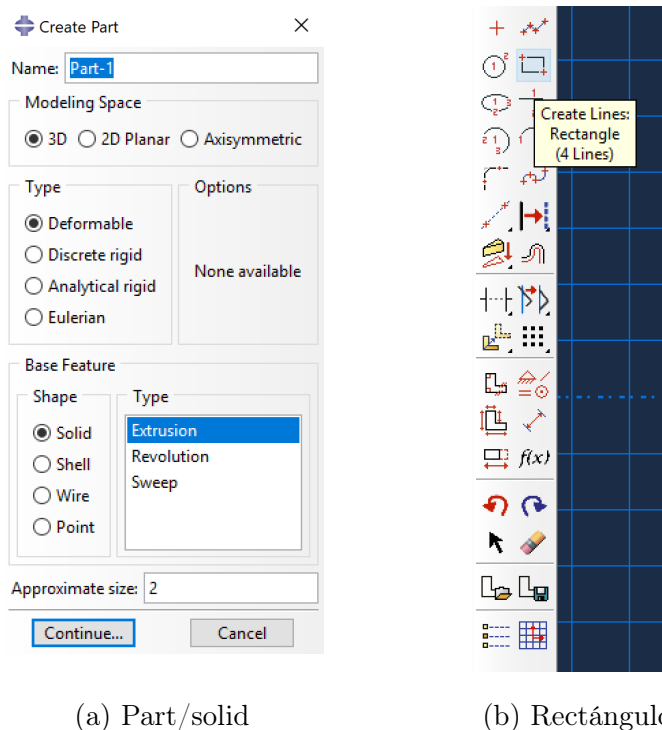


Figura 5: Creación de geometría

Una vez definido, hacemos clic en *done* y pasamos a definir la longitud de la barra en la dirección Z , que en nuestro caso es 10 mm. Deberíamos obtener un cuerpo similar al del la figura 7.

4.2. Módulo Property

Se activa el icono para crear material nuevo, (figura 8), se selecciona el material elástico lineal y se introducen las propiedades $E = 1000 \text{ MPa}$, $\nu = 0$. Hemos de tener en cuenta que al trabajar en mm y N para la carga, debemos introducir el módulo elástico en $\text{MPa} = \text{N/mm}^2$. A continuación se crea una “sección”, en la que se definen sus propiedades, en este caso solo el tipo de material descrito anteriormente, tipo *solid*-homogéneo (figura 9a). A continuación se asigna la sección a la parte creada (figura 9b), seleccionando todo el volumen y haciendo clic en *Done*.

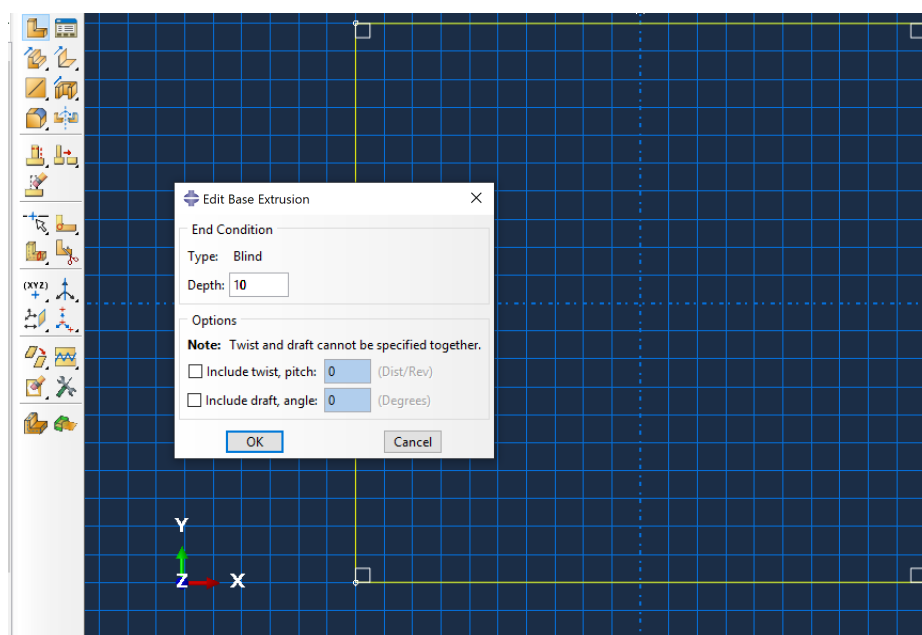


Figura 6: Extrusión de la sección.

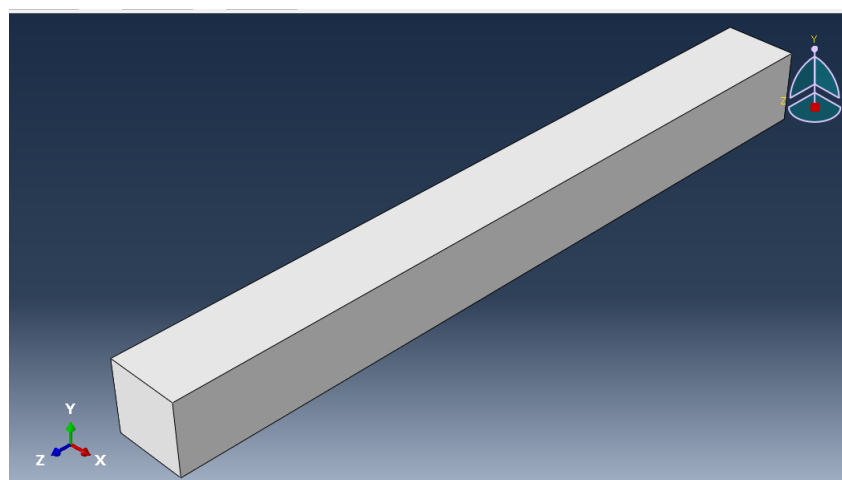


Figura 7: Barra final

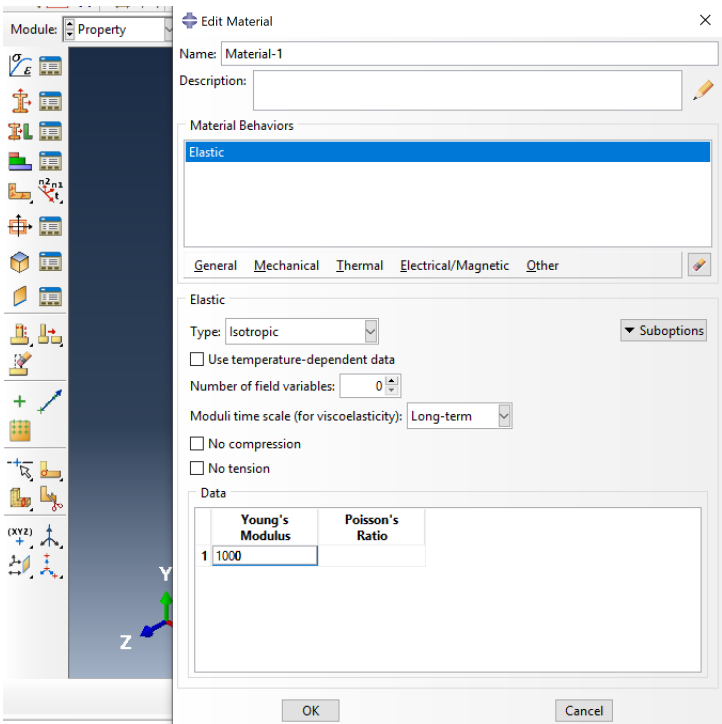
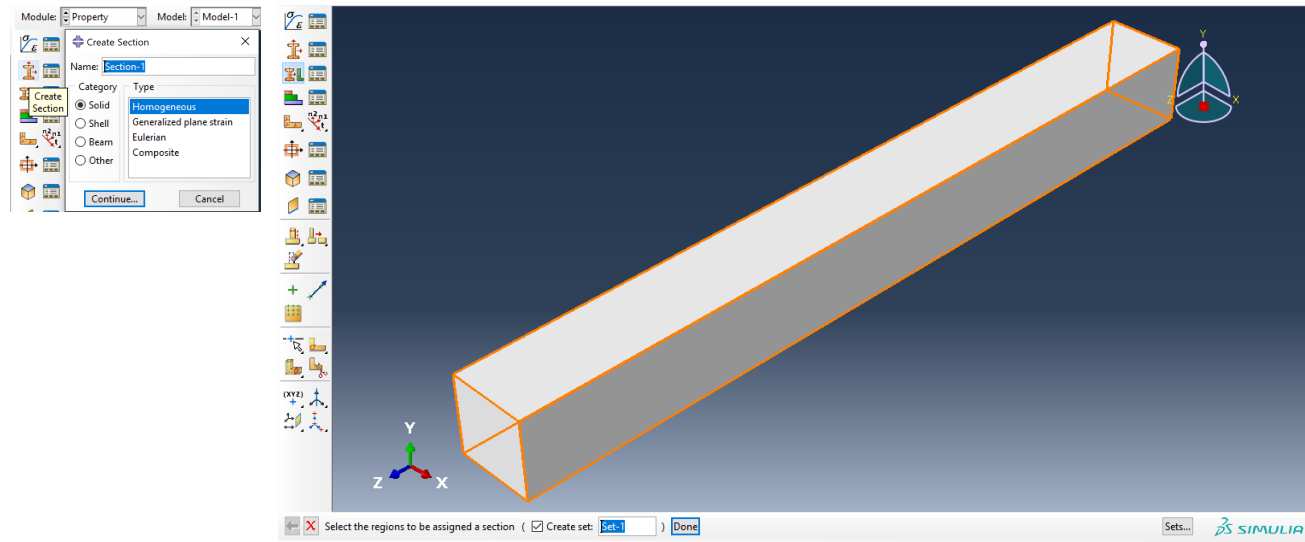


Figura 8: Material elástico.



