

Un algoritmo numérico para localización de tumores

Tesis presentada por Juan Pablo Pereyra para el grado de Licenciado en Ciencias de la Computación en la Universidad Nacional de Córdoba



FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y COMPUTACIÓN

Universidad Nacional de Córdoba

Directores:
Agnelli, Juan Pablo y Luque, Franco Martín



Esta obra se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 2.5 Argentina](https://creativecommons.org/licenses/by-nc-sa/2.5/arg/).

*Dedicado a
familiares y amigos.*

Agradecimientos

Ya que este trabajo representa el final de la carrera, aprovecho para agradecer por la ayuda muy importante durante la carrera a mis padres, a mis familiares, a mis amigos, a los compañeros de aula durante la carrera, a los profesores que tuve durante la carrera tanto de la UNC como de la UNS y de la UNLP, a la institución FaMAF, finalmente también a mis directores de tesis, a todos por la confianza en mis decisiones, la paciencia y la perseverancia.

Resumen

El objetivo de este trabajo es desarrollar un algoritmo numérico capaz de estimar la localización y tamaño de un tumor inmerso en el tejido corporal utilizando como información perfiles de temperatura medidos sobre la superficie corporal. Desde el punto de vista matemático este tipo de problema se enmarca en el área de problemas inversos.

El problema es abordado en dos etapas. En una primera instancia, se modela la transferencia de calor en tejidos vivos por medio de ecuaciones diferenciales en derivadas parciales. Una vez definido el problema de borde o problema directo, se implementa y aplica el método de elementos finitos para obtener una solución numérica aproximada al problema directo.

Luego, en una segunda etapa, se define de manera precisa el problema inverso y se lo plantea como un problema de optimización. Para hallar la solución a este problema de optimización proponemos un algoritmo numérico basado en el método del gradiente.

Los resultados obtenidos en los diferentes casos de estudio, reflejan un gran desempeño de la estrategia numérica propuesta. La implementación fue realizada en el lenguaje Python con el paquete para computación científica llamado numpy.

Palabras claves: Método de los elementos finitos, Ecuación del calor, Método del gradiente, Localización de tumores.

Abstract

The goal of this work is the development of a numerical algorithm to estimate the location and size of an embedded (melanoma) tumor using temperature profiles measured on the skin surface. From a mathematical point of view this type of problem is framed in the area of inverse problems.

The problem is addressed in two stages. In the first instance, heat transfer in living tissue is modeled by partial differential equations. Once the boundary problem or direct problem is defined, the finite element method is implemented and applied in order to obtain an approximate numerical solution.

Then, in a second stage, the inverse problem is stated in precise way and is also reformulated as an optimization problem. To find the solution to this optimization problem we propose a numerical algorithm based on the gradient method.

The results obtained in the different case studies reflect a strong performance of the proposed numerical strategy. The implementation was done in the Python language with scientific computing package called numpy.

Keywords: finite element method, heat equation, gradient method, localization of tumors.

Índice general

Agradecimientos	III
Resumen	V
1. Introducción	1
2. Transferencia de calor en tejidos vivos	3
2.1. Derivación de la ecuación del calor	3
2.2. Ecuación del calor en tejidos vivos	7
2.3. Modelo matemático	8
2.3.1. Condiciones de contorno sobre la región de estudio o dominio	8
3. Resolución numérica vía el MEF	11
3.1. Método de elementos finitos	11
3.1.1. Triangulación	12
3.1.2. El problema como un sistema de ecuaciones lineales	13
3.1.3. Ensamble de la matriz A y el vector f	14
3.2. MEF aplicado al problema (2.22) (PD)	17
4. Resolución del problema inverso	19
4.1. Qué es un problema inverso (PI)	19
4.2. Definición de nuestro PI	19
4.3. El PI como problema de optimización	20
4.3.1. Minimización y método del gradiente	20
4.4. Un algoritmo para la solución	22
5. Implementación y casos de estudio	23
5.1. Arquitectura y diseño del software	23
5.2. Descripción de los módulos	23
5.2.1. Módulo principal (elementosF.py)	24
5.2.2. Cálculo de los coeficientes (coeficientes.py)	25
5.2.3. Cálculo de la temperatura (problema_directo.py)	26
5.2.4. Cálculo de J (costo.py)	27
5.2.5. Cálculo de ∇J (derivada_de_forma.py)	28
5.2.6. Triangulación (generar_malla_rectangular.py)	28
5.2.7. Estructura del paquete de software realizado	29
5.3. Herramientas y librerías externas	29
5.4. Paralelización	29
5.5. Casos de estudio y resultados	30

5.5.1. Resultados en cuanto al PD	30
5.5.2. Resultados en cuanto al PI (localización de tumores)	30
6. Conclusiones y trabajo futuro	37
6.1. Conclusiones	37
6.2. Trabajo futuro	38

Capítulo 1

Introducción

La relación entre la temperatura del cuerpo humano y la aparición de alguna enfermedad ha sido estudiada desde hace años. A grandes rasgos se puede decir que la temperatura corporal y superficial de una persona está principalmente controlada por la circulación sanguínea, el metabolismo local y el intercambio que se da entre la superficie corporal y el medio ambiente en el que se encuentra la persona [1, 2, 3]. Cualquier anormalidad que exista dentro del organismo humano que implique algún tipo de cambio en alguno de estos parámetros puede producir variaciones en la temperatura y en el flujo de calor existente en la superficie corporal.

La particular estructura interna que poseen los tumores y el proceso de angiogénesis originan un incremento de la temperatura. Inflamación, aumento de la actividad metabólica, anormal morfología de nuevos vasos sanguíneos y falta de respuesta a señales de homeostasis son algunas de las características que hacen que un tumor se comporte de manera anormal con respecto al tejido sano en términos de producción y disipación de calor.

Se ha determinado que la presencia de un tumor de mamas o melanoma de piel (tumor maligno que deriva de la transformación o proliferación de los melanocitos y que se encuentra predominantemente en la piel), produce un incremento de la temperatura en las zonas que lo rodean y en la superficie corporal próxima a este [4, 5, 6, 7].

En resumen, la detección de un perfil de temperatura anormal sobre la piel puede indicar la presencia interna de un tumor y esta información puede ser utilizada para predecir la localización y tamaño del tumor, también para determinar valores de parámetros térmicos asociados a la región tumoral.

El objetivo de este trabajo es desarrollar una metodología (algoritmo numérico) para estimar la localización y tamaño de un tumor inmerso en el tejido corporal usando como información perfiles de temperatura.

Para lograr este objetivo el trabajo fue dividido en dos etapas. Una primera instancia, la cual denominamos problema directo, consistió en modelar el problema de transferencia del calor en tejidos vivos mediante ecuaciones en derivadas parciales. Este problema directo luego es resuelto de manera numérica utilizando el método de elementos finitos. Cabe señalar que el uso de este método numérico es una novedad para la resolución de este problema.

La segunda etapa, la cual llamamos el problema inverso consiste en la localización del tumor. Para esto definimos un funcional costo que representa el

error entre la temperatura experimental dada y la solución numérica del problema directo suponiendo una localización aproximada del tumor. El mínimo del funcional costo se da justamente cuando ambas temperaturas coinciden y por lo tanto habremos hallado la posición y tamaño del tumor. Es decir, planteamos el problema inverso como un problema de optimización. Para minimizar el funcional costo usamos el método del máximo descenso, el cual implica calcular el gradiente del funcional costo con respecto a los parámetros geométricos asociados al tumor.

A continuación se exponen brevemente los contenidos de este trabajo.

En el capítulo 2 el objetivo es plantear nuestro modelo matemático, el cual es un modelo de la transferencia de calor en tejidos vivos. Para ver cómo derivamos este modelo comenzamos por cómo se deriva la ecuación de calor en un sólido y luego introducimos la ecuación propuesta por Pennes en el año 1948 [8], la cual es una variante de la ecuación de transferencia de calor pero específica para tejidos vivos.

En el capítulo 3 se introduce brevemente el Método de Elementos Finitos (FEM), método numérico para la solución de ecuaciones diferenciales. Primero a modo introductorio aplicaremos el método a un ejemplo simple y luego se especificará cómo aplicar el método a nuestro caso de estudio.

En el capítulo 4 se trata en concreto el problema de localización del tumor. Primero se introduce el concepto de problema inverso (P.I.) y se plantea el problema de la localización como un P.I. al que luego se lo planteará como un problema de optimización. Finalmente se presenta un algoritmo para la resolución.

En el capítulo 5 se describe el paquete de software realizado para estimar la localización y radio de tumores. También se describen las bibliotecas externas usadas como numpy, el lenguaje que utilizamos para programar (Python) y mostramos algunos casos de estudio con sus resultados.

Finalmente en el capítulo 6 se exponen las conclusiones del trabajo, las pruebas realizadas al software y una sección de trabajo futuro.

Capítulo 2

Modelado de la transferencia de calor en tejidos vivos

El objetivo de este capítulo es presentar el modelo matemático propuesto para modelar la transferencia de calor en tejidos vivos.

En la primer sección, a modo introductorio veremos la derivación de la ecuación de calor en un sólido. Luego, en la segunda sección presentaremos la ecuación de calor en tejidos vivos. Finalmente en la tercer sección se describe y plantea el modelo matemático con el cual trabajaremos.

2.1. Derivación de la ecuación del calor en una barra unidimensional

Como dijimos, en esta primer sección del capítulo derivaremos la ecuación de calor en un sólido, una barra unidimensional para ser más específico. Este es un simple ejemplo para luego ver y entender la ecuación de calor en tejidos vivos. La ecuación de calor es muy importante en este trabajo ya que para la localización de un tumor usaremos como información un perfil de temperatura medido sobre la piel, por lo tanto necesitamos conocer cómo el calor se conduce en el tejido tanto en la zona del tumor como en el tejido sano.

El ejemplo y la información en esta sección está desarrollada en base a la referencia [9].

Consideremos una barra de longitud L , orientada en la dirección x (desde $x = 0$ hasta $x = L$, ver Figura 2.1). Sea A una sección transversal de la barra, la cual tiene área constante y además propiedades térmicas constantes; con esto último suponemos que la barra es unidimensional. Nuestro objetivo es conocer la temperatura para cada punto de la barra y en cada instante de tiempo, es decir la distribución de temperatura respecto del tiempo t y del espacio x . Tengamos en cuenta que la dependencia respecto de x y t corresponde a una situación en la que la barra no se calienta de manera uniforme, es decir la densidad de energía térmica varía de una sección transversal a otra.

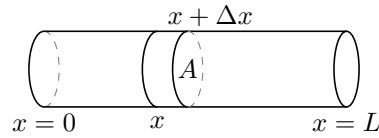


Figura 2.1: Barra unidimensional de longitud L con sección transversal o rebanada de longitud Δx .

A continuación explicamos algunos conceptos físicos que nos serán de utilidad para derivar la ecuación del calor.

