











(8). Se obtienen 56 ecuaciones. A modo de ejemplo, si se toma como nodo a optimizar el 10 se obtiene:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{u}} \left\{ \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{F} + \right. \\ \left. \lambda_{x_{10}} (u_{10} - u_{18} + \varepsilon_x \Delta x_{10-18}) + \right. \\ \left. \lambda_{y_{10}} (v_{10} - v_{18} + \frac{\gamma_{xy}}{2} \Delta x_{10-18}) \right\} = 0 \\ \frac{\partial}{\partial \mathbf{u}} \left\{ \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{F} + \right. \\ \left. \lambda_{x_{10}} (u_{10} - u_{18} + g_{x_{10}}) + \right. \\ \left. \lambda_{y_{10}} (v_{10} - v_{18} + g_{y_{10}}) \right\} = 0 \end{aligned} \quad (12)$$

Estas ecuaciones se organizan de la siguiente manera:

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{u}} = 0 \rightarrow \mathbf{K} \mathbf{u} + \lambda_{x_{10}} \begin{pmatrix} \vdots \\ 1 \\ 0 \\ \vdots \\ \vdots \\ -1 \\ 0 \\ \vdots \end{pmatrix} + \lambda_{y_{10}} \begin{pmatrix} \vdots \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ -1 \\ \vdots \end{pmatrix} = \mathbf{F} \\ \frac{\partial f}{\partial \lambda} = 0 \rightarrow \begin{matrix} u_{10} - u_{18} = -g_{x_{10}} \\ v_{10} - v_{18} = -g_{y_{10}} \end{matrix} \end{aligned} \quad (13)$$

Por tanto, para cada caso de deformación media, se construyen las matrices  $\mathbf{C}$  y el vector  $\mathbf{g}$  y se resuelve el sistema mostrado en la ecuación (14).

$$\begin{pmatrix} \mathbf{K} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{g} \end{pmatrix} \quad (14)$$

### 3.3. Redes neuronales

Una red neuronal es un procesador distribuido en paralelo que tiene la capacidad natural de almacenar conocimientos experimentales y hacerlos disponibles para su uso [12].

El aprendizaje es un proceso por el cual los parámetros libres de la red neuronal se adaptan mediante la estimulación del entorno en el que está embebida la red.

El presente trabajo se centrará en el uso de redes neuronales entrenadas mediante aprendizaje supervisado. Éste asume un conjunto de datos de aprendizaje consistentes en  $N$  pares de datos de entrada-salida (objetivo):

$$T = (\mathbf{x}_i, \mathbf{d}_i)_{i=1}^N \quad (15)$$

donde  $\mathbf{x}_i$  es el vector de datos de entrada  $i$ -ésimo,  $\mathbf{d}_i$  es el vector deseado de salida (objetivo) al vector de entrada  $\mathbf{x}_i$ , y  $N$  es el tamaño de la muestra.

Por tanto, se desea, dada una muestra  $T$ , calcular los parámetros libres de la red neuronal para reducir una función error que mide la diferencia entre la salida  $\mathbf{y}_i$  a un vector de entrada  $\mathbf{x}_i$  y la salida deseada  $\mathbf{d}_i$  para todos los  $i$  de una manera estadística. En el presente trabajo se ha utilizado el error cuadrático medio (*ECM*) [13]:

$$ECM(n) = \frac{1}{N} \sum_{q=1}^N (d_q - y_q)^2 \quad (16)$$

En la figura 6 se muestra un ejemplo de una red neuronal prealimentada con  $p$  elementos de entrada, una capa de neuronas oculta consistente de 3 neuronas de tipo función logística, y una única salida.

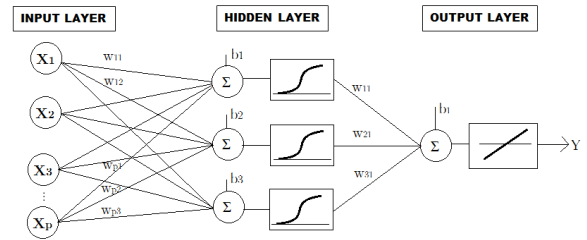


Figura 6: Ejemplo de red neuronal prealimentada con  $p$  entradas, una capa oculta de  $N$  (en este caso tres) neuronas y  $S$  (una única) salidas.

Se observa que el número de conexiones (sinapsis) entre la capa de entrada y la capa oculta es de  $Np$ , siendo  $N$  el número de neuronas ocultas. Por tanto, en esta primera sinapsis los coeficientes libres a optimizar serán  $N \cdot p$  coeficientes llamados pesos sinápticos  $w$ , así como  $N$  niveles de activación  $b$  (uno por cada neurona de la capa oculta).

Los coeficientes libres se organizan de forma matricial, y se obtiene (17), donde *sigmoide* es la función de transferencia sigmoidea:

$$\mathbf{a}^{Nx1} = \text{sigmoide}(\mathbf{W}_1^{Nxp} \cdot \mathbf{x}^{px1} + \mathbf{b}_1^{Nx1}) \quad (17)$$

En la segunda sinapsis entre la capa oculta y la capa de salida, los coeficientes libres son  $N \cdot s$  pesos sinápticos  $w$ , así como  $s$  constantes; sea  $s$  el número de salidas (en la figura 6 sólo una).

La salida se obtiene de la siguiente manera:

$$\mathbf{y}^{sx1} = \mathbf{W}_2^{sxN} \cdot \mathbf{a}^{Nx1} + \mathbf{b}_2^{sx1} \quad (18)$$

Por tanto, en una red neuronal prealimentada con una única capa oculta, el número de coeficientes libres a ajustar es  $N \cdot p + N + N \cdot s + s = N \cdot (p + s + 1) + s$ , es decir, el número de coeficientes a ajustar aumenta de forma lineal con el número de neuronas de la capa oculta en este

ejemplo, donde  $N$  es el número de neuronas de la capa oculta, y  $p$  y  $s$  son el números de elementos de cada par de datos de entrada y salida respectivamente.

### 3.3.1. Entradas. Matriz $C$

Una imagen de 64 píxeles como en el caso que nos ocupa, tiene 64 pesos de su transformada DCT (Discrete Cosine Transform). En este primer análisis utilizaremos solamente tres de estos coeficientes ( $C_{11}$ ,  $C_{12}$  y  $C_{21}$ ) que representaran la media y la información contenida en la frecuencia más baja en sentido horizontal y vertical.

Para reducir el número de entradas, se utilizarán como entradas  $C_{12}/C_{11}$  así como  $C_{21}/C_{11}$ , y la matriz de constantes elásticas resultante se multiplicará por el peso  $C_{11}$  para obtener la matriz buscada. Por tanto, la capa de entrada tendrá 2 neuronas.

### 3.3.2. Salidas. Matriz $D$

En cuanto a la salida, se buscarán 6 coeficientes de la matriz  $D$  ( $D_{11}$ ,  $D_{12}$ ,  $D_{13}$ ,  $D_{22}$ ,  $D_{23}$  y  $D_{33}$ ), al ser ésta simétrica. Por tanto, la capa de salida tendrá 6 neuronas.

### 3.3.3. Aprendizaje

Desde que el algoritmo de propagación hacia atrás (*Back-propagation algorithm*) fue obtenido por Henry J. Kelley en 1960, han aparecido numerosos métodos para acelerar la convergencia del algoritmo. Estos se pueden clasificar en técnicas *ad hoc* y métodos quasi-Newton.

Los métodos quasi-Newton se consideran más eficientes, si bien los requisitos computacionales aumentan con el cuadrado del tamaño de la red neuronal [12]. Sin embargo, en el presente trabajo, la red neuronal apenas tendrá unos cuantos coeficientes libres, por lo que este segundo criterio será el adecuado. El algoritmo de Marquardt-Levenberg usado por el *software* está descrito en [14].

La red neuronal utilizada en *Matlab* tendrá por defecto un aprendizaje de tipo Marquardt-Levenberg con ratios de entrenamiento, validación y test serán del 70 % - 15 % - 15 % de los pares de datos de entrada-salida [14].