(a) Crear sección

(b) Asignar sección

Figura 9: Selección de la sección

4.3. Módulo Assembly

En este módulo tan solo hay que crear una “instancia” a partir de la parte, mediante las opciones por defecto:

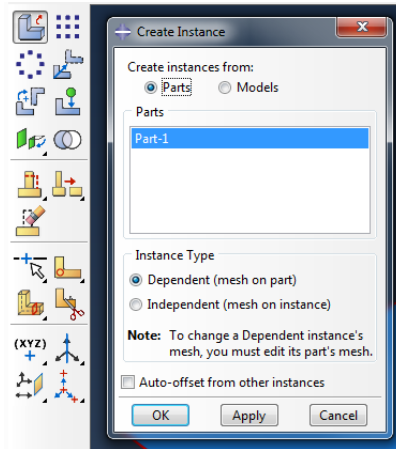


Figura 10: Assembly: crea una instancia de la parte

4.4. Módulo Step

Se crea un “step” para el procedimiento de cálculo “Static, general”. Se toman las opciones por defecto (figura 11). No es necesario editar el “field output”, pues el conjunto de variables de salida por defecto incluye ya las que necesitaremos en nuestro análisis.

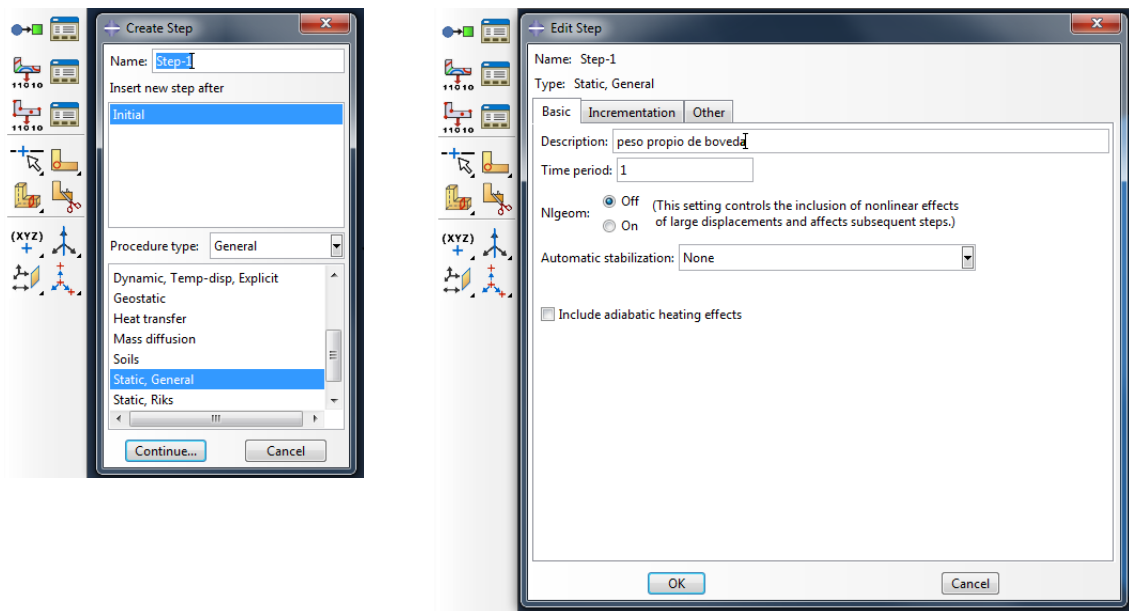


Figura 11: Crear y definir el “step”

4.5. Módulo Load

Debemos definir en este módulo ambos tipos de condiciones de contorno. En primer lugar, en $Z=0$, empotramos la barra. Para ello creamos *Boundary Condition*, de categoría *Mechanical* y tipo *Encastre* (figura 12).

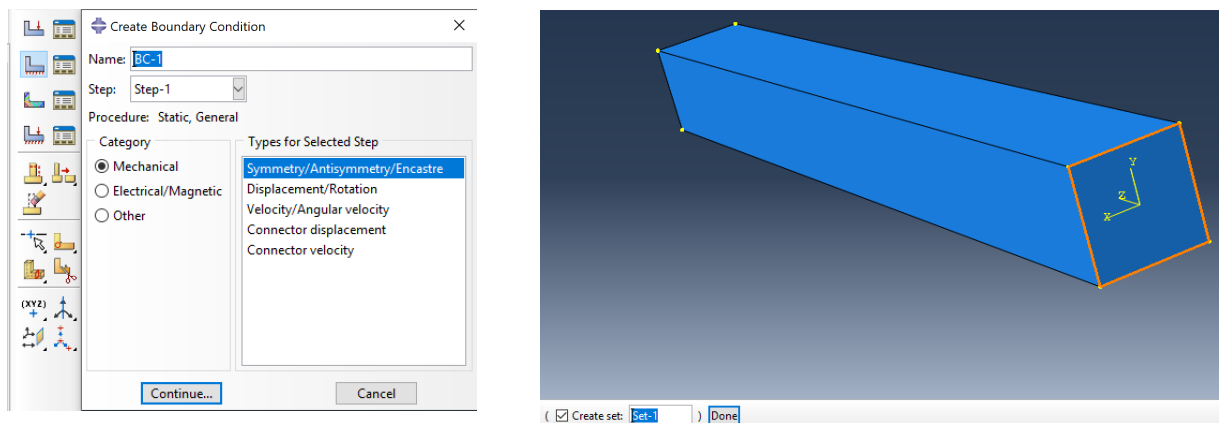


Figura 12: Creación del empotramiento

Una vez seleccionada la superficie a empotrar y habiendo hecho clic en *Done*, debemos especificar el tipo de condición a asignar dentro de este subgrupo, por lo que seleccionaremos la última de esta nueva pestaña (figura 13), donde menciona la opción *Encastre*.