Densidad de energía térmica

Introducimos temporalmente la cantidad de energía térmica por unidad de volumen como una variable desconocida y la llamamos densidad de energía térmica:

$$\boxed{e(x, t) \equiv \text{densidad de energía térmica}} \tag{2.1}$$

Energía térmica

Consideremos una rebanada delgada de la barra, contenida entre x y $x + \Delta x$. Si la densidad de energía térmica es constante en todo el volumen, entonces la energía total en la rebanada es el producto de la densidad de energía térmica y el volumen. En general, la densidad de energía no es constante. Sin embargo, si Δx es sumamente pequeño, entonces $e(x, t)$ se puede aproximar como una constante en todo el volumen de manera que:

$$\boxed{\text{energía térmica} = e(x, t)A\Delta x} \tag{2.2}$$

ya que el volumen de una rebanada es $A\Delta x$.

Conservación de la energía de calor

Los cambios de energía térmica en el tiempo entre x y $x + \Delta x$ sólo se deben a la energía térmica que fluye a través de los bordes (x y $x + \Delta x$) y a la energía térmica generada dentro (debido a fuentes positivas o negativas de energía térmica). El proceso fundamental de flujo de calor se describe por la siguiente ecuación de palabras:

tasa de cambio de la energía térmica = en el tiempo	energía térmica que fluye a través de los bordes por unidad de tiempo	+	energía térmica generada en el in- terior por unidad de tiempo	(2.3)
---	---	---	--	-------

Flujo de calor y fuentes de calor

Estos dos conceptos corresponden a los términos de la derecha de la ecuación de palabras (2.3). Para representar el flujo de energía térmica en la barra

unidimensional utilizamos el flujo de calor.

$$\boxed{\Phi(x, t) = \begin{array}{l} \text{flujo de calor (la cantidad de energía} \\ \text{t\u00e9rmica por unidad de tiempo que fluye} \\ \text{hacia la derecha por unidad de superficie)} \end{array}} \quad (2.4)$$

Si $\Phi(x, t) < 0$ significa que la energ\u00eda t\u00e9rmica est\u00e1 fluyendo hacia la izquierda. Por el contrario $\Phi(x, t) > 0$ significa que la energ\u00eda t\u00e9rmica est\u00e1 fluyendo hacia la derecha. Entonces, el flujo total que se tiene en una rebanada de longitud Δx , est\u00e1 dado por:

$$\Phi(x, t)A - \Phi(x + \Delta x, t)A.$$

Tambi\u00e9n permitimos fuentes internas de energ\u00eda t\u00e9rmica (fuentes de calor):

$$\boxed{q(x, t) = \begin{array}{l} \text{energ\u00eda t\u00e9rmica por unidad de volumen} \\ \text{generada por unidad de tiempo} \end{array}} \quad (2.5)$$

Podemos asumir que $q(x, t)$ es aproximadamente constante en el espacio en una rebanada delgada, y por lo tanto la energ\u00eda t\u00e9rmica total generada por unidad de tiempo en el corte o rebanada es de aproximadamente $q(x, t)A\Delta x$.

La velocidad de cambio de la energ\u00eda de calor es debido a la energ\u00eda t\u00e9rmica que fluye a trav\u00e9s de los l\u00edmites y de las fuentes internas.

Luego, restringi\u00e9ndonos a una rebanada de longitud Δx , de acuerdo a la ecuaci\u00f3n (2.3) y utilizando los t\u00e9rminos (2.4) y (2.5), obtenemos la aproximaci\u00f3n:

$$\frac{\partial}{\partial t} (e(x, t)A\Delta x) \approx \Phi(x, t)A - \Phi(x + \Delta x, t)A + q(x, t)A\Delta x \quad (2.6)$$

esta ecuaci\u00f3n se vuelve cada vez m\u00e1s precisa con Δx tendiendo a cero. Si, ahora dividimos la ecuaci\u00f3n (2.6) por $A\Delta x$ obtenemos:

$$\frac{\partial}{\partial t} (e(x, t)) \approx \frac{\Phi(x, t) - \Phi(x + \Delta x, t)}{\Delta x} + q(x, t) \quad (2.7)$$

Finalmente, tomando l\u00edmite con ($\Delta x \rightarrow 0$) en (2.7), obtenemos la siguiente ecuaci\u00f3n de conservaci\u00f3n de energ\u00eda:

$$\frac{\partial e}{\partial t} = \lim_{\Delta x \rightarrow 0} \frac{\Phi(x, t) - \Phi(x + \Delta x, t)}{\Delta x} + q(x, t), \quad (2.8)$$

En este proceso de l\u00edmite, t se mantiene fijo, por lo tanto obtenemos la siguiente ecuaci\u00f3n de conservaci\u00f3n de energ\u00eda:

$$\boxed{\frac{\partial e}{\partial t} = -\frac{\partial \Phi}{\partial x} + q} \quad (2.9)$$

Temperatura y el calor espec\u00edfico

Usualmente describimos a los materiales por su temperatura,

$$\boxed{u(x, t) = \text{temperatura}} \quad (2.10)$$

no por su densidad térmica. Distinguir entre los conceptos de temperatura y energía térmica no es necesariamente una tarea trivial. Sólo alrededor de 1700 se descubrió que para dos materiales diferentes puede tomar diferentes cantidades de energía térmica elevar de una temperatura a otra temperatura mayor [9]. Esto exige la introducción del calor específico (o capacidad calorífica).

$$\boxed{\begin{array}{l} \text{calor específico (la energía térmica que debe ser su-} \\ c = \text{ministrada a una unidad de masa de una sustancia} \\ \text{para elevar su temperatura una unidad)} \end{array}} \quad (2.11)$$

Dado que el calor específico es independiente de la temperatura, la energía térmica por unidad de masa es sólo $c(x)u(x, t)$, y para poder calcular la energía térmica con estos datos introducimos la densidad de masa:

$$\boxed{\rho(x) = \text{densidad de masa (masa por unidad de volumen)}} \quad (2.12)$$

La masa total de una rebanada de sección transversal es $\rho A \Delta x$. Por lo tanto la energía térmica total es $c(x)u(x, t) \cdot \rho A \Delta x$, entonces

$$e(x, t) A \Delta x = c(x)u(x, t) \cdot \rho A \Delta x \quad (2.13)$$

De este modo hemos explicado la relación básica entre la energía térmica y la temperatura. Dicho de otra forma ya podemos ver qué es la energía térmica:

$$\boxed{e(x, t) = c(x)\rho(x)u(x, t)} \quad (2.14)$$

Luego reemplazando en la ecuación de conservación de la energía 2.9 la densidad de energía térmica usando la ecuación anterior 2.14 obtenemos:

$$\boxed{c(x)\rho(x) \frac{\partial u}{\partial t} = -\frac{\partial \Phi}{\partial x} + q} \quad (2.15)$$

La ley de Fourier de conducción de calor

La ley de la conducción de calor, también conocida como ley de Fourier, establece que el flujo de calor a través de un material es proporcional al negativo del gradiente de la temperatura. Esto es:

$$\boxed{\Phi = -\sigma \frac{\partial u}{\partial x}} \quad (2.16)$$

donde σ es el coeficiente de conductividad térmica.

La conductividad térmica es una propiedad física de los materiales que mide la capacidad de conducción de calor, es decir de transferir la energía cinética de sus moléculas a otras adyacentes.

Ecuación del calor

A partir de las dos ecuaciones anteriores (2.15) y (2.16) obtenemos una ecuación en derivadas parciales:

$$\boxed{c\rho \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(\sigma \frac{\partial u}{\partial x} \right) + q} \quad (2.17)$$

Como nosotros tomaremos como información perfiles de temperatura correspondientes a imágenes termográficas, no consideramos variación temporal. Esto puede pensarse como si el sistema evolucionó una cierta cantidad de tiempo de manera tal que ya no hay un cambio de temperatura temporal sino sólo espacial. En este caso la ecuación de calor estacionaria es:

$$-\frac{\partial}{\partial x} \left(\sigma \frac{\partial u}{\partial x} \right) = q \quad (2.18)$$

Para el caso de dos o tres dimensiones espaciales la derivación de la ecuación del calor es similar, obteniendo:

$$-div(\sigma \nabla u) = q \quad (2.19)$$

donde $\nabla u = \left(\frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n} \right)$ es el gradiente de u y div denota la divergencia de un campo vectorial, la cual está definida por $div(F) = \frac{\partial F}{\partial x_1} + \dots + \frac{\partial F}{\partial x_n}$ para $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

2.2. Ecuación del calor en tejidos vivos: ecuación de Pennes

En el trabajo de Harry Pennes “Analysis of tissue and arterial blood temperatures in the resting human forearm”[8], publicado en 1948, se analiza la distribución de temperatura en el cuerpo humano y en este trabajo se desarrolló un modelo matemático muy bueno para describir la transferencia de calor en tejidos humanos, el cual se usó en adelante como referencia para trabajos que involucran esta problemática.

La principal contribución que realizó Pennes fue considerar la tasa de transferencia de calor existente entre la sangre y el tejido corporal como proporcional al producto de la tasa de perfusión por unidad de volumen de tejido y la diferencia entre la temperatura arterial y la temperatura del tejido local, o sea:

$$h_b = G_b \rho_b c_b (T_b - u) \quad (2.20)$$

donde h_b representa la tasa de transferencia de calor por unidad de volumen de tejido, G_b es la tasa de perfusión por unidad de volumen de tejido, ρ_b es la densidad de la sangre, c_b es el calor específico de la sangre, T_b es la temperatura de sangre en arteria y u es la temperatura local del tejido.

Fisiológicamente, la perfusión es el mecanismo por el cual las células, y por tanto los órganos, son alimentados en oxígeno y en nutrientes, es la sangre la que vehicula todos estos elementos esenciales.

La ecuación propuesta incluye un término que representa la transferencia de calor por conducción a través del tejido y un término que representa la

generación de calor metabólico. Estos dos términos sumados al término que representa la transferencia de calor debida a la circulación de sangre dan lugar a la siguiente ecuación de Pennes estacionaria:

$$- \operatorname{div}(\sigma(x)\nabla u(x)) + k(x)(u(x) - T_b) = q(x), \quad x \in \Omega \subset \mathbb{R}^n, \quad n = 2, 3. \quad (2.21)$$

donde σ es el coeficiente de conductividad térmica, $k = G_b \rho_b c_b$ es el coeficiente de perfusión, q es la fuente de calor metabólico y T_b es la temperatura de la sangre que suponemos de valor constante.

2.3. Modelo matemático

Teniendo en cuenta el hecho de que los valores correspondientes a la conductividad, la perfusión y la actividad metabólica en la región tumoral son significativamente mayores a los valores encontrados en tejido sano, vamos a modelar estos coeficientes como funciones continuas a trozos. Por ejemplo, el coeficiente de conductividad térmica lo definimos como:

$$\sigma(x) = \begin{cases} \sigma_1, & \text{si } x \in \Omega - \bar{\omega} \\ \sigma_0, & \text{si } x \in \bar{\omega} \end{cases} \quad (2.22)$$

donde $\bar{\omega}$ representa la región tumoral y $\Omega - \bar{\omega}$ el tejido sano. Así la ecuación de Pennes en nuestro caso queda:

$$\begin{cases} -\sigma_1 \Delta u_1 + k_1 u_1 = q_1 + k_1 T_b, & \text{en } \Omega - \bar{\omega} \\ -\sigma_0 \Delta u_0 + k_0 u_0 = q_0 + k_0 T_b, & \text{en } \bar{\omega} \end{cases} \quad (2.23a)$$

$$(2.23b)$$

donde hemos utilizado el hecho que para σ_i constante vale:

$$- \operatorname{div}(\sigma_i \nabla u) = -\sigma_i \operatorname{div}(\nabla u) = -\sigma_i \sum_{j=1}^n \frac{\partial u}{\partial x_j} = -\sigma_i \Delta u. \quad (2.24)$$

2.3.1. Condiciones de contorno sobre la región de estudio o dominio

Estas condiciones varían en los diferentes bordes del dominio de manera de describir en forma aproximada la realidad del fenómeno que estamos considerando. En el borde inferior, el cual denotamos como Γ_b , suponemos una condición Dirichlet no homogénea:

$$u(x) = T_b, \quad x \in \Gamma_b \quad (2.25)$$

Esto se debe a que este borde se encuentra inmerso en el cuerpo y por lo tanto suponemos que la temperatura corporal interior es constante, más precisamente de $37^\circ C$.