En la figura 7 se muestra un esquema de la red neuronal generada, con una capa oculta compuesta de 4 nodos o neuronas cuya función de transferencia es en este caso la tangente hiperbólica.

Para la resolución de un problema de elementos finitos mediante la técnica propuesta, una vez entrenada la red neuronal se seguirá el siguiente esquema:

1. Se agrupan los píxeles en conjuntos de  $8 \times 8$ .
2. Se aplica la transformada DCT al conjunto de píxeles para obtener los 64 coeficientes  $C_{uv}$ .

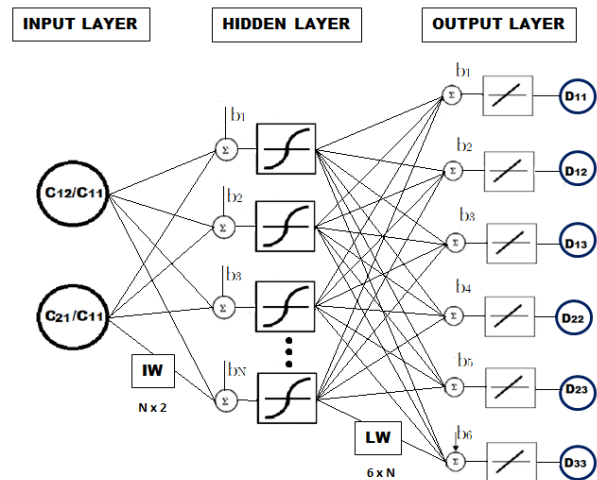


Figura 7: Esquema de la red neuronal generada.

3. Se evalúa la red neuronal con los ratios  $\{C_{12}/C_{11}$  y  $C_{21}/C_{11}\}$  para obtener la matriz  $D$  del nuevo elemento. Este proceso es mucho más rápido que resolver el problema de homogeneización.
4. Se obtiene con ella la matriz de rigidez  $K_e$  del elemento.
5. Ensamblado de las matrices de rigidez. Obtención de  $K_{Global}$
6. Obtención del vector de fuerzas nodales  $f$  adaptado a los nodos de los nuevos elementos.
7. Resolución del sistema de ecuaciones.

## 4. RESULTADOS

En este apartado se ha probado el uso de las redes neuronales para acelerar el proceso de homogeneización. En primer lugar, se ha realizado un análisis para escoger la configuración de red neuronal más apropiada para la homogeneización de propiedades elásticas de  $8 \times 8$  píxeles de una imagen de espumas de aluminio (sección 4.1). A continuación, se ha resuelto un problema a partir de una imagen real (sección 4.2).

### 4.1. Caracterización mecánica de espumas de aluminio usando 3 pesos de la transformada DCT

En esta sección se ha tratado de escoger la configuración de la red neuronal más ventajosa para la caracterización mecánica de espumas metálicas. Para esto diferentes configuraciones han sido analizadas, variando el número de capas ocultas así como el número de neuronas en cada capa.

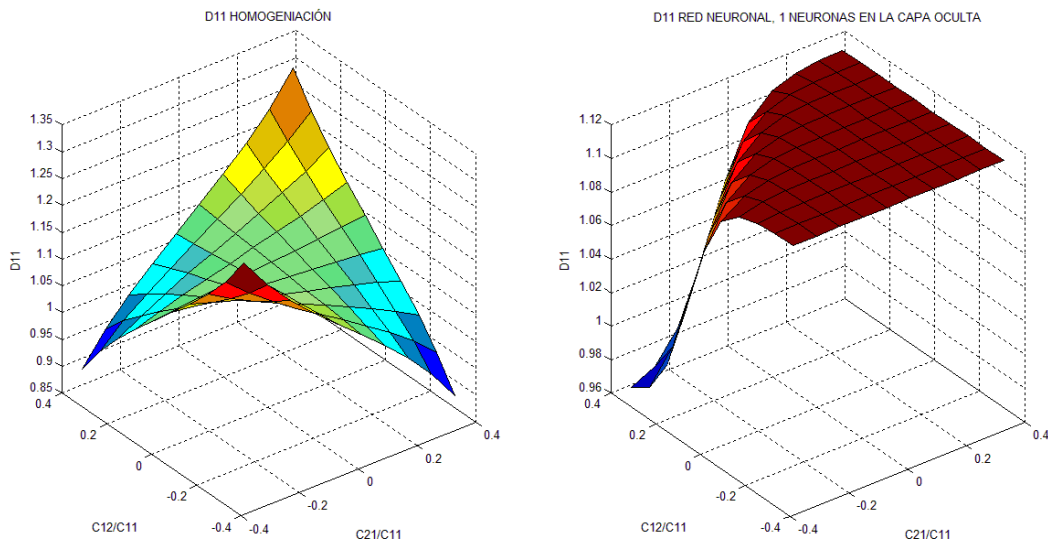


Figura 8: Coeficiente  $D_{11}$ . Real vs red neuronal con 1 neurona en la capa oculta.

#### 4.1.1. Entrenamiento A: Muestras de todo el espacio de entrada. Influencia del número de neuronas en la capa oculta y número de muestras sobre el ECM

En la Tabla 1 se muestra el error cuadrático medio (ECM) obtenido en todo el espacio  $\{C_{12}/C_{11}, C_{21}/C_{11}\}$  para varias redes neuronales con una capa oculta y nodos desde 1 hasta 8, así como dos capas ocultas y nodos desde 1/1 hasta 8/8. Como entradas a la red neuronal se proporcionan 100 conjuntos entrada-salida. Las mismas salidas (deseadas) han sido comparadas con la salida de la red neuronal para hallar el ECM. Cada caso se ha repetido 3 veces y se ha hallado la media del ECM.

Tabla 1. Influencia del Número de Capas Ocultas y Neuronas.

| 1 Capa Oculta |                       | 2 Capas Ocultas |                       |
|---------------|-----------------------|-----------------|-----------------------|
| Nodos         | ECM                   | Nodos           | ECM                   |
| 1             | $1.979 \cdot 10^{-3}$ | 1/1             | $2.099 \cdot 10^{-3}$ |
| 2             | $1.237 \cdot 10^{-3}$ | 2/2             | $1.136 \cdot 10^{-3}$ |
| 3             | $6.773 \cdot 10^{-4}$ | 3/3             | $3.115 \cdot 10^{-4}$ |
| 4             | $8.313 \cdot 10^{-6}$ | 4/4             | $1.500 \cdot 10^{-6}$ |
| 5             | $3.161 \cdot 10^{-6}$ | 5/5             | $1.218 \cdot 10^{-6}$ |
| 6             | $2.497 \cdot 10^{-6}$ | 6/6             | $1.425 \cdot 10^{-6}$ |
| 7             | $2.594 \cdot 10^{-6}$ | 7/7             | $2.029 \cdot 10^{-6}$ |
| 8             | $2.211 \cdot 10^{-6}$ | 8/8             | $4.312 \cdot 10^{-6}$ |

Como vemos en la Tabla 1, a partir de 4 neuronas, el rendimiento de la red neuronal es muy bueno, y el ECM es de un orden despreciable.

En segundo lugar, se observa las ventajas del uso de una

red neuronal de 1 sola capa oculta ya que con menos parámetros libres y un menor tiempo de cálculo se ha obtenido una precisión semejante a modelos con más capas ocultas.

Como se ve en las figuras 8, 9, 10, 11 y 12, para las salidas  $D_{11}$ ,  $D_{12}$ ,  $D_{22}$  y  $D_{33}$ , una capa oculta con una, dos o tres neuronas no modela bien las funciones deseadas de los coeficientes de salida. Sin embargo, a partir de cuatro neuronas en la capa oculta la red neuronal es capaz de modelar la curva de manera adecuada.