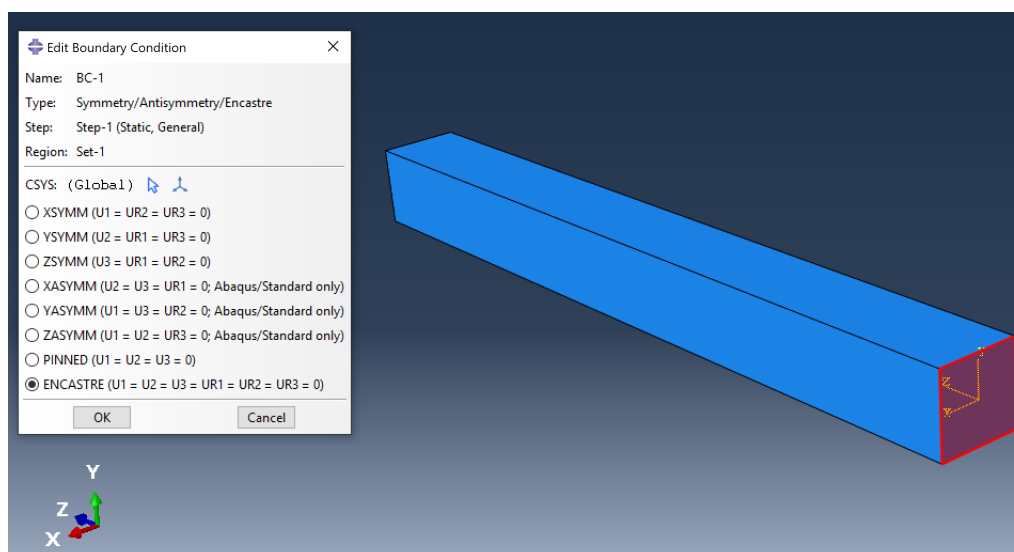
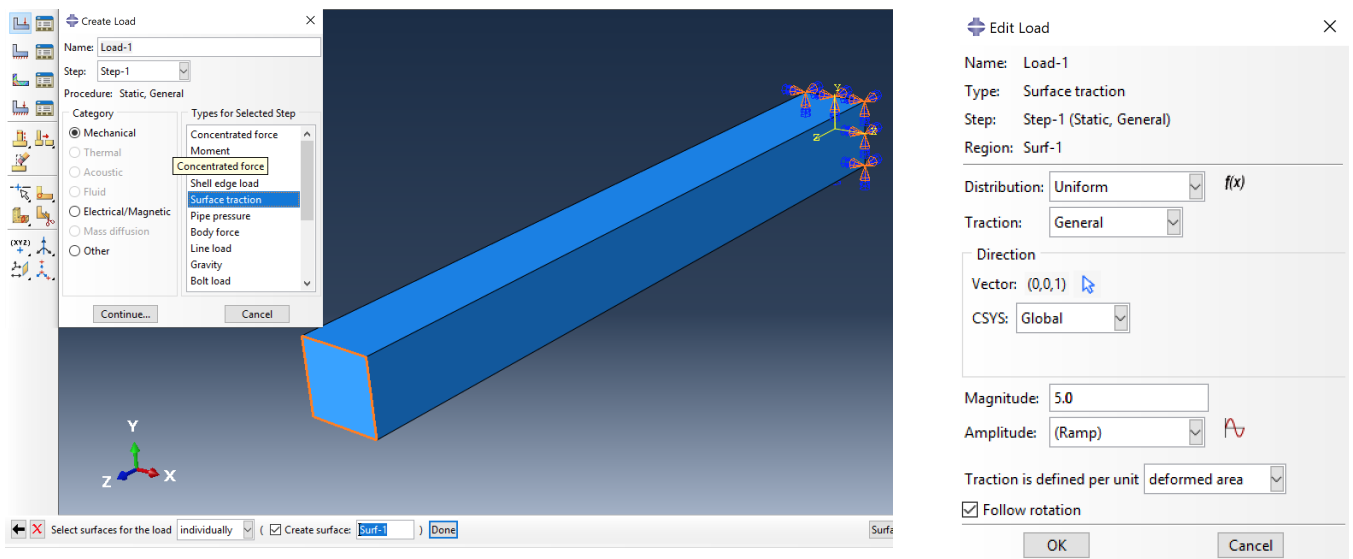


Figura 13: Tipo de *Boundary Condition*: *Encastre*

Pasamos entonces a definir las dos condiciones de carga que aplican en nuestro problema: carga puntual y volumétrica. Con respecto a la primera se puede realizar de diversas formas. La que consideramos más inmediata es aplicar una tracción a la superficie en $Z=L$, teniendo en cuenta que en **Abaqus** es una carga de categoría *Mechanical* y tipo *Surface Traction* (figura 14a). Definimos a continuación la superficie donde se va a aplicar y se nos abrirá la pestaña de edición de carga (figura 14b). La carga es de distribución uniforme, y la tracción de tipo *General*. Para la dirección debemos dibujar en el modelo un vector que siga la dirección creciente del eje Z , por lo que se nos debe escribir un vector unitario $(0,0,1)$. La magnitud será de 5 N dividido de la superficie donde se aplica, 1 mm^2 , es decir, 5 MPa.

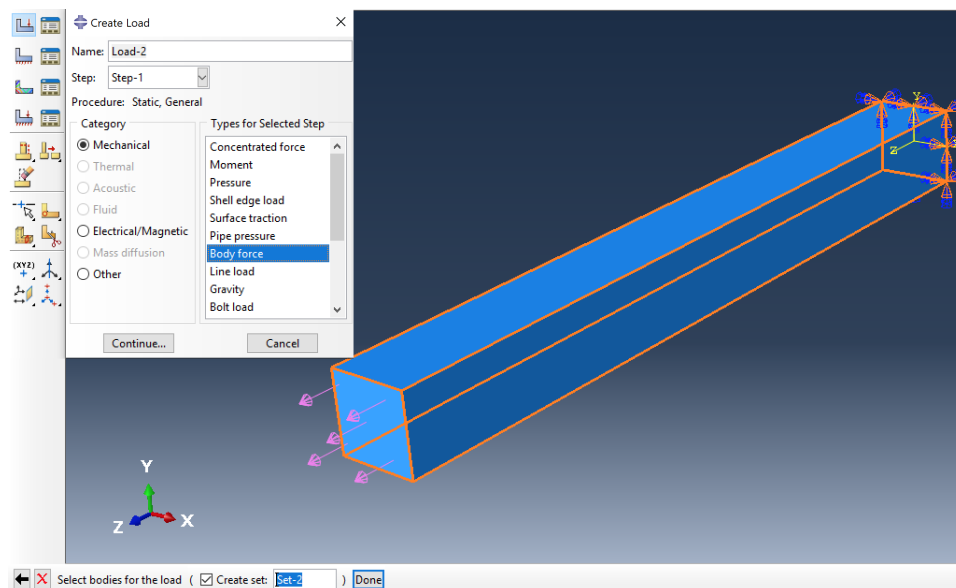


(a) Creación de la carga y asignación

(b) Definición

Figura 14: Carga puntual

En lo que respecta a la carga volumétrica repartida en todo el cuerpo se ha de seleccionar el tipo *Body Force* en el apartado de creación de cargas y posteriormente asignársela a todo el cuerpo (figura 15).

Figura 15: Creación de la carga *Body Force*

A continuación hemos de definir dicha carga. Para ello hemos de hacer clic primero en un botón con el símbolo $f(x)$, puesto que no será un valor puntual sino una función. Para la resolución de este problema optamos por asignar una expresión. Se nos abra una ventana de creación de *Expression Field*. Teniendo en cuenta que el número de matrícula asignado en esta práctica es 1124, deberemos crear una carga que siga la ecuación $q = 0.2 + 0.04 * Z$, puesto que nuestra barra sigue dicha dirección en el modelo de **Abaqus**. Será esta la ecuación que deberemos escribir en el campo correspondiente. Una vez creado, en el campo *Distribution*, en la ventana *Edit Load*, asignaremos el *Expression Field* creado, en nuestro caso *AnalyticalField-1*. La componente será la (0,0,1), es decir, siguiendo la dirección Z .

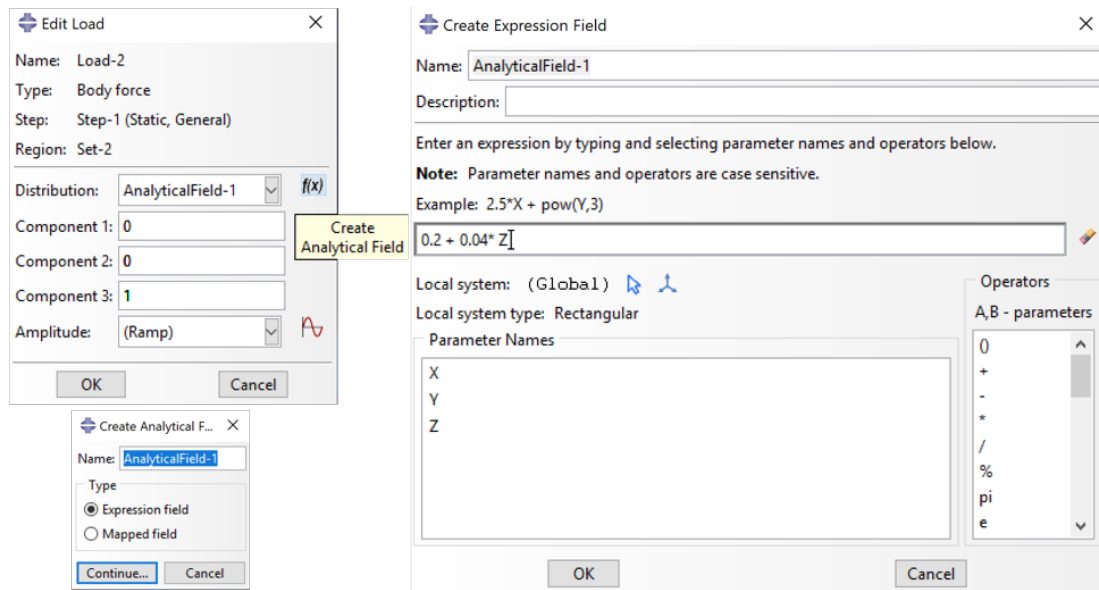


Figura 16: Definición de la distribución de carga con una expresión

Una vez definida esta segunda carga, nuestro modelo debe tener una apariencia como la que vemos a continuación:

