



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Simulación computacional con elementos finitos y  
redes neuronales de problemas multifísicos.  
Aplicación al cocinado de carne.

Finite elements and neural network modelling in  
multiphysics problems. Application to meat cooking  
simulation.

Autor

Pedro Miguel Laborda

Director

Jorge Grasa Orús

ESCUELA DE INGENIERÍA Y ARQUITECTURA  
2020





DECLARACIÓN DE  
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Pedro Miguel Laborda

con nº de DNI 78754308 P en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado \_\_\_\_\_, (Título del Trabajo)

Simulación computacional con elementos finitos y redes neuronales de problemas multifísicos. Aplicación al cocinado de carne.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 26 de Junio de 2020

Fdo: Pedro Miguel Laborda



# AGRADECIMIENTOS

Como preámbulo a este Trabajo Fin de Grado (TFG) quiero agradecer su apoyo y colaboración a todas aquellas personas que lo han hecho posible.

En primer lugar, agradecer a Don Jorge Grasa Orús, Director de este Proyecto Final de Grado, su interés y esfuerzo constantes. Sus conocimientos y experiencia han constituido un elemento fundamental para el resultado obtenido, sin olvidar el buen trato recibido, la paciencia y la confianza en mí depositada. Sin duda alguna su labor y apoyo durante estos meses han sido esenciales para la consecución de este proyecto.

Agradecer también a todos los profesores y compañeros de clase que durante toda esta etapa de formación me han acompañado, su aportación a mi enriquecimiento personal e intelectual.

Por último, no puedo terminar estas líneas sin agradecer a mi familia su apoyo incondicional. Ellos han hecho posible que haya llegado hasta aquí al dedicar buena parte de su vida y recursos a mi formación y desarrollo personal.

A todos ellos, muchas gracias.



# **SIMULACIÓN COMPUTACIONAL CON ELEMENTOS FINITOS Y REDES NEURONALES DE PROBLEMAS MULTIFÍSICOS. APLICACIÓN AL COCINADO DE CARNE.**

## **RESUMEN**

Este Trabajo Fin de Grado se enmarca en un proyecto de colaboración entre la Universidad de Zaragoza y la empresa B/S/H España. Con el objetivo del desarrollo de una metodología computacional capaz de sentar las bases del cocinado autónomo.

En primer lugar, se describe la física que rige el cocinado de carne y se realiza una breve introducción del aprendizaje profundo y las redes neuronales. Una buena forma de comprobar, a priori, el potencial y validez de las redes neuronales, es resolviendo una serie de ejemplos de aplicación relacionados cada uno, con las físicas que intervienen en el cocinado de carne: Mecánica estructural, Transferencia de calor y Transferencia de masa. Para ello se han simulado dichos problemas en un software de elementos finitos, obteniendo una población de resultados utilizados para entrenar a las redes neuronales y que estas sean capaces de predecir estos resultados, en tiempo real y con exactitud.

En segundo lugar, se ha abordado la simulación del cocinado de carne utilizando los resultados de modelos 1D y 3D para entrenar redes neuronales que permitan obtener predicciones disminuyendo el coste computacional. En este caso se han considerado como variables de salida la distribución de la temperatura y de la pérdida de masa del alimento. Un aspecto que se ha tenido en cuenta para valorar la metodología propuesta, reside en las principales características de las redes neuronales como: variables de entrada, número de capas ocultas o tamaño de la población de entrenamiento. En este trabajo se ha realizado un estudio detallado de la influencia de estas características.

Por último, se han comparado las predicciones obtenidas mediante las redes neuronales con los resultados obtenidos en el software de elementos finitos, con el objetivo de comprobar la validez de las redes neuronales en la aplicación al cocinado de carne.



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Objetivos . . . . .	3
1.3. Contenidos . . . . .	4
<b>2. Fundamentos teóricos</b>	<b>5</b>
2.1. Física del cocinado de la carne . . . . .	5
2.1.1. Transferencia de calor . . . . .	6
2.1.2. Transferencia de masa . . . . .	9
2.1.3. Formulación de la mecánica del sólido deformable . . . . .	16
2.2. Aprendizaje profundo (“Deep Learning”) y el método de los elementos finitos . . . . .	21
2.2.1. Inteligencia artificial y aprendizaje profundo. Redes neuronales	22
2.2.2. Ejemplos de aplicación . . . . .	26
2.3. Simulación del cocinado de carne . . . . .	35
2.3.1. Modelo de elementos finitos en 1D . . . . .	35
2.3.2. Modelo de elementos finitos en 3D . . . . .	38
<b>3. Simulación del modelo y validación</b>	<b>45</b>
3.1. Modelo 1D de elementos finitos . . . . .	46
3.1.1. Generación población modelos . . . . .	46
3.1.2. Entrenamiento de las redes . . . . .	48
3.1.3. Resultados y validación . . . . .	49
3.2. Modelo 3D de elementos finitos . . . . .	51
3.2.1. Generación población modelos . . . . .	51
3.2.2. Entrenamiento de las redes . . . . .	52
3.2.3. Resultados y validación . . . . .	53
<b>4. Conclusiones y líneas futuras</b>	<b>57</b>
4.1. Conclusiones . . . . .	57

4.2. Líneas futuras . . . . .	59
<b>5. Bibliografía</b>	<b>61</b>
<b>Lista de Figuras</b>	<b>63</b>
<b>Lista de Tablas</b>	<b>69</b>
<b>Anexos</b>	<b>71</b>
<b>A. Ejemplos aplicación</b>	<b>73</b>
A.1. Mecánica estructural . . . . .	73
A.1.1. Redes neuronales con 100 muestras de entrenamiento . . . . .	77
A.1.2. Redes neuronales con 1000 muestras de entrenamiento . . . . .	78
A.1.3. Redes neuronales con 10000 muestras de entrenamiento . . . . .	79
A.2. Transferencia de calor . . . . .	80
A.2.1. Redes neuronales con 10 modelos de entrenamiento . . . . .	84
A.2.2. Redes neuronales con 100 modelos de entrenamiento . . . . .	85
A.2.3. Redes neuronales con 1000 modelos de entrenamiento . . . . .	86
A.3. Transferencia de masa . . . . .	88
A.3.1. Redes neuronales con 10 modelos de entrenamiento . . . . .	91
A.3.2. Redes neuronales con 100 modelos de entrenamiento . . . . .	92
A.3.3. Redes neuronales con 1000 modelos de entrenamiento . . . . .	94
<b>B. Resultados y validación de los modelos 1D y 3D</b>	<b>97</b>
B.1. Modelo 1D de elementos finitos . . . . .	97
B.1.1. Redes neuronales obtenidas con una población de 10 modelos . .	97
B.1.2. Redes neuronales obtenidas con una población de 100 modelos .	100
B.1.3. Redes neuronales obtenidas con una población de 1000 modelos	103
B.2. Modelo 3D de elementos finitos . . . . .	106
B.2.1. Redes neuronales obtenidas con una población de 10 modelos . .	106
B.2.2. Redes neuronales obtenidas con una población de 100 modelos .	112

# Capítulo 1

## Introducción

Este Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación en el que colaboran la Universidad de Zaragoza y la empresa B/S/H Electrodomésticos España.

La sociedad se dirige hacia hogares inteligentes conectados, en los que se podrá manejar todos los electrodomésticos desde la pantalla de dispositivos móviles. Como respuesta a esta realidad, las empresas y los grandes grupos tecnológicos han sabido identificarla, y tienen la intención de desarrollar sistemas que conecten estos dispositivos electrónicos con todo lo que nos rodea. Por eso el grupo B/S/H en colaboración con la Universidad de Zaragoza, lleva varios años trabajando para aplicarlo a las cocinas de inducción con el fin de buscar la máxima eficiencia de los equipos, así como un cocinado asistido como parte del hogar inteligente.

Sin embargo, la conectividad no es más que uno de los objetivos que se pretende alcanzar. También se está trabajando en desarrollar un cocinado guiado semi-autónomo, de alta calidad y prestaciones, y eficiente energéticamente, que permita al consumidor tomar decisiones correctas a la hora de seleccionar la técnica y los parámetros de cocinado más adecuados.

A día de hoy, el grupo B/S/H ya comercializa equipos de cocina en los que es posible seleccionar el alimento que se va a cocinar, para que se regulen algunos de los parámetros del cocinado. Sin embargo, para predecir, de alguna manera, las condiciones finales tras el cocinado, en base a las preferencias del usuario, es necesario desarrollar modelos numéricos capaces de modelar no sólo el proceso de cocinado, sino también aquellos fenómenos relacionados con la propia calidad del mismo.

## 1.1. Motivación

El desarrollo de modelos numéricos para la simulación del cocinado de alimentos, representa una gran ventaja frente a la experimentación tradicional. El análisis computacional de los procesos supone un ahorro de tiempo y dinero a la hora de recabar los datos necesarios para caracterizar el cocinado de los alimentos. Un buen ejemplo es la simulación del cocinado de carne, al tratarse de un proceso en el que se pueden tener en cuenta distintos parámetros del cocinado con tan solo ajustar unas pocas propiedades del modelo.

El siguiente paso, tras la simulación computacional, es implementarlo en cocinas inteligentes con interfaz de usuario que permitan predecir el cocinado de la carne, en función de varios parámetros de entrada. Por lo que, la principal ventaja que suponen estos modelos numéricos, es la de tener acceso a todas las variables que intervienen en el cocinado en cualquier punto de la pieza y en tiempo real. Esto permitiría llegar a crear un sistema realimentado, en el que la propia cocina fuese capaz de calcular la potencia a suministrar en cada momento. Y esto, tan solo, proporcionando a esta cocina inteligente como parámetros de entrada las características de la carne a cocinar.

A priorí, este método tiene mucho potencial, ya que incorporándolo a las cocinas de hoy en día, se lograría una gran independencia del usuario a la hora de cocinar. Es decir, el usuario solo tendría que indicarle a la cocina inteligente los parámetros de la carne a cocinar, y ésta es la que se encargaría de todo el proceso. Esto supondría un avance muy interesante en el cocinado autónomo, e incorporaría un cambio fundamental en la vida de los usuarios: ahorro de tiempo. Los usuarios podrían destinar el tiempo que dedican a cocinar un filete, a otra cosa, mientras su comida se está preparando de forma autónoma. Además este sistema, sería muy sencillo e intuitivo de usar, y más hoy en día, que estamos acostumbrados a manejar todo, o casi todo, desde una pantalla, ya sea de ordenador o de teléfono móvil.

A estos beneficios a nivel de comodidad de vida, hay que añadirles el beneficio energético que tendría esta cocina inteligente. Ella propia, se encargaría de suministrar la potencia necesaria, en cada momento, al filete de carne, por lo que la eficiencia energética del proceso se maximiza. Algo que cada vez es más importante, que los electrodomésticos tengan una mayor eficiencia energética.

## 1.2. Objetivos

El objetivo global de este proyecto es el de desarrollar una metodología computacional, que permita predecir el nivel del cocinado de un filete de carne, en tiempo real. Todo esto en base a unos parámetros de entrada, que no son más que las características del filete. Este nivel de cocinado, puede ser obtenerse conociendo la temperatura en puntos característicos del filete.

Esta metodología computacional se basa en el método de los elementos finitos y redes neuronales. De manera que se generarán una serie de modelos de entrenamiento validados mediante un software de elementos finitos. Estos modelos servirán de entrenamiento a cada red neuronal, con el objetivo de que las predicciones de las redes se parezcan lo máximo posible al resultado obtenido en el software. Consiguiendo, así, predecir el comportamiento del filete de carne, en tiempo real.

Para simular el proceso de cocinado, se ha de resolver un problema multifísico acoplado de transferencia de calor, transferencia de masa y de mecánica de sólidos. Por ello se ha optado por utilizar el software comercial COMSOL Multiphysics 5.2a.

Para entrenar y desarrollar las redes neuronales, se ha usado el software MATLAB R2020a.

Como subobjetivos de este TFG, se tienen:

- Evaluación de la capacidad de las redes neuronales en predecir resultados en problemas multifísicos. Para llevar a cabo esta validación, se han propuesto tres ejemplos de aplicación, cada uno asociado a una física de las tres que componen el modelo global.
- Adaptación de un modelo en 1D con el objetivo de generar una población de modelos que permitan entrenar una red neuronal.
- Adaptación de un modelo en 3D con el objetivo de generar una población de modelos que permitan entrenar una red neuronal.
- Análisis de los resultados numéricos obtenidos y validación del modelo final.

## 1.3. Contenidos

La memoria de este Trabajo Fin de Grado está estructurada en cuatro capítulos y dos anexos. El primer capítulo es la introducción, donde se aborda la motivación y los objetivos de este trabajo. En el segundo capítulo, por un lado se detallan los fundamentos teóricos del cocinado de carne, en donde se describen las físicas que intervienen, y se establecen los parámetros y variables, la geometría, la malla y las condiciones de contorno de los modelos de elementos finitos en 1D y 3D. Por otro lado, se aborda el tema de las redes neuronales, explicando en qué consiste la inteligencia artificial y el aprendizaje profundo o “Deep Learning”. El tercer capítulo se ha dedicado a la simulación del modelo, y a la descripción del proceso de entrenamiento de las redes neuronales, comenzando por la generación de las respectivas poblaciones de modelos y finalizando por su resultado y validación. Por último, en el cuarto capítulo se exponen las conclusiones, así como futuras líneas de investigación, tomando como punto de partida este proyecto. En el anexo A, se abordan, en detalle, los tres ejemplos de aplicación estudiados, comentando el modelo de elementos finitos usado, así como el proceso de entrenamiento de las redes neuronales. Y por último en el Anexo B, se detallan los resultados y la validación de las predicciones de las redes neuronales de los modelos 1D y 3D de elementos finitos.

# Capítulo 2

## Fundamentos teóricos

Este capítulo se divide en tres partes. En primer lugar, se van a describir las ecuaciones que permiten modelar las físicas que intervienen en el cocinado de la carne, que como ya se ha expuesto anteriormente, son la transferencia de calor, la transferencia de masa y la mecánica de sólidos deformables. En segundo lugar, se explicará en qué consiste el concepto de aprendizaje profundo (“Deep Learning”) y redes neuronales. También en este apartado, se han incluido tres ejemplos, uno de cada física que interviene en el cocinado de carne, para comprobar la precisión y el potencial que tienen las redes neuronales. Por último, se explican los modelos de elementos finitos usados para la simulación del cocinado de carne.

### 2.1. Física del cocinado de la carne

La carne está formada fundamentalmente por agua, en torno a un 70-75 % en peso, y proteínas, alrededor del 20 %, y en menor medida, por otros componentes (grasa, carbohidratos). El agua sin embargo, queda confinada dentro de la propia red proteica. Esta configuración, así como sus propiedades termofísicas, permiten tratar a la carne como un material cuya matriz sólida está saturada en contenido de agua (Bernad Pérez, 2019).

Durante el calentamiento, se va produciendo la desnaturalización de las proteínas. Este proceso comienza a unos 35°C, pero se intensifica en el rango de los 40-60°C. Este proceso libera el agua de la red y lleva consigo también la retracción de las fibras que forman la matriz, lo cual genera unos gradientes de presión que se traducen en el transporte de agua por el interior de la carne. Esta retracción de las fibras es la que se aprecia a escala macroscópica en toda la pieza de la carne, según avanza el cocinado. El agua libre que llega a la superficie se expelle de dos maneras: por evaporación, proceso

que absorbe gran cantidad de energía; o por medio de goteo, sobre todo en las paredes laterales en las que la temperatura no es tan alta. Las consecuencias de la pérdida desigual de agua en las distintas partes de la carne, se traduce en una mayor retracción en aquellas zonas donde los flujos son mayores.

El cocinado de la carne se puede modelar mediante un problema donde es necesario acoplar tres físicas: transferencia de calor, transferencia de masa y la mecánica del sólido. El modelo a implementar en este Trabajo Fin de Grado está basado en los modelos recogidos en la bibliografía realizados por Dhall and Datta (2011) y por Bernad Pérez (2019). El modelo se implementa en el software COMSOL Multiphysics 5.2a, basado en el cálculo por el método de los elementos finitos (FEM). En este software podemos implementar directamente las ecuaciones a resolver, así como las condiciones iniciales y de contorno, y las geometrías realistas de los alimentos a modelar, en este caso, del filete de ternera sobre la sartén.

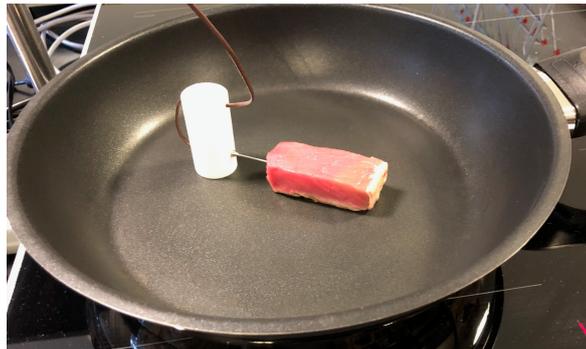


Figura 2.1: Cocinado de carne en sartén.

A continuación se procede a describir brevemente las ecuaciones implementadas para cada una de las físicas que rigen el cocinado de carne.

### 2.1.1. Transferencia de calor

Para el modelado de la transferencia de calor en el proceso de cocinado de carne se tendrán en cuenta las siguientes consideraciones:

- No hay generación interna de calor y no se produce evaporación en el interior del filete de carne, por lo que el agua en el interior de la carne aparece solo en estado líquido.
- La distribución inicial de temperatura y humedad es uniforme.
- Tanto la matriz sólida como el agua líquida en cada punto poseen la misma temperatura.

- La distribución de temperatura en la sartén es uniforme y constante en el tiempo, gracias al sistema de control del equipo de inducción.

Así, el proceso de transferencia de calor en el interior del producto, asumiendo equilibrio térmico, se puede modelar con un único balance energético para todo el producto:

$$(\rho_{eff}C_{p,eff})\frac{\partial T}{\partial t} + (\vec{n}_{w,G} \cdot \nabla(C_{p,w}T)) = \nabla \cdot (\lambda_{eff}\nabla T) \quad (2.1)$$

donde  $\rho_{eff}$  es la densidad del producto en  $[\text{kg}/\text{m}^3]$ ,  $C_{p,eff}$  es el calor específico del producto en  $[\text{J}/(\text{kg}\cdot\text{K})]$ ,  $\lambda_{eff}$  es la conductividad térmica del producto en  $[\text{W}/(\text{m}\cdot\text{K})]$ ,  $C_{p,w}$  es el calor específico del agua en  $[\text{J}/(\text{kg}\cdot\text{K})]$  y  $\vec{n}_{w,G}$  es el flujo másico de agua (observador estacionario) en  $[\text{kg}/(\text{m}^2\text{s})]$

El primer término de la ecuación de transferencia de calor, es el término de acumulación que representa el incremento de temperatura. El segundo término representa la transferencia de calor por convección debida al transporte de agua líquida en el interior del sólido. Y el tercer término corresponde al efecto de la conducción térmica desde las zonas más calientes hacia las más frías, en la que aparece el laplaciano de la temperatura.

### 2.1.1.1. Condiciones de contorno para la transferencia de calor

La ecuación general para la transferencia de calor en los contornos es la siguiente:

$$q_{surf} = h(T_{amb} - T_{surf}) - \lambda n_{w,s,surf} - \vec{n}_{w,G}c_{p,w}T \cdot \vec{N}_{surf} \quad (2.2)$$

donde  $h$  es el coeficiente de convección térmica en  $[\text{W}/(\text{m}^2\text{K})]$ ,  $T_{amb}$  es la temperatura del aire circundante a la carne en  $[\text{K}]$ ,  $T_{surf}$  es la temperatura de la carne en la superficie en la que se está evaluando la condición de contorno,  $\lambda$  es el calor latente de vaporización del agua en  $[\text{J}/\text{kg}]$  y  $c_{p,w}$  es el calor específico del agua en  $[\text{J}/(\text{kg}\cdot\text{K})]$ . El significado físico de cada uno de los términos es el siguiente: el primero corresponde a la convección natural externa, el segundo representa la energía consumida en el proceso de evaporación del agua, y el último el calor que transportan los flujos de agua salientes.