En los laterales del dominio, la condición de borde considerada es de tipo Neumann homogénea o de flujo cero, es decir:

$$-\sigma(x) \frac{\partial u(x)}{\partial n} = 0, \quad x \in \Gamma_l \quad (2.26)$$

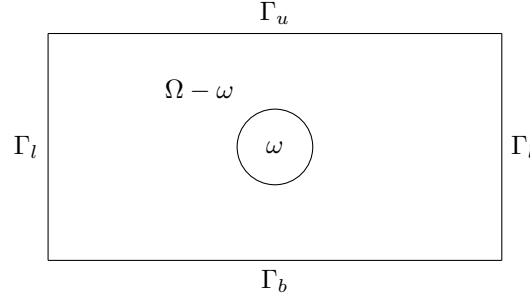


Figura 2.2: Gráfico del dominio donde se señalan los nombres que usaremos para el contorno.

donde Γ_l denota los lados laterales del dominio para $n = 2$ y las caras laterales en el caso $n = 3$. El motivo para considerar condiciones adiabáticas en los laterales se debe a que suponemos que en lugares lejos del centro del dominio, o más precisamente alejados del centro de la región tumoral, la temperatura no es afectada por ninguna fuente de calor existente dentro o fuera del dominio, por lo cual no existe un flujo de intercambio de calor.

Por último, en el borde superior del dominio, que representa la superficie corporal y denotamos por Γ_u suponemos la siguiente condición convectiva o también conocida como Ley de enfriamiento de Newton:

$$-\sigma(x) \frac{\partial u(x)}{\partial n} = \alpha(u(x) - T_a), \quad x \in \Gamma_u \quad (2.27)$$

donde α es el coeficiente de transferencia de calor y T_a es la temperatura ambiente. Esta ecuación simula el intercambio de calor que se da entre el cuerpo y el medio ambiente.

Teniendo en cuenta todas las condiciones anteriores, el problema de borde que consideraremos para modelar la transferencia de calor en tejidos vivos es el siguiente:

$$\left\{ \begin{array}{ll} -\sigma_1 \Delta u_1 + k_1 u_1 = q_1 + k_1 T_b, & \text{en } \Omega - \bar{\omega} \quad (2.28a) \\ -\sigma_0 \Delta u_0 + k_0 u_0 = q_0 + k_0 T_b, & \text{en } \bar{\omega} \quad (2.28b) \\ u_1 = u_0, & \text{sobre } \partial\omega \quad (2.28c) \\ -\sigma_1 \frac{\partial u_1}{\partial n} = -\sigma_0 \frac{\partial u_0}{\partial n}, & \text{sobre } \partial\omega \quad (2.28d) \\ -\sigma_1 \frac{\partial u_1}{\partial n} = \alpha(u_1 - T_a), & \text{sobre } \Gamma_u \quad (2.28e) \\ -\sigma_1 \frac{\partial u_1}{\partial n} = 0, & \text{sobre } \Gamma_l \quad (2.28f) \\ u_1 = T_b, & \text{sobre } \Gamma_b \quad (2.28g) \end{array} \right.$$

A este problema de borde lo llamaremos a lo largo de este trabajo el problema directo (PD). Es sabido que este problema admite soluciones (ver [10]), no obstante no se conoce una expresión analítica de su solución. En el capítulo siguiente

introduciremos un método numérico para obtener una solución aproximada del (PD).

Capítulo 3

Resolución numérica vía el MEF

El objetivo de este capítulo es presentar el método de elementos finitos (MEF), método numérico que usaremos para resolver nuestro problema directo presentado en el capítulo anterior. A modo introductorio primero aplicaremos el método a un simple ejemplo y luego se especificará cómo aplicarlo a nuestro caso de estudio.

3.1. Método de elementos finitos

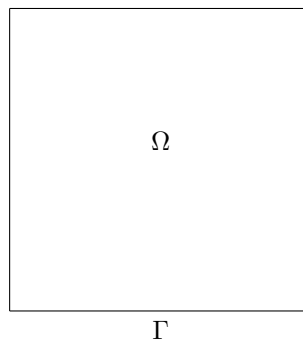
El método de elementos finitos (FEM) es un método numérico para la solución de ecuaciones diferenciales, principalmente ecuaciones en derivadas parciales. Fue introducido por ingenieros en la década del 60 y se ha convertido en uno de los métodos numéricos más utilizados para la resolución de ecuaciones diferenciales de tipo elípticas y parabólicas.

El método se basa en la formulación débil o variacional del problema de borde. Luego en lugar de buscar la solución en un espacio de dimensión infinita se consideran subespacios de dimensión finita y es allí donde se buscará la solución aproximada.

Para entender cómo funciona el FEM veamos el siguiente ejemplo. Consideremos el problema

$$\begin{cases} -\operatorname{div}(\sigma \nabla u) + ku = q & \text{en } \Omega, & (3.1a) \\ u = 0 & \text{sobre } \Gamma & (3.1b) \end{cases}$$

donde σ , k y q son funciones suaves y $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ y $\Gamma = \partial\Omega$.



Si multiplicamos la ecuación (3.1a) por una función suave v e integramos en

Ω obtenemos

$$\int_{\Omega} (-\operatorname{div}(\sigma \nabla u)v + kuv) dx = \int_{\Omega} qv dx$$

luego, utilizando el teorema de la divergencia [11], obtenemos

$$\int_{\Omega} (\sigma \nabla u \cdot \nabla v) + kuv) dx = \int_{\Omega} qv dx.$$

Consideremos la forma bilineal

$$B[u, v] = \int_{\Omega} \sigma \nabla u \cdot \nabla v + kuv dx, \quad (3.2)$$

y el funcional lineal

$$F[v] = \int_{\Omega} qv dx. \quad (3.3)$$

La formulación débil del problema (3.1) consiste en hallar $u \in H_0^1(\Omega)$ tal que

$$B[u, v] = F(v), \quad \forall v \in H_0^1(\Omega) \quad (3.4)$$

donde $H_0^1(\Omega) = \{v \in L^2(\Omega) : \nabla v \in L^2(\Omega)\}$ ver [11].

Como veremos a continuación el método de elementos finitos nos permitirá hallar una aproximación de u solución de (3.4).

3.1.1. Triangulación

Consideremos una triangulación \mathcal{T} del dominio $\Omega \subset \mathbb{R}^2$. Esto es, una partición de Ω en triángulos tales que si dos de estos se intersectan lo hacen en un vértice o una arista completa de ambos elementos, ver la figura (3.1).

Vamos a denotar por $\mathbb{V}^{\mathcal{T}}$ al conjunto de funciones continuas en $\bar{\Omega}$ y que además son lineales en cada elemento o triángulo T de la triangulación, esto es

$$\mathbb{V}^{\mathcal{T}} = \{v \in C(\bar{\Omega}) : v|_T \text{ es lineal}, \forall T \in \mathcal{T}\}$$

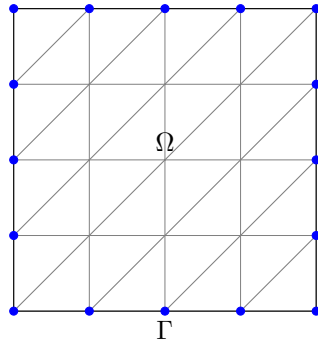
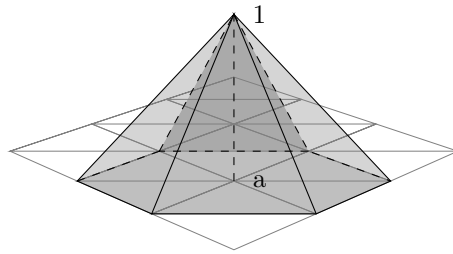
Observemos que cada función $v \in \mathbb{V}^{\mathcal{T}}$ queda unívocamente determinada por el valor que toma en cada vértice de la triangulación.

La formulación de elementos finitos que nos interesa resolver consiste en hallar $u_{\mathcal{T}} \in \mathbb{V}_0^{\mathcal{T}}$ tal que

$$B[u_{\mathcal{T}}, v_{\mathcal{T}}] = F(v_{\mathcal{T}}), \quad \forall v_{\mathcal{T}} \in \mathbb{V}_0^{\mathcal{T}} \quad (3.5)$$

donde $\mathbb{V}_0^{\mathcal{T}} = \{v \in \mathbb{V}^{\mathcal{T}} : v(x) = 0, \text{ para todo } x \in \Gamma\}$, B es la forma bilineal definido por (3.2) y F es el funcional lineal definido por (3.3).

Notemos que la formulación (3.5) es similar al problema (3.4) pero en este caso la solución se encuentra en el subespacio $\mathbb{V}_0^{\mathcal{T}} \subset H_0^1$, es decir, buscamos una solución aproximada en un espacio de dimensión finita.

Figura 3.1: Triangulación de Ω Figura 3.2: Ejemplo de un ϕ en este caso ϕ_a en una triangulación

3.1.2. El problema como un sistema de ecuaciones lineales

Ya presentada la formulación de elementos finitos (3.5) el próximo objetivo será hallar la solución $u_{\mathcal{T}}$.

Consideremos el conjunto de funciones $\{\phi_j\}_{j=1}^{N_{\mathcal{T}}}$ que forman una base de $\mathbb{V}^{\mathcal{T}}$ y tienen la propiedad

$$\phi_j \in \mathbb{V}^{\mathcal{T}} \text{ y } \phi_j(x_i) = \delta_{ij} \text{ para } i, j = 1, 2, \dots, N_{\mathcal{T}}$$

donde $N_{\mathcal{T}}$ es el número de vértices de la triangulación \mathcal{T} y $x_i, i = 1, 2, \dots, N_{\mathcal{T}}$ son los vértices de la triangulación \mathcal{T} .

Si tomamos como ejemplo la figura 3.2, $\phi_a(x_a) = 1$ y para los demás vértices ϕ_a es 0. Por lo tanto, si $v \in \mathbb{V}^{\mathcal{T}}$ tenemos que

$$v(x) = \sum_{i=1}^{N_{\mathcal{T}}} v_i \phi_i(x) = \sum_{i=1}^{N_{\mathcal{T}}} v(x_i) \phi_i(x).$$

Observemos que para cada $x \in \Omega$, a lo sumo tres términos pueden resultar no nulos en la sumatoria anterior.