Como se ve en las figuras 13, 14 y 15, para las salidas  $D_{13}$  y  $D_{23}$ , el uso de redes neuronales con una capa oculta de una, dos o tres neuronas tampoco simula correctamente las funciones de los coeficientes de salida. Para cuatro neuronas, la red neuronal, como se ve en la figura 16, es capaz de proporcionar una función similar a la del coeficiente de salida, mientras que para un número mayor de neuronas, no se ha obtenido una mayor precisión (figura 17). Esto puede ser debido en parte a un exceso de coeficientes libres, además de al hecho de que el valor de los coeficientes  $D_{13}$  y  $D_{23}$  oscila en un rango de valores pequeño y muy cercano a cero (en el caso de materiales isotrópicos así como ortótropos,  $D_{13}$  y  $D_{23}$  es cero por definición, y en este caso no estamos considerando variaciones cruzadas de las propiedades del material). Por tanto, la red neuronal trata de ajustar un valor prácticamente nulo, y se produce un error de tipo numérico y sin una gran aportación al error cuadrático medio total.



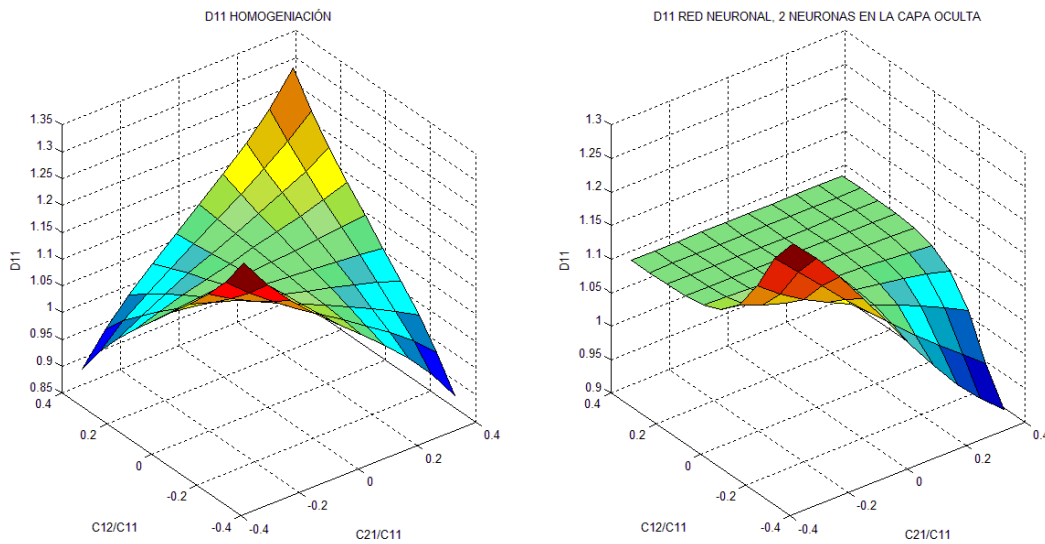


Figura 9: Coeficiente  $D_{11}$ . Real vs red neuronal con 2 neuronas en la capa oculta.

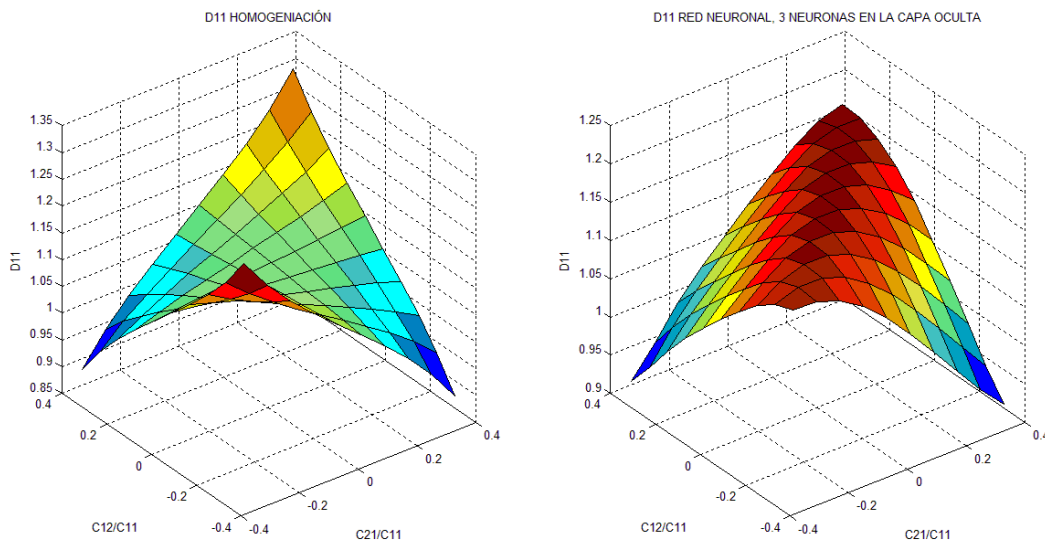


Figura 10: Coeficiente  $D_{11}$ . Real vs red neuronal con 3 neuronas en la capa oculta.

#### 4.1.2. Entrenamiento B: Muestras aleatorias. Influencia del número de neuronas en la capa oculta y número de muestras sobre el ECM

En esta sección, se han generado conjuntos de 8x8 píxeles aleatorios, para después obtener los pesos de la transformada DCT (entradas) y generar conjuntos de entrada-salida para el entrenamiento. El error cuadrático medio se ha evaluado en todo el espacio al igual que en el caso anterior. Cada caso se ha evaluado tres veces.

En la Tabla 2 se observa que para un número de neuronas en la capa oculta mayor, la mayor precisión se alcanza con un mayor número de muestras.

Como se observa en la Tabla 2, en el caso de 3 pesos de la DCT, el entrenamiento con muestras aleatorias ha resultado menos eficiente que el realizado con muestras de todo el espacio muestral. Las muestras generadas aleatorias tienen más contenido en altas frecuencias de variación del color que las imágenes reales. Estas frecuencias

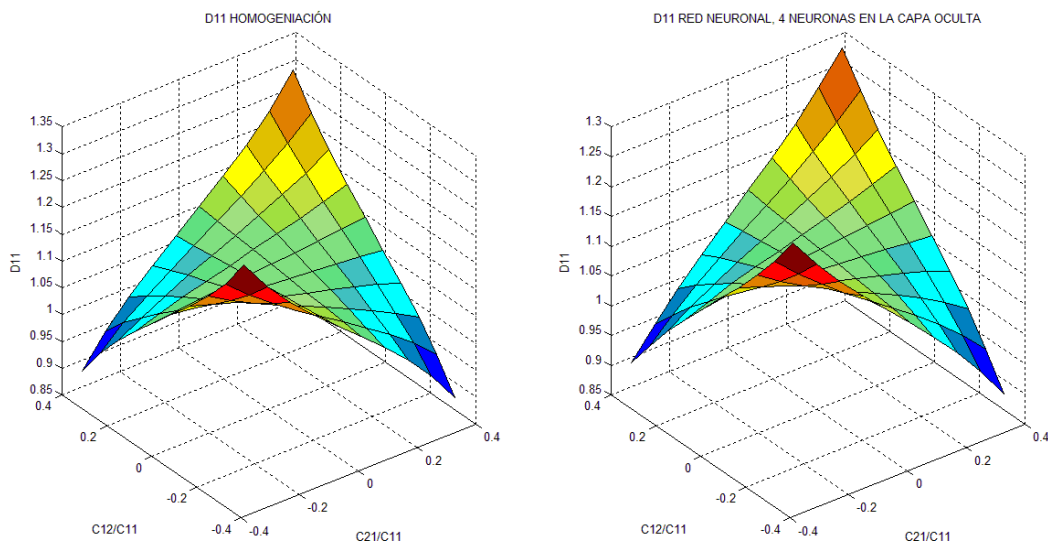


Figura 11: Coeficiente  $D_{11}$ . Real vs red neuronal con 4 neuronas en la capa oculta.