Sin embargo, ha de considerarse que no en todas las fronteras tienen lugar los mismos procesos. En la superficie inferior tiene lugar tanto el pase a fase vapor como el goteo de agua líquida; en las paredes laterales, se reduce al goteo; y en la cara superior no sucede ninguno de los dos fenómenos. Esto se refleja en los flujos de calor en esas superficies.

También hay que considerar la transferencia de calor por contacto con la sartén para la cara inferior de la carne. Como se ha explicado anteriormente, el sistema inductor de la cocina asegura una temperatura constante en la sartén. Esto se traduce en un flujo de calor para las dos superficies de contacto definido por la siguiente ecuación:

$$-\lambda_s \frac{\partial T}{\partial z} \Big|_{z \text{ sartén}=0} = -\lambda_c \frac{\partial T}{\partial z} \Big|_{z \text{ carne}=0} = H_c(T_s - T_{surf}) \quad (2.3)$$

donde  $T_s$ ,  $T_{surf}$  y  $\lambda_s$ ,  $\lambda_c$  son las temperaturas y conductividades térmicas de la sartén y de la cara inferior del trozo de carne en [K] y [W/(m·K)], respectivamente. El parámetro  $H_c$  representa la conductancia térmica del contacto entre ambas superficies en [W/(m<sup>2</sup>K)].

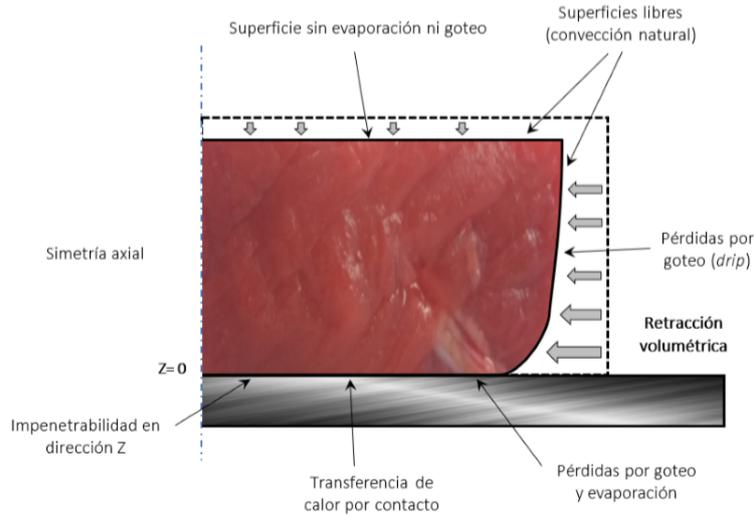


Figura 2.2: Condiciones contorno en el cocinado de carne.

### 2.1.1.2. Cálculo de las propiedades térmicas del producto

Las propiedades térmicas del producto son función de la temperatura y de la concentración de agua. Por ello su valor, cambia en cada punto de la pieza y su cálculo vendrá dado en función de la temperatura en ese punto y la fracción másica o volumétrica de agua, según la propiedad. Para el cálculo de las propiedades térmicas del producto se establecerá que tan solo está formado por proteínas y agua (Choi and Okos, 1986).

En primer lugar, se define la densidad del producto  $\rho_{eff}$ , aplicando la siguiente ecuación:

$$\rho_{eff}(T) = \varepsilon_w \rho_w(T) + \varepsilon_s \rho_s(T) \quad (2.4)$$

donde  $\rho_w$  y  $\rho_s$  son las densidades en  $[\text{kg}/\text{m}^3]$  del agua y de la parte sólida del producto, es decir, de la proteína.

La capacidad calorífica  $C_{p,eff}$  y la conductividad térmica  $\lambda_{eff}$  se calculan de la siguiente manera:

$$C_{p,eff}(T) = y_w C_{p,w}(T) + y_s C_{p,s}(T) \quad (2.5)$$

$$\lambda_{eff}(T) = g \lambda_{perpendicular}(T) + (1 - g) \lambda_{paralela}(T) \quad (2.6)$$

donde  $g$  es el parámetro de conductividad del calor del modelo, y  $\lambda_{perpendicular}(T)$ ,  $\lambda_{paralela}(T)$  se calculan como:

$$\lambda_{perpendicular} = \frac{1}{\frac{y_w}{\lambda_w(T)} + \frac{y_s}{\lambda_s(T)}} \quad (2.7)$$

$$\lambda_{paralela} = y_w \lambda_w(T) + y_s \lambda_s(T) \quad (2.8)$$

donde se representan las respectivas propiedades para cada componente (subíndice  $w$  para el agua y  $s$  para la proteína). Las fracciones másicas se calculan en base a sus fracciones volumétricas definidas en las siguientes ecuaciones:

$$y_w = \frac{\varepsilon_w \rho_w}{\rho_{eff}} \quad y_s = \frac{\varepsilon_s \rho_s}{\rho_{eff}} \quad (2.9)$$

## 2.1.2. Transferencia de masa

### 2.1.2.1. Definición de variables y ecuaciones de conservación de masa

El modelado del cocinado de un alimento considerando su deformación, ha sido habitualmente tratado en la bibliografía para distintos alimentos como un medio poroso multifásico. En la figura 2.3 se muestra la disposición teórica de las distintas fases presentes en el alimento. En el caso general, se considera el transporte de dos fases, agua líquida y gas, esta última formada por vapor de agua y aire, en un medio poroso no saturado.

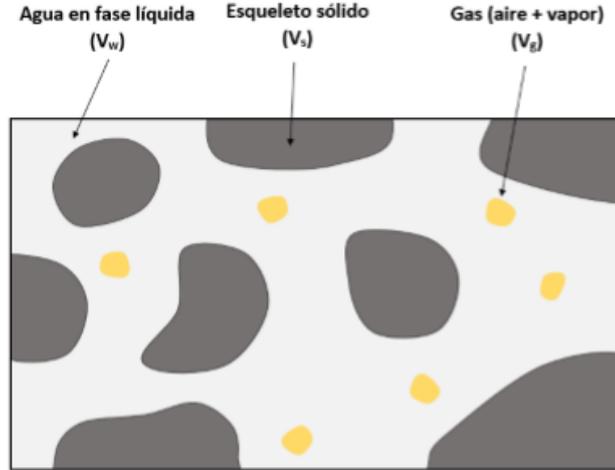


Figura 2.3: Esquema de la estructura del alimento en el que se muestran las distintas fases presentes y la nomenclatura para los respectivos volúmenes de cada una.

Una de las variables con la que se va a trabajar en el desarrollo de los fundamentos teóricos, es la porosidad del producto,  $\phi$ . Esta, representa la fracción volumétrica que ocupan la fase líquida y gaseosa respecto al volumen total  $V_T$ , el cual se define como la suma de los volúmenes de todos los componentes: del agua ( $V_w$ ), de la matriz sólida ( $V_s$ ) y del gas ( $V_g$ ). En lo sucesivo, se hará referencia al volumen total de carne y fluidos como producto.

$$V_T = V_w + V_s + V_g \quad (2.10)$$

$$\phi = \frac{V_w + V_g}{V_T} \quad (2.11)$$

Análogamente, se pueden definir las fracciones volumétricas de cada componente por separado:

$$\varepsilon_w = \frac{V_w}{V_T} \quad \varepsilon_g = \frac{V_g}{V_T} \quad \varepsilon_s = \frac{V_s}{V_T} \quad (2.12)$$

La fracción volumétrica de la fase sólida se puede expresar también en función de la porosidad:

$$\varepsilon_s = 1 - \phi \quad (2.13)$$

Estas variables serán posteriormente empleadas para calcular las propiedades del producto en cada instante de tiempo, en función del contenido de cada componente.

Otra variable del problema, que en este caso se obtiene actualizando a lo largo de la simulación, es la concentración de las especies,  $c_i$ , que expresa los kg que hay de la misma por  $\text{m}^3$  de producto. Resulta interesante definir también la variable  $M$  que representa el contenido de agua líquida en base seca (kg de agua/kg de carne):

$$M = \frac{c_w}{(1 - \phi)\rho_s} \quad (2.14)$$

donde  $\rho_s$  es la densidad del material del esqueleto sólido en  $[\text{kg}/\text{m}^3]$ .

Las ecuaciones de conservación de masa para cada componente, agua líquida, aire y vapor, para el problema en el que se considera el transporte de las dos fases en un medio poroso no saturado, se define mediante:

$$\frac{\partial c_w}{\partial t} + \nabla \cdot \vec{n}_{w,G} = -\dot{I} \quad (2.15)$$

$$\frac{\partial c_g}{\partial t} + \nabla \cdot \vec{n}_{g,G} = \dot{I} \quad (2.16)$$

$$\frac{\partial c_g \omega_v}{\partial t} + \nabla \cdot \vec{n}_{v,G} = \dot{I} \quad (2.17)$$

donde  $c_w$  y  $c_g$  son las concentraciones del agua líquida y de la fase gaseosa respectivamente en  $[\text{kg}/\text{m}^3]$  y  $\omega_v$  es la fracción másica de vapor. Los términos primeros de la izquierda representan la acumulación temporal de unas y otras especies en cada punto. La divergencia del flujo másico  $\vec{n}_{i,G}$  del segundo término expresa la cantidad de materia que llega al punto o volumen de control en cada instante de tiempo. Y el término de la derecha es la tasa de sustancia que se genera o desaparece para convertirse en otra fase.

Sin embargo, la carne se ha tratado como un material saturado desde el inicio, es decir, no existe fase gaseosa en su interior, y el mecanismo dominante es el del transporte de agua líquida (no se considera el transporte de otras especies como grasa o carbohidratos, ya que al trabajar con ternera, la proporción de estos es muy baja). Además dentro del grupo en el que se clasifica a la carne durante el cocinado, figura 2.4, lo que sufre es un proceso de deshidratación de la matriz sólida, en el que el frente de evaporación se limita a la superficie del material, por lo que se ha considerado en primera instancia, que no existe tampoco generación interna de vapor (Dhall and Datta, 2011). Esto supone que la fracción volumétrica de gas  $\varepsilon_g$ , y la concentración de gas  $c_g$ , en todo momento en el interior del producto sean nulas.

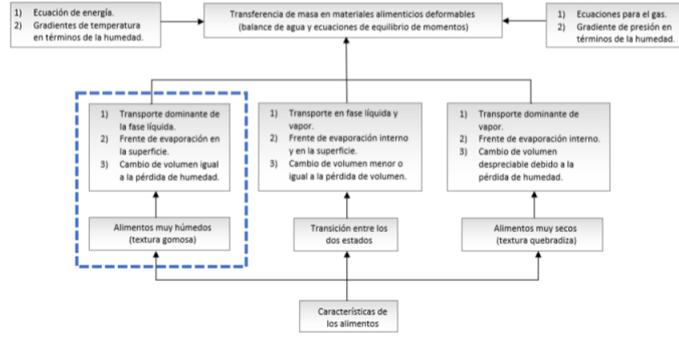


Figura 2.4: Esquema de las distintas tipologías de alimentos tras su cocinado y los fenómenos que lo rigen. Para el caso de la carne, la tipología corresponde a la remarcada en el cuadro azul.

Con todo ello, para este caso, las ecuaciones de conservación de masa en el interior del producto se reducen a la concerniente al agua líquida:

$$\frac{\partial c_w}{\partial t} + \nabla \cdot \vec{n}_{w,G} = 0 \quad (2.18)$$

Hay que tener en cuenta que dado que el problema atañe a un medio deformable, el sólido tiene una velocidad de deformación  $\vec{v}_{s,G}$ , función del gradiente de deformación, que es a su vez función de la variable de estado, concentración de agua. Por tanto, el flujo másico de agua respecto al observador estacionario se calcula con la siguiente suma:

$$\vec{n}_{w,G} = \vec{n}_{w,s} + c_w \vec{v}_{s,G} \quad (2.19)$$

donde el primer sumando corresponde al flujo de agua respecto al sólido, y el segundo al flujo debido al movimiento del sólido.

### 2.1.2.2. Ecuaciones de transferencia de masa

Las ecuaciones que rigen la transferencia de masa deben de modelar que con el incremento de la temperatura, se produce una caída de la capacidad de retención de agua debido a la desnaturalización de las proteínas. Es necesario mencionar, que en un proceso como el del cocinado en sartén, en el que la transferencia de calor es por contacto, los tiempos de cocinado son mucho menores que los del transporte de humedad. Esto se traduce en que la concentración en una parte del filete sea mayor que la capacidad de retención de ésta, en equilibrio.

Para cuantificar el transporte de humedad por difusión, se define la presión del agua líquida. Dicha presión, ha sido estimada empleando la teoría de Flory-Rehner, y en ella se tiene en cuenta la caída de la capacidad de retención de agua de la carne durante el calentamiento (der Sman, 2007). Tomando esta presión como nula para la fracción volumétrica de humedad en el equilibrio, y realizando una linealización de la expresión de Flory-Rehner en puntos próximos al equilibrio, se obtiene que la presión de agua líquida en la carne es proporcional a la diferencia entre la concentración de humedad en ese punto y la concentración de humedad en el equilibrio a esa temperatura,  $c_{w,eq}(T)$ :

$$p_w = C(c_w - c_{w,eq}(T)) \quad (2.20)$$

$C$  es una constante de proporcionalidad que, aunque tomada constante en este caso, puede ser dependiente de la temperatura:

$$C = \frac{D_w \mu_w}{\rho_w k_w} \quad (2.21)$$

donde  $D_w$  es la difusividad del agua en  $[\text{m}^2/\text{s}]$ ,  $\mu_w$  es la viscosidad dinámica del agua en  $[\text{Pa}\cdot\text{s}]$ ,  $\rho_w$  es la densidad del agua en  $[\text{kg}/\text{m}^3]$  y  $k_w$  es la permeabilidad del agua en  $[\text{m}^2]$ .

Sin embargo, por la tipología del problema, no es posible establecer una expresión analítica de la presión  $p_w$ , que se pueda introducir en la ley de Darcy para el estudio del flujo del agua en un medio poroso.

$$\vec{n}_{w,S} = -(\nabla p_w - \rho_w \vec{g}) \quad (2.22)$$

La ley de Darcy en la forma de la ecuación (2.22), expresa que los flujos de agua en un medio poroso son debidos al gradiente de presiones en el interior de la matriz sólida y la gravedad. Considerando despreciable el efecto de la gravedad, por no resultar relevante como mecanismo de transporte en este problema, e introduciendo la expresión obtenida para  $p_w$  en (2.20), se obtiene el flujo de agua líquida en el interior de la matriz sólida. Agrupando se obtiene:

$$\vec{n}_{w,S} = -(D_w \nabla c_w - D_{w,T} \nabla T) \quad (2.23)$$

En la ecuación (2.23), se pueden distinguir dos términos de difusión: el primero de la derecha tiene que ver con el gradiente de concentración de agua líquida en el material y el segundo, es debido al gradiente de temperatura. En consecuencia, se definen dos

difusividades, cada una asociada a uno de los componentes: la primera de ellas,  $D_w$ , asociada al contenido de humedad de la carne; y la segunda  $D_{w,T}$ , con la variación de la concentración de agua en el material en el equilibrio. Esta última se define como:

$$D_{w,T} = D_w \frac{\partial c_{w,eq}}{\partial T} \quad (2.24)$$

De esta manera, sustituyendo en la ecuación (2.18) para la conservación de masa del agua líquida, la ecuación (2.19) para el flujo de agua respecto al observador estacionario, donde el flujo de humedad se expresa mediante la ecuación (2.23), se obtiene:

$$\frac{\partial c_w}{\partial T} + \nabla \cdot (c_w \vec{v}_{s,G}) = \nabla \cdot (D_w \nabla c_w + D_{w,T} \nabla T) \quad (2.25)$$

### 2.1.2.3. Desnaturalización de las proteínas

Durante el calentamiento, las proteínas de la carne se desnaturalizan, lo que causa el transporte de humedad y cambios estructurales, como se ha expuesto anteriormente. De esta manera, se puede establecer la relación de estos fenómenos con la variable de la capacidad de retención de agua, Water Holding Capacity (WHC) o  $c_{w,eq}$ , que representa para una temperatura dada el contenido de agua que le correspondería en equilibrio termodinámico. En esta variable se recoge la forma en que evoluciona el contenido de agua conforme se produce la desnaturalización de las proteínas. Esta curva con forma de sigmoide, depende del tipo de carne analizada.

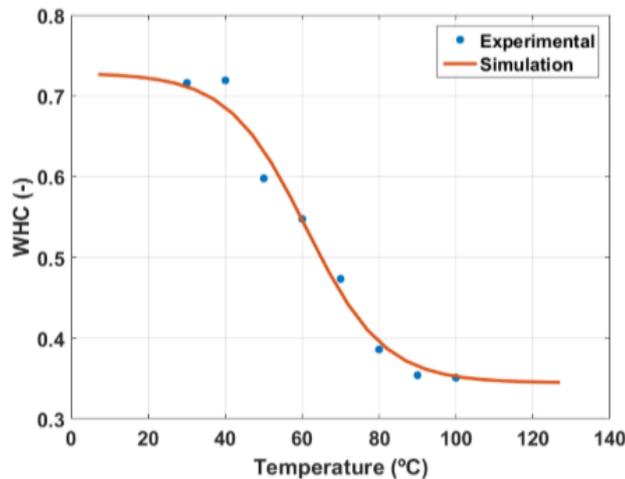


Figura 2.5: Curva de retención de agua ajustada para la ternera obtenida mediante experimentación.

$c_{w,eq}$  se ha obtenido de manera experimental en Bernad Pérez (2019). Su evolución viene dada por la siguiente ecuación:

$$c_{w,eq}(T) = 0,7284 - \frac{0,3838}{1 + 0,6166 \exp(-0,09927(T - 338,9))} \quad (2.26)$$

#### 2.1.2.4. Condiciones de contorno para la transferencia de humedad

En el instante inicial, el contenido en agua en la pieza de carne, como es lógico es el mismo que en el equilibrio, es decir, el que marca la curva WHC de equilibrio a la temperatura inicial. En lo que respecta a las condiciones de contorno, la pérdida de agua en la carne, tiene lugar sobre todo en la parte en contacto con la sartén, por ser la zona que adquiere mayor temperatura. También se produce en las paredes laterales de la carne, aunque en menor medida y disminuyendo progresivamente conforme aumenta la distancia a la sartén. En la cara superior, ya que la temperatura es baja, se considera contorno cerrado para la pérdida de humedad. En cuanto a las formas en que se pierde este agua, se pueden diferenciar dos distintas, como se ha mencionado anteriormente. Por un lado aparece el flujo de agua líquida, goteo, y por otro el flujo de vapor de agua.