En particular, si escribimos $u_{\mathcal{T}} = \sum_{j=1}^{N_{\mathcal{T}}} u_j \phi_j$, el problema (3.5) se reescribe como: hallar $u_j \in \mathbb{R}, j = 1, \dots, N_{\mathcal{T}}$ tal que

$$B \left[\sum_{j=1}^{N_{\mathcal{T}}} u_j \phi_j, \phi_i \right] = F(\phi_i) \quad i = 1, \dots, N_{\mathcal{T}} \quad \text{y} \quad x_i \notin \Gamma \quad (3.6)$$

$$u_i = 0 \quad i = 1, \dots, N_{\mathcal{T}} \quad \text{y} \quad x_i \in \Gamma, \quad (3.7)$$

donde hemos utilizado $v = \phi_i$ y el hecho que $\{\phi_i\}_{i=1}^{N_{\mathcal{T}}}$ es una base de $\mathbb{V}^{\mathcal{T}}$. Recordando que B es bilineal entonces (3.6) es equivalente a

$$\sum_{j=1}^{N_{\mathcal{T}}} u_j B[\phi_j, \phi_i] = F(\phi_i) \quad i = 1, \dots, N_{\mathcal{T}} \quad \text{y} \quad x_i \notin \Gamma$$

$$u_i = 0 \quad i = 1, \dots, N_{\mathcal{T}} \quad \text{y} \quad x_i \in \Gamma.$$

Lo cual equivale a resolver el siguiente sistema lineal, de tamaño $N_{\mathcal{T}} \times N_{\mathcal{T}}$:

$$Au = f \quad (3.8)$$

donde

$$A_{ij} = B[\phi_j, \phi_i] = \int_{\Omega} \sigma \nabla \phi_j \cdot \nabla \phi_i + k \phi_j \phi_i \, dx \quad \text{si } x_i \notin \Gamma$$

$$A_{ij} = \delta_{ij} \quad \text{si } x_i \in \Gamma$$

$$f_i = F(\phi_i) = \int_{\Omega} q \phi_i \, dx \quad \text{si } x_i \notin \Gamma$$

$$f_i = 0 \quad \text{si } x_i \in \Gamma.$$

Notemos que aplicado el método de elementos finitos hemos pasado de una ecuación diferencial a un sistema lineal. La solución del sistema lineal es una aproximación numérica de la solución de la ecuación diferencial.

3.1.3. Ensamble de la matriz A y el vector f

Hasta ahora hemos presentado la formulación de elementos finitos desde un punto de vista teórico, el objetivo es llevarlo a la práctica. Para esto desearíamos tener un programa computacional, tal que:

1. Dado Ω nos proporcione una triangulación \mathcal{T} y defina cuáles son los nodos pertenecientes al interior del dominio y cuáles pertenecen a la frontera.
2. Que reciba todos los coeficientes de la ecuación σ , k , q y los datos de borde si es que son no nulos.
3. Ensamble de la matriz A y el vector f .
4. Resuelva el sistema lineal $Au = f$.

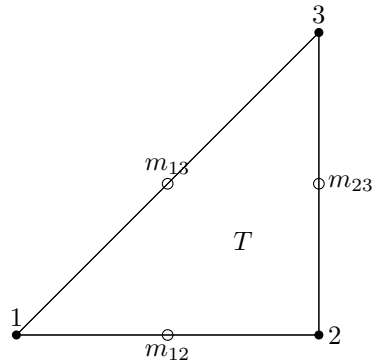


Figura 3.3: Elemento tipo de la triangulación \mathcal{T} y m_{12}, m_{23}, m_{13} puntos medios de cada lado.

A continuación nos concentraremos en el paso (3) del algoritmo.

Para comenzar, consideremos el ensamble del lado derecho f . Anteriormente vimos que

$$f_i = \int_{\Omega} q\phi_i dx \text{ si } x_i \notin \Gamma \quad \text{o} \quad f_i = 0 \text{ si } x_i \in \Gamma.$$

Observemos que $\int_{\Omega} q\phi_i dx = \sum_{T \in \mathcal{T}} \int_T q\phi_i dx$.

Dado $T \in \mathcal{T}$, vamos a calcular $\int_T q\phi_i$ utilizando la regla del punto medio, regla de cuadratura que es exacta para polinomios de grado menor o igual que 2 y está definida por

$$\int_T q\phi_i dx \approx \frac{|T|}{3} [q(m_{12})\phi_i(m_{12}) + q(m_{23})\phi_i(m_{23}) + q(m_{31})\phi_i(m_{31})], \quad (3.9)$$

donde $|T|$ denota la medida o área del triángulo y m_{12}, m_{23} y m_{31} son los puntos medios de cada uno de los lados de T (ver figura 3.3).

Para ensamblar el vector f (y también para el caso de la matriz A) en lugar de iterar sobre los vértices x_i , lo haremos sobre los elementos $T \in \mathcal{T}$ y en cada elemento calcularemos las integrales no nulas, luego estos valores se agregarán a al vector f ó a la matriz A según corresponda. Veamos este procedimiento para el caso del vector f .

Dado un elemento $T \in \mathcal{T}$, sus vértices tendrán la numeración global dada por la malla, por ejemplo 25, 47, 11. Redefinimos en este elemento una numeración local 1, 2, 3 para sus vértices y hacemos la identificación

$$\begin{array}{lcl} 25 & \longrightarrow & 1 \\ 47 & \longrightarrow & 2 \\ 11 & \longrightarrow & 3 \end{array}$$

Entonces tenemos que

$$\begin{aligned}\phi_{25|_T} &= \varphi_T^1 \\ \phi_{47|_T} &= \varphi_T^2 \\ \phi_{11|_T} &= \varphi_T^3,\end{aligned}$$

donde φ_T^i es la función lineal en T tal que, vale 1 en el i -ésimo vértice local y 0 en los demás. Luego, para cada φ_T^i con $i = 1, 2, 3$, reemplazando en la fórmula (3.9) obtenemos

$$\begin{aligned}i = 1: \quad \int_T q \varphi_T^1 dx &= \frac{|T|}{3} [q(m_{12}) \frac{1}{2} + q(m_{31}) \frac{1}{2}] = \frac{|T|}{6} (q(m_{12}) + q(m_{31})). \\ i = 2: \quad \int_T q \varphi_T^2 dx &= \frac{|T|}{6} (q(m_{12}) + q(m_{23})). \\ i = 3: \quad \int_T q \varphi_T^3 dx &= \frac{|T|}{6} (q(m_{23}) + q(m_{31})).\end{aligned}$$

Por lo tanto, la contribución del elemento $T \in \mathcal{T}$ al vector f es:

$$f_T = \frac{|T|}{6} \begin{pmatrix} (q(m_{12}) + q(m_{31})) \\ (q(m_{12}) + q(m_{23})) \\ (q(m_{23}) + q(m_{31})) \end{pmatrix}.$$

Cada componente del vector f_T se debe sumar al valor de la correspondiente componente del vector f , esto es $f_T(1)$ se sumará al valor de $f(25)$, $f_T(2)$ al valor de $f(47)$ y $f_T(3)$ al valor de $f(11)$, por ejemplo

$$f(25) = f(25) + f_T(1).$$

Por su parte, para ensamblar la matriz A se procede de manera similar. Por ejemplo para cargar los valores aportados por el término $\int_T k \varphi_T^j \varphi_T^i dx$ utilizamos también la regla del punto medio obteniendo

$$\begin{aligned}\int_T k \varphi_T^j \varphi_T^i \approx \frac{|T|}{3} [k(m_{12}) \varphi_T^j(m_{12}) \varphi_T^i(m_{12}) + k(m_{23}) \varphi_T^j(m_{23}) \varphi_T^i(m_{23}) + \\ k(m_{31}) \varphi_T^j(m_{31}) \varphi_T^i(m_{31})]\end{aligned}$$

tendremos que considerar todos las combinaciones entre los productos $\varphi_T^j \cdot \varphi_T^i$ con $i, j = 1, 2, 3$. En total son 9 casos, a modo de ejemplo desarrollamos los siguientes dos:

$$\begin{aligned}i = j = 1: \quad \int_T k \varphi_T^1 \varphi_T^1 dx &= \frac{|T|}{3} [k(m_{12}) \frac{1}{2} \frac{1}{2} + k(m_{31}) \frac{1}{2} \frac{1}{2}] = \frac{|T|}{12} (k(m_{12}) + k(m_{31})) \\ i = 1, j = 2: \quad \int_T k \varphi_T^2 \varphi_T^1 dx &= \frac{|T|}{3} (k(m_{12}) \frac{1}{2} \frac{1}{2}) = \frac{|T|}{12} k(m_{12})\end{aligned}$$

Los demás casos son análogos. Finalmente la contribución del elemento T a la matriz A es:

$$A_T = \frac{|T|}{12} \begin{pmatrix} (k(m_{12}) + k(m_{31})) & k(m_{12}) & k(m_{31}) \\ k(m_{12}) & (k(m_{12}) + k(m_{23})) & k(m_{23}) \\ k(m_{31}) & k(m_{23}) & (k(m_{23}) + k(m_{31})) \end{pmatrix}.$$

Al igual que con el vector f , cada entrada de la matriz A_T se sumará a la entrada correspondiente de la matriz global A . En este ejemplo tendríamos:

$$\begin{aligned} A(25, 25) &= A(25, 25) + A_T(1, 1), \\ A(25, 47) &= A(25, 47) + A_T(1, 2), \\ &\vdots \\ A(11, 11) &= A(11, 11) + A_T(3, 3). \end{aligned}$$

3.2. MEF aplicado al problema (2.22) (PD)

Para aplicar el MEF al problema de borde (2.22) primero debemos dar su formulación débil o variacional. De acuerdo a [12] la formulación débil del problema (2.22) es:

Hallar $u \in H_{\Gamma_b,0}^1(\Omega)$ tal que

$$B[u, v] = F(v), \quad \forall v \in H_{\Gamma_b,0}^1(\Omega) \quad (3.10)$$

donde

$$\begin{aligned} H_{\Gamma_b,0}^1(\Omega) &= \{v \in H^1(\Omega) : v = 0, \text{ sobre } \Gamma_b\} \\ B[u, v] &= \int_{\Omega} \sigma(x) \nabla u \cdot \nabla v \, dx + \int_{\Omega} k(x) uv \, dx + \alpha \int_{\Gamma_u} uv \, d\Gamma \\ F(v) &= \int_{\Omega} q(x)v \, dx + \alpha(T_a - T_b) \int_{\Gamma_u} v \, d\Gamma. \end{aligned}$$

Observemos que la forma bilineal B y el funcional lineal F son similares a los presentados en el ejemplo introductorio. Salvo que en este caso aparecen además términos relacionados a la condición de borde de tipo Robin considerada sobre el borde del dominio Γ_u o sea sobre la "superficie".

Teniendo en cuenta que

$$\int_{\Gamma_u} v \, d\Gamma = \sum_{S \subset \Gamma_u} \int_S v \, d\Gamma$$

donde S son los segmentos (lados de los triángulos) que se encuentran en el borde Γ_u . Luego para calcular las integrales sobre la frontera Γ_u , iteramos sobre los segmentos S , calculamos las contribuciones de cada segmento y luego ese valor es sumado a la entrada de la matriz A o del vector f según corresponda.