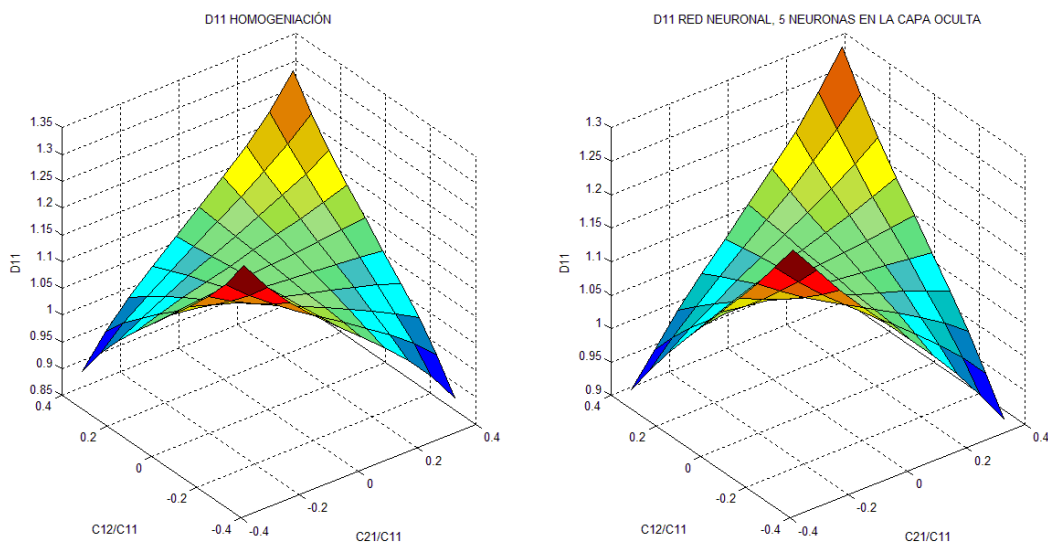


Figura 12: Coeficiente  $D_{11}$ . Real vs red neuronal con 5 neuronas en la capa oculta.

no se toman en cuenta en nuestro análisis. Sin embargo, como se ve en las figuras 18 y 19, las muestras aleatorias tienden a presentar unas variaciones de color del píxel en las bajas frecuencias más suaves ( $C_{12}/C_{11}$ ,  $C_{21}/C_{11}$ ), por lo que las muestras de entrenamiento no cubren todo el espacio muestral y no son apropiadas para entrenar la red neuronal.

Se ha observado que el ECM crece por tanto, en las regiones donde los pesos asociados a las bajas frecuencias

( $C_{12}/C_{11}$ ,  $C_{21}/C_{11}$ ) son más altos.

#### 4.2. Imagen real

En esta sección se ha planteado un problema de elementos finitos, modelando una espuma de aluminio a partir de una imagen 2D de una tomografía axial computarizada (TAC). Las soluciones de referencia y la obtenida con el uso de redes neuronales han sido comparadas en cuanto a coste computacional y precisión de los resultados.

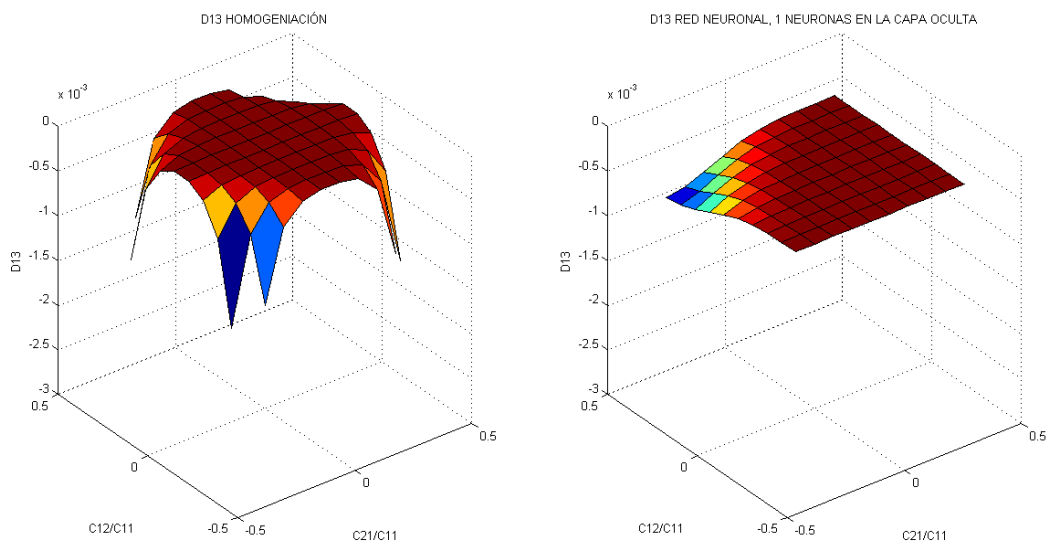


Figura 13: Coeficiente  $D_{13}$ . Real vs red neuronal con 1 neurona en la capa oculta.

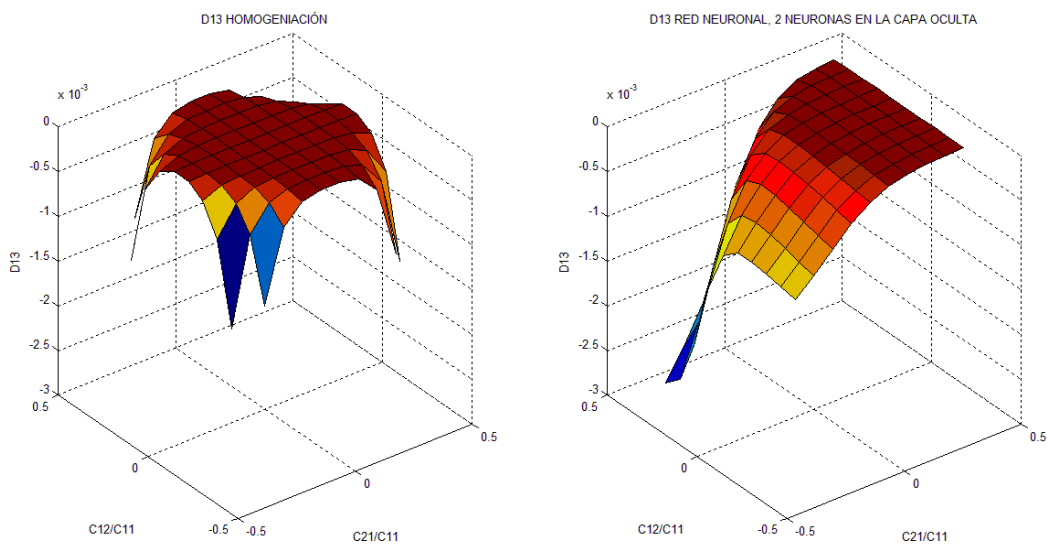


Figura 14: Coeficiente  $D_{13}$ . Real vs red neuronal con 2 neuronas en la capa oculta.

#### 4.2.1. Planteamiento del problema

En la figura 20 se muestra una imagen de  $512 \times 512$  píxeles obtenida con tomografía axial computarizada para obtener una solución de referencia. El modelo ha sido construido asociando a cada píxel un elemento finito cuyo módulo de Young varía linealmente con el color del píxel. El color más claro ha sido modelizado con  $E = 72 \text{ GPa}$ , mientras que el color más oscuro, asociado a los huecos de la espuma, ha sido modelizado con

$E = 1000 \text{ Pa}$ . El coeficiente de Poisson ha sido supuesto de 0.3 en todo el dominio de la geometría.