##### Flujos por evaporación

Se considera que existe evaporación en la superficie de la carne en contacto con la sartén desde el primer instante, debido a las altas temperaturas. La magnitud del flujo de vapor en el proceso de evaporación viene dado de la siguiente manera:

$$n_{w,s,surfE} = h_m(\rho_{v,surf} - \rho_{v,amb}) \quad (2.27)$$

donde  $h_m$  es el coeficiente de transferencia de masa por convección en [m/s]. Los términos  $\rho_{v,surf}$  y  $\rho_{v,amb}$  son la densidad del vapor en la superficie y la densidad del vapor en el aire circundante respectivamente en [kg/m<sup>3</sup>]. En ambos casos, las densidades se calculan considerando el vapor como gas ideal. De esta manera, se toma la temperatura del sólido en la superficie del contorno evaluado en el primer caso, y la temperatura ambiente en el segundo, ambos a presión ambiente:

$$\rho_{v,surf} = M_w \frac{P_{amb}}{RT} \quad (2.28)$$

$$\rho_{v,amb} = M_w \frac{P_{amb}}{RT_{amb}} \quad (2.29)$$

donde  $M_w$  es la masa molar del agua en [kg/mol],  $P_{amb}$  es la presión ambiente en [Pa] y  $R$  es la constante de los gases en [m<sup>3</sup>Pa/(mol·K)].

### Flujos por goteo (drip)

El flujo por goteo sucede únicamente cuando la concentración de humedad en la superficie,  $c_{w,surf}$ , es mayor que la capacidad de retención de agua de la carne,  $c_{w,eq}$ . Este valor viene representado en la curva de retención de agua WHC. Así las pérdidas por goteo,  $n_{w,s,surf D}$ , bajo estas condiciones son iguales al flujo que llega a la superficie menos el flujo por la evaporación superficial,  $n_{w,s,surf E}$ :

$$n_{w,s,surf D} = \vec{n}_{w,s} \cdot \vec{N}_{surf} - h_m(\rho_{v,surf} - \rho_{v,amb}) \quad (2.30)$$

El término  $\vec{n}_{w,s}$  ya se ha desarrollado en la ecuación (2.23), y representa el flujo de agua en el interior de la carne. De este modo, el flujo total de humedad en la superficie, desde el punto de vista del observador estacionario, es igual a la suma del flujo por goteo, la pérdida por evaporación, y el flujo debido al movimiento de la superficie de por sí.

$$n_{w,G,surf} = n_{w,s,surf D} + n_{w,s,surf E} + c_w \vec{v}_{s,G} \cdot \vec{N}_{surf} \quad (2.31)$$

## 2.1.3. Formulación de la mecánica del sólido deformable

Para poder incorporar el efecto del cambio de volumen que se produce en la carne como consecuencia del cocinado, es necesario considerar el filete como un material deformable.

### 2.1.3.1. Análisis de grandes deformaciones

Durante el cocinado en la sartén, el filete de carne puede llegar a encoger en torno a un 30% respecto a su volumen inicial (Dhall and Datta, 2011). Esto requiere un planteamiento de grandes deformaciones para el análisis de la deformación del sólido.

La deformación viene dada por el tensor gradiente de deformación  $\mathbf{F}$ . En este caso, se puede aplicar una descomposición multiplicativa para diferenciar los cambios de volumen asociados al cambio de humedad y los debidos al comportamiento deformable (Vujosevic and Lubarda, 2002).

$$\mathbf{F} = \mathbf{F}_M \mathbf{F}_{el} \quad (2.32)$$

donde  $\mathbf{F}_M$  y  $\mathbf{F}_{el}$  son los tensores de gradiente de deformación debido a la humedad y a efectos elásticos, respectivamente.

Las tensiones debidas al cambio de volumen se relacionan con el jacobiano elástico  $J_{el}$ , que se obtiene como el ratio del cambio de volumen total y el cambio de volumen debido a la pérdida de humedad (figura 2.6)

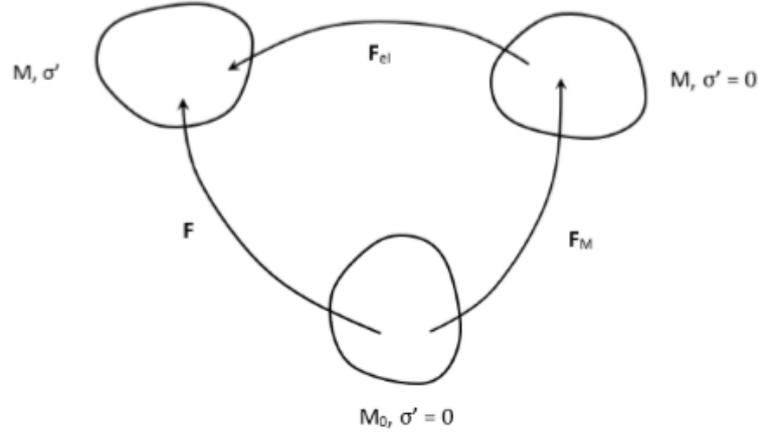


Figura 2.6: Descomposición multiplicativa del tensor gradiente de deformación, distinguiendo los efectos mecánicos y de la humedad.

### 2.1.3.2. Ecuaciones de equilibrio

Para grandes deformaciones es habitual emplear una formulación Lagrangiana para plantear las ecuaciones de equilibrio. En este tipo de coordenadas, la ecuación de equilibrio de momentos es:

$$\nabla_X \cdot (\mathbf{S}'' \mathbf{F}_{el}^T) = 0 \quad (2.33)$$

donde  $\mathbf{S}''$  es el segundo tensor de tensiones de Piola-Kirchhoff y  $\mathbf{F}_{el}$  es el tensor gradiente de deformación elástica. El término de la derecha es nulo, ya que la evaporación del contenido de agua se limita a la superficie de la carne y por tanto, no hay un aumento de presión interna por la generación de una fase gaseosa. Por ello, el término gradiente de la presión puede ser despreciado en este balance. Las tensiones del tensor  $\mathbf{S}''$ , se pueden relacionar con las tensiones del tensor de Cauchy,  $\sigma''$  de la siguiente manera:

$$\mathbf{S}'' = J\mathbf{F}_{el}^{-1}\sigma''\mathbf{F}_{el}^{-T} \quad (2.34)$$

La segunda ley de la termodinámica permite establecer la relación entre las tensiones de  $\mathbf{S}''$  y el tensor de Green-Lagrange para deformaciones elásticas,  $\mathbf{E}_{el}$ , a través de la función de energía de deformación,  $W_S$ :

$$\mathbf{S}'' = \frac{\partial W_S}{\partial \mathbf{E}_{el}} \quad (2.35)$$

### 2.1.3.3. Ley constitutiva

Es necesario establecer una ecuación constitutiva para la densidad de energía de deformación elástica,  $W_S$ , para caracterizar la deformación que sufre el material. Debido a la naturaleza gomosa con la que se trata a la carne como material, el esqueleto sólido se puede tratar como un material hiperelástico. Sin embargo, a pesar de que las fibras de la carne están orientadas de una manera concreta, y por ello, poseen distintas propiedades a lo largo y transversal a las mismas, en este primer trabajo se va a considerar que las propiedades mecánicas promedio del material son isotropas. El modelo constitutivo que se establece corresponde a un modelo Neo-Hookeano modificado:

$$W_{el} = \frac{K}{2}(J_{el} - 1)^2 + \frac{\mu}{2}(\bar{I}_1 - 3) \quad (2.36)$$

donde  $K$  es el módulo de compresibilidad del material en [Pa] y  $\mu$  es su módulo de elasticidad transversal en [Pa]. El primer sumando de la derecha corresponde a la función de densidad de energía de deformación asociada al cambio de volumen. El segundo término corresponde a la función de densidad de energía de deformación isocórica, o de cambio de forma. Como ya se ha definido,  $J_{el}$  es el jacobiano elástico, e  $\bar{I}_1$  es el primer invariante del tensor derecho de Cauchy Green modificado,  $\bar{\mathbf{C}} = (\bar{\mathbf{F}}_{el}^T \bar{\mathbf{F}}_{el})$ , para la parte desviadora del gradiente de deformación elástica,  $\bar{\mathbf{F}}_{el}$ .

La parte desviatoria del gradiente de deformación elástica se relaciona con el gradiente de deformación elástica,  $\mathbf{F}_{el}$ , y su parte volumétrica,  $J_{el}^{1/3}$  a través de la siguiente ecuación:

$$\mathbf{F}_{el} = J_{el}^{1/3} \bar{\mathbf{F}}_{el} \quad (2.37)$$

Sin embargo, el Jacobiano elástico  $J_{el}$  viene marcado por el cambio de volumen debido a la pérdida de humedad, que se recoge en el llamado  $J_M$  (Moisture Jacobian), determinante del tensor  $\mathbf{F}_M$ :

$$\mathbf{F}_M = J_M^{1/3} \mathbf{I} \quad (2.38)$$

Para estimar el valor de  $J_{el}$ , y ante la ausencia de condiciones externas de tensión, la carne tan solo se contrae en función de la fracción de humedad perdida. Considerando  $V(t)$  el volumen total del producto en cada instante, y  $\varepsilon_w$  la fracción volumétrica de agua, se establece el siguiente balance:

$$V(t) - V_0 = \varepsilon_w(t)V - \varepsilon_{w,0}V_0 \quad (2.39)$$

$$J_M(t) = \frac{V}{V_0} = \frac{1 - \varepsilon_{w,0}}{1 - \varepsilon_w} \quad (2.40)$$

Como ya se ha comentado en el apartado de transferencia de masa, la matriz sólida está saturada de agua en todo momento, así que la porosidad del esqueleto,  $\phi$ , coincide con la fracción volumétrica de agua,  $\varepsilon_w$ . Por ello y de manera análoga, se puede calcular el valor de la porosidad en cada instante de tiempo  $\phi(t)$ , considerando la incompresibilidad del esqueleto sólido, a partir de lo que este representa en cada instante respecto al volumen total:

$$(1 - \phi(t))V = (1 - \phi_0)V_0 \quad (2.41)$$

$$\phi(t) = 1 - \frac{1 - \phi_0}{V(t)/V_0} = 1 - \frac{1 - \phi_0}{J(t)} \quad (2.42)$$

De este modo, el Jacobiano debido a la pérdida de humedad,  $J_M$ , es una función de estado, dependiente del contenido de humedad, variable del problema. Por otro lado, la porosidad,  $\phi(t)$ , es una variable del proceso, dependiente del Jacobiano total del sólido,  $J(t)$ , representación de la variación volumétrica total.

Además a la hora de definir en el módulo de mecánica de sólidos la ley constitutiva para la función de energía, el programa calcula internamente los valores del jacobiano elástico,  $J_{el}$ , y del primer invariante del tensor derecho de Cauchy Green,  $\bar{I}_1$ . Sin embargo, éste de por sí, no tiene en cuenta la humedad como causa de la deformación, y por tanto, no la considera en el cálculo de estas dos variables. Por ello, ha sido necesario definir dos variables independientes a las del propio programa para que hagan su función considerando el cambio de volumen debido a la pérdida de humedad:

$$J_{el} = \frac{J_M}{J} \quad (2.43)$$

$$\bar{I}_1 = J_{el}^{-2/3} J_M^{-2/3} tr(\mathbf{F}^T \mathbf{F}) \quad (2.44)$$

donde  $J$  es el jacobiano total calculado por el programa y  $tr(\mathbf{F}^T \mathbf{F})$  es el primer invariante del tensor derecho de Cauchy Green calculado internamente por COMSOL.

#### 2.1.3.4. Condiciones de contorno para la mecánica del sólido deformable

En el aspecto mecánico de las condiciones de contorno existe un acoplamiento entre los dos cuerpos que componen el problema, sartén y filete de carne. En un estado inicial y de manera ideal durante el resto del cocinado, debería existir un contacto permanente entre la superficie inferior de la carne y la base de la sartén. Esto se traduce en una condición de contacto mecánico en la dirección axial.

$$u_z|_{z_{carne}=0} = u_z|_{z_{sartén}=0} = 0 \quad (2.45)$$

Sin embargo, durante el análisis visual en los experimentos realizados en Bernad Pérez (2019), se ha podido comprobar que este contacto no es perfecto y aparece un hueco o gap entre la sartén y la base del filete. La razón del mismo está en la contracción desigual de las fibras durante el cocinado, debido a la heterogénea composición de la carne, Esto afecta notablemente a la transferencia de calor por contacto. Por simplicidad, se ha modelado y simulado un contacto perfecto durante todo el proceso de cocinado.

### 2.1.3.5. Cálculo del módulo elástico

El módulo elástico de la carne, se ha obtenido a partir de valores experimentales de  $G$  a distintas temperaturas obtenidos en Bernad Pérez (2019) a través de la siguiente ecuación:

$$E(T) = \frac{G(T)}{2(1 + \nu)} \quad (2.46)$$

De acuerdo a la bibliografía (Aberham Hailu Feyissa and Adler-Nissen, 2013), en el rango de los 50-80 °C, el módulo elástico de la carne se puede ajustar a una sigmoide. Por regresión lineal de los datos experimentales la expresión ajustada de  $E$  en [kPa] es:

$$E(T) = 36,4 + \frac{85,32}{1 + \exp(-0,3386(T - 68,04))} \quad (2.47)$$

La curva ajustada a los valores experimentales se muestra en la siguiente figura.

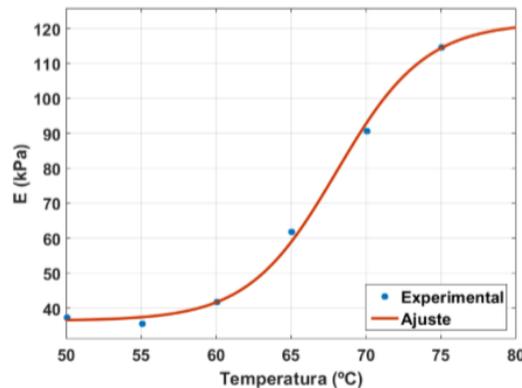


Figura 2.7: Ajuste del módulo elástico en función de la temperatura  $E(T)$  para la ternera.

## 2.2. Aprendizaje profundo (“Deep Learning”) y el método de los elementos finitos

En esta sección se van a explicar los conceptos de inteligencia artificial y aprendizaje profundo. Así como su relación con las redes neuronales, que es lo que se va a aplicar, al fin y al cabo, en el cocinado de carne. Además se van a resolver una serie de ejemplos de aplicación a través de redes neuronales para comprobar la validez y precisión de su uso en la resolución de problemas de este estilo.

### 2.2.1. Inteligencia artificial y aprendizaje profundo. Redes neuronales

En primer lugar, se va a comenzar definiendo el concepto de aprendizaje profundo. El aprendizaje profundo (“Deep Learning”), DL, es un subcampo del aprendizaje automático (“Machine Learning”), ML, en la inteligencia artificial (“Artificial Intelligence”), AI, que lidia con algoritmos inspirados en la estructura biológica y el funcionamiento del cerebro humano (Moolayil, 2019). En esta definición, se pueden observar tres términos importantes: DL, ML y AI.

La inteligencia artificial, AI, puede ser definida como el atributo de inteligencia que ha sido introducido en una máquina. Las máquinas no son inteligentes, así que para hacerlas más listas, se les induce un poco de inteligencia de manera que puedan tomar decisiones de forma independiente. Esta inteligencia es provocada de forma artificial, por consiguiente se tiene el nombre de inteligencia artificial. Otro punto a remarcar, es que esta inteligencia es, explícitamente, programada, es decir, una gran lista de reglas condicionales (if-else). Pero lo que se necesita, es introducir esa inteligencia en la máquina sin realizar una programación, algo con lo que la máquina pueda aprender por ella misma. Esta es la base del aprendizaje automático.

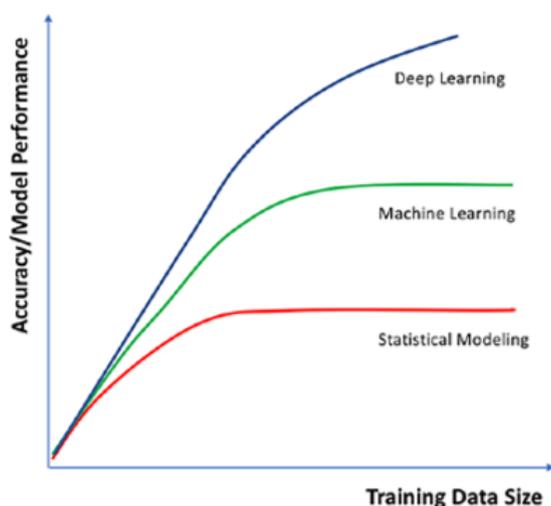


Figura 2.8: Comportamiento del DL, ML y SM en función del tamaño de los datos de entrenamiento (Moolayil, 2019).

El aprendizaje automático, ML, puede ser definido como el proceso de provocar inteligencia en un sistema o máquina sin programación explícita. Un ejemplo podría ser un sistema capaz de predecir si un estudiante va a aprobar o suspender un examen, aprendiendo de sus exámenes anteriores y sus propios atributos. Aquí, el sistema no esta

programado con una serie de reglas que deciden si el estudiante aprueba o suspende, sino que aprende por sí solo basándose en los patrones previos. Entonces, ¿Dónde se sitúa el aprendizaje profundo, DL, en este contexto? Mientras que, el ML trabaja bien en una variedad de problemas, falla en distinguir algunos casos específicos, aparentemente, sencillos para los seres humanos, por ejemplo: clasificar una imagen de un gato o de un perro, o distinguir un clip de voz masculino o femenino. El aprendizaje automático actúa de manera pobre con una imagen y otro tipo de datos desestructurados. La búsqueda de las razones de este pobre rendimiento, nos conduce a la idea de la imitación de los procesos biológicos del cerebro humano, compuesto de billones de neuronas conectadas y coordinadas para adaptar el aprendizaje de cosas nuevas.

En un camino paralelo, las redes neuronales han sido un tema de estudio durante varios años, pero solo se ha conseguido un pequeño progreso debido a las limitaciones computacionales y de datos. Tras alcanzar la cúspide del ML y las redes neuronales, se ha llegado al campo del aprendizaje profundo, el cual ha sido encuadrado mediante el desarrollo de las redes neuronales profundas (“Deep Neural Networks”). Las DNNs son redes neuronales con muchas más capas. Por lo que, se ha llegado a la conclusión de que el DL sobresale en nuevas fronteras, donde el ML se queda atrás. En su momento, investigaciones adicionales permitieron llegar a entender donde podía tomar ventaja el DL frente a las tareas del ML y esperar un mejor comportamiento, siempre y cuando hubiera un excedente de datos disponible. Por lo tanto, el aprendizaje profundo se convirtió en un extendido campo para resolver problemas predictivos, más que para estar limitado a clasificar una imagen o distinguir un clip de audio.

Hoy en día, se tiene la certeza de que con el uso del aprendizaje profundo para la mayoría de casos que antes se resolvían con el aprendizaje automático, se van a conseguir mejores resultados. Lo mismo sucedía, hace unos años, con el debate entre el aprendizaje automático y los modelos estadísticos, (“Statistical Models”), SM. En la figura 2.8, se representa el comportamiento en función del tamaño de los datos de entrenamiento, de los conceptos descritos: “Deep Learning“, “Machine Learning“ y “Statistical Modeling“.

Por lo que ahora, si se repasa la definición, explicada al comienzo de la sección, se pueden describir los conceptos de AI, ML y DL, usando un diagrama de Venn como el de la figura 2.9

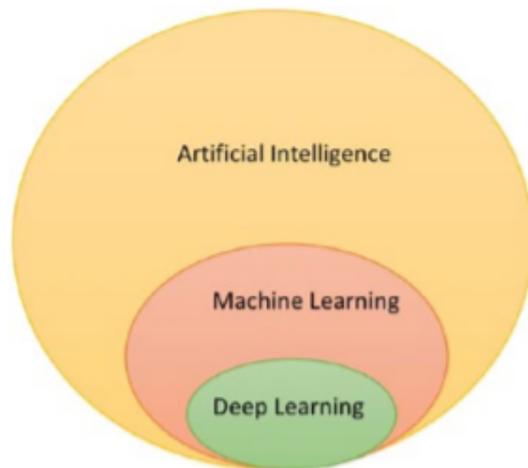


Figura 2.9: Diagrama de Venn de AI, ML y DL (Moolayil, 2019).