Para calcular la integral en cada segmento utilizaremos la regla de Simpson, regla de cuadratura exacta para polinomios de grado menor o igual a 3, la cual está definida por

$$\int_a^b g \, dx \approx \frac{b-a}{6} \left(g(a) + 4g\left(\frac{a+b}{2}\right) + g(b) \right)$$

Entonces, por ejemplo, para ensamblar el vector f tendremos que calcular las contribuciones de:

$$\int_{\Omega} q(x)\varphi_T^i dx \quad \text{y de} \quad \alpha(T_a - T_b) \int_{\Gamma_u} \varphi_T^i d\Gamma$$

Nuevamente en lugar de iterar sobre los vértices, en este caso iteramos sobre los segmentos S . Entonces, fijado un segmento S y nuevamente dando una numeración local a los vértices de ese segmento, debemos calcular

$$\begin{aligned} \int_S \varphi_T^1 d\Gamma &= \frac{|S|}{6} \left[1 + 4 \frac{1}{2} + 0 \right] = \frac{|S|}{2} \\ \int_S \varphi_T^2 d\Gamma &= \frac{|S|}{6} \left[0 + 4 \frac{1}{2} + 1 \right] = \frac{|S|}{2} \end{aligned}$$

Entonces, por ejemplo, la contribución de $\alpha(T_a - T_b) \int_{\Gamma_u} \phi_i d\Gamma$ para un segmento S es:

$$f_S = \begin{pmatrix} \frac{|S|}{2} \\ \frac{|S|}{2} \end{pmatrix} \alpha(T_a - T_b)$$

y esta debe ser sumada a la componente (con numeración global) del vector f .

La contribución $\int_{\Omega} q(x)\varphi_T^i dx$ de cada triángulo $T \in \mathcal{T}$ es análoga al ejemplo de la sección anterior.

Para ensamblar la matriz A tendremos que calcular 3 contribuciones de cada triángulo $T \in \mathcal{T}$ y son las siguientes:

$$\int_{\Omega} \sigma(x) \nabla \varphi_T^i \cdot \nabla \varphi_T^j dx, \quad \int_{\Omega} k(x) \varphi_T^i \varphi_T^j dx, \quad \alpha \int_{\Gamma_u} \varphi_T^i \varphi_T^j d\Gamma$$

Las primeras dos integrales se calculan igual a la mencionada en la sección anterior.

A modo de ejemplo mostraremos la contribución $\alpha \int_{\Gamma_u} \phi_j \phi_i d\Gamma$ de un triángulo T , también usando la regla de Simpson:

$$\begin{aligned} i = j = 1 : \quad \int_S \phi_1^2 d\Gamma &= \frac{|S|}{6} \left(1 + 4 \cdot \frac{1}{2} \cdot \frac{1}{2} + 0 \right) = \frac{|S|}{3} \\ i = 1, j = 2 : \quad \int_S \phi_2 \phi_1 d\Gamma &= \frac{|S|}{6} \left(1 \cdot 0 + 4 \cdot \frac{1}{2} \cdot \frac{1}{2} + 0 \cdot 1 \right) = \frac{|S|}{6} \end{aligned}$$

el caso de $i = j = 2$ también da como resultado $\frac{|S|}{3}$ y el caso de $i = 2, j = 1$ da $\frac{|S|}{6}$, por lo tanto la contribución es:

$$|S| \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \alpha$$

la cual debe ser sumada a las entradas correspondientes de la matriz A teniendo en cuenta la numeración global.

En los próximos capítulos se darán más detalles del algoritmo para la solución de nuestro problema.

Capítulo 4

Resolución del problema inverso: Localización de tumores

Los objetivos de este capítulo son: en primer lugar definir el concepto de problema inverso (PI) y formalizar el problema de localización como un PI. Luego plantear el PI como un problema de optimización. Finalmente, daremos un algoritmo de minimización para obtener la solución del problema de localización.

4.1. Qué es un problema inverso (PI)

Según Joseph Keller en [13] se pueden llamar a dos problemas inversos el uno del otro si la formulación de cada uno implica la totalidad o parte de la solución del otro. Usualmente por razones históricas, uno de los dos problemas ha sido estudiado ampliamente durante algún tiempo, mientras que el otro es más reciente y no tan bien entendido. En estos casos el primero se llama problema directo, mientras que el segundo se llama el problema inverso. Algunos ejemplos son: el cálculo de una imagen en la tomografía computarizada, o el cálculo de la densidad de la Tierra a partir de mediciones de su campo de gravedad.

El término problema inverso se refiere a que comienza con los resultados y luego calcula las causas. Éste es el inverso de un problema directo, que comienza con las causas y luego calcula los resultados.

4.2. Definición de nuestro PI

Según los trabajos [4, 5, 6, 7], un tumor cancerígeno produce un incremento de la temperatura corporal en la zona que lo rodea, debido al incremento de la perfusión sanguínea y la actividad metabólica local. En este trabajo nos enfocamos en tumores que están cercanos a la superficie del cuerpo como los melanomas (tumores de piel), y tumores mamarios ya que estos llegan a causar un perfil de temperatura anormal en la superficie (ver [7]), este perfil anormal de temperatura se puede detectar con una cámara infrarroja.

Nuestro problema inverso es detectar la localización y tamaño del tumor, conociendo el perfil de temperatura sobre la piel. En términos matemáticos, el problema que deseamos resolver es:

Conocido el valor de todos los parámetros fisiológicos, las temperaturas T_b y T_a y dado un perfil de temperatura u_{exp} medido sobre Γ_u , hallar $\omega^ \subset \Omega$ tal que la solución u_{ω^*} al problema de borde (2.22) verifique $u_{\omega^*} = u_{exp}$ en Γ_u .*

El ω^* óptimo se alcanza cuando la función costo:

$$J(\omega^*) = \Psi(\omega, u_{exp}) = \frac{1}{2} \int_{\Gamma_u} (u_{\omega^*} - u_{exp})^2 d\Gamma \quad (4.1)$$

se anula, esto significa que J tiene un mínimo en ω^* .

La función costo $J(\omega)$ representa la diferencia entre el perfil de temperatura medido u_{exp} y la solución numérica u_ω , obtenida mediante el método de elementos finitos, del problema de transferencia de calor (2.22), asumiendo una posición del tumor ω .

En este trabajo nos concentramos en el caso de dos dimensiones. Además como la motivación de este trabajo es aplicar esta metodología para melanomas nodulares y cáncer de mamas, asumiremos que el tumor tiene una forma esférica, por lo tanto para predecir la posición y tamaño es necesario determinar el centro y radio de la circunferencia. Entonces hallar ω equivale a hallar una terna (x_c, y_c, R) que determina el centro y radio del tumor, si el problema estuviera en tres dimensiones debe hallarse (x_c, y_c, z_c, R) .

En otras palabras nuestro problema directo es: dado ω definido por la terna (x_c, y_c, R) es decir el centro y radio del tumor, calculamos la temperatura del tejido que incluye a la temperatura sobre la piel. Nuestro problema inverso es: dado el perfil de temperatura sobre la piel queremos conseguir el centro y radio del tumor, es decir la ubicación del tumor en el tejido.

4.3. El PI como problema de optimización

Debido a que necesitamos hallar el ω^* óptimo y como vimos en la sección anterior esto equivale a hallar el mínimo de la función J , en esta sección veremos primero un poco sobre métodos de minimización y luego el método de minimización elegido y aplicado a nuestro problema inverso.

4.3.1. Minimización y método del gradiente

El problema de hallar la localización y tamaño del tumor puede ser planteado como el siguiente problema de optimización:

$$\begin{aligned} & \text{Minimizar } J(x_c, y_c, R) \\ & \text{sujeto a} \\ & (x_c, y_c, R) \in C \end{aligned}$$

donde C es un conjunto cerrado y convexo en \mathbb{R}^3 definido por

$$C = \{(x, y, z) \in \mathbb{R}^3 : x_l \leq x \leq x_u, y_l \leq y \leq y_u, z_l \leq z \leq z_u\}$$

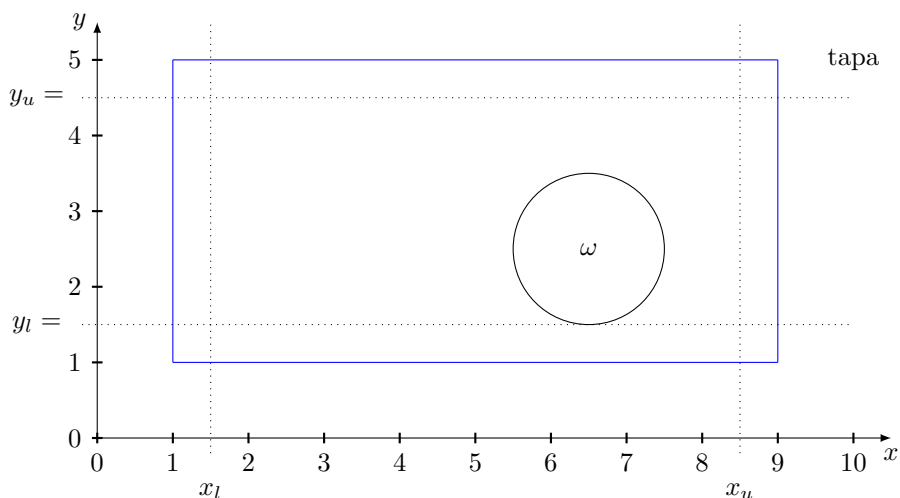


Figura 4.1: Tumor en el tejido.

donde los x_l y x_u representan valores mínimo y máximo para la coordenada x del centro del tumor, similarmente y_l e y_u son los valores mínimo y máximo que puede tomar la coordenada y del tumor, mientras que z_l y z_u corresponden al mínimo y máximo valor que puede tomar el radio del tumor, de manera tal que siempre $\omega \subset \Omega$ observar la figura (4.1).

Como ya dijimos anteriormente para hallar el subdominio ω^* óptimo, definido por la terna (x^*, y^*, r^*) , debemos hallar el mínimo de la función J . Por lo tanto nos abocaremos a la minimización de la función J .

A grandes rasgos los métodos iterativos de minimización u optimización son generalmente descendentes en el sentido que a partir de un valor inicial x_0 se elige un nuevo valor x_1 (perteneciente a un conjunto permitido) tal que $f(x_1) \leq f(x_0)$. Así iterando varias veces, la idea es converger al mínimo local más próximo.

Podemos dividir los métodos de minimización en dos grupos:

1. Métodos que no utilizan información de las derivadas de la función.
2. Métodos que sí utilizan información de las derivadas de la función.

Los métodos del primer grupo se denominan de orden cero mientras que los métodos del segundo grupo se denominan de orden 1 si sólo usan información de la primera derivada de la función o de orden 2 si usan también información de la segunda derivada de la función. Generalmente estos últimos métodos son más rápidos que los métodos del primer grupo.

En este trabajo utilizamos un método de optimización de primer orden, es decir que hace uso de la primera derivada de la función J .