Se ha tomado como dimensiones de la espuma  $1 \times 1$  unidades de longitud ya que esto no influye sobre el análisis de datos. El problema a resolver consta en la aplicación de una fuerza distribuida de forma lineal sobre la cara superior de la espuma. Se han restringido los desplazamientos verticales en la cara inferior así como los horizontales en el nodo inferior izquierdo para evitar mo-

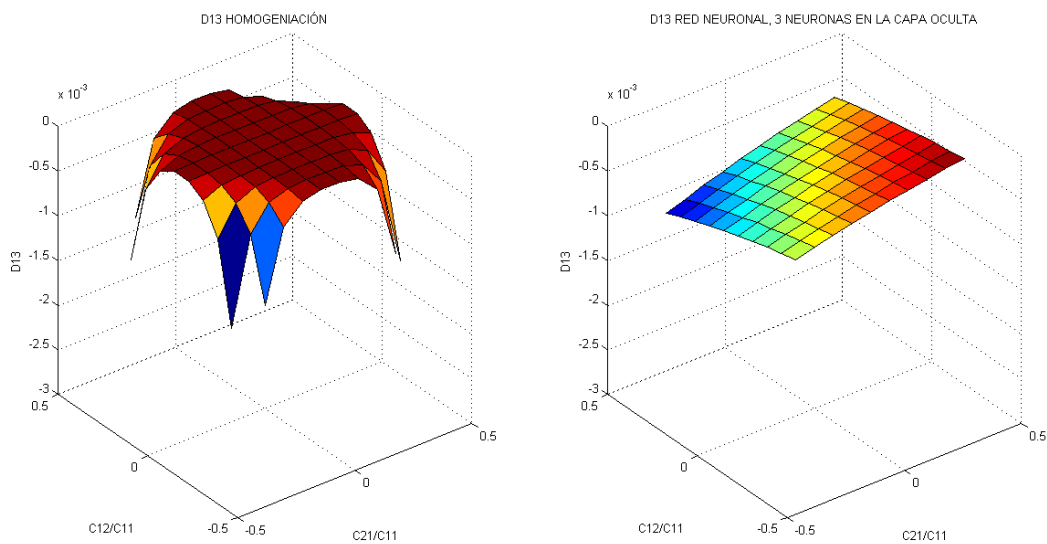


Figura 15: Coeficiente  $D_{13}$ . Real vs red neuronal con 3 neuronas en la capa oculta.

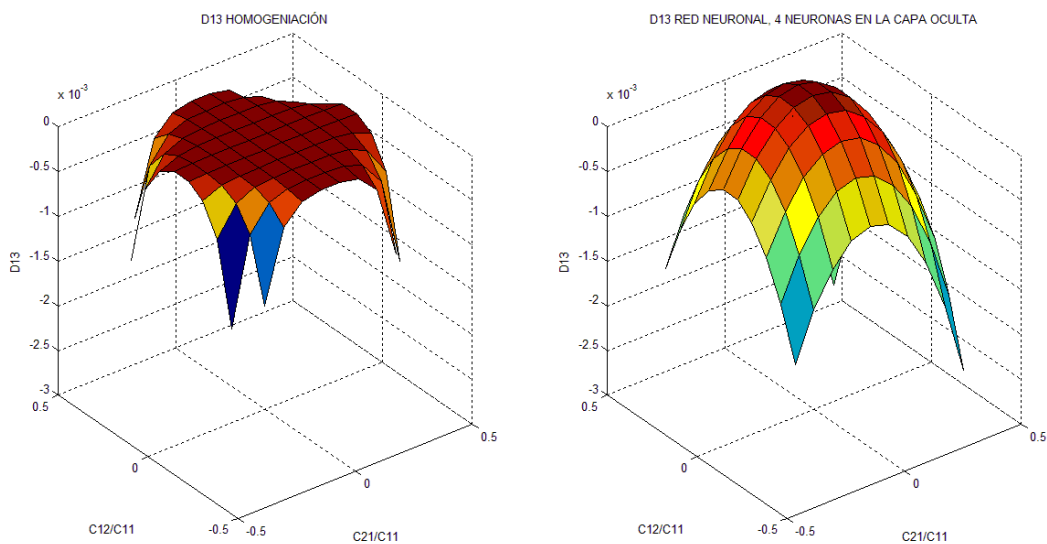


Figura 16: Coeficiente  $D_{13}$ . Real vs red neuronal con 4 neuronas en la capa oculta.

vimientos de sólido rígido, tal y como se muestra en la figura 21.

#### 4.2.2. Modelo 0. Solución de referencia

En las figuras 22, 23 y 24 se representa la distribución de deformaciones para la solución de referencia (asociando un elemento a cada píxel).

Esta solución, por tanto, ha sido la utilizada para validar el modelo con redes neuronales. En la Tabla 3 se ha calculado la energía de deformación de referencia.

#### 4.2.3. Modelo 1. Solución con 1 coeficiente JPEG

En general el tratamiento de imágenes píxel a píxel no es viable. Para el caso en 2D, se han tenido que resolver 262.144 elementos. Para analizar un TAC 3D de resolución 512 píxeles por dimensión se requerirían un total de 134.217.728 elementos lo que hace esto inviable para los PC's actuales de uso doméstico.

Una de las técnicas para aliviar esta dificultad es la de utilizar el promedio del valor del color en una malla cuyos

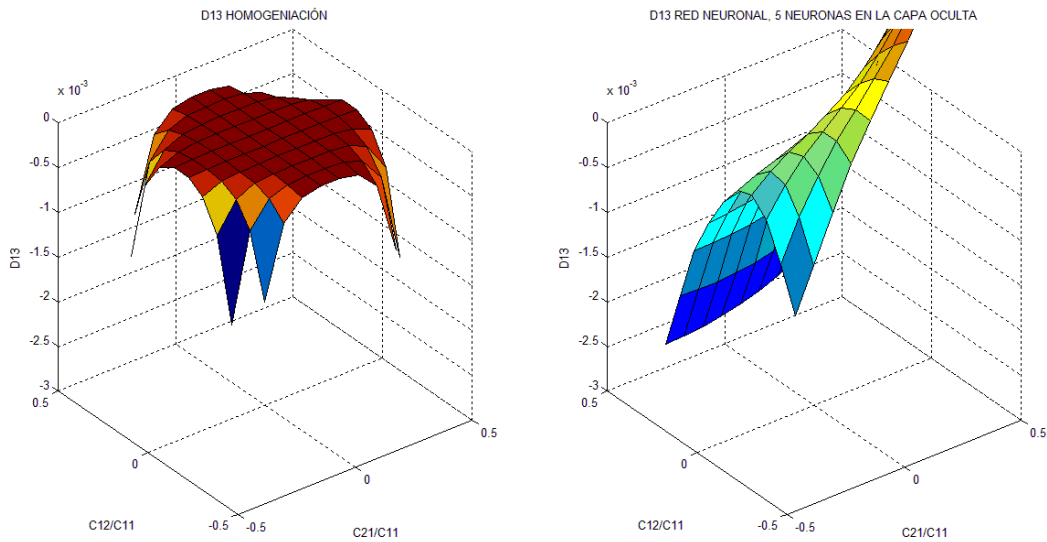


Figura 17: Coeficiente  $D_{13}$ . Real vs red neuronal con 5 neuronas en la capa oculta.

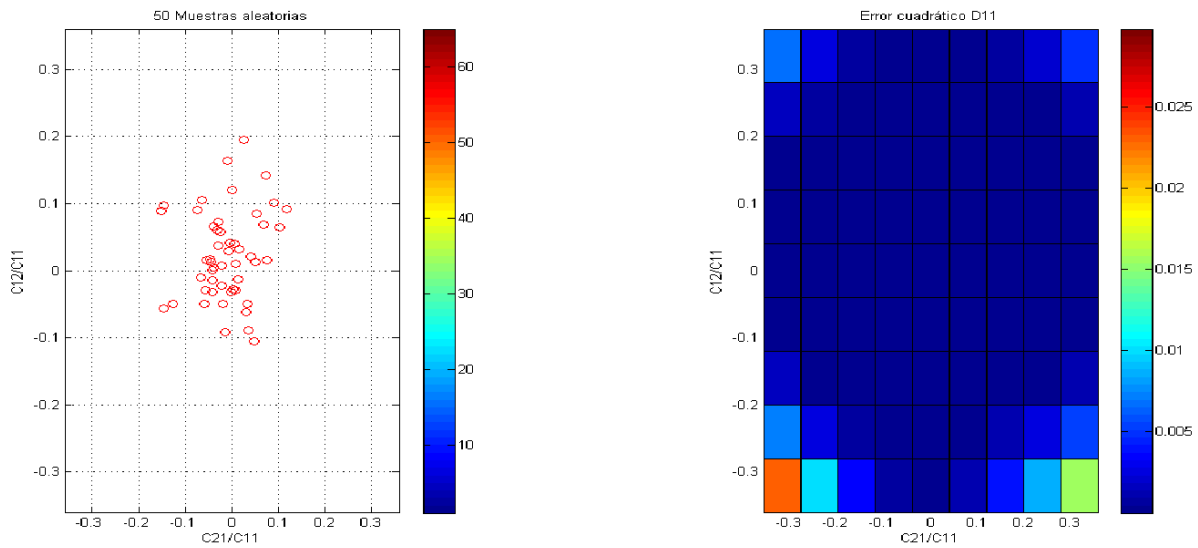


Figura 18: Coeficiente  $D_{11}$ . Distribución de ECM con 4 neuronas en la capa oculta tomando 50 muestras aleatorias.