A modo de conclusión, se puede indicar que la AI es el campo que provoca inteligencia en una máquina o sistema artificialmente, con o sin programación explícita. El ML es un subcampo de la inteligencia artificial, en el que la inteligencia es provocada sin ninguna programación. Por último, el DL es un campo dentro del ML, en el que la inteligencia en los sistemas es provocada usando algoritmos inspirados en el funcionamiento biológico del cerebro humano (Moolayil, 2019).

#### **2.2.1.1. Redes neuronales**

En su forma más básica, los modelos de aprendizaje profundo están diseñados usando la arquitectura de las redes neuronales. Una red neuronal es una organización jerárquica de neuronas (similar a las neuronas del cerebro) con conexiones con otras neuronas. Estas transmiten un mensaje o señal a otras neuronas, basándose en la señal de entrada que reciben y construyen una red compleja, que aprende con un mecanismo de retroalimentación, todo esto mencionado en Moolayil (2019).

Como se puede observar en la figura 2.10, los datos de entrada son consumidos por las neuronas en la primera capa oculta (“hidden layer”), que proporciona una salida para la capa siguiente hasta, finalmente, llegar a la última. Cada capa puede tener una o varias neuronas, y cada una de ellas calculará una pequeña función. La conexión entre dos neuronas de capas consecutivas tendrá asociada un peso. El peso define la influencia que cada una tendrá sobre la siguiente, y finalmente sobre la última salida. En una red neuronal, los pesos iniciales son obtenidos de manera aleatoria durante el entrenamiento del modelo, pero esos pesos son actualizados de forma iterativa,

para aprender a predecir la salida correcta. Descomponiendo la red, se pueden definir los bloques: neurona, capa, peso, entrada, salida, función de activación dentro de la neurona, etc.

En la red neuronal, la información de entrada (input) es consumida por las neuronas de la primera capa, y se calcula una función de activación dentro de cada una. Basada en una regla simple, la neurona envía una salida a la siguiente. Cuanto mayor sea la salida de una neurona, mayor será la importancia de esa dimensión de entrada. Estas dimensiones se combinan en la siguiente capa para formar nuevas dimensiones adicionales. El sistema lo aprende intuitivamente. El proceso, cuando se reproduce varias veces, desarrolla una compleja red con conexiones.

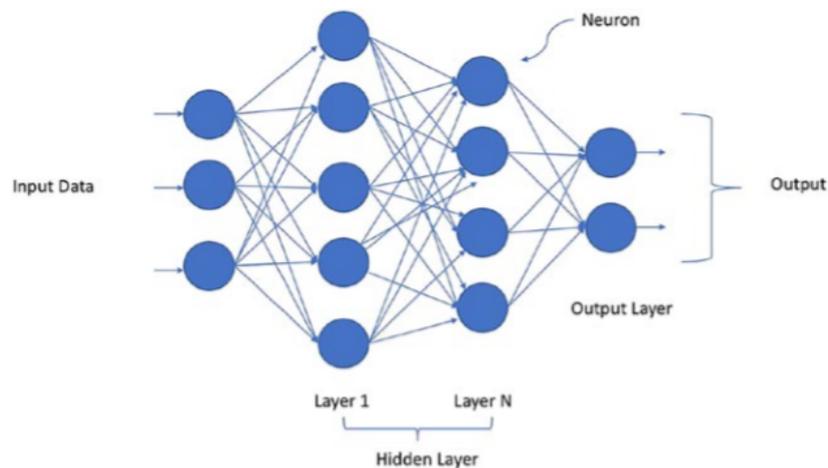


Figura 2.10: Estructura básica de una red neuronal (Moolayil, 2019).

Una vez explicada la estructura de la red neuronal, se va a detallar el proceso de aprendizaje. Cuando se le proporcionan los datos de entrada a la red, la salida final es una predicción, que podrá estar bien o mal. Basándose en la salida, si se realimenta a la red para que se adapte mejor utilizando algún medio para hacer una mejor predicción, el sistema aprende actualizando el peso de las conexiones. Para conseguirlo, se utiliza un algoritmo llamado “backpropagation”. Iterar el proceso varias veces paso a paso, con más y más datos, ayuda a la red a actualizar los pesos de manera adecuada, creando un sistema capaz de tomar una decisión prediciendo la salida. Dicho sistema se basa en las reglas internas que ha creado por sí solo, a través de los pesos y las conexiones.

El nombre de “Deep Neural Networks”, DNN, mencionado anteriormente, deriva del uso de muchas capas ocultas, convirtiéndola en una red amplia para aprender patrones más complejos.

Para finalizar, es interesante remarcar que el desarrollo del aprendizaje profundo, estos últimos años, se debe a la evolución que ha habido a nivel computacional. Ya que, entrenar una red neuronal es computacionalmente pesado y necesita grandes cantidades de datos.

### 2.2.2. Ejemplos de aplicación

Una buena forma de comprobar el potencial y precisión que tienen las redes neuronales, es resolviendo problemas sencillos y comprobando la validez de las predicciones. Por lo que, para realizar estos ejemplos de aplicación, se han escogido las diferentes físicas que intervienen en el cocinado de carne, y se han resuelto pequeños problemas de forma independiente. La metodología de resolución de estos problemas es la siguiente: En primer lugar, se realiza un modelo en el software COMSOL de cada problema, en el que solo interviene la física en cuestión. En segundo lugar, se generan las poblaciones de muestras pertinentes, que servirán de entrenamiento a la red neuronal. A continuación, estableciendo las variables de entrada correspondientes, se entrenan las redes neuronales por medio del software MATLAB. Por último, se comprueba la validez de las predicciones de cada red, comparándolas con la solución del problema obtenida en el software.

#### 2.2.2.1. Mecánica estructural

Para comprobar la validez y la precisión de las redes neuronales dentro de la mecánica estructural, se ha elegido el estudio de una viga en voladizo por medio de la teoría de Timoshenko, figura 2.11

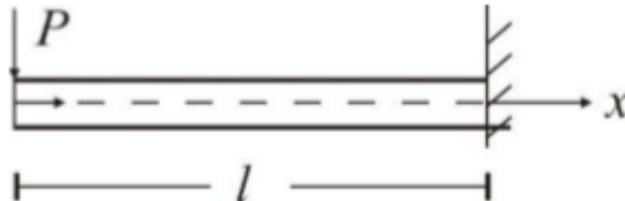


Figura 2.11: Viga en voladizo de sección rectangular y con carga puntual en el extremo libre.

Como parámetros del problema, se tienen los recogidos en la tabla 2.1:

Parámetro	Descripción	Unidad
P	Carga puntual en el extremo libre	N
L	Longitud de la viga	mm
E	Módulo de Young	N/mm <sup>2</sup>
k	Factor de corrección de corte	-
A	Sección de la viga	mm <sup>2</sup>
b	Ancho de la viga	mm
h	Espesor de la viga	mm
$\nu$	Coefficiente Poisson	-

Tabla 2.1: Parámetros ejemplo de Mecánica Estructural.

Para la generación de muestras, se han escogido las variables de la tabla 2.2, que serán los parámetros de entrada de la red neuronal. Cabe destacar, que se va a usar una distribución uniforme al azar, para poder generar la población de modelos.

Variable	Unidad	Descripción
P	N	Carga puntual en el extremo libre
L	mm	Longitud de la viga
E	N/mm <sup>2</sup>	Módulo de Young
b	mm	Ancho de la viga
h	mm	Espesor de la viga

Tabla 2.2: Variables de entrada a la red neuronal en Mecánica Estructural.

Una vez se tienen las variables de entrada a la red, se procede a su entrenamiento. Dicho entrenamiento, se realiza mediante el software MATLAB. Como variable de salida de la red, y en este caso con solución analítica bajo las hipótesis de la Resistencia de Materiales, se tiene la flecha máxima de la viga en voladizo en el punto en el que se ejerce la carga. En este problema, se han obtenido diferentes redes neuronales, en función del tamaño de las muestras de entrenamiento (100, 1000 y 10000), y en función del número de capas ocultas (“hidden layers”) de la propia red (1, 2, 3, 5 y 10). Finalmente, se obtienen los siguientes resultados:

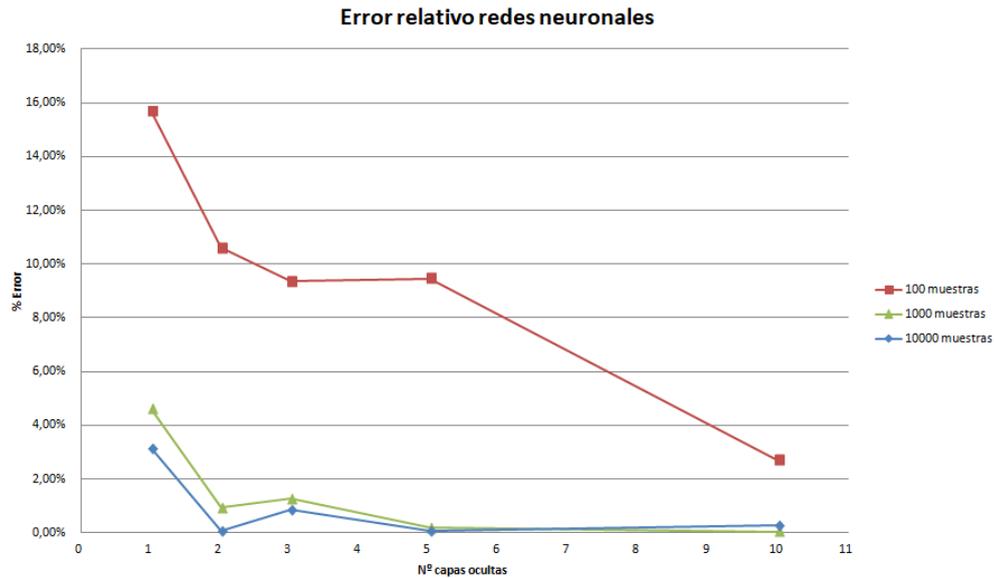


Figura 2.12: Error relativo de las predicciones de la flecha máxima de las redes neuronales.

Como se puede apreciar en la figura 2.12, a mayor número de muestras se obtiene un error relativo menor. Lo que indica que la predicción de la red, está más cerca de la solución del problema, en este caso obtenida analíticamente. Por otro lado, cabe destacar la influencia del número de capas ocultas de la propia red, se puede observar como aumentando el número de las mismas, conseguimos disminuir el error relativo. Esto se ve claramente, para el caso de las redes entrenadas con 100 muestras (línea roja), que pasa de tener un error de más de un 15%, con 1 capa oculta, a tener un error ligeramente superior al 3% con 10 capas ocultas. A mayor tamaño de muestras de entrenamiento, la influencia del número de capas ocultas no es tan significativo, puesto que el error relativo es bajo. A modo de conclusión, se puede añadir que las redes neuronales muestran una gran precisión en este problema de Mecánica Estructural cuando el tamaño de muestras de entrenamiento es superior a 1000, y las redes tienen 5 o más capas ocultas. De forma que no se supera el 1% de error con estas características de la red. El proceso completo de descripción del modelo y entrenamiento de las redes neuronales se detalla en el Anexo A.1.

### 2.2.2.2. Transferencia de calor

Para comprobar la validez y la precisión de las redes neuronales dentro de la física de la transferencia de calor, se va a modelar el siguiente problema en tres dimensiones. En este caso, y a diferencia del ejemplo de la mecánica estructural, el estudio va a ser temporal y no existe solución analítica. La resolución se llevará a cabo utilizando el software COMSOL que utiliza método de los elementos finitos. El modelo queda de la siguiente manera:

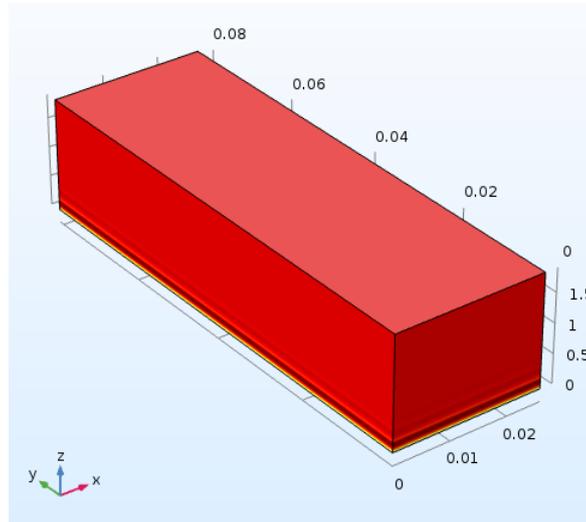


Figura 2.13: Modelo 3D con transferencia de calor en COMSOL.

Como se puede apreciar en la figura 2.13, se tiene un bloque 3D. En la parte inferior del bloque, hay una fuente de calor, simulando la presencia de una sartén. Además de esta fuente de calor, en el resto de las caras hay un flujo de calor convectivo, simulando el proceso de convección externa que existiría en una sartén con el aire que rodea al filete de carne. Los parámetros del modelo son los recogidos en la tabla 2.3.

Parámetro	Descripción	Unidad
$T_{amb}$	Temperatura ambiente	°C
$P_{amb}$	Presión ambiente absoluta	Pa
k	Conductividad térmica del bloque	W/(m·K)
$\rho$	Densidad del bloque	kg/m <sup>3</sup>
$C_p$	Capacidad térmica a presión constante	J/(kg·K)
$T_0$	Temperatura inicial	K
$h_{aire}$	Coefficiente convección del aire	W/(m <sup>2</sup> K)
$Q_{sartén}$	Potencia de la sartén	W/m <sup>2</sup>
thickness	Espesor del bloque	cm
l	Longitud del bloque	cm
a	Anchura del bloque	cm
fracw	Fracción volumétrica de agua en tanto por uno	-
$\rho_{agua}$	Densidad del agua	kg/m <sup>3</sup>
$\rho_{proteina}$	Densidad de la proteína	kg/m <sup>3</sup>

Tabla 2.3: Parámetros ejemplo de Transferencia de Calor.

Para la generación de la población de modelos, se han escogido como variables las indicadas en la tabla 2.4, que serán los parámetros que se le suministrará a la red neuronal como entrada. Para generar las muestras pertinentes, se utilizará una distribución normal para cada uno de los parámetros.

Variable	Unidad	Media	Desviación típica	Descripción
$T_0$	°C	20	2	Temperatura inicial del bloque
thickness	cm	1.9	0.19	Espesor del bloque
$Q_{sartén}$	W/m <sup>2</sup>	15000	1500	Potencia de la sartén
$\rho_{proteina}$	kg/m <sup>3</sup>	1330	133	Densidad de la proteína
fracw	-	0.7222	0.07222	Fracción volumétrica de agua

Tabla 2.4: Variables de entrada a la red neuronal en Transferencia de Calor.

Una vez se tienen las variables de entrada a la red, se procede a su entrenamiento, mediante el software MATLAB, al igual que para el ejemplo anterior. En este caso, como salida de la red se tiene  $T_{sonda}$ , que como se ha explicado en los parámetros del modelo, es la temperatura en el punto central del bloque. Al ser un problema temporal, dicha salida varía en función del tiempo. Por lo que para poder comparar las predicciones de la red (que en vez de predecir un valor mostrará una gráfica de la evolución de la temperatura en cada instante de tiempo) con un valor correcto, se resolverá en COMSOL el mismo problema, pero en vez de temporal en estado estacionario. Es decir, el programa suministrará una temperatura en el centro del bloque, que se corresponderá con el valor final que se obtiene en un estudio temporal cuando se estabiliza dicha temperatura. En este ejemplo, se han entrenado redes neuronales con 10, 100 y 1000

modelos, y en cada población de modelos, se tienen redes neuronales con 1, 2, 3, 5 y 10 capas ocultas. Los resultados que se han obtenido se muestran en la siguiente figura, que representa el % de error relativo de las predicciones en función del número de capas ocultas y del tamaño de la población de modelos.

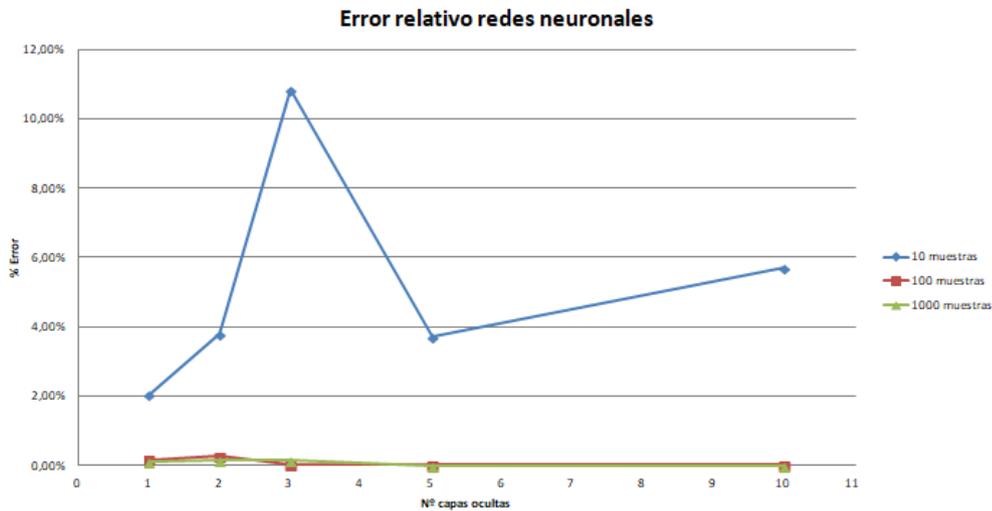


Figura 2.14: Error relativo de las predicciones de la temperatura en el centro del bloque de las redes neuronales.

Como se puede apreciar en la figura 2.14, cuanto mayor es el tamaño de la población de modelos, menor error se obtiene en la predicción, es decir, más se acerca la predicción realizada por las redes neuronales al valor objetivo. Cabe destacar, que para el caso de 10 modelos (número muy bajo) al aumentar el número de capas ocultas, no disminuye el error y además, el error alcanza un 10 %. En este caso, los resultados no guardan ningún tipo de relación en función del número de capas ocultas. Esto se debe, básicamente, a que el tamaño de la población de modelos es muy pequeño y la red no ha conseguido aprender lo suficiente. Por contra, se puede apreciar que con poblaciones de 100 y 1000 modelos, el error que se obtiene, independientemente del número de capas ocultas, es muy bajo, siendo siempre inferior al 1 %. Lo que muestra una gran precisión de las redes neuronales en la Transferencia de Calor cuando el tamaño de la población de modelos es superior a 100. El proceso completo de descripción del modelo y entrenamiento de las redes neuronales se detalla en el Anexo A.2.

### 2.2.2.3. Transferencia de masa

Para comprobar la validez y la precisión de las redes neuronales dentro de la física de la transferencia de masa, se va a modelar el siguiente problema. Al igual que en el ejemplo de transferencia de calor, el estudio va a ser temporal, por lo que no hay solución analítica. Para incluir el método de los elementos finitos, que es el que se usará en el cocinado de carne, se va a modelar el problema en COMSOL. El modelo queda de la siguiente manera:

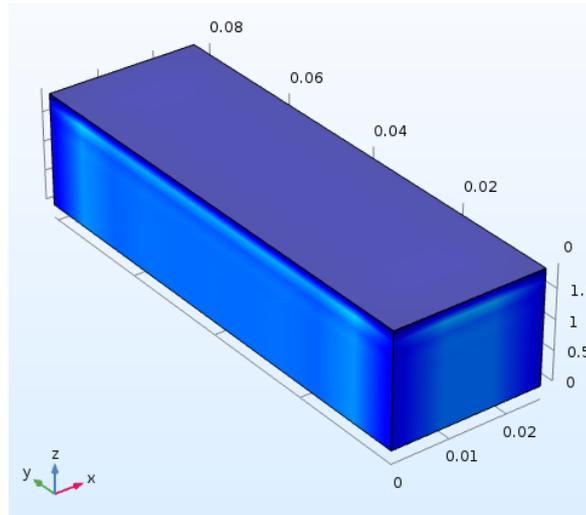


Figura 2.15: Modelo 3D con transferencia de masa en COMSOL.