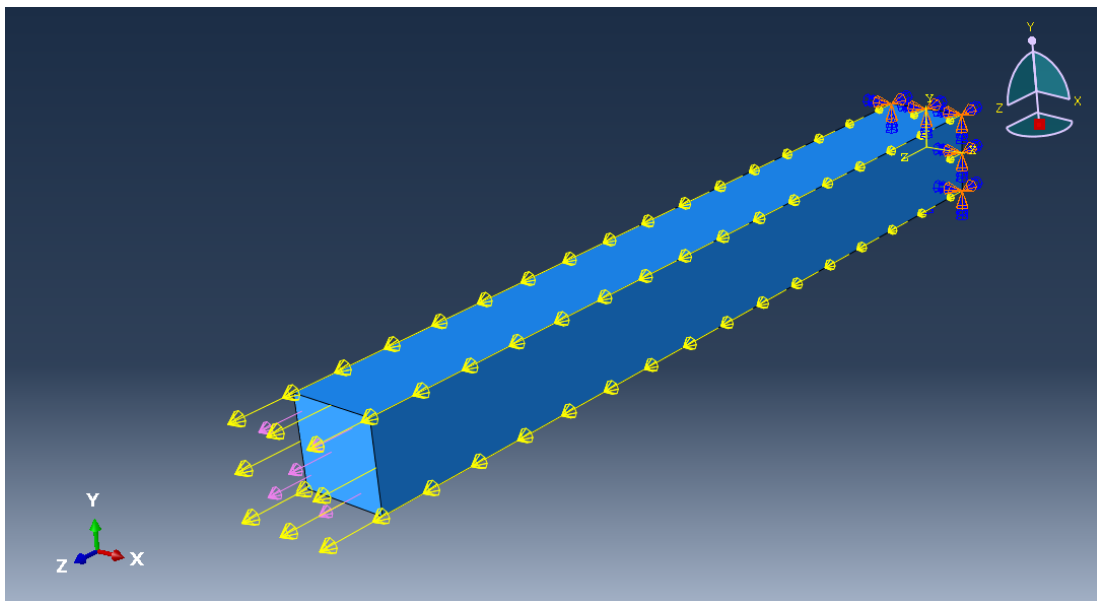


Figura 17: Distribución de cargas final

4.6. Módulo Mesh

En el módulo mesh hay que comenzar por seleccionar la parte expandiendo el árbol del modelo / parte de la izquierda, y activando con el botón derecho el icono “Mesh”: Fig. 18. Se abre en el menú superior *Mesh* → *Controls* y se selecciona “Hex” y “Structured”, para generar una malla estructurada de cuadriláteros (Fig. 19).

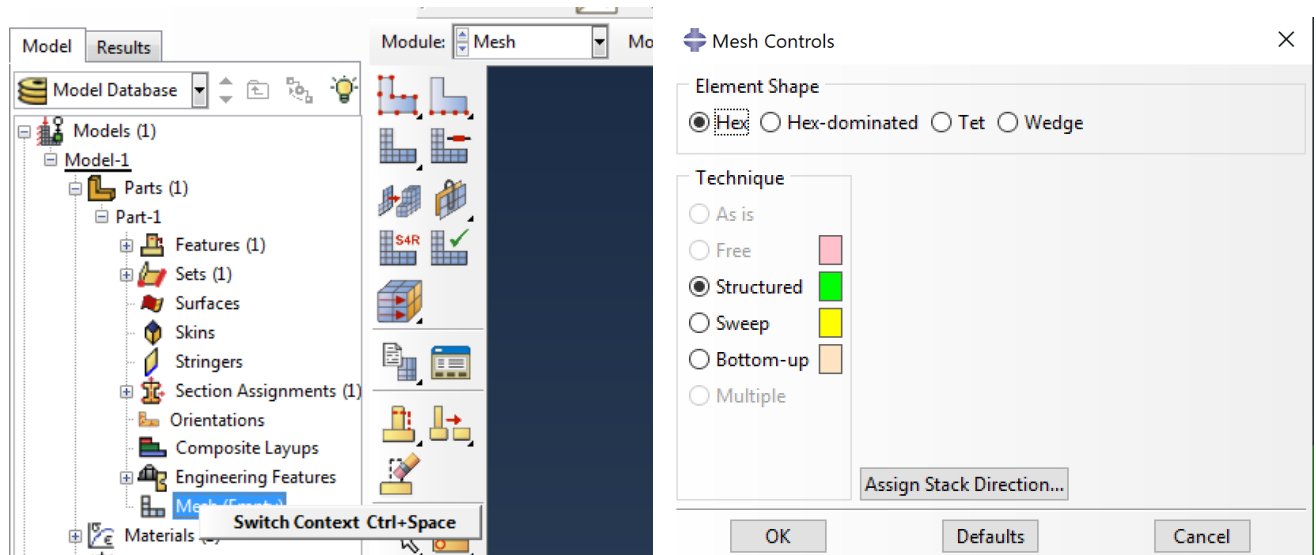


Figura 18: Seleccionar en el menú desplegado del modelo para mallar. Figura 19: Definir opciones en “Mesh controls”

A continuación se abre en el menú superior *Mesh* → *Element type* y se seleccionan las opciones “3D Stress”, “Linear”, “Reduced integration” lo que dará lugar al elemento C3D8R, suficiente para el cálculo que vamos a realizar. (Fig. 20).

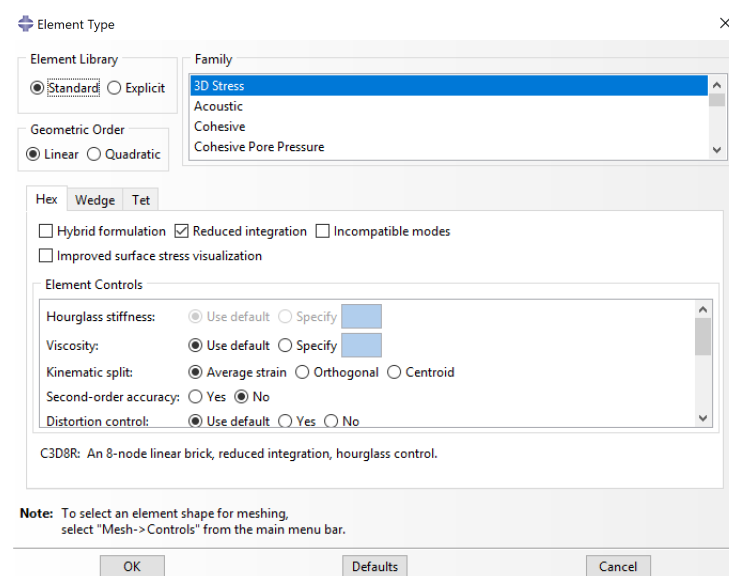


Figura 20: Seleccionar tipo de elemento en “Mesh/element”

El tamaño “Semilla” que vamos a emplear es de 1 mm en todas las direcciones del espacio, teniendo en cuenta queremos realizar un mallado de 10 elementos hexaédricos en la dirección longitudinal de la barra (Fig. 21).