Más precisamente utilizaremos el método de descenso más inclinado también conocido como método del gradiente. Este método se basa en tomar como dirección de descenso la opuesta a la dirección dada por el gradiente. Supongamos

que tenemos la función f a la cual queremos encontrar su mínimo, entonces aplicamos el método:

1. Consideramos un punto inicial $x = x^0$ en el dominio de la función, en este punto estamos en $k = 0$.
2. Elegimos una dirección de descenso $d^k = -\nabla f(x^k)$.
3. Seleccionamos un paso α_k tal que: $x^k + \alpha_k d^k$ sea un punto admisible y además $g_k(\alpha_k) = f(x^k + \alpha_k d^k) < f(x^k)$.
4. Hacemos $x^{k+1} = x^k + \alpha_k d^k$.
5. Si $\|\nabla f(x^k)\| < \epsilon$ paramos, sino $k = k + 1$ y volvemos a el paso 2.

Donde ϵ es un valor de tolerancia, como criterio de parada.

En este trabajo consideramos $\alpha_k = cte$ y en el caso de que $x^k + \alpha_k d^k$ no resulta un punto admisible, redefinimos $\alpha = \alpha/2$ hasta que $x^k + \alpha_k d^k$ resulte un punto admisible.

Por otra parte, para el cálculo de ∇J nos basamos en los resultados presentados en [12].

4.4. Un algoritmo para la solución

Como ya adelantamos en la sección anterior, el problema de encontrar el tumor y su radio lo resolveremos con un método iterativo. Dada una tolerancia para el error el cual sería el criterio de parada de la iteración el algoritmo que proponemos para resolver en PD es el siguiente

1. Dado un $\omega = \omega_{inicial} = (x_0, y_0, R_0)$, $k = 0$
2. Resolver el problema directo (2.28), obteniendo u_k temperatura en Ω .
3. Calcular la función costo, es decir $J(\omega_k)$. Si es menor al error, fin. Sino seguir.
4. Calcular el gradiente, es decir $\nabla J(\omega_k)$.
5. Actualizar $\omega_{k+1} = \omega_k - \alpha_k \nabla J(\omega_k)$.
6. $k = k + 1$. Volver al paso 2.

Con este algoritmo en pseudocódigo finaliza el capítulo 4, sólo resta la parte computacional para dar una solución, que se verá en el capítulo 5.

Capítulo 5

Implementación y casos de estudio

El objetivo de este capítulo es mostrar el proceso de desarrollo del software en el proyecto. Esto incluye el paradigma de programación usado, la arquitectura del software, la estructura de archivos y los módulos realizados. También describir las librerías y módulos externos que se usaron en la codificación y finalmente uso del software y algunos casos de estudio con sus resultados.

El sistema de control de versión utilizado fue git, hospedado en un servidor de github la dirección completa es <https://github.com/jppar/detum>.

Con respecto al proceso de desarrollo del software, el módulo se desarrolló bajo el paradigma de programación modular. Consiste de un algoritmo, esto es, un transformador de información desde el punto de vista de la ingeniería del software y sistemas. Toma unos datos de entrada, los transforma y produce unos datos de salida.

Luego de dar una descripción de cada módulo, al final del capítulo se presentan diferentes casos de estudio con el fin de mostrar el desempeño de la metodología propuesta.

5.1. Arquitectura y diseño del software

La programación modular es programación estructurada con el agregado del método "divide y vencerás" ó análisis descendente (top-down)[14]. Divide y vencerás hace referencia también a un paradigma de diseño algorítmico. Se muestra el diseño descendente del proyecto en la Figura 5.1.

A continuación describiremos con más detalle los módulos, cuales son sus APIs(Application Programming Interface) y qué APIs de los módulos externos se utilizan.

5.2. Descripción de los módulos

El modulo raíz o programa principal es elementosF.py. Los demás módulos tiene una sola función cada uno, las cuales son invocadas por el modulo principal.

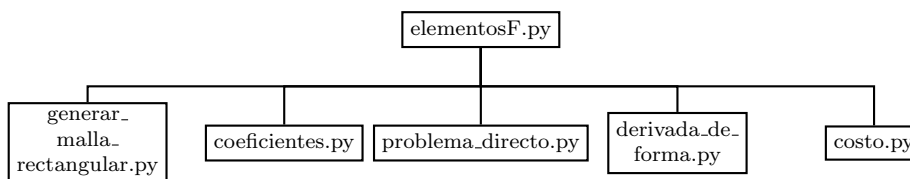


Figura 5.1: Diagrama top-down, arquitectura del software.

5.2.1. Módulo principal (elementosF.py)

En elementosF.py se encuentra el programa principal main().

```
main(argv='', **kwargs)
```

Parámetros o datos de entrada:

Los siguientes tres parámetros están relacionados al dominio, la triangulación y la temperatura experimental. Y como existe una relación entre ellos, si se ingresan, deben ingresar los tres.

- dimension: es una tupla (n,m) donde n y m son float entre 0.02 y 0.12, esta tupla representa los límites del dominio y la unidad es el metro, es un rectángulo (0, n) x (0, m). Por defecto es (0.09,0.03).
- malla: es una tupla (N,M) donde N y M son enteros entre 10 y 200 y deben ser proporcionales a n y m es decir que se tiene que cumplir que n/m es igual a N/M. Esta tupla representa a la cantidad de divisiones de la triangulación. Por defecto es (120,40)
- exp_tumor: es una tupla (x,y,r), representa un tumor donde x,y es el centro y r es el radio, del cual se obtiene la temperatura (sobre la superficie de la piel) y luego esa temperatura se toma como la temperatura experimental para que el algoritmo halle x,y,r. Sirve para testear el modulo. El tumor debe estar dentro del dominio (0, n) x (0, m). Por defecto es (0.038, 0.016, 0.0055).

Los siguientes 4 parámetros están relacionados a la ecuación de calor.

- conductividad: es una tupla (sigma1, sigma2) que representa el coeficiente de conductividad térmica en tejido sano y en tejido enfermo respectivamente. Por defecto es (0.5,0.75).
- perfusion: es una tupla (k1,k2) que representa el coeficiente de perfusión en tejido sano y en tejido enfermo respectivamente. Por defecto es (1998.1,7992.4).
- metabolismo: es una tupla (q1,q2) que representa el coeficiente de calor metabólico en tejido sano y en tejido enfermo respectivamente. Por defecto es (4200,42000).
- temperatura_ambiente: es la temperatura ambiente. Por defecto es 25.

Los siguientes tres parámetros son de criterio de parada.

- max_iter: es un número entero, cuando se llega a ese número de iteración el algoritmo se detiene. Por defecto es 50000.
- max_tiempo: es un número entero, cuando se llega a esa cantidad de segundos de ejecución el algoritmo se detiene.
- cota: es un float, y el algoritmo se detiene cuando el error entre la temperatura experimental y la aproximada es menor que este. Mientras mas chico es el error mejor aproximado esta la localización y tamaño del tumor, de el tumor real.

Datos de salida:

Esta función principal main() retorna una tupla (aprox, error) donde 'aprox' es un lista [x,y,r] con la aproximación obtenida del tumor, donde x,y es el centro y r es el radio, 'error' es el error que tiene de la localización y tamaño exacto.

Una vez definidos los parámetros de entrada, lo primero que hace este módulo principal es calcular la temperatura sobre la superficie de la piel a partir de exp_tumor (con esto se simula la temperatura experimental).

Luego con el siguiente código se elige el punto de partida para comenzar la iteración según el algoritmo visto en el capítulo 4.

```
lado_minimo = min(n,m)
posicion_inicial = [n/2, m/2, lado_minimo/4.0]
```

Luego se sigue el algoritmo visto en la sección 4.4 utilizando las APIs que veremos a continuación.

5.2.2. Cálculo de los coeficientes (coeficientes.py)

En el modulo coeficientes.py se encuentra la función coefficients(). El objetivo de esta función es calcular el valor de los parámetros de la ecuación de calor en cada punto de la triangulación, dada la triangulación, el tumor, y el valor de los coeficientes en tejido sano y en tejido enfermo.

```
coefficients(x_c, y_c, R1, vertex_coordinates,
            sigma1, sigma2, k1, k2, q1, q2)
```

Parámetros o datos de entrada:

- x_c, y_c, R1: estos primero tres parámetros corresponden a la localización y radio del tumor.
- vertex_coordinates: es un arreglo de numpy y contiene por cada vértice de la triangulación un arreglo de numpy de longitud 2 que contiene las coordenadas x,y del vertice.
- sigma1, sigma2: representa el coeficiente de conductividad térmica en tejido sano y en tejido enfermo respectivamente.
- k1, k2: representa el coeficiente de perfusión en tejido sano y en tejido enfermo respectivamente.
- q1, q2: representa el coeficiente de calor metabólico en

tejido sano y en tejido enfermo respectivamente.

Datos de salida:

- `coeff_sigma`, `coeff_k`, `coeff_q`: son tres arreglos de numpy de longitud la cantidad de vértices de la triangulación, representan respectivamente sobre la triangulación: el coeficiente de conductividad térmica, el coeficiente de perfusión y el coeficiente de fuente de calor metabólico.
- `VX1x`, `VX1y`, `VY2x`, `VY2y`, `VR1x`, `VR1y`, `VR2x`, `VR2y`, estos datos se refieren a campos vectoriales necesarios para calcular la derivada de forma.

Estos coeficientes σ, k, q que retorna la función son necesarios, se encuentran en la ecuación de calor propuesta por H. Pennes (2.21) fueron mencionados en torno a ella en la sección 2.3, y por lo tanto se encuentran en nuestro problema (PD) planteado en (2.28).

Por otro lado los datos de salida que se refieren a los campos vectoriales con respecto al eje x (VX) al eje y (VY) y al radio (VR) los calculamos según [10]. Y serán utilizados para calcular la derivada de forma que es la que nos dirá como cambiar la forma del tumor para que su temperatura se aproxime a la dada.

5.2.3. Cálculo de la temperatura (problema_directo.py)

En el modulo `problema_directo.py` se encuentra la función `direct_problem()`. El objetivo de esta función es calcular la temperatura en el tejido a partir de la ubicación y radio del tumor.

```
direct_problem(elem_vertices, vertex_coordinates, dirichlet, robin,
               u_exp, coef_sig, coef_k, coef_q, alpha, Ta, Tb)
```

Parámetros o datos de entrada:

Los siguiente 4 parámetros están relacionados a la triangulación.

- `elem_vertices`: es un arreglo de numpy que contiene los elementos de la triangulación, es decir los triángulos, cada elemento es de la forma `[a,b,c]` donde `a,b` y `c` son vértices.
- `vertex_coordinates`: es un arreglo de numpy que contiene los vértices de la triangulación cada vértice es de la forma `[a,b]` donde `a` y `b` son coordenadas.
- `dirichlet`: contiene cuales son los vértices correspondientes a los datos de tipo `dirichlet`.
- `robin`: contiene cuales son los vértices correspondientes a los datos de tipo `robin`.

- `u_exp`: es la temperatura experimental.

Los siguientes 6 parámetros están relacionados a la ecuación de calor.

- `coef_sig`, `coef_k`, `coef_q`: son los mismos tres parámetros que retorna la función `coefficients` explicada anteriormente.
- `alpha`: es el coeficiente de transferencia de calor de la piel.
- `Ta`: es la temperatura ambiente.
- `Tb`: es la temperatura corporal.

Datos de salida:

- `u_d`: es un arreglo de numpy el cual contiene por cada vértice (índice del arreglo) su temperatura, en particular tiene la temperatura de los vértices que corresponden a la superficie de la piel.
- `u_adj`: también es un arreglo de longitud la cantidad de vértices de la triangulación y para calcularlo se usa el dato de entrada `u_exp`.