Como se puede apreciar en la figura 2.15, se tiene el mismo bloque 3D que para transferencia de calor. En la parte inferior del bloque estaría la sartén, por lo que en esa cara no hay transferencia de masa. En el resto de las caras hay una transferencia de masa basada en la diferencia de concentraciones que existe en el exterior del bloque y dentro del mismo. Los parámetros del modelo son los recogidos en la tabla 2.5.

Parámetro	Descripción	Unidad
thickness	Espesor del bloque	cm
l	Longitud del bloque	cm
a	Anchura del bloque	cm
$c_{w,0}$	Concentración inicial de agua	kg/m <sup>3</sup>
Mw	Peso molecular del agua	kg/mol
$h_m$	Coefficiente de transferencia de masa	m/s
D	Coefficiente difusividad del agua	m <sup>2</sup> /s
$c_{in}$	Concentración inicial dentro del bloque	mol/m <sup>3</sup>
$c_{out}$	Concentración fuera del bloque	mol/m <sup>3</sup>

Tabla 2.5: Parámetros ejemplo de Transferencia de Masa.

Para poder generar las diferentes poblaciones de modelos, se han escogido como variables las indicadas en la tabla 2.6, que serán los parámetros de entrada a la red neuronal. Al igual que en el bloque con transferencia de calor, se ha utilizado una distribución normal para generar las muestras pertinentes de las variables.

Variable	Unidad	Media	Desviación típica	Descripción
$h_m$	m/s	0.008	0.0008	Coefficiente de transferencia de masa
D	m <sup>2</sup> /s	$1 \cdot 10^{-9}$	$1 \cdot 10^{-10}$	Coefficiente difusividad del agua
$c_{in}$	mol/m <sup>3</sup>	50000	5000	Concentración inicial dentro del bloque
$c_{out}$	mol/m <sup>3</sup>	610000	1000	Concentración fuera del bloque

Tabla 2.6: Variables de entrada a la red neuronal en Transferencia de Masa.

Una vez definidas las variables de entrada a la red, se procede a su entrenamiento, mediante el software MATLAB, al igual que para el ejemplo anterior. En este caso, como salida de la red se tiene la pérdida de peso relativa del bloque en tanto por uno. Al ser un problema temporal, dicha salida varía en función del tiempo. Por lo que para poder comparar las predicciones de la red (que en vez de predecir un valor mostrará una gráfica de la evolución de la pérdida de peso en cada instante de tiempo) con un valor correcto, se resolverá en COMSOL el mismo problema, pero en vez de temporal en estacionario. Es decir, el programa suministrará la pérdida de peso del bloque, que se corresponderá con el valor final que se obtiene en un estudio temporal cuando se llega a régimen permanente. En este ejemplo, se han entrenado redes neuronales con 10, 100 y 1000 modelos, y en cada población de modelos, se tienen redes neuronales con 1, 2, 3, 5 y 10 capas ocultas. Los resultados que se han obtenido se muestran en la siguiente figura, que representa el % de error relativo de las predicciones en función del número de capas ocultas y del tamaño de la población de modelos.

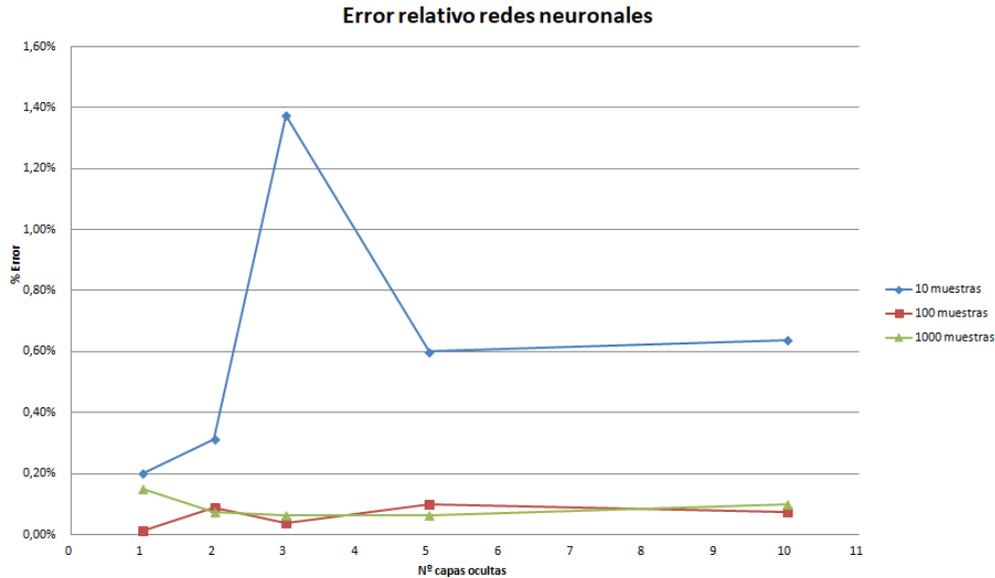


Figura 2.16: Error relativo de las predicciones de la pérdida de peso del bloque de las redes neuronales.

Como se puede apreciar en la figura 2.16, la tendencia en el error de las predicciones de la pérdida de peso del bloque es similar a la de las predicciones de la temperatura del ejemplo anterior (figura 2.14). A mayor tamaño de población de modelos, menor error se obtiene en la predicción. No obstante, cabe destacar, que para este ejemplo, en el caso de la población de 10 modelos se obtiene un error como mucho del 1,4 %, mientras que para el ejemplo anterior, con una población de 10 modelos se llegaba hasta un 10 %. También es verdad, que al aumentar el número de capas ocultas, no disminuye el error. En este caso, los resultados no guardan ningún tipo de relación en función del número de capas ocultas. Esto se debe, básicamente, a que el tamaño de la población de modelos es muy pequeño. Por contra, se puede apreciar que con poblaciones de 100 y 1000 muestras, el error que se obtiene, independientemente del número de capas ocultas, es muy bajo, siendo siempre inferior al 0.15 %. Lo que muestra una gran precisión de las redes neuronales en la Transferencia de Masa, tanto para redes entrenadas con 10 modelos, como para redes entrenadas con más de 100 modelos, donde la precisión es la más elevada de los tres ejemplos descritos. El proceso completo de descripción del modelo y entrenamiento de las redes neuronales se detalla en el Anexo A.3.

## 2.3. Simulación del cocinado de carne

Una vez comprobada la validez que tiene el uso de las redes neuronales en la resolución de problemas basados en las físicas de este estudio, se van a aplicar para predecir el cocinado de la carne. Por lo que, antes de proceder a su simulación y comprobar su validez, es preciso describir los modelos de elementos finitos usados para simular el proceso del cocinado de la carne en el programa multifísico COMSOL. Para ello, se han realizado dos modelos de elementos finitos: uno en 1D y otro en 3D, que serán descritos a continuación. Cabe destacar que ambos se basan en Dhall and Datta (2011).

### 2.3.1. Modelo de elementos finitos en 1D

En la construcción del modelo de elementos finitos en 1D, capaz de simular el proceso de cocinado en sartén tan solo se ha considerado el filete de carne. El modelo simula la línea central del filete a lo largo del espesor, única dimensión.

Para el modelado de los distintos fenómenos que tienen lugar durante el cocinado se han empleado los módulos físicos del software de “Transferencia de Calor en sólidos”, haciendo referencia a la transferencia de calor existente, “Transporte de Especies diluidas”, haciendo referencia a la transferencia de masa. Las variables del problema a resolver son, respectivamente para cada módulo: la temperatura  $T$  en [°C] en una sonda situada en medio del espesor y la concentración molar de agua en el sólido  $c_w$  en [mol/m<sup>3</sup>] en cada punto de la malla. Aún así, para que la transferencia de masa sea más fácil de interpretar, en lugar de representarla con  $c_w$ , se va a representar mediante una variable introducida en el modelo: Pérdida de Peso en %. Esta pérdida de peso se obtiene fácilmente mediante la siguiente ecuación:

$$PP = \left( 1 - \frac{\int_V c_w dV}{\int_V c_{w0} dV} \right) \cdot 100 \quad (2.48)$$

Una vez explicadas las físicas que se han implementado en el modelo, en la tabla 2.7 se adjuntan los parámetros y variables que intervienen en el cálculo del modelo y de las propiedades térmicas del producto explicadas en el punto 2.1.1.2 de esta memoria. Basados en el modelo validado de Bernad Pérez (2019).

Parámetro	Valor	Unidad	Descripción
$C_{p,s}$	2008.2	J/(kg·K)	Capacidad calorífica de la proteína
$C_{p,w}$	4128.9	J/(kg·K)	Capacidad calorífica del agua
$c_{w0}$	773.98	kg/m <sup>3</sup>	Concentración inicial de agua líquida en el filete
$c_{w,eq}$	Ecuación (2.26)	mol/m <sup>3</sup>	Capacidad de retención del agua del filete
D	$1 \cdot 10^{-9}$	m <sup>2</sup> /s	Coefficiente difusividad del agua
$\varepsilon_w$	0.7222	-	Fracción volumétrica de agua en el filete de carne
g	0.45	-	Parámetro del modelo de conductividad del calor
$h_{aire}$	10	W/(m <sup>2</sup> K)	Coefficiente de convección del aire
$h_m$	0.008	m/s	Coefficiente de transferencia de masa
$\lambda_s$	0.17881	W/(m·K)	Conductividad térmica de la proteína
$\lambda_w$	0.57	W/(m·K)	Conductividad térmica del agua
$\lambda$	$2.26 \cdot 10^6$	J/kg	Calor latente de vaporización del agua
$M_w$	0.018015	kg/mol	Peso molecular del agua
$P_{amb}$	101325	Pa	Presión ambiente
$Q_{sartén}$	15000	W/m <sup>2</sup>	Potencia suministrada por la sartén
$\rho_s$	1330	kg/m <sup>3</sup>	Densidad de la proteína
$\rho_w$	997.2	kg/m <sup>3</sup>	Densidad del agua
$\rho_{v,amb}$	0.73636	kg/m <sup>3</sup>	Densidad del vapor en el aire circundante
$\rho_v$	Ecuación (2.28)	kg/m <sup>3</sup>	Densidad del vapor en la superficie del filete
$T_{inicial}$	20	°C	Temperatura inicial del filete de carne
$T_{amb}$	25	°C	Temperatura ambiente
$t_{cocinado}$	300	s	Tiempo de cocinado del filete de carne
$t_{vuelta}$	200	s	Tiempo en el que se le da la vuelta al filete de carne
Espesor	1.9	cm	Espesor filete de carne

Tabla 2.7: Parámetros y variables del modelo 1D.

En este caso, la geometría del modelo es muy sencilla al ser 1D. Tan solo, consiste en una recta de longitud el parámetro espesor, que simula una línea en el centro del filete.



Figura 2.17: Geometría del modelo 1D.

Esta simplificación en la geometría, al ser un modelo 1D, también se ve reflejada en las condiciones de contorno para las dos físicas que intervienen. De manera que se tiene:

## Transferencia de calor

- Fuente de calor de contorno en un extremo de la geometría simulando el contacto filete-sartén, controlada por el parámetro  $Q_{sarten}$ .
- Flujo de calor en el mismo extremo de la geometría simulando el calor que absorbe el agua al evaporarse en el contacto filete-sartén.
- Flujo de calor en el extremo opuesto simulando la convección en la parte superior del filete, controlado por el parámetro  $h_{aire}$  y  $T_{amb}$ .

Cabe destacar, que al simularse también el cocinado de la otra cara del filete, la fuente y el flujo de calor situados en un extremo deberán situarse justo en el otro en cuánto se llegue al parámetro de simulación  $t_{vuelta}$ . Lo mismo ocurrirá con el flujo de calor convectivo.

## Transferencia de masa

- Reacciones en todo el dominio, simulando las reacciones de difusión y convección dentro del filete de carne, definidas por el parámetro D.
- Flujo de masa en uno de los extremos, simulando la evaporación de agua que se da en el contacto filete-sartén, controlado por los parámetros  $h_m$  y  $\rho_{v,amb}$ .

Al igual que para el caso de la transferencia de calor, el flujo de masa situado en un extremo, se situará en el otro para simular el cocinado de la otra cara del filete una vez el tiempo pase del parámetro de simulación  $t_{vuelta}$ .

En definitiva, en el modelo 1D se eliminan los flujos por goteo y los flujos de calor de convección laterales que estarán presentes en el modelo 3D, ya que al tener en cuenta solo la dimensión del espesor no tiene sentido físico incluirlos, puesto que suceden en las caras laterales.

Por último, la malla que se ha elegido para simular este modelo es una controlada por la física con un tamaño de elemento normal, tal y como se puede apreciar en la siguiente figura. Obteniendo 15 elementos y 16 nodos.



Figura 2.18: Malla del modelo 1D.

En cuanto al estudio temporal, se ha escogido un tiempo de cocinado igual a 300 s y un tiempo de vuelta igual a 200 s, tal y como se aprecia en la tabla de parámetros del modelo. Estos valores se corresponden a un punto de cocinado de la carne “medium rare”.

### 2.3.2. Modelo de elementos finitos en 3D

En la construcción del modelo de elementos finitos en 3D capaz de simular el proceso de cocinado en sartén se han considerado dos geometrías: el filete de carne y la sartén. Cabe mencionar que la modelización de esta última, se ha realizado teniendo en cuenta una temperatura constante en toda la superficie de la misma. Por otro lado, se ha comprobado la escasa influencia sobre el cocinado que ejercen las paredes de la sartén, siendo el único efecto considerable el aumento de la temperatura del aire que rodea a la pieza de carne.

Para el modelado de los distintos fenómenos que tienen lugar durante el cocinado de carne, se han empleado los módulos físicos del software de “Transferencia de Calor en sólidos”, haciendo referencia a la transferencia de calor existente, “Transporte de Especies diluidas”, haciendo referencia a la transferencia de masa, y “Mecánica del sólido”, haciendo referencia a la deformación que sufre el filete de carne durante el cocinado. Las variables del problema a resolver son: la temperatura  $T$  en  $[\text{°C}]$  y la concentración molar de agua en el sólido  $c_w$  en  $[\text{mol}/\text{m}^3]$ , ambas en cada nodo de la malla.

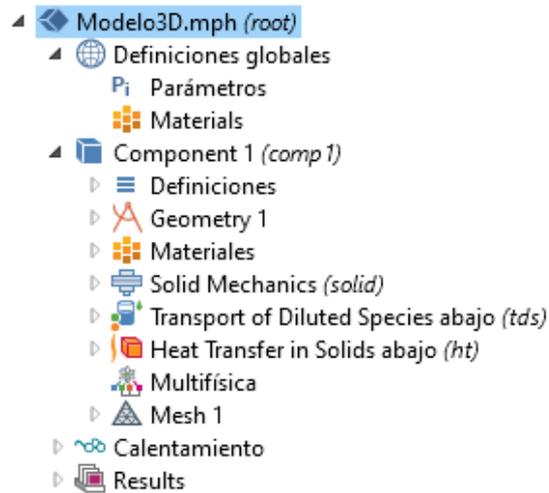


Figura 2.19: Físicas empleadas en COMSOL para el modelo 3D.

Una vez explicadas las físicas que se han implementado en el modelo, en las siguientes tablas se adjuntan los parámetros y variables, respectivamente, que intervienen en el cálculo del modelo y de las propiedades térmicas del producto explicadas en el punto 2.1.1.2. de esta memoria. Basados en el modelo validado de Bernad Pérez (2019).

Parámetro	Valor	Unidad	Descripción
$C_{p,sarten}$	475	J/(kg·K)	Capacidad calorífica de la sartén de acero
$c_{w0}$	773.98	kg/m <sup>3</sup>	Concentración inicial de agua líquida en el filete de carne
D	$1 \cdot 10^{-9}$	m <sup>2</sup> /s	Coefficiente difusividad del agua
$\varepsilon_w$	0.7222	-	Fracción volumétrica de agua en el filete de carne
$\varepsilon_{w,0}$	0.77615	-	Fracción volumétrica de agua inicial en el filete de carne
g	0.45	-	Parámetro del modelo de conductividad del calor
$h_{aire}$	10	W/(m <sup>2</sup> K)	Coefficiente de convección del aire
$h_m$	0.008	m/s	Coefficiente de transferencia de masa
$H_C$	120	W/(m <sup>2</sup> K)	Coefficiente de conductancia entre sartén y filete
$\lambda_{sarten}$	44.5	W/(m·K)	Conductividad térmica de la sartén de acero
$\lambda_w$	0.57	W/(m·K)	Conductividad térmica del agua
$\lambda$	$2.26 \cdot 10^6$	J/kg	Calor latente de vaporización del agua
$M_w$	0.018015	kg/mol	Peso molecular del agua
$P_{amb}$	101325	Pa	Presión ambiente
$\rho_s$	1330	kg/m <sup>3</sup>	Densidad de la proteína
$\rho_w$	997.2	kg/m <sup>3</sup>	Densidad del agua
$\rho_{v,amb}$	0.73636	kg/m <sup>3</sup>	Densidad del vapor en el aire circundante
$\rho_{sarten}$	7850	kg/m <sup>3</sup>	Densidad de la sartén de acero
$T_{inicial}$	20	°C	Temperatura inicial del filete de carne
$T_{amb}$	25	°C	Temperatura ambiente
$T_{sarten}$	215	°C	Temperatura de la sartén
$T_{amb,lateral}$	50	°C	Temperatura ambiente en los laterales del filete de carne
$t_{cocinado}$	300	s	Tiempo de cocinado del filete de carne
$t_{vuelta}$	200	s	Tiempo en el que se le da la vuelta al filete de carne
Largo	8.1	cm	Longitud filete de carne
Ancho	2.6	cm	Anchura filete de carne
Espesor	1.9	cm	Espesor filete de carne
$Espesor_{sarten}$	0.5	cm	Espesor sartén
$\phi_{sarten}$	10	cm	Diámetro sartén
$K_w$	$2.2 \cdot 10^9$	Pa	Módulo de compresibilidad del agua
$\nu$	0.49	-	Coefficiente de Poisson del filete de carne
$\nu_{sarten}$	0.30	-	Coefficiente de Poisson de la sartén de acero
$E_{sarten}$	$200 \cdot 10^6$	kPa	Módulo de Young de la sartén de acero

Tabla 2.8: Parámetros del modelo 3D.

Variable	Expresión	Unidad	Descripción
$C_{p,s}$	$2,0082 + 1,2089 \cdot 10^{-3} \cdot T - 1,3129 \cdot 10^{-6} \cdot T^2$	J/(kg·K)	Capacidad calorífica proteína
$C_{p,w}$	$4,1289 - 9,0864 \cdot 10^{-5} \cdot T + 5,4731 \cdot 10^{-6} \cdot T^2$	J/(kg·K)	Capacidad calorífica agua
$c_{w,eq}$	Ecuación (2.26)	mol/m <sup>3</sup>	Capacidad retención agua
$\lambda_s$	$1,7881 \cdot 10^{-1} + 1,1958 \cdot 10^{-3} \cdot T - 2,7178 \cdot 10^{-6} \cdot T^2$	W/(m·K)	Conductividad térmica proteína
$\rho_v$	Ecuación (2.28)	kg/m <sup>3</sup>	Densidad vapor en superficie
$E$	Ecuación (2.47)	kPa	Módulo de Young filete
$G$	Ecuación (2.46)	kPa	Módulo cizalladura filete
$\phi$	Ecuación (2.42)	-	Porosidad filete carne
$M$	Ecuación (2.14)	$\frac{kg_{agua}}{kg_{carne}}$	Contenido agua líquida base seca
$J_{el}$	Ecuación (2.43)	-	Jacobiano elástico filete carne
$J_M$	Ecuación (2.40)	-	Jacobiano pérdida de humedad
$K$	$\frac{K_w}{c_w}$	Pa	Módulo compresibilidad filete

Tabla 2.9: Variables del modelo 3D.