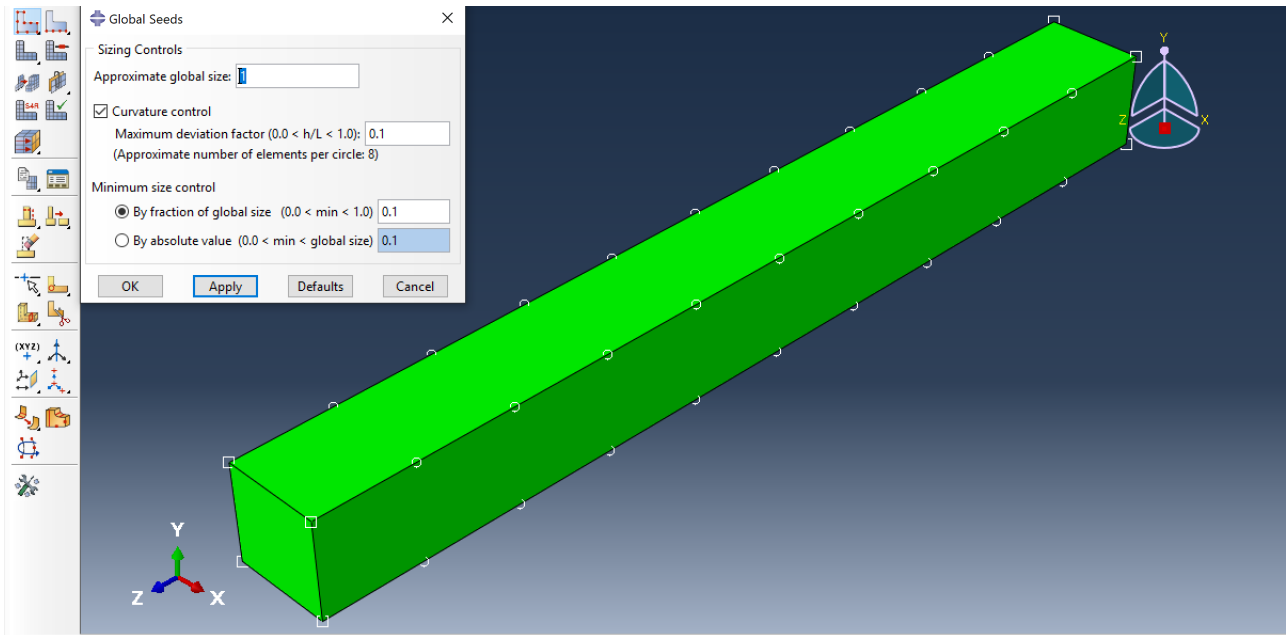


Figura 21: “Global Seeds” empleado

Por tanto, la malla que resulta es la que vemos a continuación

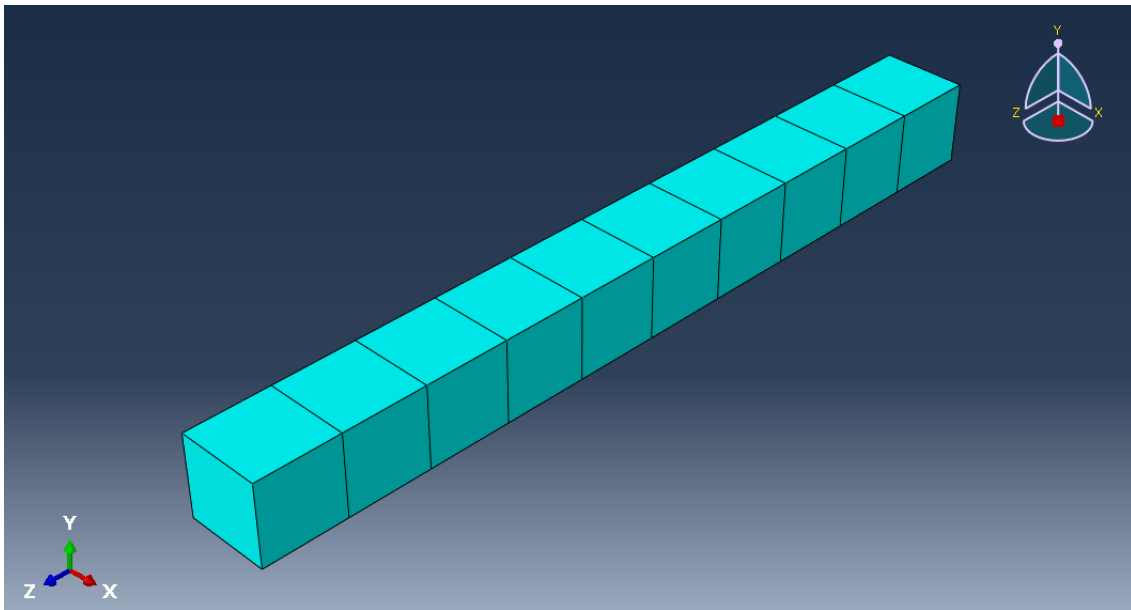


Figura 22: Malla obtenida

4.7. Módulo Job

Una vez completado el modelo, se crea un “Job” con las opciones por defecto (figura 23).

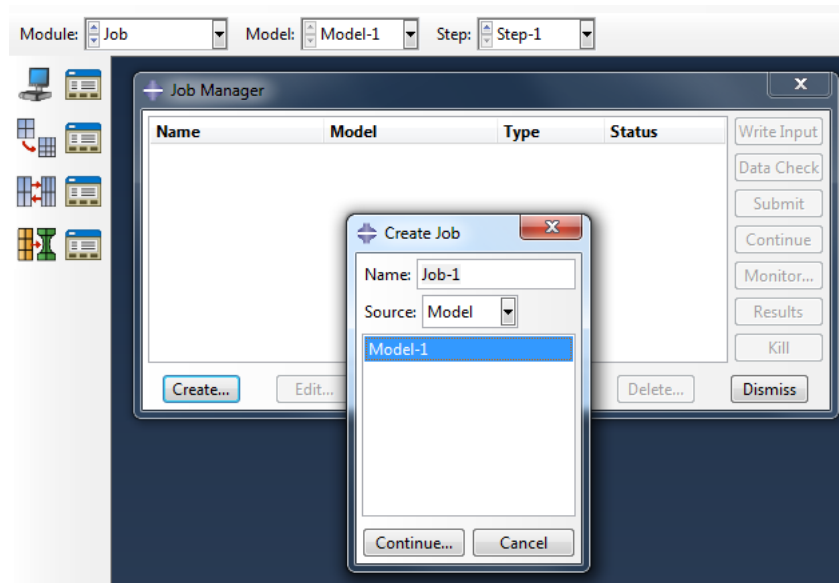


Figura 23: Creación del “Job”

Y se envía para calcular mediante “Submit”. El *Status* va cambiando de “Submitted” → “Running” → “Completed”. (figura 24) Si no hay mensaje de error el problema está acabado y

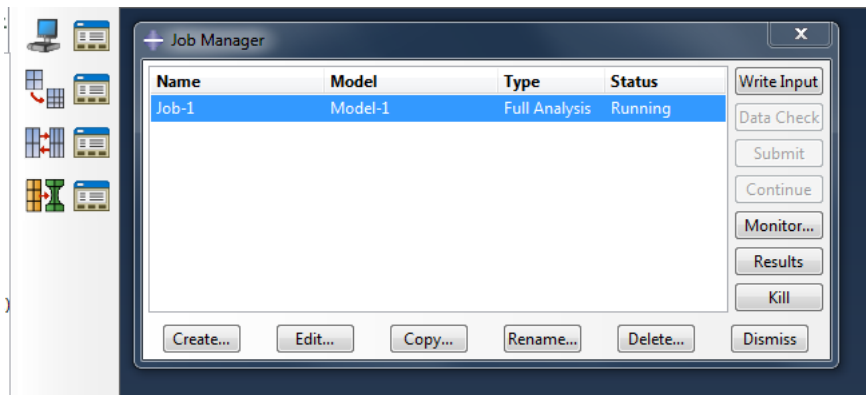


Figura 24: Envío del “Job”

se pasa al módulo de visualizar los resultados.

4.8. Módulo Visualization

A modo de comprobación, el primer campo a visualizar es el del desplazamiento en Z . Vemos que el desplazamiento máximo es de 0.073 mm, similar al obtenido en **MatLab** (figura 25). Vamos a dibujar la distribución del desplazamiento con respecto a la posición en Z .

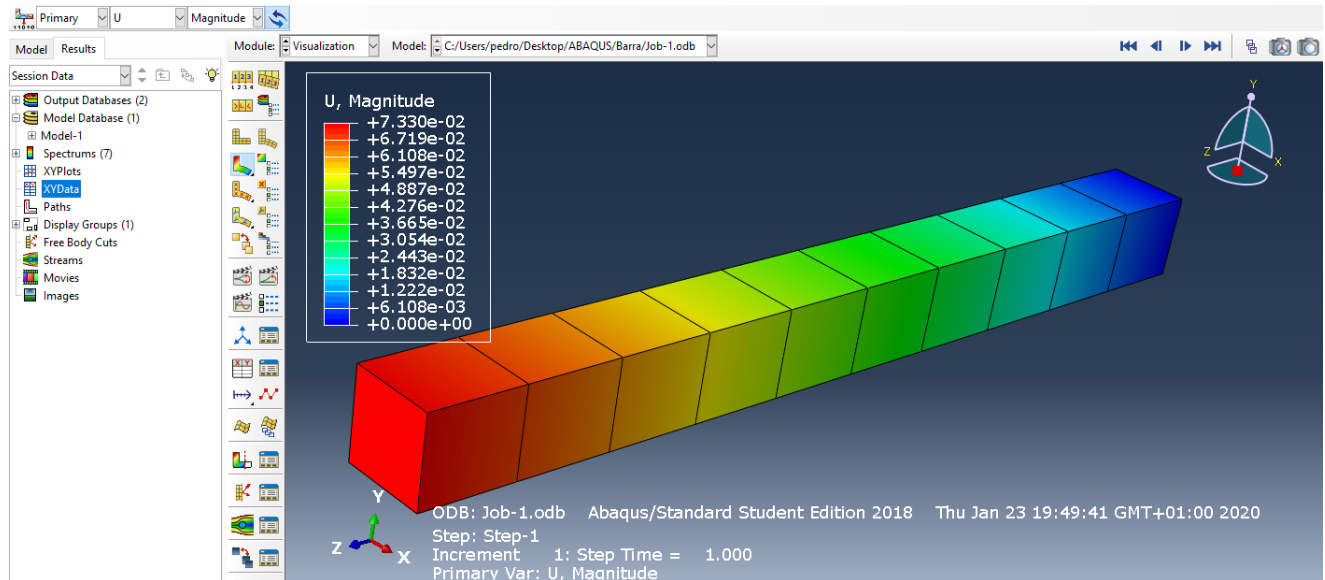


Figura 25: Campo de desplazamientos

Antes de nada hemos de indicar un *Path* que tenga nodos distribuidos en Z . Para ello seleccionamos path y con el botón derecho abrimos un desplegable donde debemos seleccionar *create* (figura 26). Nuestro tipo de path será creado seleccionándolos de una lista de nodos. Nuestro modo de selección será según avanzamos en el eje Z , por lo que seleccionamos *Add After* y debemos cuidar de escoger primero un nodo en $Z = 0$ y el siguiente nodo en $Z = L$. Podemos elegir cualquier arista, pues todas poseen la misma información. Cómo queda este path definido puede verse en la figura 27.