elementos contengan gran cantidad de píxeles. A continuación se ha resuelto el problema agrupando los píxeles en elementos de  $8 \times 8$ . A cada nuevo elemento le ha sido asociado el módulo de elasticidad medio del conjunto de  $8 \times 8$  píxeles, es decir, el primer peso de la compresión JPEG, como se muestra en la imagen 25. En las figuras 26, 27 y 28 se representa la solución obtenida. Como se observa, la homogeneización de constantes elásticas y posterior discretización del problema en elemen-

tos cuadriláteros de primer orden de  $8 \times 8$  píxeles no permite obtener valores máximos de las deformaciones. Esto es debido, entre otros motivos, a que la aproximación se realiza en una malla más basta y con una reducción considerable del coste computacional (con nuestro portátil Asus K53SJ, y con nuestro código, el *speed up* fue de 240 veces). Esto provoca que la energía de deformación se subestime en todos los casos analizados, lo cual no nos pone en el lado de la seguridad.

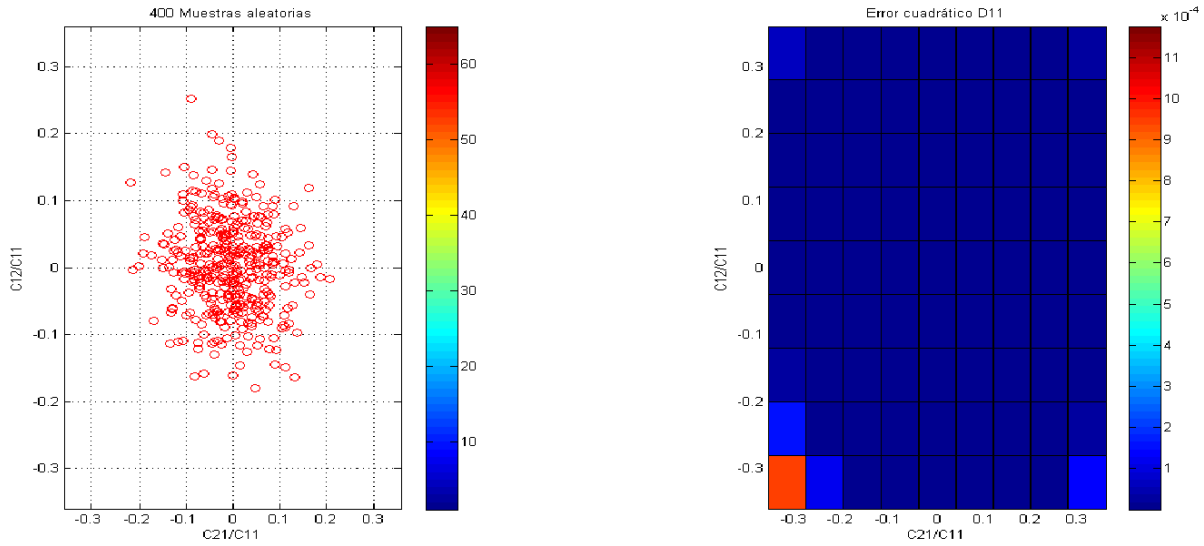


Figura 19: Coeficiente  $D_{11}$ . Distribución de ECM con 4 neuronas en la capa oculta tomando 400 muestras aleatorias.

Tabla 2. Influencia del Número de Muestras Aleatorias de Entrenamiento, así como del Número de Nodos de la Capa Oculta.

| Nodos capa oculta | Error cuadrático medio ( $\times 10^{-5}$ ) |       |       |       |
|-------------------|---|-------|-------|-------|
|                   | Número de muestras aleatorias               |       |       |       |
|                   | 50  | 100   | 200   | 400   |
| 4                 | 62.56                                       | 89.35 | 35.94 | 56.67 |
| 6                 | 116.6                                       | 37.85 | 11.09 | 25.31 |
| 8                 | 148.5                                       | 33.74 | 25.53 | 8.658 |
| 10                | 97.99                                       | 73.63 | 27.47 | 9.293 |

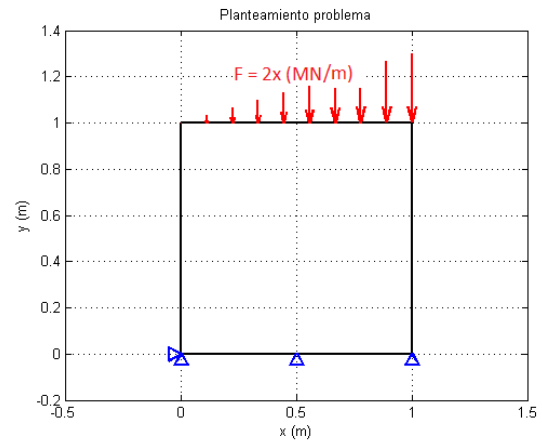


Figura 21: Planteamiento del problema.

Tabla 4. Solución con 1 Peso de la DCT

| Energía de deformación (MJ) |                   |
|-----------------------------|-------------------|
| 1 coef                      | 1.1920 (-12.82 %) |

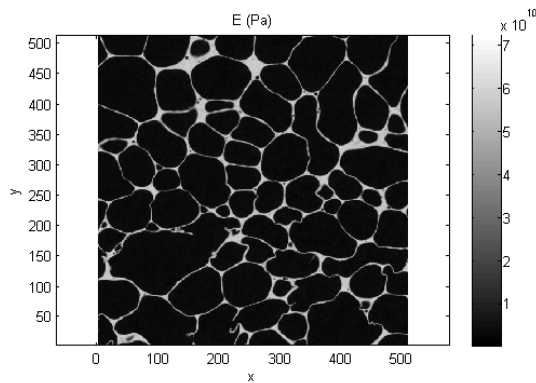


Figura 20: Imagen de la espuma metálica a analizar.

Tabla 3. Solución de Referencia

| Energía de deformación (MJ) |        |
|-----------------------------|--------|
| Solución de referencia      | 1.3673 |

#### 4.2.4. Modelo 2. Solución con 3 coeficientes JPEG

En este punto se ha planteado realizar un estudio de la técnica propuesta. En este caso, en vez de utilizar únicamente la media del color en cada elemento formado por 8x8 píxeles, se han considerado también las variaciones horizontales y verticales del color, a través de los coeficientes  $C_{12}$  y  $C_{21}$ . Con respecto a la anterior sección esto no lleva ningún coste computacional en la fase *online* ya que las matrices  $D$  están precalculadas gracias al entrenamiento en la fase *offline* de la red neuronal. Notar que

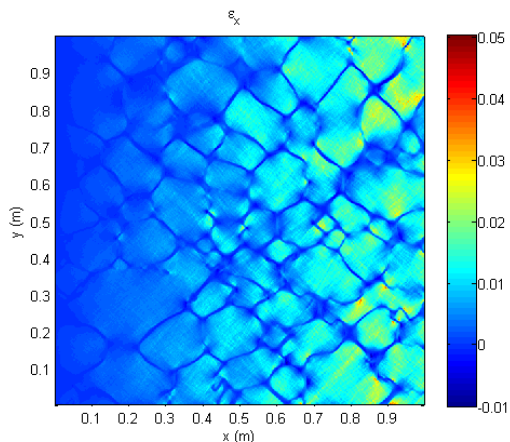


Figura 22: Solución de referencia. Deformación en dirección x ( $\epsilon_x$ ).