En lo que respecta a la geometría del modelo, como se ha indicado anteriormente, se ha incluido la presencia de la sartén. Cabe destacar que se han diseñado dos sartenes, una arriba y otra abajo para simular el cocinado de ambas caras, por lo que cuando una esté activa la otra no lo estará. Además, para reducir el coste computacional de los cálculos, se ha optado por aplicar simetría en ambos ejes del plano horizontal del modelo (ejes x e y de la figura 2.20), por lo que en cada física se tendrá en cuenta dicha simetría. El modelo queda de la siguiente manera:

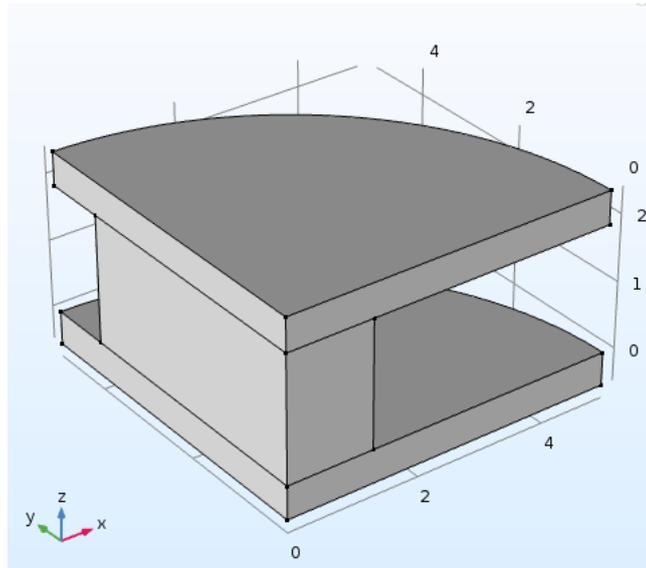


Figura 2.20: Geometría del modelo 3D.

A continuación se describen las condiciones de contorno para cada física, basadas en la figura 2.2.

### Transferencia de calor

- Fuente de calor en todo el dominio del filete.
- Flujo de calor convectivo en las paredes laterales del filete simulando la convección existente con el aire que rodea el filete por sus caras laterales, controlado por los parámetros  $h_{aire}$  y  $T_{amb,lateral}$ .
- Flujo de calor convectivo en la cara superior del filete simulando la convección existente con el aire en la cara de arriba del filete, controlado por los parámetros  $h_{aire}$  y  $T_{amb}$ .
- Par de contacto térmico entre sartén y filete definido por un flujo de calor de conductancia  $H_C$  y a una temperatura  $T_{sarten}$ .
- Flujo de calor en la cara de contacto sartén-filete simulando el calor que absorbe el agua al evaporarse en dicho contacto.

Cabe destacar, que al simularse también el cocinado de la otra cara del filete; el flujo de calor convectivo en la cara superior, el par de contacto térmico y el flujo de calor en el contacto sartén-filete deberán situarse justo en el otro lado en cuánto se llegue al parámetro de simulación  $t_{vuelta}$ .

### Transferencia de masa

- Reacciones en todo el dominio del filete, simulando las reacciones de difusión y convección dentro del filete de carne, definidas por el parámetro  $D$ .
- Flujos de masa en las caras laterales del filete simulando el goteo de agua líquida que se produce en ellas, controlados por los parámetros  $h_m$ ,  $\rho_{v,amb}$  y  $\rho_v$ , entre otros.
- Flujo de masa en el contacto sartén-filete, simulando la evaporación de agua que se da en esa cara, y controlado por los parámetros  $h_m$ ,  $\rho_{v,amb}$  y  $\rho_v$ .

Al igual que para el caso de la transferencia de calor, el flujo de masa situado en el contacto sartén-filete, se situará en el otro lado para simular el cocinado de la otra cara del filete una vez el tiempo de simulación exceda del parámetro de simulación  $t_{vuelta}$ .

## Mecánica del sólido

- Se ha tenido en cuenta al filete de carne como un material hiperelástico, para ello se han definido las variables  $G$ ,  $J_{el}$ ,  $J_M$  y  $K$ .
- Condición de contorno de rodillo en el contacto sartén-filete.

Por último, la malla que se ha elegido para simular este modelo es la de la siguiente figura. En ella, se puede apreciar como se han usado diferentes elementos para el filete y la sartén. Por un lado, para el filete de carne se han usado 300 elementos hexaédricos. Mientras que para la sartén, se han usado 224 elementos tetraédricos.

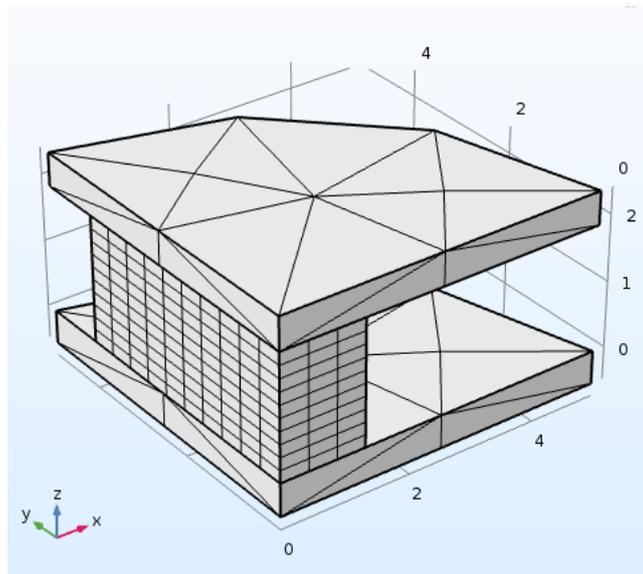


Figura 2.21: Malla del modelo 3D.

En cuanto al estudio temporal, se ha escogido un tiempo de cocinado igual a 300 s y un tiempo de vuelta igual a 200 s, tal y como se aprecia en la tabla de parámetros del modelo. Estos valores se corresponden a un punto de cocinado de la carne “medium rare”.



# Capítulo 3

## Simulación del modelo y validación

Una vez descritos los modelos de elementos finitos que se han implementado en el software COMSOL Multiphysics<sup>®</sup> 5.2a para la simulación del modelo, en este capítulo se va a explicar el proceso de generación de las poblaciones de modelos y de entrenamiento de las redes neuronales para cada modelo, mediante el software MATLAB. Una vez entrenadas, a las redes neuronales se les suministrarán una serie de variables de entrada, y estas serán capaces de predecir el cocinado del filete de carne, mediante parámetros como la temperatura del filete o la pérdida de peso que este sufre. Para la validación de las predicciones, estas se compararán con el resultado obtenido para temperatura y pérdida de peso, mediante el programa multifísico COMSOL.

Por lo que, el esquema de este capítulo queda de la siguiente manera. En primer lugar, se explicará el proceso de generación de la población de modelos. Para el modelo 1D se han escogido poblaciones de 10, 100 y 1000 modelos, mientras que para el modelo 3D se han escogido poblaciones de 10 y 100 modelos, debido al elevado coste computacional que supone el cálculo del modelo en tres dimensiones. Cabe destacar, en este apartado, la selección de las variables de entrada, puesto que la generación de la población de modelos se basa en modificar una serie de variables para, así, conseguir tener una población. En segundo lugar, se describirá el proceso de entrenamiento que tienen las redes neuronales, para conseguir que las predicciones sean óptimas. Por último, las predicciones obtenidas se compararán con el resultado proporcionado por el método de los elementos finitos, comprobando, así, su validez.

## 3.1. Modelo 1D de elementos finitos

### 3.1.1. Generación población modelos

La generación de una población de modelos consiste en realizar tantos estudios temporales como tamaño de la población de modelos se precise, mediante el software COMSOL. Para ello, es necesario definir los parámetros que se van a modificar. La generación de muestras será uniforme, para que todos los valores tengan la misma probabilidad de aparición. Esta distribución estará acotada, es decir, oscilará entre un valor máximo y uno mínimo. Estos parámetros que se van a modificar para originar la población de modelos, serán, posteriormente, las variables de entrada a cada una de las redes neuronales.

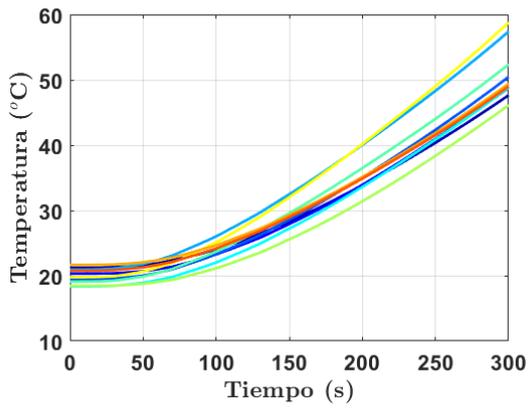
#### 3.1.1.1. Variables de entrada

Para el modelo 1D de elementos finitos, se han elegido los parámetros que se adjuntan en la tabla 3.1. El criterio de selección se ha basado en la importancia de cada parámetro en el desarrollo del problema. De esta forma se han escogido 3 parámetros fundamentales en la Transferencia de Masa, y 5 parámetros fundamentales en la Transferencia de Calor.

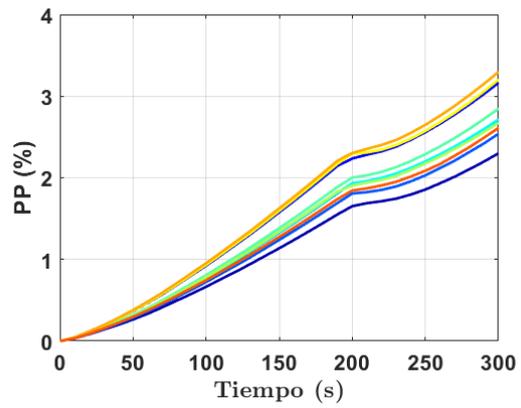
Variable	Valor mínimo	Valor máximo	Unidad	Descripción
D	$9 \cdot 10^{-10}$	$11 \cdot 10^{-10}$	m <sup>2</sup> /s	Coficiente difusividad del agua
$h_m$	0.0072	0.0088	m/s	Coficiente de transferencia de masa
$c_{w0}$	696.58	851.38	kg/m <sup>3</sup>	Concentración inicial de agua líquida en el filete
$Q_{sartén}$	13500	16500	W/m <sup>2</sup>	Potencia suministrada por la sartén
$T_{inicial}$	18	22	°C	Temperatura inicial del filete de carne
Espesor	1.71	2.09	cm	Espesor filete de carne
g	0.405	0.495	-	Parámetro del modelo de conductividad del calor
$\varepsilon_w$	0.65	0.79	-	Fracción volumétrica de agua en el filete de carne

Tabla 3.1: Variables de entrada a la redes neuronales en el modelo 1D.

A continuación se adjuntan las poblaciones de 10, 100 y 1000 modelos, con sus respectivos resultados en Temperatura y Pérdida de Peso, que son las que van a servir de entrenamiento para las redes neuronales.

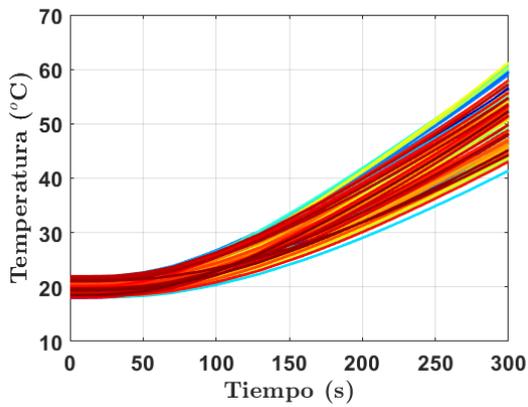


(a) Temperatura

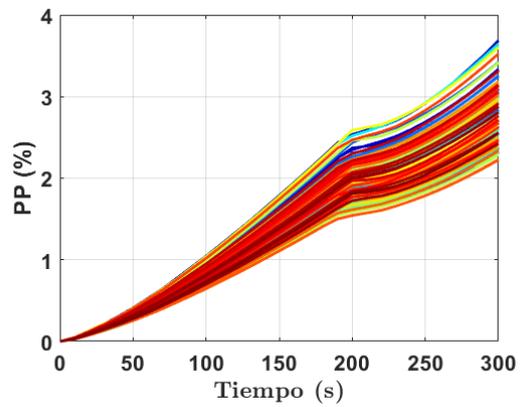


(b) Pérdida de Peso

Figura 3.1: Población de 10 modelos usados como entrenamiento para las redes neuronales en el modelo 1D.

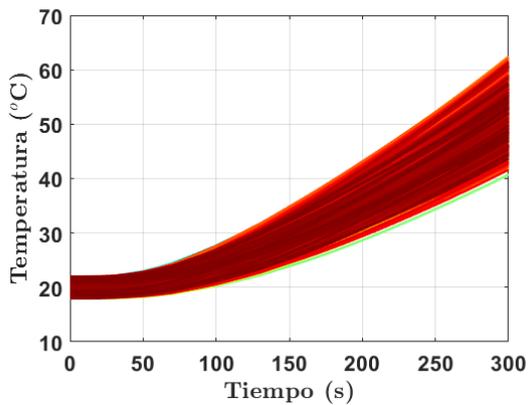


(a) Temperatura

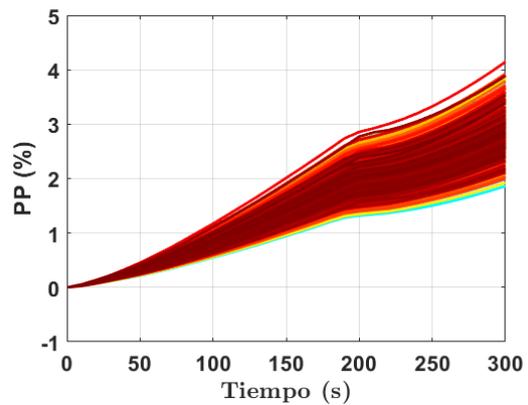


(b) Pérdida de Peso

Figura 3.2: Población de 100 modelos usados como entrenamiento para las redes neuronales en el modelo 1D.



(a) Temperatura



(b) Pérdida de Peso

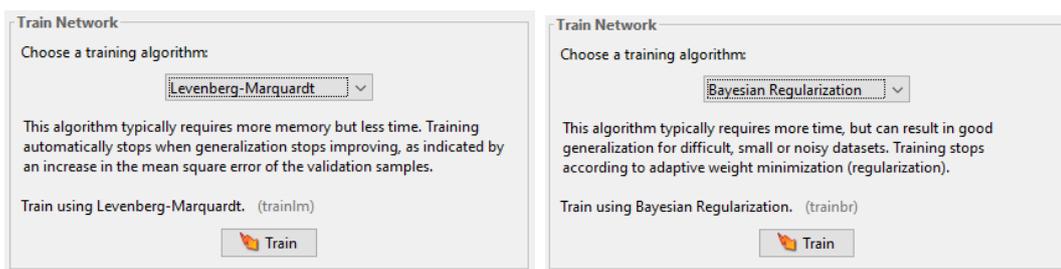
Figura 3.3: Población de 1000 modelos usados como entrenamiento para las redes neuronales en el modelo 1D.

### 3.1.2. Entrenamiento de las redes

El entrenamiento de las redes neuronales se ha llevado a cabo con el software MATLAB. Para ello, se necesitan unas variables de entrada a la red. Estas variables son las descritas anteriormente, en la tabla 3.1. Las salidas (predicciones) de cada red se corresponderán con la temperatura en el punto central del espesor y la pérdida de peso, respectivamente, obteniendo una red para cada salida. En este caso, como el tiempo de simulación son 300 s, la red proporcionará un valor de las salidas cada 10 s, es decir, se obtendrán 31 valores (teniendo en cuenta el valor inicial) a lo largo del tiempo.

Un aspecto fundamental a considerar en el entrenamiento de las redes neuronales, está en la elección del número de capas ocultas de la propia red. En este caso, tanto para las redes que van a predecir la temperatura, como para las que van a predecir la pérdida de peso, se han elegido 1, 2, 3, 5 y 10 capas ocultas.

El último aspecto a considerar en el entrenamiento de las redes neuronales tras el tamaño de la población de modelos, las variables de entrada, las salidas de la red y el número de capas ocultas, es el algoritmo de entrenamiento. Como se puede ver en la siguiente figura, se han escogido dos algoritmos de entrenamiento para cada red. Por un lado, está el algoritmo Levenberg-Marquardt, que requiere una mayor memoria pero menor tiempo. Por otro lado, está el algoritmo Bayesian Regularization, que requiere mayor tiempo pero puede ser bastante bueno para un conjunto de datos difícil, pequeño o con bastante ruido.



(a) Levenberg-Marquardt

(b) Bayesian Regularization

Figura 3.4: Algoritmos de entrenamiento para las redes neuronales en el modelo 1D.

### 3.1.3. Resultados y validación

Antes de comparar las predicciones obtenidas con las diferentes redes neuronales creadas, es preciso indicar qué valores de las variables de entrada se han tomado para obtener la solución en el software COMSOL, que son los indicados en la tabla 2.7 de parámetros y variables del modelo 1D. Estos valores serán los mismos, que se le suministrarán a cada red como variables de entrada, para que hagan su predicción.

De tal forma, que con las variables y parámetros mencionados de la tabla 2.7, se obtienen los siguientes resultados en COMSOL para temperatura y pérdida de peso, que serán con los que se van a comparar las predicciones de las redes neuronales.

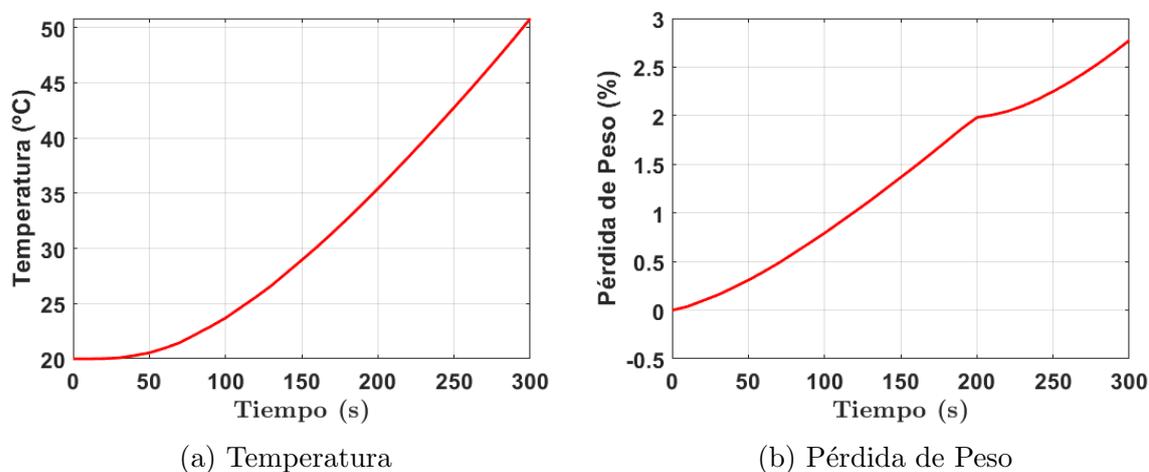


Figura 3.5: Resultado obtenido en COMSOL para el modelo 1D.