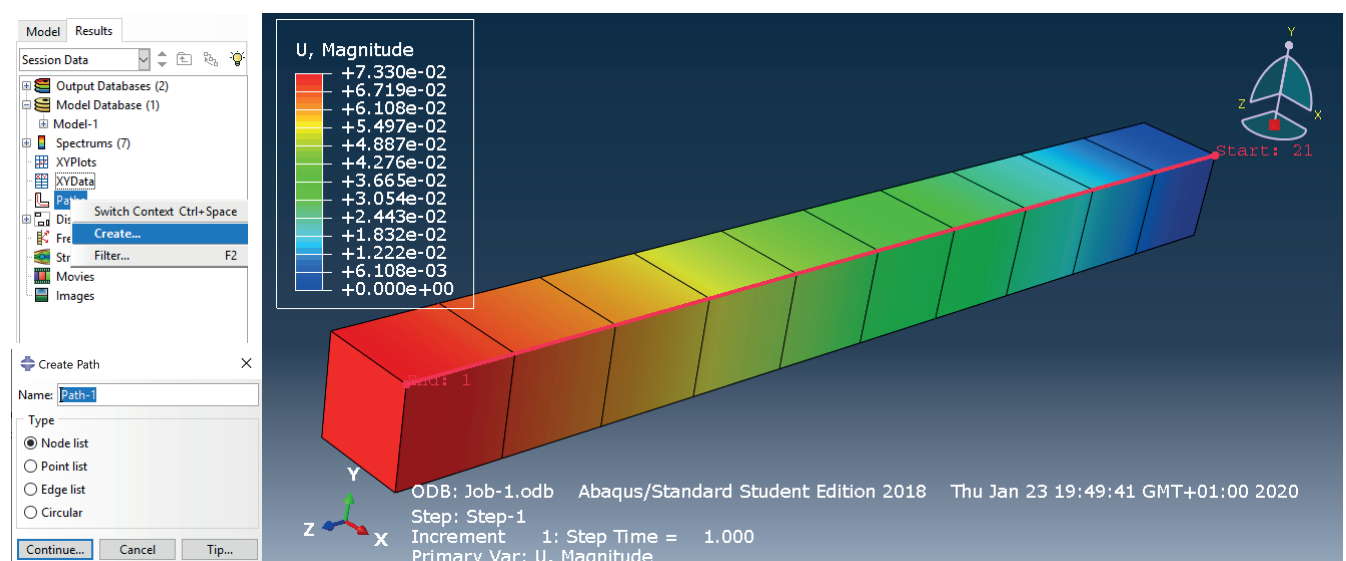


Figura 26: Definición de un path

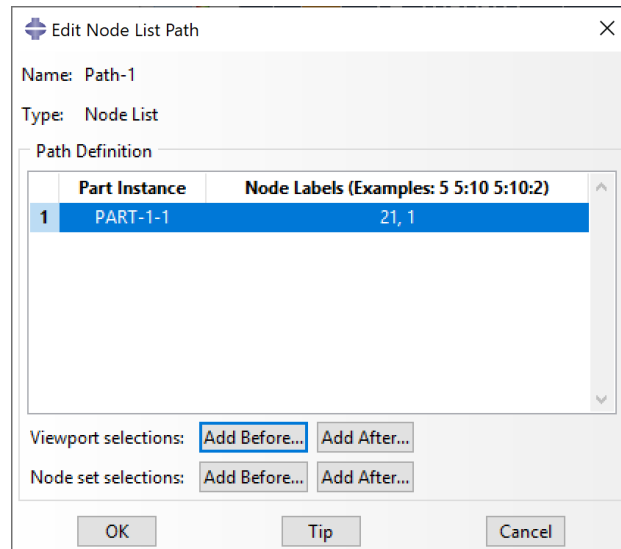


Figura 27: Lista de nodos de un path

Ya es posible crear un gráfico del desplazamiento en Z a lo largo de la barra. Para ello, necesitamos crear un *XY-Data*. En dicho campo, con el botón derecho se nos abre un desplegable donde, al seleccionar *Create*, podemos crear dicho *XY-Data* desde el un Path ya definido. En nuestro caso el campo ha de ser el desplazamiento en Z, *U3*. Con este campo seleccionado debemos, en la ventana *XY Data from Path* (figura 28), seleccionar el Path creado anteriormente (en su configuración no deformada) y seleccionar la opción de incluir las intersecciones, para añadir así los valores de los nodos intermedios. Finalmente, optamos por elegir, como valor en X de nuestro gráfico, la coordenada Z.

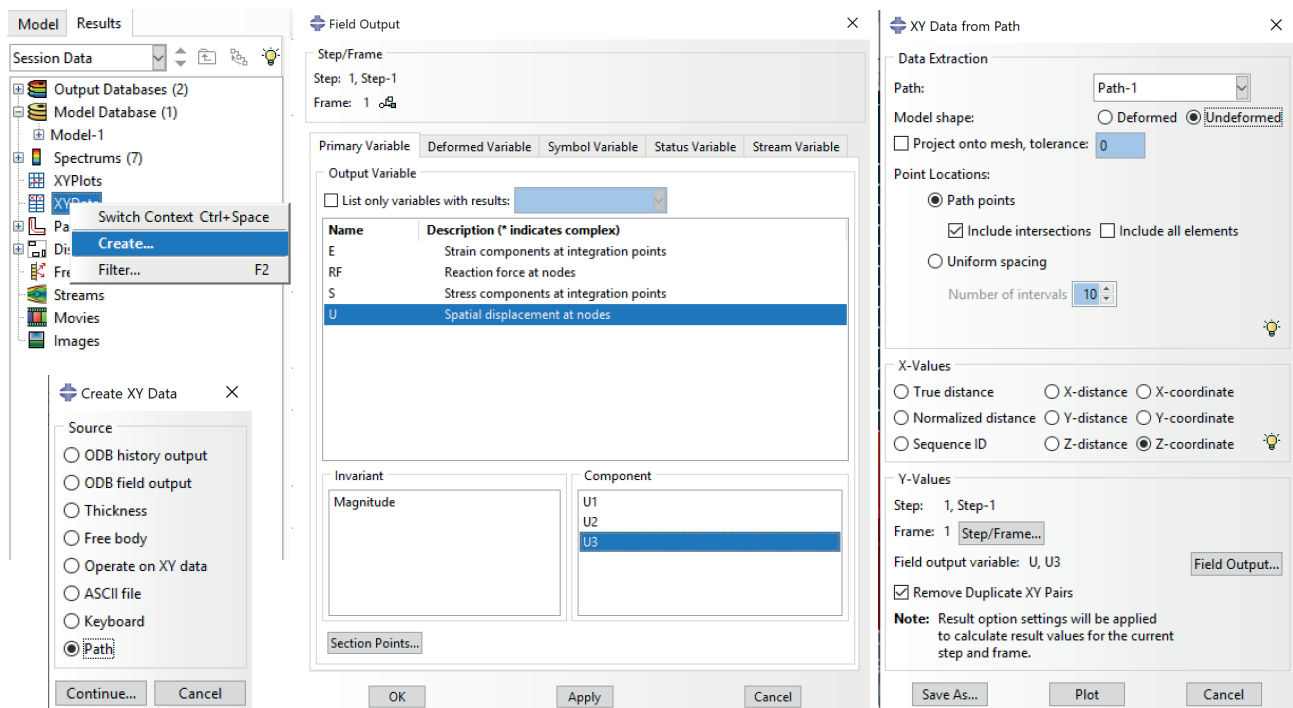


Figura 28: Creación de un XY-data

Si dibujamos dicho gráfico obtenemos una figura similar a la que se puede ver en la figura 29. Comparando la forma y valores de esta gráfica, con la obtenida en **MatLab** vemos una correspondencia bastante alta, si bien necesitaremos superponer ambas gráficas.