El parámetro de entrada `u_exp` no sería necesario para resolver el problema directo, pero se usa para resolver el problema adjunto, la solución de este problema se retorna en el parámetro de salida `u_adj`, este ultimo es un dato que luego se usará para el cálculo de la derivada de forma según [10], no ahondaremos más en este trabajo sobre los campos vectoriales y los detalles sobre el cálculo de la derivada de forma, para ver detalles sobre esto ver [10].

En este módulo se usan las siguientes APIs de numpy:

1. `numpy.identity(n)`: retorna la matriz identidad de tamaño $n \times n$, es una matriz cuadrada con unos en la diagonal principal.
2. `transpose()`: Metodo de un arreglo numpy, sirve para calcular la transpuesta de una matriz.
3. `numpy.linalg.det(a)`: calcula el determinante de una matriz.
4. `numpy.linalg.solve(a, b)`: calcula la solución x para el sistema de ecuaciones lineales $ax = b$.
5. `numpy.dot(a, b)`: realiza el producto de las matrices a, b .
6. `numpy.linalg.norm(x)`: calcula la norma de un vector o matriz x .

5.2.4. Cálculo de J (`costo.py`)

En el modulo `costo.py` se encuentra la función `calcularJ()`. El objetivo de esta función es calcular el error entre la temperatura experimental y la temperatura calculada a partir de la aproximación de tumor vía el problema directo. De esta forma podemos tener una idea de qué tan lejos esta nuestra aproximación de la localización y tamaño real del tumor.

`calcularJ(u_exp, u_d, robin, vertex_coordinates)`

Parámetros o datos de entrada:

- `u_exp`: temperatura experimental.
- `u_d`: temperatura a partir del tumor aproximado
- `robin`: datos de tipo robin (indica que vértices son de tipo robin).
- `vertex_coordinates`: coordenadas de los vértices.

Datos de salida:

- `J`: el error, de tipo float.

5.2.5. Cálculo de ∇J (derivada_de_forma.py)

En el módulo `derivada_de_forma.py` se encuentra la función `shape_derivative_J()`. El objetivo de esta función es calcular la derivada de forma que según la notación usada en el algoritmo dado en la sección 4.4 corresponde a ∇J es decir el gradiente de J .

```
shape_derivative_J(elem_vertices, vertex_coordinates, sigma1, k1,
q1, ud, wa, VX1x, VX1y, VY2x, VY2y, VR1x, VR1y, VR2x, VR2y)
```

Parámetros o datos de entrada:

- `elem_vertices`: contiene los elementos de la triangulación.
- `vertex_coordinates`: contiene los vértices con sus coordenadas.
- `sigma1`: coeficiente de conductividad térmica en tejido sano.
- `k1`: coeficiente de perfusión en tejido sano.
- `q1`: coeficiente de calor metabólico en tejido sano.
- `ud`: temperatura obtenida vía el problema directo.
- `wa`: solución obtenida del problema adjunto.
- `VX1x, VX1y, VY2x, VY2y, VR1x, VR1y, VR2x, VR2y`: campos vectoriales con los cuales se calculan las derivadas con respecto a x, y, r .

Datos de salida:

- `grad_J`: es un arreglo numpy de dimension (3,1), representa el gradiente de J .

En este módulo se usan las mismas APIs de numpy que se usan en el modulo `problema_directo.py` descrito en la subsección 5.2.3.

5.2.6. Triangulación (generar_malla_rectangular.py)

En el modulo `generar_malla_rectangular.py` se encuentra la función homónima. El objetivo de este modulo es triangular el dominio del problema.

```
generar_malla_rectangular(xmax, ymax, N, M)
```

Parámetros o datos de entrada:

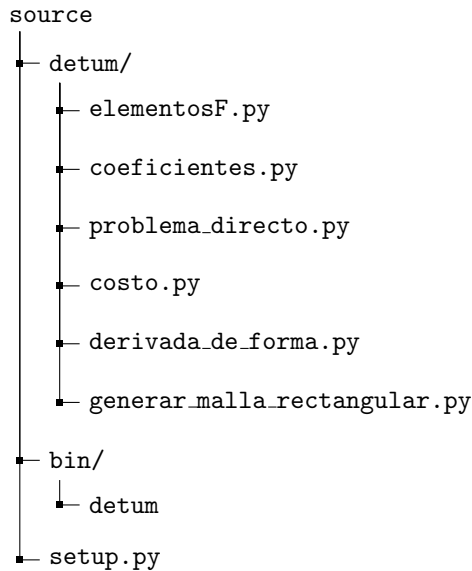
- `xmax, ymax`: representan al borde lateral derecho y al borde superior del dominio respectivamente, la unidad de medida que se adopta es el metro.
- `N, M`: son números enteros y se utilizan para dividir el dominio y hacer la triangulación, por ejemplo si $N=9$ y $M=3$ el dominio se divide en 9 verticalmente y en 3 horizontalmente.

Datos de salida:

Los siguientes cuatro son los mismos que se usan como parámetro de entrada en la función ya vista `direct_problem()`.

- `elem_vertices`: contiene los elementos de la triangulación.
- `vertex_coordinates`: contiene los vértices de la triangulación.
- `dirichlet`: contiene cuales son los vértices de tipo dirichlet.
- `robin`: contiene cuales son los vértices de tipo robin.

5.2.7. Estructura del paquete de software realizado



5.3. Herramientas y librerías externas

Módulos adicionales de Python usados para resolver el problema:

1. NumPy: es un paquete fundamental para computación científica con Python, provee un gran soporte para trabajar cómodamente con matrices y vectores, contiene una biblioteca de funciones matemáticas de alto nivel para operar con matrices y vectores. Usamos este módulo para soportar la triangulación, también para construir el sistema de ecuaciones lineales y resolverlo obteniendo el vector u de temperatura.
2. Matplotlib: es una biblioteca de Python y compatible con NumPy para la generación de gráficos a partir de datos contenidos en matrices y vectores.

5.4. Paralelización

Para paralelizar la solución se usaron los siguientes módulos:

1. ViennaCL: es una librería de álgebra lineal, libre, de open-source para cálculos computacionales en arquitecturas de múltiples núcleos (GPUs, MIC) y CPUs multinucleos. La librería está escrita en C++ y soporta CUDA, OpenCL, y OpenMP.
2. PyViennaCL: es un módulo en Python que es un wrapper (envoltura) de python para ViennaCL, es decir que permite usar las herramientas de ViennaCL para paralelizar cálculos de álgebra lineal, escribiendo código en Python y compatible con NumPy.
3. OpenCL: Para paralelizar ViennaCL brinda una API, pero necesita conectarse con un backend para funcionar, los backend compatible son: OpenCL,

OpenMP, CUDA. Elegimos OpenCL ya que es más compatible con distintos hardware. OpenCL (Open Computing Language) es un framework que brinda un lenguaje de programación basado en C99 y una API (application programming interface). Juntos permiten crear aplicaciones con paralelismo a nivel de datos y de tareas que pueden ejecutarse tanto en CPUs (central processing units) como en GPUs (graphics processing units)

Se corrieron pruebas del algoritmo del problema directo en una placa de video GEFORCE GT 740 de 2GB de memoria DDR3. No se observaron mejoras en los tiempos de ejecución. No pudimos encontrar las causas, quedando como trabajo futuro o posible extensión.

5.5. Casos de estudio y resultados

El objetivo de esta sección es, por un lado mostrar los resultados de resolver el PD es decir el calculo de la temperatura en el tejido dada la localización del tumor. Por otro lado mostrar cuatro casos de estudio y sus resultados para la localización de tumores.

Para la persona interesada en el software, en el repositorio: <https://github.com/jppar/detum> en el archivo README hay un instructivo de instalación, una vez instalado se puede ejecutar:

```
detum -h
```

5.5.1. Resultados en cuanto al PD

Las Figuras 5.2, 5.3 y 5.4 muestran el resultado de resolver el problema directo con el método de elementos finitos, con una malla/triangulación de 120x40. El tumor está en las coordenadas 4,5cm y 1,5cm (justo en el centro) y es de radio 0,75cm. Podemos observar que la temperatura en zonas cercanas a la región tumoral es mayor. En estos gráficos se aprecia la eficacia de este método para el problema.

Por un lado en las figuras 5.2 y 5.3 podemos ver la distribución de temperatura sobre todo el dominio. Ésta presenta un importante incremento en la región donde se localiza el tumor. Por otro lado en la Figura 5.4 se ilustra el perfil de temperatura sobre la superficie de la piel. Podemos observar una diferencia de temperatura de aproximadamente 1°C entre la región que está situada encima del tumor (de aproximadamente $36,45^{\circ}\text{C}$) y las regiones situadas encima del tejido sano (de aproximadamente $35,45^{\circ}\text{C}$). Esto concuerda con el hecho de que la presencia de un tumor puede inducir un incremento de la temperatura en la superficie corporal.

5.5.2. Resultados en cuanto al PI (localización de tumores)

En la Figura 5.5 se ilustran los casos de estudio ejecutados. El dominio es un rectángulo de 9cm x 3cm. No olvidar que la unidad de medida utilizada es el metro. El círculo del centro representa a la posición inicial con la que inicia el algoritmo para estos casos, y los demás círculos son los casos de estudio.

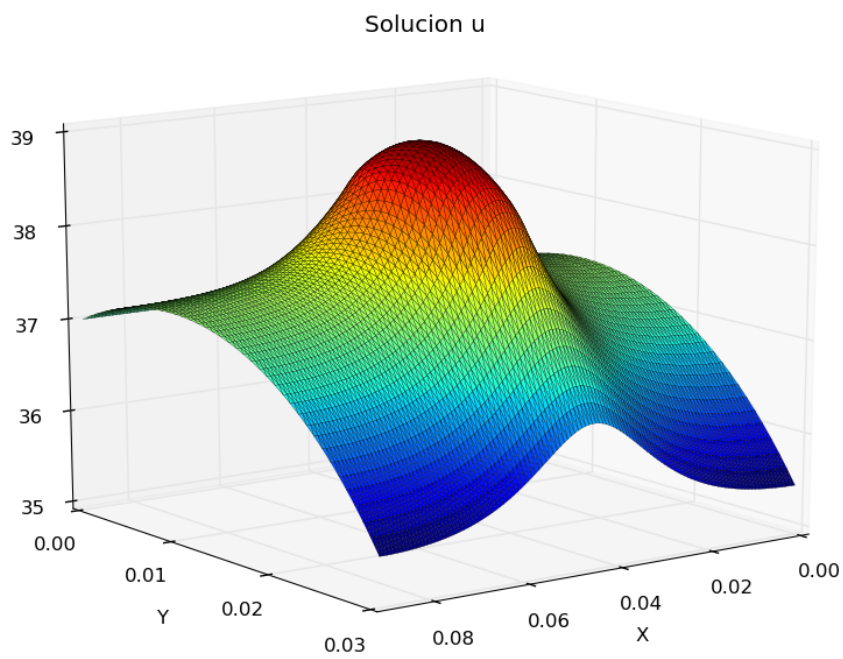


Figura 5.2: Gráfico de la distribución de temperatura obtenido a través del problema directo considerando un tumor centrado en $(0.045, 0.015)$ y de radio 0.0075.