Una vez obtenidos los resultados objetivo, para poder comparar las predicciones de las redes neuronales y comprobar su validez, con el objetivo que la extensión de la memoria no sea elevada, se escogerán las predicciones de la red neuronal entrenada con una población de 100 modelos, por estar entre 10 y 1000 modelos; y con el algoritmo de entrenamiento L-M. El proceso completo de comparación y validación de los resultados obtenidos con el resto de redes neuronales, para todos los tamaños de población de modelos y con ambos algoritmos de entrenamiento se detalla en el anexo B.1.

### Redes neuronales entrenadas con una población de 100 modelos y con el algoritmo de entrenamiento Levenberg-Marquardt

Los resultados en temperatura y pérdida de peso, en comparación con el valor obtenido en COMSOL, obtenidos con redes neuronales de 1, 2, 3, 5 y 10 capas ocultas y entrenadas con un algoritmo de entrenamiento Levenberg-Marquardt, son los siguientes.

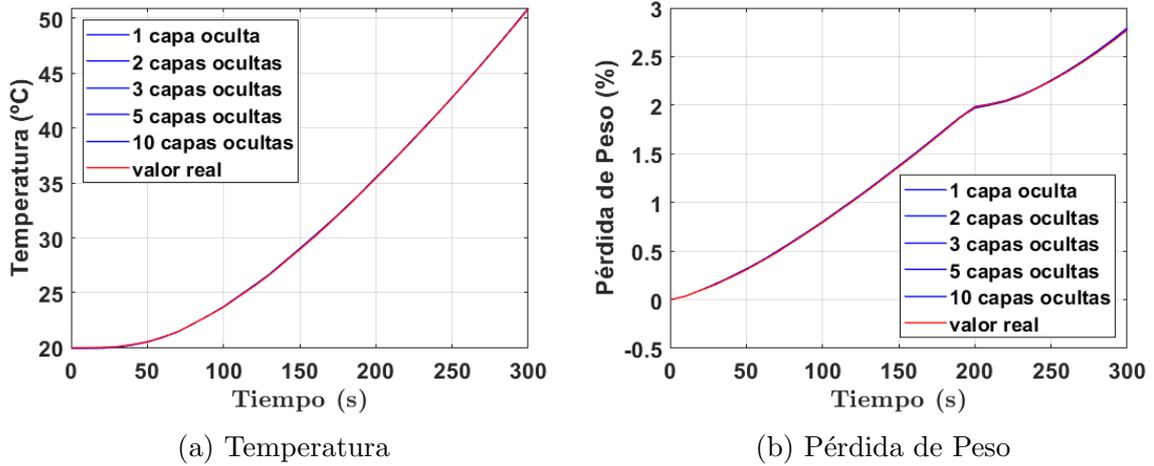


Figura 3.6: Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Levenberg-Marquardt y una población de 100 modelos para el modelo 1D.

Al aumentar el tamaño de la población de modelos de entrenamiento, las redes neuronales son capaces de predecir, tanto los valores de la temperatura como los valores de la pérdida de peso, de forma muy exacta. Como se puede ver en la figura B.6, solo se aprecia una línea cuando debería haber 6, ya que hay 5 redes neuronales y el valor en COMSOL. Esto, significa que las predicciones de las redes son muy uniformes y el efecto del número de capas ocultas, no se manifiesta (en teoría, a mayor número de capas ocultas, mayor precisión). Comparando las predicciones con el valor obtenido en COMSOL, se puede apreciar la elevada precisión obtenida. Por lo que, a modo de conclusión, se puede afirmar la validez de las predicciones obtenidas.

## 3.2. Modelo 3D de elementos finitos

### 3.2.1. Generación población modelos

El inconveniente que tiene el modelo 3D, reside en el tiempo de simulación. Y es por eso, que a la hora de generar una población de modelos, solo se han tenido en cuenta los tamaños de 10, cuyo tiempo de simulación es superior a 2h, y de 100, cuyo tiempo de simulación supera las 24h. Tal y como se ha explicado para el modelo 1D, la generación de una población de modelos consiste en simular en el software COMSOL tantos modelos del problema planteado, como tamaño de la población se requiera. Un aspecto importante a considerar, es el tamaño de dichas poblaciones. En este caso, a diferencia del modelo 1D, se han escogido poblaciones de 10 y 100 modelos, puesto que el coste computacional de simulación del modelo en tres dimensiones es muy superior al del modelo en una dimensión. Otro aspecto fundamental, está en la definición de las variables a modificar. Al igual que para el modelo 1D, se ha utilizado una distribución uniforme, de manera que todos los valores, acotados entre un mínimo y un máximo, tengan la misma probabilidad de aparición.

#### 3.2.1.1. Variables de entrada

Para el modelo 3D de elementos finitos se han elegido los parámetros que se adjuntan en la tabla 3.2. El criterio de selección de los parámetros que se ha seguido, es similar al del modelo 1D, escogiendo 3 parámetros fundamentales en la Transferencia de Masa, y 5 parámetros fundamentales en la Transferencia de Calor. No se ha escogido ningún parámetro relacionado directamente con la Mecánica Estructural, puesto que la mayoría ya dependen de la temperatura y del material de la sartén, supuesto acero en todo momento.

Variable	Valor mínimo	Valor máximo	Unidad	Descripción
D	$9 \cdot 10^{-10}$	$11 \cdot 10^{-10}$	$\text{m}^2/\text{s}$	Coficiente difusividad del agua
$h_m$	0.0072	0.0088	m/s	Coficiente de transferencia de masa
$c_{w0}$	696.58	851.38	$\text{kg}/\text{m}^3$	Concentración inicial de agua líquida en el filete
$H_C$	108	132	$\text{W}/(\text{m}^2\text{K})$	Coficiente de conductancia entre sartén y filete
$T_{inicial}$	18	22	$^{\circ}\text{C}$	Temperatura inicial del filete de carne
Espesor	1.71	2.09	cm	Espesor filete de carne
g	0.405	0.495	-	Parámetro del modelo de conductividad del calor
$\varepsilon_w$	0.65	0.79	-	Fracción volumétrica de agua en el filete de carne

Tabla 3.2: Variables de entrada a las redes neuronales en el modelo 3D.

### 3.2.2. Entrenamiento de las redes

Al igual que para el modelo 1D, el entrenamiento de las redes neuronales se ha llevado a cabo mediante el software MATLAB. Como variables de entrada a las redes neuronales se tienen las descritas en la tabla 3.2. Las salidas (predicciones) de cada red se corresponderán con la temperatura y la concentración molar de agua en cada nodo de la malla y en cada instante de tiempo, obteniendo una red para cada predicción. En este caso, como el tiempo de simulación es de 300 s, cada red proporcionará 524 valores (524 nodos) cada 10 s, es decir, se obtendrán 524 x 31 valores (teniendo en cuenta el valor inicial) a lo largo de la simulación.

Un aspecto fundamental a considerar en el entrenamiento de las redes neuronales, tras el tamaño de la población de modelos, las variables de entrada, las salidas de la red, es el algoritmo de entrenamiento. Al igual que para el modelo 1D, se han elegido los algoritmos de entrenamiento de la figura 3.4. Por un lado está el algoritmo Levenberg-Marquardt, que se va a usar para entrenar redes neuronales con 1 y 10 capas ocultas. Por otro lado, está el algoritmo Bayesian Regularization, que al requerir un mayor tiempo de entrenamiento, se ha usado para entrenar redes neuronales con 1 y 2 capas ocultas. Se han escogido estos números de capas ocultas, ya que al ser un modelo 3D el entrenamiento ya cuesta de por sí, por lo que para el caso del algoritmo L-M se compararán resultados con 1 y 10 capas ocultas, y para el caso del algoritmo BR, debido a su elevado tiempo de entrenamiento, se compararán resultados con 1 y 2 capas ocultas.

### 3.2.3. Resultados y validación

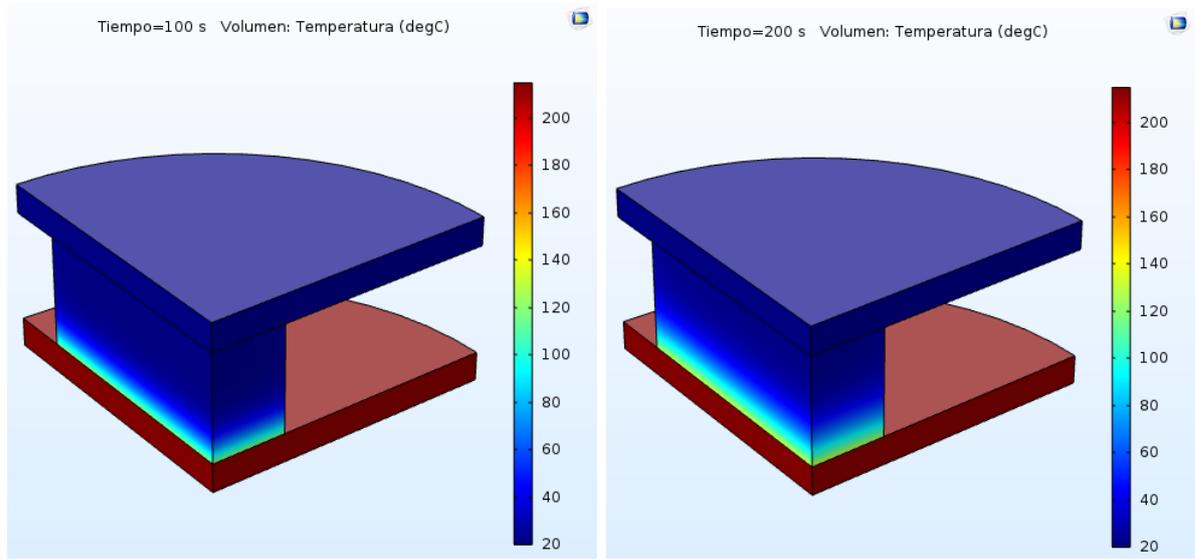
Para el modelo 3D, a diferencia del modelo 1D, se tienen 524 valores de temperatura y concentración molar de agua diferentes para cada discretización en el tiempo, 31 en este caso. Por lo que para poder representar las predicciones de cada red neuronal y poder compararlas con el resultado obtenido en COMSOL, se mapearán estas predicciones en la geometría del modelo obteniendo 31 dibujos diferentes. Para que la comparación de resultados no se haga pesada, se escogerán instantes de tiempo específicos, de las predicciones por un lado, y del software COMSOL por otro, eligiendo los instantes temporales de la tabla 3.3. Además, para añadir un componente numérico a la comparación, para la predicción de cada red neuronal, se adjuntará el mayor de los errores obtenidos en las predicciones en cada discretización temporal con respecto al valor obtenido en el software. Este error máximo es denominado como RMSE.

Tiempo	Descripción
100 s	Instante tras 100 s de cocinado por una cara
200 s	Instante final de cocinado de una cara
250 s	Instante tras 50 s de cocinado por la otra cara
300 s	Instante final cocinado

Tabla 3.3: Instantes temporales escogidos para el modelo 3D

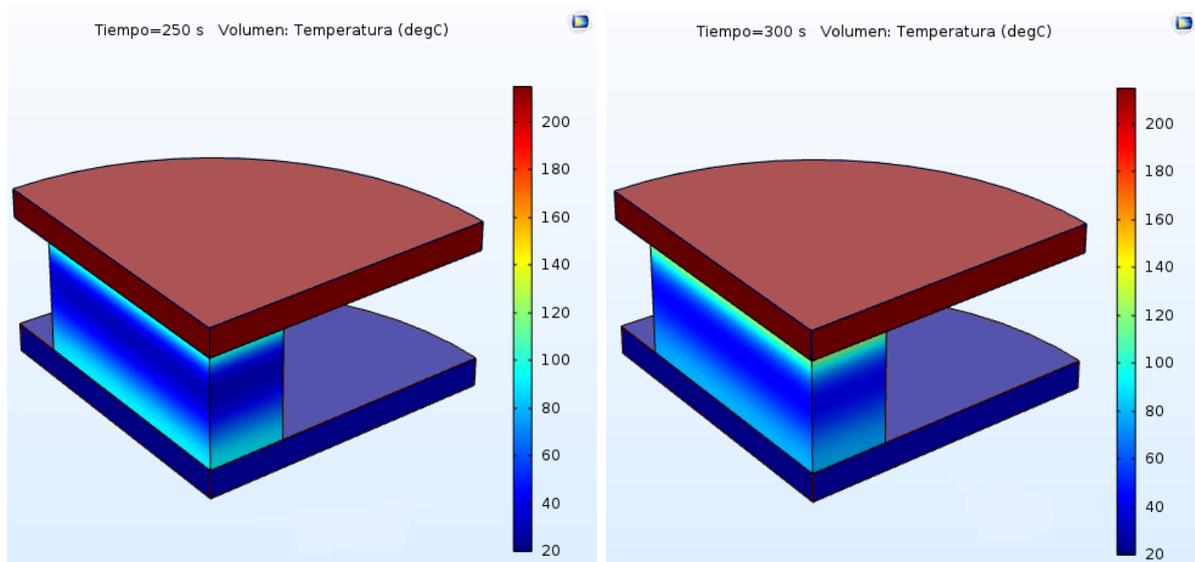
Antes de comparar las predicciones obtenidas con las diferentes redes neuronales creadas, es preciso indicar qué valores de las variables de entrada se han tomado para obtener la solución en el software COMSOL, que son los indicados en la tabla 2.8 de parámetros del modelo 3D. Estos valores serán los mismos, que se le suministrarán a cada red como variables de entrada, para que hagan su predicción.

De tal forma, que se obtienen los siguientes resultados en COMSOL para temperatura y concentración molar de agua.



(a) Temperatura a los 100 s

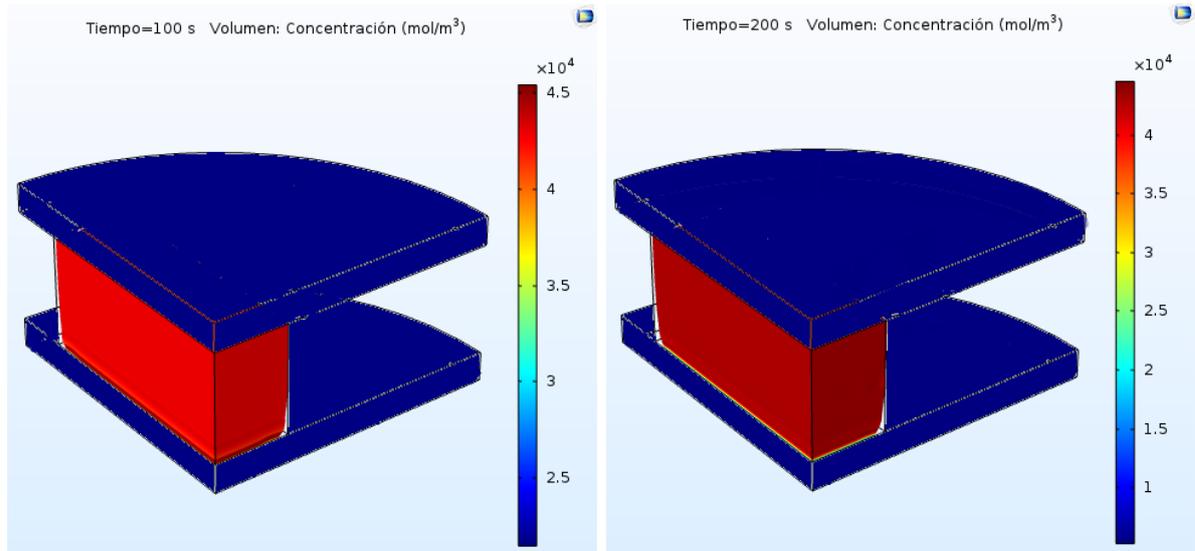
(b) Temperatura a los 200 s



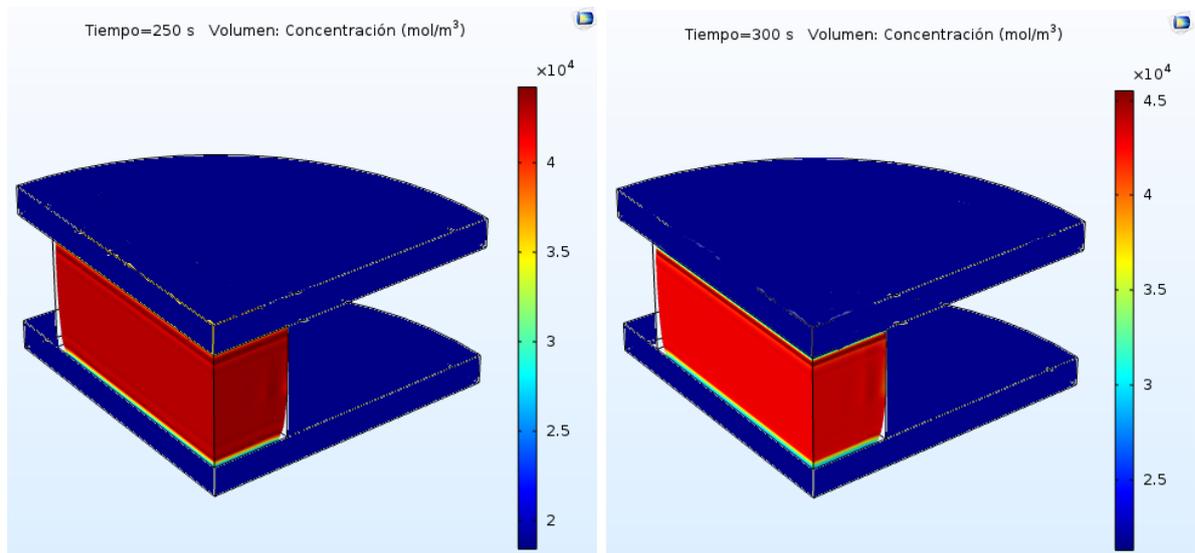
(a) Temperatura a los 250 s

(b) Temperatura a los 300 s

Figura 3.8: Resultados de la temperatura en COMSOL del modelo 3D.



(a) Concentración molar de agua a los 100 s    (b) Concentración molar de agua a los 200 s



(a) Concentración molar de agua a los 250 s    (b) Concentración molar de agua a los 300 s

Figura 3.10: Resultados de la concentración molar de agua en COMSOL del modelo 3D.

Para poder comparar las predicciones de las redes neuronales y comprobar su validez, pero que la extensión de la memoria no sea elevada. En las tablas 3.4 y 3.5, se pueden observar los distintos RMSE obtenidos con las predicciones de las redes neuronales. El proceso completo de comparación y validación de los resultados obteniendo el mapa de color de las predicciones de las redes neuronales se detalla en el anexo B.2.

Red neuronal	RMSE (°C)	RMSE (mol/m <sup>3</sup> )
Algoritmo L-M con 1 capa oculta	2.84	2487
Algoritmo L-M con 10 capas ocultas	4.39	1548
Algoritmo BR con 1 capa oculta	3.68	1952
Algoritmo BR con 2 capas ocultas	7.56	2156

Tabla 3.4: RMSE obtenidos con redes neuronales entrenadas con una población de 10 modelos.

Como conclusión con respecto a las predicciones con una población de entrenamiento de 10 modelos, se puede añadir que debido a este tamaño, el aumento del número de capas ocultas no significa una mayor precisión en la predicción de las redes neuronales. Aun así, los mayores errores obtenidos (RMSE) son bastante inferiores a un 10 %, por lo que las redes neuronales son válidas.