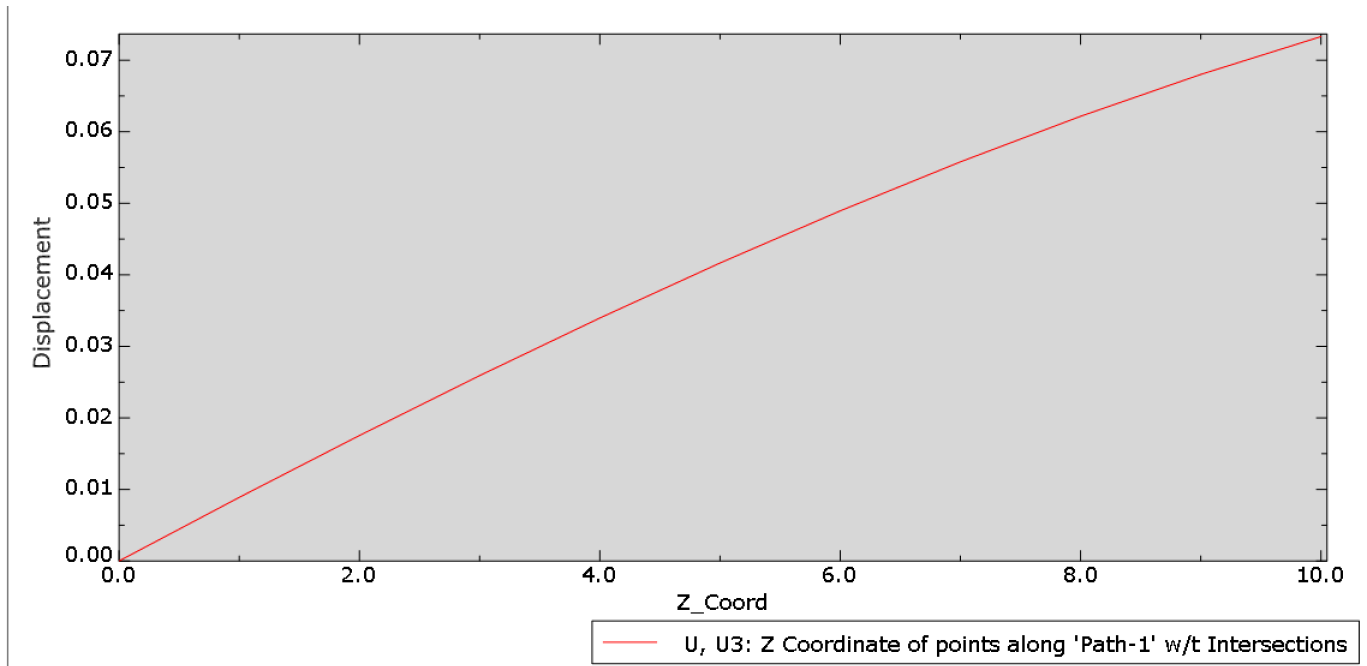
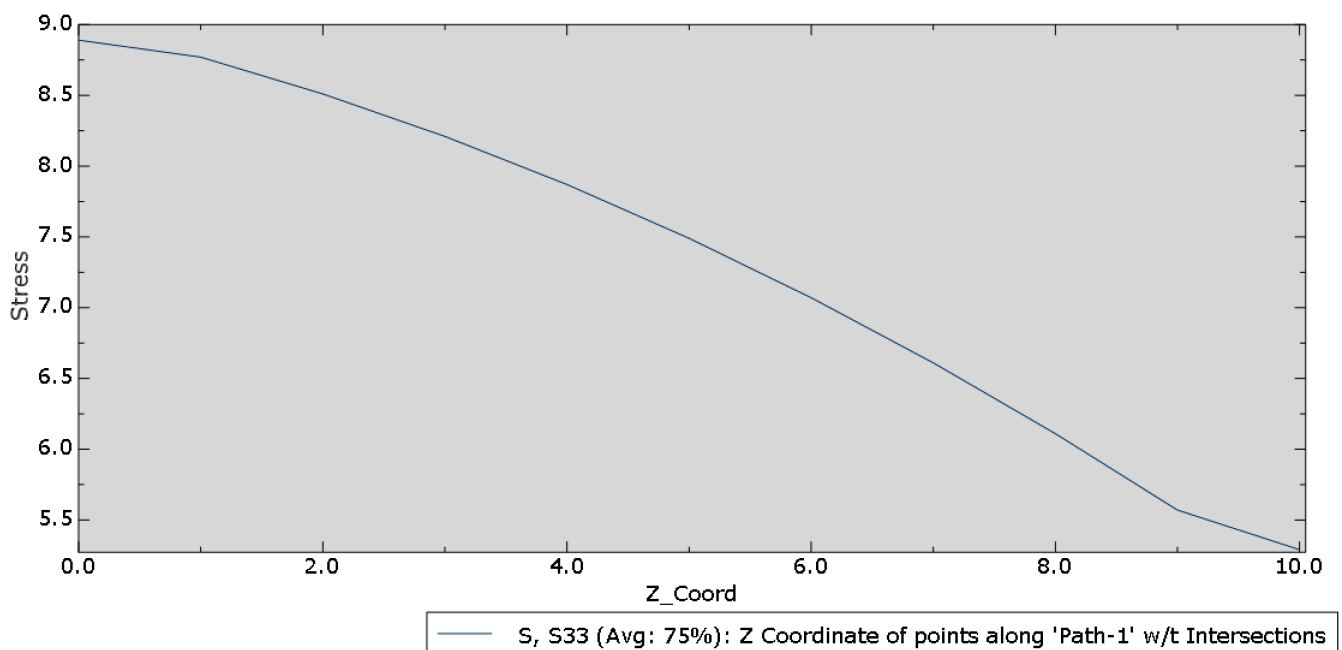


Figura 29: Desplazamiento en en eje Z

Podemos guardar estos datos, para posteriormente exportarlos a **MatLab** para su comparativa con la solución analítica, haciendo clic en el botón *Save as...* Si intentásemos cambiar el campo de dibujo al de las tensiones, en este caso S33 que es la que se corresponde con σ_z , observamos, en la figura 30. En cambio, aunque la tendencia y valores son cercanos, en los extremos ocurren cosas raras. Esto se debe a que la tensión es un dato correspondiente a los puntos de integración, no a los nodos, que es donde se puede hacer este path. Por tanto, para la comparativa, deberemos extraer los datos de otra manera.

Figura 30: Tension σ_z a lo largo del eje Z

4.9. Comparativa con solución analítica en MatLab / Octave

En primer lugar debemos dejar preparado el código de MatLab / Octave para introducir los datos procedentes de *Abaqus*. A continuación se presenta el código, similar al de postproceso empleado anteriormente.

Código 5: Esquema Matlab-Abaqus

```
% E: ABAQUS
%-----

xn_ab=[

];

un_ab=[

];

xe_ab=[

];

sigma_ab=[

];

% % 1. Dibuja la solucion del desplazamiento en x
figure
plot(xn_ab,un_ab,'-+',xc,uc,'-', 'LineWidth',2, 'MarkerSize',10)
%hold
title(['Ejercicio de fibra elástica 1D MEF, Nmat=' num2str(Nmat)])
legend({'Solución MEF Abaqus','Solución analítica'}, 'Location','Northwest')
xlabel('Coordenada x (mm)')
ylabel('Desplazamiento u (mm)')

% 2. Dibuja la solucion de la tensión en x
figure
plot(xe_ab,sigma_ab,'o',xc,sigmac,'-', 'LineWidth',2, 'MarkerSize',10)
hold
xs=linspace(L,0,Nele+1);
stairs(xs,[sigma_ab;sigma_ab(Nele)], 'LineWidth',2)
title(['Ejercicio de fibra elástica 1D MEF, Nmat=' num2str(Nmat)])
legend({'Solución MEF Abaqus','Solución analítica'}, 'Location','Northeast')
xlabel('Coordenada x (mm)')
ylabel('Tensión \sigma (N/mm^2)')
```

Como vemos, dejamos huecos para los vectores que provengan de los resultados de *Abaqus*. Los dos primeros vectores pueden ser rellenados rápidamente, pues son datos que ya poseemos. Para ello debemos abrir un desplegable con el botón derecho desde el XYData que hemos guardado anteriormente y seleccionar *Edit...* Directamente obtendremos dos columnas donde la X será la coordenada Z y la Y el valor de U3 (figura 31). Podemos pegar ambos en los huecos *xn_ab* y *un_ab* respectivamente.