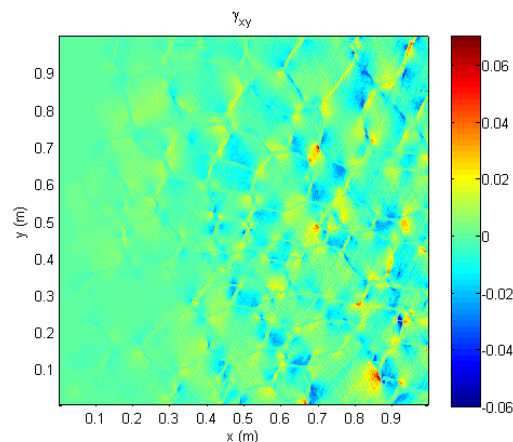


Figura 24: Solución de referencia. Deformación angular ( $\gamma_y$ ).

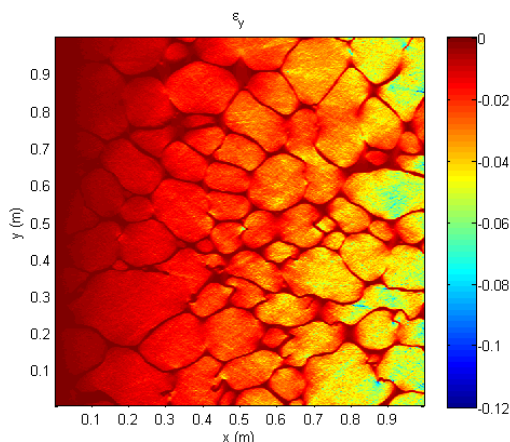


Figura 23: Solución de referencia. Deformación en dirección y ( $\epsilon_y$ ).

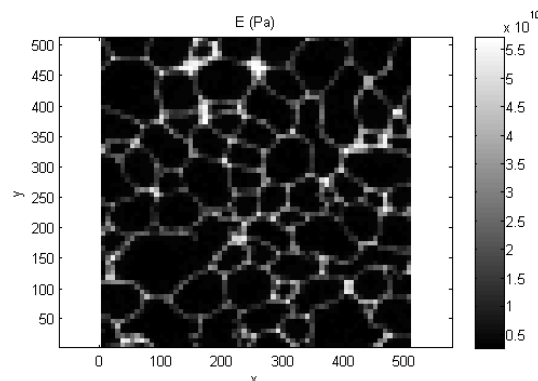


Figura 25: Imagen resultante tras la compresión JPEG con el primer peso.

este entrenamiento es independiente del problema y se ha realizado una única vez. El objetivo final es obtener una precisión mayor que el método anterior manteniendo el coste computacional gracias al uso de una red neuronal. Se ha usado para ello una red neuronal con una capa oculta de 4 neuronas, que ha sido entrenada con  $N$  muestras, que o bien han sido tomadas de la imagen (entrenamiento A), o generadas como un conjunto de  $8 \times 8$  píxeles aleatorios (entrenamiento B).

### Entrenamiento A

En este primer modelo de entrenamiento, se han elegido  $N$  conjuntos de  $8 \times 8$  píxeles de la imagen. Los resultados en cuanto a energía de deformación del problema se muestran en la Tabla 5. Cada caso ha sido evaluado tres veces y se ha hallado la media de la energía de deforma-

ción.

TABLA 5. ENERGÍA DE DEFORMACIÓN (MJ). INFLUENCIA DEL NÚMERO DE MUESTRAS.

| Energía de deformación (MJ) |                   |
|-----------------------------|-------------------|
| N=10                        | 1.2238 (-10.50 %) |
| N=50                        | 1.2110 (-11.43 %) |
| N=100                       | 1.2485 (-8.69 %)  |
| N=500                       | 1.3217 (-3.33 %)  |

Como vemos en la Tabla 5, tomando 3 coeficientes de la DCT, el número de muestras óptimo ha sido el mayor posible, ya que ello permite cubrir el mayor espacio muestral posible.

Al igual que en la sección anterior, todos los experimentos tienden a subestimar la energía de deformación. En las figuras 29, 30 y 31 se representa las distribuciones

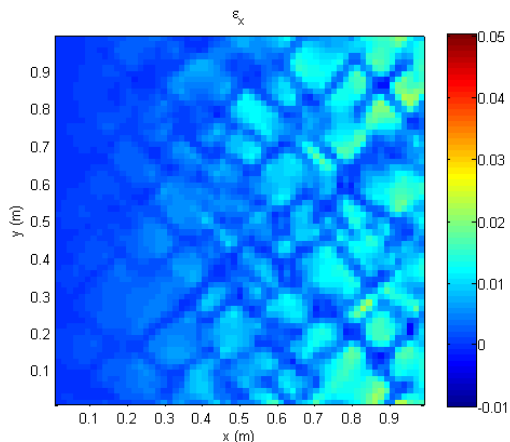


Figura 26: Solución con 1 peso de la DCT.

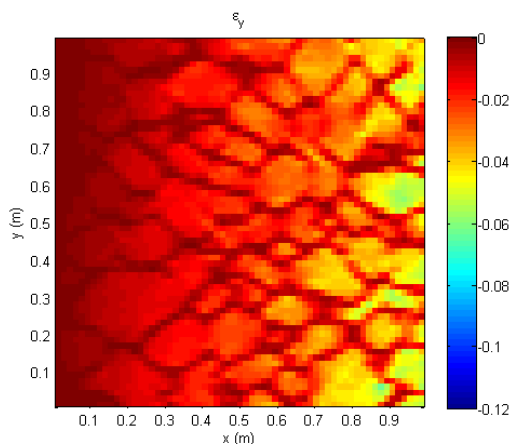


Figura 27: Solución con 1 peso de la DCT.

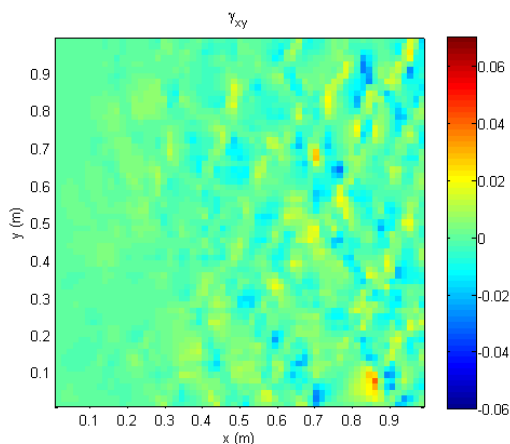


Figura 28: Solución con 1 peso de la DCT.

de deformaciones, para 3 coeficientes de la DCT y 500

muestras de entrenamiento.

Volvemos a observar que no se han obtenido los valores pico de la solución de referencia. No obstante vemos como la calidad de la solución mejora con respecto a la propuesta clásica (media del color, 1 peso JPEG) con un *speed up* de también 240 respecto a la solución de referencia.

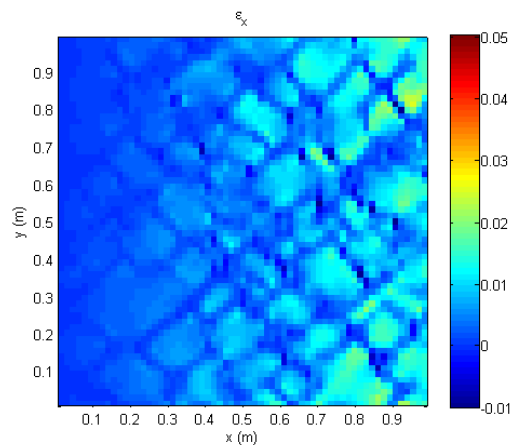


Figura 29: Solución con 3 pesos de la DCT y 500 muestras.