Red neuronal	RMSE (°C)	RMSE (mol/m <sup>3</sup> )
Algoritmo L-M con 1 capa oculta	0.24	257
Algoritmo L-M con 10 capas ocultas	0.17	79.84
Algoritmo BR con 1 capa oculta	0.19	178.56
Algoritmo BR con 2 capas ocultas	0.13	173.72

Tabla 3.5: RMSE obtenidos con redes neuronales entrenadas con una población de 100 modelos.

Como conclusión con respecto a las predicciones con una población de entrenamiento de 100 modelos, se puede añadir que a diferencia de lo ocurrido con un tamaño de población de 10 modelos, el aumento del número de capas ocultas significa una mayor precisión en la predicción de las redes neuronales, y por tanto un menor RMSE. Cabe destacar como la red neuronal con 1 capa oculta entrenada con BR tiene un menor RMSE que la entrenada con L-M. Aun así, ambos métodos de entrenamiento dan lugar a redes neuronales muy precisas y con un error bastante inferior al 1 % entre los valores proporcionados por COMSOL y la predicciones de las respectivas redes neuronales.

# Capítulo 4

## Conclusiones y líneas futuras

En este último capítulo, se exponen las conclusiones finales del presente Trabajo Fin de Grado, así como una serie de posibles líneas futuras a desarrollar a partir del mismo.

### 4.1. Conclusiones

Tal y como se ha comentado en el apartado 1.2. de esta memoria, el objetivo global de este proyecto consistía en desarrollar una metodología computacional con la capacidad de poder predecir el cocinado de un filete de carne en tiempo real. Es en este capítulo, donde cabe destacar que la metodología de las redes neuronales es apropiada para esta aplicación. Pero, para poder llegar a esta conclusión y afirmar la validez de las redes neuronales, se han realizado dos diferentes estudios.

Por un lado, se ha diseñado un modelo de elementos finitos en 1D. Aunque este modelo, no sea el que mejor represente el cocinado de carne en una sartén, principalmente por las simplificaciones realizadas, consistía en una buena toma de contacto, tras la resolución de ejemplos por separado mediante la metodología de redes neuronales. Como conclusión con respecto al modelo 1D, se puede añadir la elevada exactitud obtenida en las predicciones de las redes neuronales.

Por otro lado, se ha diseñado un modelo de elementos finitos en 3D, bastante más complejo que el 1D, y más adecuado a la realidad del cocinado de carne en sartén. Al ser un modelo más complejo, cabía la posibilidad de que las predicciones otorgadas por las redes neuronales no fuesen lo suficientemente precisas. No obstante, se ha podido comprobar, pese a la complejidad del modelo en 3D, el elevado potencial que tienen las redes, puesto que se han obtenido unas predicciones muy exactas con respecto al

modelo simulado en el software COMSOL Multiphysics 5.2a.

Un aspecto clave a reseñar, reside en las características de las redes neuronales. Una de ellas es el tamaño de la población de modelos, puesto que son los que sirven de entrenamiento a las redes. Cuanto mayor sea la población de entrenamiento de las redes, mejores resultados se obtienen en lo que respecta a exactitud. Una afirmación clave a destacar en esta conclusión. También, es necesario añadir, que cuanto mayor es la población de modelos, más tiempo le lleva a la red el proceso de entrenamiento. Además, la elección de las variables de entrada a la red tiene una elevada importancia, ya que conviene que sean parámetros cuya variación afecte de forma considerable a la predicción final, en este caso se ha hablado de temperatura y concentración molar de agua, relacionada con la pérdida de peso. Si se consigue elegir unas variables de entrada importantes, la red se ajustará más, que si las variables de entrada son triviales. Para cerrar este apartado, realizando una conclusión sobre las características de las redes neuronales, no hay que olvidarse del número de capas ocultas de la red. A mayor número de capas ocultas, la red es más compleja, por lo que su entrenamiento es más costoso y en teoría se obtienen mejores predicciones. Algo que se ha podido comprobar en este proyecto, está relacionado con el número de capas ocultas y el tamaño de la población de modelos. A lo largo de los diferentes resultados obtenidos, tanto para el modelo 1D como para el modelo 3D, se ha observado que cuando el tamaño de la población es reducido, un aumento del número de capas ocultas, no se traduce en una mejora en la precisión de las predicciones. Algo, desde luego, a tener en cuenta.

Por último, para cerrar esta conclusión es necesario destacar que este Trabajo Fin de Grado ha sido el primero en abordar el problema, como tal, de la simulación en tiempo real del cocinado de carne mediante el uso de redes neuronales. Lo cual, sienta las bases para el futuro, del desarrollo del cocinado autónomo mediante el uso de redes neuronales capaces de predecir el cocinado de carne en tiempo real.

## 4.2. Líneas futuras

Las futuras líneas de investigación propuestas, a partir de este Trabajo Fin de Grado son las siguientes:

- Inclusión en el estudio de más variables de entrada con las que llevar a cabo el entrenamiento de las redes neuronales.
- Mejora del cálculo y análisis del error en las predicciones obtenidas por las redes neuronales.
- Exploración de otros tipo de redes más complejas, con el objetivo de completar este estudio y obtener predicciones aún más acertadas.
- Estudio del cocinado de carne mediante otras variables, como puede ser el color como complemento a la temperatura o la pérdida de peso relativa.
- Investigación de posibles modelos de contacto entre sartén y filete, con el objetivo de conseguir un modelo más real.



# Capítulo 5

## Bibliografía

- José Ángel Bernad Pérez. Simulación computacional de la retracción que se produce en el cocinado de la carne mediante modelos multifísicos. Trabajo fin de grado, Universidad de Zaragoza, 2019.
- Ashish Dhall and Ashim K. Datta. Transport in deformable food materials: A poromechanics approach. *Chemical Engineering Science*, 2011.
- Y. Choi and M. R. Okos. Effects of temperature and composition on thermal properties of foods.. *Journal of Food Process and Applications*, 1986.
- RGM Van der Sman. Moisture transport during cooking of meat: An analysis based on flory–rehner theory. *Meat science*, 2007.
- L Vujosevic and VA Lubarda. Finite-strain thermoelasticity based on multiplicative decomposition of deformation gradient. *Theoretical and applied mechanics*, 2002.
- Krist V Gernaey Aberham Hailu Feyissa and Jens Adler-Nissen. 3d modelling of coupled mass and heat transfer of a convection-oven roasting process. *Meat science*, 2013.
- Jojo Moolayil. *Learn Keras for Deep Neural Networks*. Springer Science+Business Media New York, 2019.
- Charles Chinwuba. Timoshenko beam theory for the flexural analysis of moderately thick beams. *Department of Civil Engineering, Enugu State University of Science & Technology, Enugu State, Nigeria*, 2019.
- F. Rabeler and A.H. Feyissa. Kinetic modeling of texture and color changes during thermal treatment of chicken breast meat. *Food and Bioprocess Technology*, 2018b.

A. Salvador E. Chambers IV S. McGraw M. M. López, G. Hough and S. Fiszman. Beef's optimum internal cooking temperature as seen by consumers from different countries using survival analysis statistics. *Food Quality and Preference*, 2008.

Irene Pérez Salesa. Simulación numérica del cocinado de carne en plancha. Trabajo fin de grado, Universidad de Zaragoza, 2019.

# Lista de Figuras

2.1. Cocinado de carne en sartén. . . . .	6
2.2. Condiciones contorno en el cocinado de carne. . . . .	8
2.3. Esquema de la estructura del alimento en el que se muestran las distintas fases presentes y la nomenclatura para los respectivos volúmenes de cada una. . . . .	10
2.4. Esquema de las distintas tipologías de alimentos tras su cocinado y los fenómenos que lo rigen. Para el caso de la carne, la tipología corresponde a la remarcada en el cuadro azul. . . . .	12
2.5. Curva de retención de agua ajustada para la ternera obtenida mediante experimentación. . . . .	14
2.6. Descomposición multiplicativa del tensor gradiente de deformación, distinguiendo los efectos mecánicos y de la humedad. . . . .	17
2.7. Ajuste del módulo elástico en función de la temperatura $E(T)$ para la ternera. . . . .	21
2.8. Comportamiento del DL, ML y SM en función del tamaño de los datos de entrenamiento (Moolayil, 2019). . . . .	22
2.9. Diagrama de Venn de AI, ML y DL (Moolayil, 2019). . . . .	24
2.10. Estructura básica de una red neuronal (Moolayil, 2019). . . . .	25
2.11. Viga en voladizo de sección rectangular y con carga puntual en el extremo libre. . . . .	26
2.12. Error relativo de las predicciones de la flecha máxima de las redes neuronales. . . . .	28
2.13. Modelo 3D con transferencia de calor en COMSOL. . . . .	29
2.14. Error relativo de las predicciones de la temperatura en el centro del bloque de las redes neuronales. . . . .	31
2.15. Modelo 3D con transferencia de masa en COMSOL. . . . .	32
2.16. Error relativo de las predicciones de la pérdida de peso del bloque de las redes neuronales. . . . .	34

2.17. Geometría del modelo 1D. . . . .	36
2.18. Malla del modelo 1D. . . . .	38
2.19. Físicas empleadas en COMSOL para el modelo 3D. . . . .	39
2.20. Geometría del modelo 3D. . . . .	41
2.21. Malla del modelo 3D. . . . .	43
3.1. Población de 10 modelos usados como entrenamiento para las redes neuronales en el modelo 1D. . . . .	47
3.2. Población de 100 modelos usados como entrenamiento para las redes neuronales en el modelo 1D. . . . .	47
3.3. Población de 1000 modelos usados como entrenamiento para las redes neuronales en el modelo 1D. . . . .	47
3.4. Algoritmos de entrenamiento para las redes neuronales en el modelo 1D.	48
3.5. Resultado obtenido en COMSOL para el modelo 1D. . . . .	49
3.6. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Levenberg-Marquardt y una población de 100 modelos para el modelo 1D. . . . .	50
3.8. Resultados de la temperatura en COMSOL del modelo 3D. . . . .	54
3.10. Resultados de la concentración molar de agua en COMSOL del modelo 3D. . . . .	55
A.1. Viga voladizo de sección rectangular y con carga puntual en el extremo libre. . . . .	73
A.2. Código en MATLAB para la generación de muestras de entrenamiento.	74
A.3. Selección de las entradas y salidas de la red en la aplicación Neural Net Fitting en el ejemplo de Mecánica Estructural. . . . .	75
A.4. Selección del número de capas ocultas de la red en la aplicación Neural Net Fitting. . . . .	75
A.5. Selección del algoritmo de entrenamiento de la red en la aplicación Neural Net Fitting. . . . .	76
A.6. Valores de la flecha máxima para las diferentes redes neuronales entrenadas con 100 muestras. . . . .	77
A.7. Valores del error relativo de las predicciones para las diferentes redes neuronales entrenadas con 100 muestras. . . . .	77
A.8. Valores de la flecha máxima para las diferentes redes neuronales entrenadas con 1000 muestras. . . . .	78
A.9. Valores del error relativo de las predicciones para las diferentes redes neuronales entrenadas con 1000 muestras. . . . .	78

A.10.Valores de la flecha máxima para las diferentes redes neuronales entrenadas con 10000 muestras. . . . .	79
A.11.Valores del error relativo de las predicciones para las diferentes redes neuronales entrenadas con 10000 muestras. . . . .	79
A.12.Fuente de calor de contorno en COMSOL. . . . .	81
A.13.Flujo de calor convectivo en COMSOL. . . . .	81
A.14.Mallado del bloque 3D con Transferencia de Calor. . . . .	81
A.15.Comando MATLAB para la generación de una distribución normal en las variables de Transferencia de Calor. . . . .	82
A.16.Selección de las entradas y salidas de la red en la aplicación Neural Net Fitting en el ejemplo de Transferencia de Calor. . . . .	83
A.17.Población de 10 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Calor. . . . .	84
A.18.Respuesta temporal de la predicción de la temperatura en el centro del bloque de las redes neuronales entrenadas con 10 modelos. . . . .	84
A.19.Población de 100 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Calor. . . . .	85
A.20.Respuesta temporal de la predicción de la temperatura en el centro del bloque de las redes neuronales entrenadas con 100 modelos. . . . .	86
A.21.Población de 1000 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Calor. . . . .	87
A.22.Respuesta temporal de la predicción de la temperatura en el centro del bloque de las redes neuronales entrenadas con 1000 modelos. . . . .	87
A.23.Flujo de masa en COMSOL. . . . .	89
A.24.Comando MATLAB para la generación de una distribución normal en las variables de Transferencia de Masa. . . . .	89
A.25.Selección de las entradas y salidas de la red en la aplicación Neural Net Fitting en el ejemplo de Transferencia de Masa. . . . .	90
A.26.Población de 10 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Masa. . . . .	91
A.27.Respuesta temporal de la predicción de la pérdida de peso del bloque de las redes neuronales entrenadas con 10 modelos. . . . .	92
A.28.Población de 100 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Masa. . . . .	93
A.29.Respuesta temporal de la predicción de la pérdida de peso del bloque de las redes neuronales entrenadas con 100 modelos. . . . .	93

A.30. Población de 1000 modelos usados como entrenamiento para las redes neuronales en el ejemplo de Transferencia de Masa. . . . .	94
A.31. Respuesta temporal de la predicción de la pérdida de peso del bloque de las redes neuronales entrenadas con 1000 modelos. . . . .	94
B.1. Predicciones con algoritmo de entrenamiento Levenberg-Marquardt y con una población de 10 modelos para el modelo 1D. . . . .	97
B.2. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Levenberg-Marquardt y una población de 10 modelos para el modelo 1D. . . . .	98
B.3. Predicciones con algoritmo de entrenamiento Bayesian Regularization y con una población de 10 modelos para el modelo 1D. . . . .	99
B.4. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Bayesian Regularization y una población de 10 modelos para el modelo 1D. . . . .	99
B.5. Predicciones con algoritmo de entrenamiento Levenberg-Marquardt y con una población de 100 modelos para el modelo 1D. . . . .	100
B.6. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Levenberg-Marquardt y una población de 100 modelos para el modelo 1D. . . . .	101
B.7. Predicciones con algoritmo de entrenamiento Bayesian Regularization y con una población de 100 modelos para el modelo 1D. . . . .	102
B.8. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Bayesian Regularization y una población de 100 modelos para el modelo 1D. . . . .	102
B.9. Predicciones con algoritmo de entrenamiento Levenberg-Marquardt y con una población de 1000 modelos para el modelo 1D. . . . .	103
B.10. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Levenberg-Marquardt y una población de 1000 modelos para el modelo 1D. . . . .	104
B.11. Predicciones con algoritmo de entrenamiento Bayesian Regularization y con una población de 1000 modelos para el modelo 1D. . . . .	105
B.12. Comparación del valor obtenido en COMSOL con predicciones con algoritmo de entrenamiento Bayesian Regularization y una población de 1000 modelos para el modelo 1D. . . . .	105
B.14. Resultados de la predicción de la temperatura de la red neuronal con 1 capa oculta entrenada con L-M y con una población de 10 modelos. . .	106

B.16. Resultados de la predicción de la concentración molar de agua de la red neuronal con 1 capa oculta entrenada con L-M y con una población de 10 modelos. . . . .	107
B.18. Resultados de la predicción de la temperatura de la red neuronal con 10 capas ocultas entrenada con L-M y con una población de 10 modelos. .	108
B.20. Resultados de la predicción de la concentración molar de agua de la red neuronal con 10 capas ocultas entrenada con L-M y con una población de 10 modelos. . . . .	108
B.22. Resultados de la predicción de la temperatura de la red neuronal con 1 capa oculta entrenada con BR y con una población de 10 modelos. . . .	109
B.24. Resultados de la predicción de la concentración molar de agua de la red neuronal con 1 capa oculta entrenada con BR y con una población de 10 modelos. . . . .	110
B.26. Resultados de la predicción de la temperatura de la red neuronal con 2 capas ocultas entrenada con BR y con una población de 10 modelos. . .	110
B.28. Resultados de la predicción de la concentración molar de agua de la red neuronal con 2 capas ocultas entrenada con BR y con una población de 10 modelos. . . . .	111
B.30. Resultados de la predicción de la temperatura de la red neuronal con 1 capa oculta entrenada con L-M y con una población de 100 modelos. .	112
B.32. Resultados de la predicción de la concentración molar de agua de la red neuronal con 1 capa oculta entrenada con L-M y con una población de 100 modelos. . . . .	113
B.34. Resultados de la predicción de la temperatura de la red neuronal con 10 capas ocultas entrenada con L-M y con una población de 100 modelos.	114
B.36. Resultados de la predicción de la concentración molar de agua de la red neuronal con 10 capas ocultas entrenada con L-M y con una población de 100 modelos. . . . .	114
B.38. Resultados de la predicción de la temperatura de la red neuronal con 1 capa oculta entrenada con BR y con una población de 100 modelos. . .	115
B.40. Resultados de la predicción de la concentración molar de agua de la red neuronal con 1 capa oculta entrenada con BR y con una población de 100 modelos. . . . .	116
B.42. Resultados de la predicción de la temperatura de la red neuronal con 2 capas ocultas entrenada con BR y con una población de 100 modelos. .	117

B.44. Resultados de la predicción de la concentración molar de agua de la red neuronal con 2 capas ocultas entrenada con BR y con una población de 100 modelos. . . . . 117

# Lista de Tablas

2.1. Parámetros ejemplo de Mecánica Estructural. . . . .	27
2.2. Variables de entrada a la red neuronal en Mecánica Estructural. . . . .	27
2.3. Parámetros ejemplo de Transferencia de Calor. . . . .	30
2.4. Variables de entrada a la red neuronal en Transferencia de Calor. . . . .	30
2.5. Parámetros ejemplo de Transferencia de Masa. . . . .	32
2.6. Variables de entrada a la red neuronal en Transferencia de Masa. . . . .	33
2.7. Parámetros y variables del modelo 1D. . . . .	36
2.8. Parámetros del modelo 3D. . . . .	40
2.9. Variables del modelo 3D. . . . .	41
3.1. Variables de entrada a la redes neuronales en el modelo 1D. . . . .	46
3.2. Variables de entrada a las redes neuronales en el modelo 3D. . . . .	51
3.3. Instantes temporales escogidos para el modelo 3D . . . . .	53
3.4. RMSE obtenidos con redes neuronales entrenadas con una población de 10 modelos. . . . .	56
3.5. RMSE obtenidos con redes neuronales entrenadas con una población de 100 modelos. . . . .	56
A.1. Valores de las variables de entrada a las redes neuronales en Mecánica Estructural. . . . .	76
A.2. Media y desviación típica de las variables entrada a la red neuronal en Transferencia de Calor. . . . .	82
A.3. Valores de las variables de entrada a las redes neuronales en Transferencia de Calor. . . . .	83
A.4. Valores de las predicciones de la temperatura en el centro del bloque en t = 24000 s para las redes neuronales entrenadas con 10 modelos. . . . .	84
A.5. Valores de las predicciones de la temperatura en el centro del bloque en t = 24000 s para las redes neuronales entrenadas con 100 modelos. . . . .	86

A.6. Valores de las predicciones de la temperatura en el centro del bloque en $t = 24000$ s para las redes neuronales entrenadas con 1000 modelos. . . . .	87
A.7. Media y desviación típica de las variables entrada a la red neuronal en Transferencia de Masa. . . . .	89
A.8. Valores de las variables de entrada a las redes neuronales en Transferencia de Masa. . . . .	91
A.9. Valores de las predicciones de la pérdida de peso del bloque en $t = 240000$ s para las redes neuronales entrenadas con 10 modelos. . . . .	92
A.10. Valores de las predicciones de la pérdida de peso del bloque en $t = 240000$ s para las redes neuronales entrenadas con 100 modelos. . . . .	93
A.11. Valores de las predicciones de la pérdida de peso del bloque en $t = 240000$ s para las redes neuronales entrenadas con 1000 modelos. . . . .	94