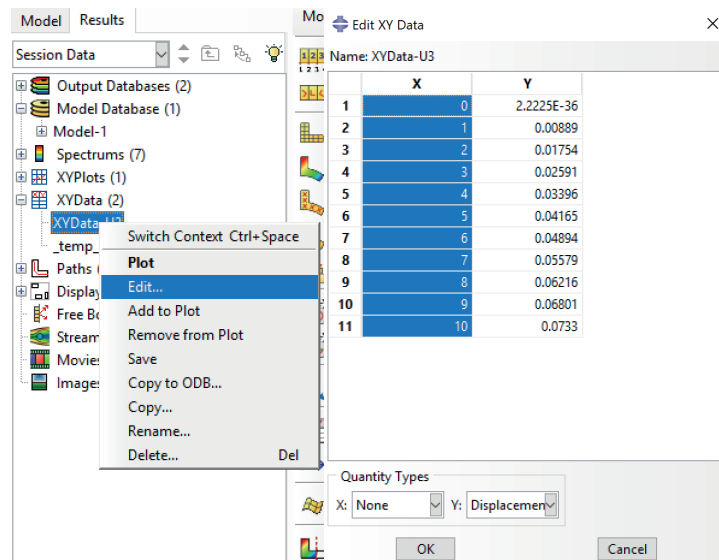
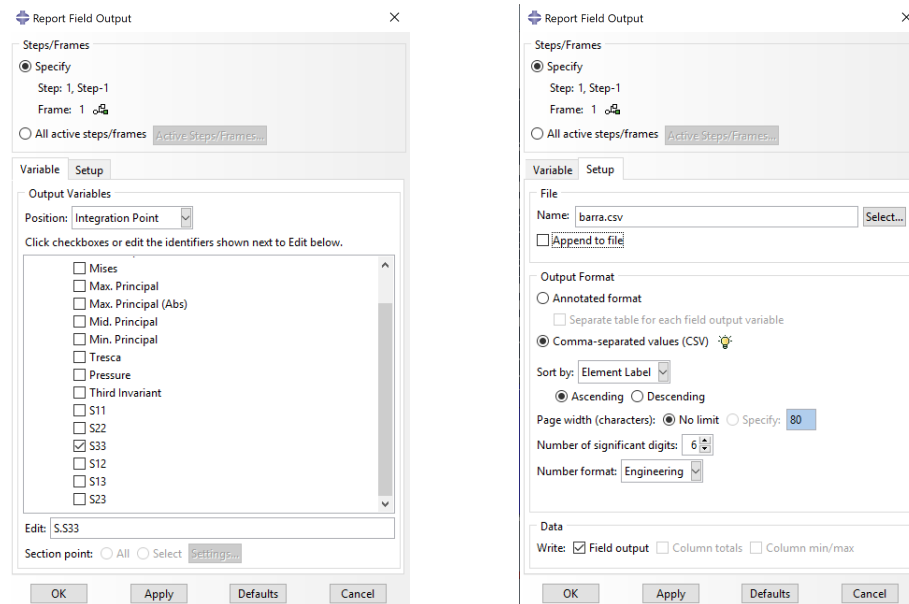


Figura 31: Exportación de un XY-data

Por otro lado, como hemos visto anteriormente, los resultados que tenemos con este método de la tensión no son válidos para la comparativa. Es por ello que lo realizamos de manera distinta al tratarse de información que está en los puntos de integración. Debemos escoger, en la pestaña “Report/” la opción *Field Output*. Se nos abrirá una ventana donde, en la pestaña *Variable* deberemos seleccionar el Field S33 en los puntos de integración (figura 32a), mientras que en la pestaña *Setup* debemos modificar el tipo de archivo a un *.csv (figura 32b).

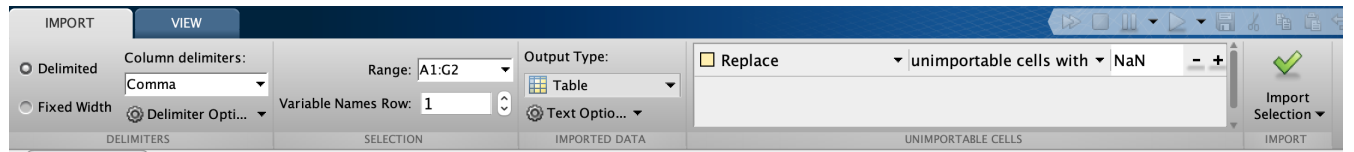


(a) Variable

(b) Setup

Figura 32: Field Output

Finalmente, el archivo *.csv generado ha de ser trasladado a la carpeta donde se encuentra nuestro archivo de MatLab / Octave, puesto que este programa posee su propio intérprete de este tipo de archivos. Si abrimos el archivo desde el programa MatLab veremos una ventana similar a la de la figura 33. La importación debe ser delimitada por comas. Tendremos por tanto, una columna con la localización de los puntos de integración así como otra con los valores de S33. Debemos seleccionar ambas por separado y pegarlas en los huecos `xe_ab` y `sigma_ab` respectivamente.



	A	B	C	D	E	F	G	H	I	J	K	L	M
	barra												
	ODBName	Step	Frame	PartInsta...	ElementL...	IntPt	X	Y	Z	SectionN...	MaterialN...	Section	
	Number	Number	Categorical	Number	Number	Number	Number	Number	Number	Categorical	Number	Text	Number
1	ODB Name	Step	Frame	Part Inst...	Element ...	IntPt	X	Y	Z	Section ...	Material ...	Section P...	S-S...
2	C:/Users...	Step-1	Incremen...	PART-1-1	1	1	0.	0.	0.	9.5	ASSEMBL...	MATERIA...	5.29...
3	C:/Users...	Step-1	Incremen...	PART-1-1	2	1	0.	0.	0.	8.5	ASSEMBL...	MATERIA...	5.85...
4	C:/Users...	Step-1	Incremen...	PART-1-1	3	1	0.	0.	0.	7.5	ASSEMBL...	MATERIA...	6.37...
5	C:/Users...	Step-1	Incremen...	PART-1-1	4	1	0.	0.	0.	6.5	ASSEMBL...	MATERIA...	6.85...
6	C:/Users...	Step-1	Incremen...	PART-1-1	5	1	0.	0.	0.	5.5	ASSEMBL...	MATERIA...	7.29...
7	C:/Users...	Step-1	Incremen...	PART-1-1	6	1	0.	0.	0.	4.5	ASSEMBL...	MATERIA...	7.69...
8	C:/Users...	Step-1	Incremen...	PART-1-1	7	1	0.	0.	0.	3.5	ASSEMBL...	MATERIA...	8.05...
9	C:/Users...	Step-1	Incremen...	PART-1-1	8	1	0.	0.	0.	2.5	ASSEMBL...	MATERIA...	8.37...
10	C:/Users...	Step-1	Incremen...	PART-1-1	9	1	0.	0.	0.	1.5	ASSEMBL...	MATERIA...	8.65...
11	C:/Users...	Step-1	Incremen...	PART-1-1	10	1	0.	0.	0.	500....	ASSEMBL...	MATERIA...	8.89...

Figura 33: Importación de valores en MatLab

El resultado de la comparativa se ve en la figura 34. Se puede comprobar que el resultado es idéntico al obtenido con el programa 1D en Matlab, figura 4, y como anteriormente también coinciden de forma exacta los valores discretos en los nodos y en el centro de los elementos con la solución analítica.

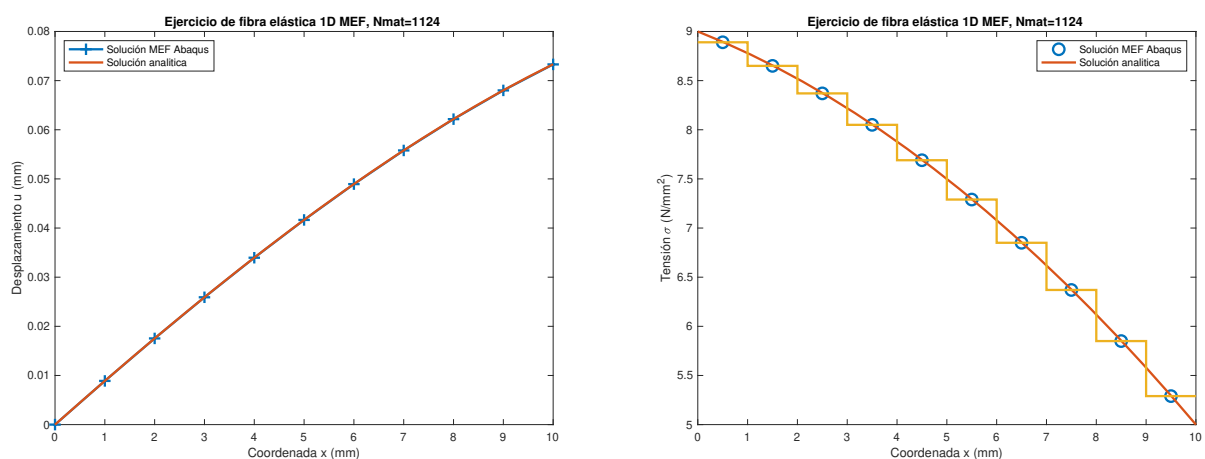


Figura 34: Comparación entre la solución obtenida con Abaqus y la solución analítica.

Observación importante.— El hecho de que los valores en los puntos discretos, tanto en tensiones como en desplazamientos, salgan exactamente iguales que los de la solución analítica, es un caso que se denomina “*superconvergente*”. Sin embargo, en general las soluciones numéricas

en los puntos discretos no coincidirán con la solución exacta. Dicho de otra manera, en este caso extremadamente sencillo el único error cometido por la solución de elementos finitos es para los valores intermedios en el interior de los elementos, siendo exactos los valores en los nodos y en los puntos de integración; sin embargo, en un caso general, habrá un error numérico también para los valores en los nodos y en los puntos de integración.