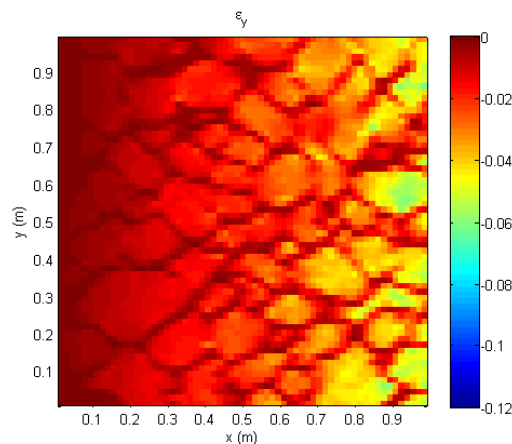


Figura 30: Solución con 3 pesos de la DCT y 500 muestras.

### Entrenamiento B

En este segundo modelo de entrenamiento, se han generado conjuntos aleatorios de 8x8 píxeles. Los resultados en cuanto a la energía de deformación del problema se muestran en la Tabla 6. Cada caso ha sido evaluado tres veces y se ha hallado la media de la energía de deformación.



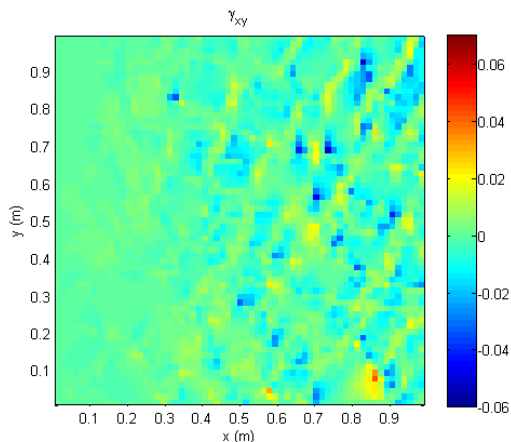


Figura 31: Solución con 3 pesos de la DCT y 500 muestras.

Tabla 6. Energía de Deformación (MJ). Influencia del Número de Muestras.

| Energía de deformación (MJ) |                   |
|-----------------------------|-------------------|
| N=10                        | 1.1908 (-12.91 %) |
| N=50                        | 1.1913 (-12.87 %) |
| N=100                       | 1.1929 (-12.76 %) |
| N=500                       | 1.1966 (-12.48 %) |

Como era previsible, este método para hallar los coeficientes ha resultado menos eficiente ya que no controlamos directamente el espacio de muestra de nuestras magnitudes de interés ( $C_{11}$ ,  $C_{12}$  y  $C_{21}$ ), por ello no se ha conseguido mejorar los resultados con respecto al método clásico (media del color).

#### 4.2.5. Modelo 3. Solución con 6 coeficientes JPEG

En esta sección se resolverá el problema tomando 6 pesos de la DCT.

#### Entrenamiento A

Los resultados en cuanto a la energía de deformación del problema se muestran en la Tabla 7.

Tabla 7. Energía de deformación (MJ). Influencia del número de muestras.

| Energía de deformación (MJ) |                  |
|-----------------------------|------------------|
| N=10                        | 1.2925 (-5.47 %) |
| N=50                        | 1.2341 (-9.74 %) |
| N=100                       | 1.2500 (-8.58 %) |
| N=500                       | 1.2711 (-7.04 %) |

En este caso, la inclusión de más parámetros no ha aportado una mejora en la calidad de la solución. Este hecho podría ser debido a la necesidad de un tamaño muestral

mayor. No obstante, si el elemento usado es lineal, la inclusión de variaciones de material de orden superior podría no tener influencia alguna en los resultados, o a lo sumo contribuir en una fuente de error numérico.

#### Entrenamiento B

Los resultados en cuanto a la energía de deformación del problema se muestran en la Tabla 8.

Tabla 8. ENERGÍA DE DEFORMACIÓN (MJ). INFLUENCIA DEL NÚMERO DE MUESTRAS.

| Energía de deformación (MJ) |                   |
|-----------------------------|-------------------|
| N=10                        | 1.1955 (-12.56 %) |
| N=50                        | 1.1978 (-12.40 %) |
| N=100                       | 1.2052 (-11.86 %) |
| N=500                       | 1.1971 (-12.45 %) |

Al igual que en el caso con 3 coeficientes JPEG, se han obtenido valores parecidos a los obtenidos con el método clásico. Esto es debido a que el entrenamiento con muestras aleatorias es menos eficiente y solo consigue capturar con precisión los valores medios.

## 5. CONCLUSIONES

Los resultados mostraron que el uso de redes neuronales puede agilizar el proceso de homogeneización de propiedades elásticas de la imagen en conjuntos de 8x8 píxeles. Además, se ha observado que las frecuencias bajas en las propiedades del material son aquellas que contienen una mayor información de interés para evaluar las propiedades elásticas del elemento.

En cuanto al uso de 6 coeficientes JPEG, no se ha observado una mejora respecto al método con 1 coeficiente JPEG, confirmando la hipótesis anterior.

En el caso 3D el uso de mallas a nivel de píxel es inviable y la mejora de las técnicas de homogeneización existentes es una necesidad evidente. Como trabajo futuro se plantea la implementación del método para una imagen en 3D.

En definitiva, en el presente trabajo se ha conseguido plantear un método de homogeneización de las propiedades elásticas a partir de imágenes que permitirá incrementar la eficiencia de la metodología cgFEM en el análisis de estructuras cuya geometría se obtiene a partir de imágenes.

## 6. AGRADECIMIENTOS

Los autores agradecen la ayuda recibida por el Ministerio de Economía y Competitividad a través del proyecto DPI2013-46317-R así como el apoyo recibido por la Generalitat Valenciana mediante el proyecto Prometeo/2016/007.

## 7. REFERENCIAS

- [1] C. Meola, S. Boccardi, and G. M. Carlomagno, "Composite materials in the aeronautical industry," *Elsevier Ltd*, 2015.
- [2] J. Banhart, "Aluminum foams for transport industry," *Materials and Design*, vol. 18, pp. 217–220, 1997.
- [3] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," *Computational Mechanics*, vol. 41, no. 1, pp. 121–133, sep 2007. [Online]. Available: <http://link.springer.com/10.1007/s00466-007-0173-y>
- [4] A. Düster, J. Parvizian, Z. Yang, and E. Rank, "The finite cell method for three-dimensional problems of solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 45-48, pp. 3768–3782, aug 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782508001163>
- [5] L. Giovannelli, J. J. Ródenas, J. M. Navarro, and M. Tur, "Direct medical image-based finite element modelling for patient-specific simulation of future implants," *Elsevier Ltd*, 2017.
- [6] E. Nadal, "Cartesian grid FEM (cgFEM): High performance h-adaptive FE analysis with efficient error control. Application to structural shape optimization," Ph.D. dissertation, Universitat Politècnica de València, 2014.
- [7] M. Tur, J. Albelda, O. Marco, and J. J. Ródenas, "Stabilized method of imposing Dirichlet boundary conditions using a recovered stress field," *Computer Methods in Applied Mechanics and Engineering*, vol. 296, pp. 352–375, aug 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782515002467>
- [8] M. Tur, J. Albelda, E. Nadal, and J. J. Ródenas, "Imposing Dirichlet boundary conditions in hierarchical cartesian meshes by means of stabilized Lagrange multipliers," *International Journal for Numerical Methods in Engineering*, vol. Accepted, 2013.
- [9] [Online]. Available: <https://es.mathworks.com/help/images/discrete-cosine-transform.html>
- [10] M. Lobo and M. E. Pérez, "Problemas de homogeneización en la ingeniería. Una experimentación numérica," *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, vol. 7, no. 4, pp. 455–472, 1991.
- [11] J. Schröder, "Constraint/boundary conditions on the micro-scale," *A numerical two-scale homogenization scheme: the FE<sup>2</sup>-method*, Udine, 2014.
- [12] M. Hagan and M. B. Menhal, "Training feedforward networks with the marquardt algorithm," *IEEE transactions on neural network*, vol. 5, no. 6, pp. 989–993, 1994.
- [13] I. W. Sandberg, "Feedforward neural networks: An introduction," *Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives*, Wiley and Sons, 2001.
- [14] H. Damuth and M. Beale, "Faster training," *Neural Network Toolbox User Guide, The MathWorks, Inc*, 2002.