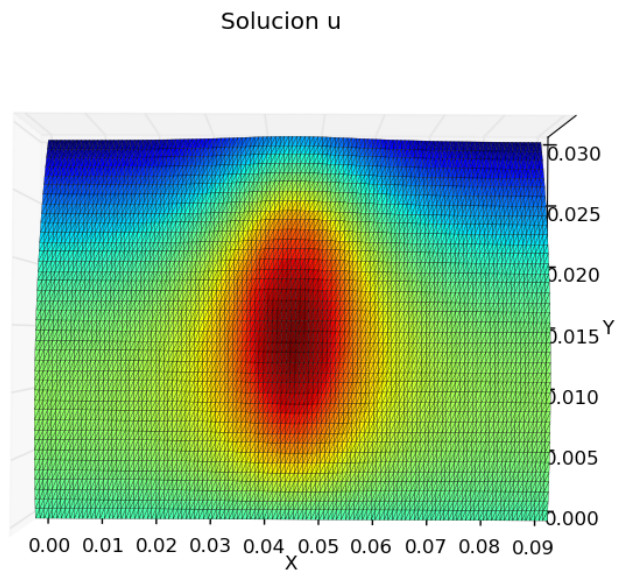


Figura 5.3: Vista en dos dimensiones de la Figura 5.2.

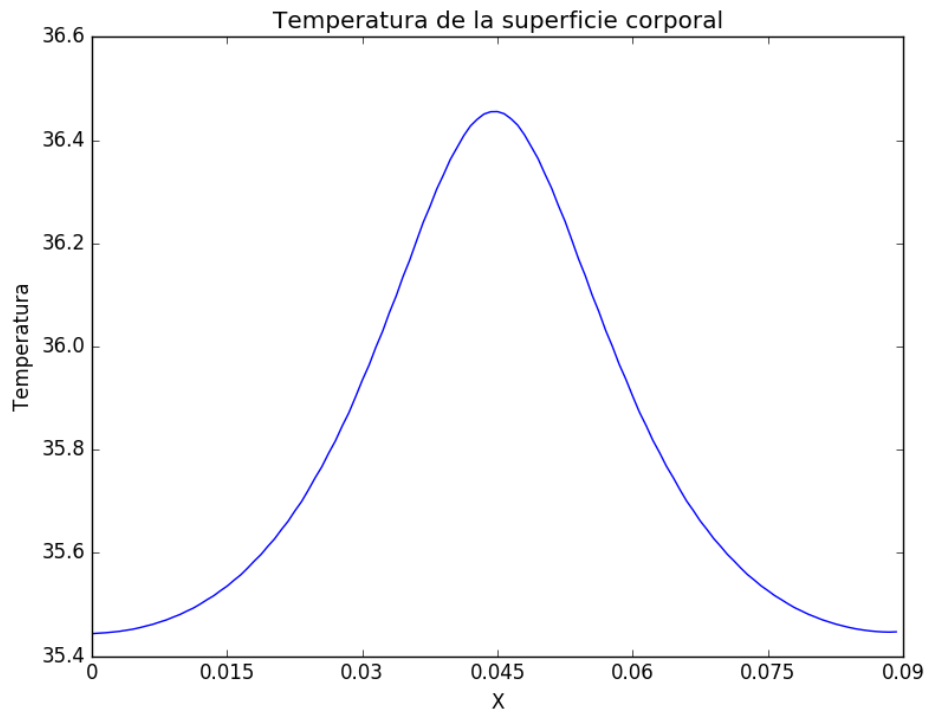


Figura 5.4: Gráfico de la temperatura sobre la piel calculada para un tumor centrado en $(0.045, 0.015)$ y de radio 0.0075.

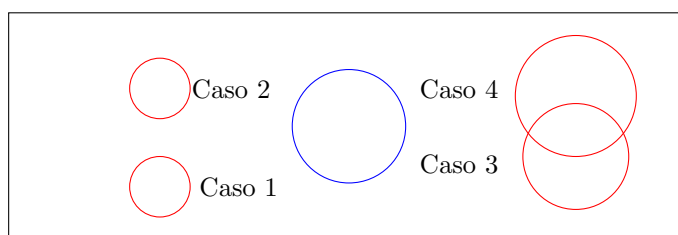


Figura 5.5: Casos de estudio, al centro la posición y radio inicial para los cuatro casos.

Caso 1: $(0.02, 0.007, 0.004)$

Caso 2: $(0.02, 0.02, 0.004)$

Caso 3: $(0.075, 0.01, 0.009)$

Caso 4: $(0.075, 0.02, 0.009)$

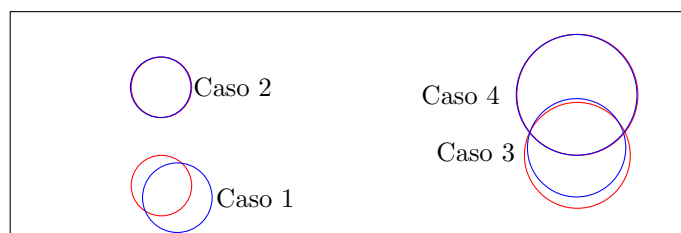


Figura 5.6: Los resultados de los cuatro casos de estudio luego de 800 iteraciones. Con azul las aproximaciones alcanzadas y con rojo la localización y radio exactos.

Aproximaciones:

Caso 1: (0.0221, 0.0054, 0.0046), error: $J(\omega_{800}) = 4 * 10^{-6}$

Caso 2: (0.0199, 0.0200, 0.0040), error: $J(\omega_{800}) = 0,0$

Caso 3: (0.0749, 0.0120, 0.0065), error: $J(\omega_{800}) = 2 * 10^{-6}$

Caso 4: (0.0749, 0.0190, 0.00798), error: $J(\omega_{800}) = 0,0$

Un ejemplo de ejecución por línea de comandos para el caso 1 es el siguiente:

```
detum -d 0.09,0.03 -m 150,50 -t 0.02,0.007,0.004
--imprimir_cada 2 --max_iter 800
```

La ejecución por línea de comandos da la opción de imprimir por salida estándar el estado en el que se encuentran los datos y la aproximación cada un cierto numero de iteración que se especifica con el parámetro `-imprimir_cada`, un ejemplo de estas impresiones es el siguiente:

```
Nro de iteracion: 200
Tiempo transcurrido en minutos: 87 : 25
Posicion inicial: 0.045 0.015 0.0075
Localizacion exacta: (0.02, 0.007, 0.004)
Localizacion aprox: 0.0181721714411 0.00965127214452 0.00319412811491
Error: 1.0135657836e-05
```

En la Figura 5.6 podemos ver los resultados de los casos de estudio con 800 iteraciones. En todos los casos se dibuja con azul la aproximación que arrojó la última iteración.

Los gráficos 5.7 y 5.8 muestran la evolución del error a través de las iteraciones. El gráfico 5.8 tiene un acercamiento en la altura 150 en el eje x que corresponde a la iteración 300 en adelante, para poder apreciar mejor las curvas del error cuando este es mas pequeño.

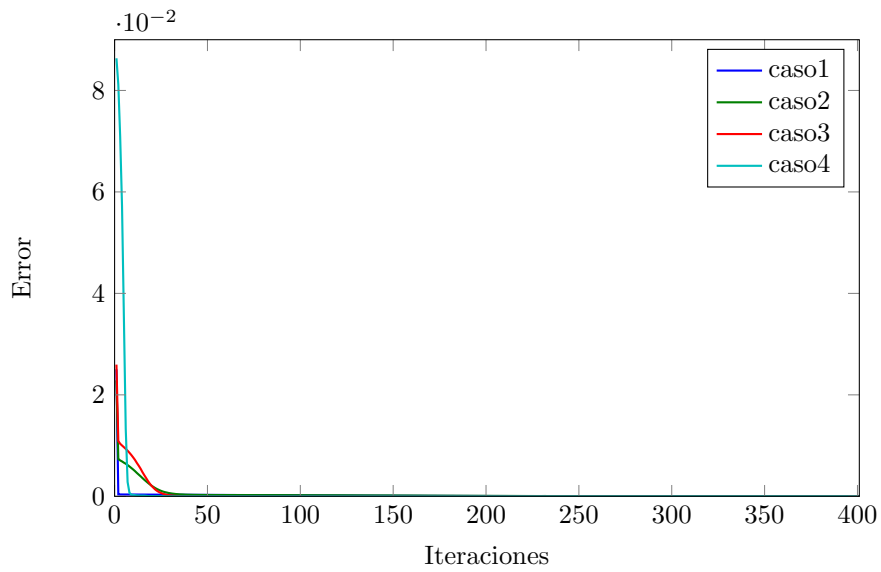


Figura 5.7: Gráfico del error de cada caso según la iteración.

Caso 1: Tiempo de ejecución: 376 minutos 19 segundos.

Caso 2: Tiempo de ejecución: 371 minutos 27 segundos.

Caso 3: Tiempo de ejecución: 376 minutos 6 segundos.

Caso 4: Tiempo de ejecución: 375 minutos 14 segundos.

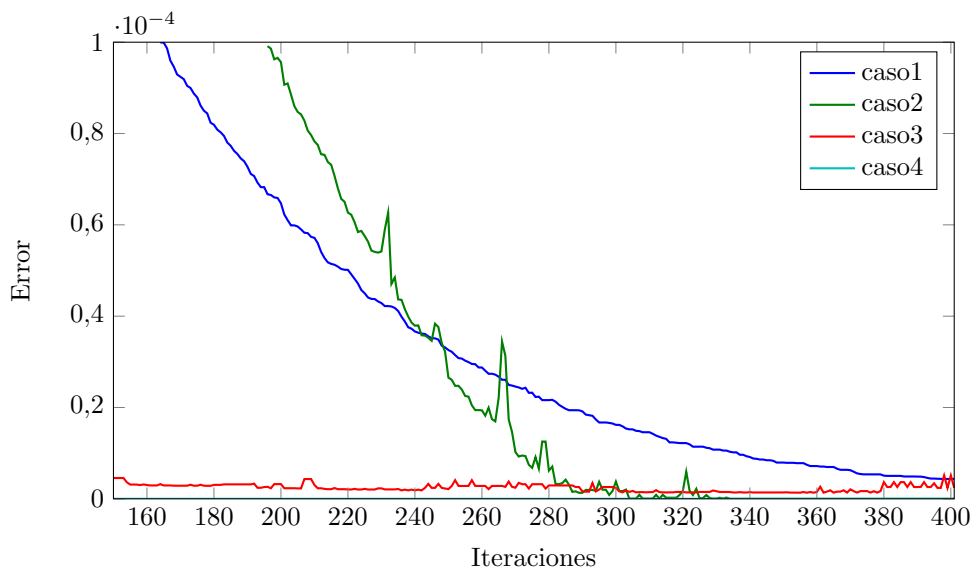


Figura 5.8: Gráfico del error según la iteración, truncado en el eje y y en el eje x .

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo repasamos los objetivos propuestos y los resultados obtenidos. Por último veremos nuevas líneas de estudio para trabajo futuro.

6.1. Conclusiones

Recapitulando, el principal objetivo de este trabajo era, desarrollar un algoritmo numérico para estimar la localización, el centro y radio de un tumor a partir de la temperatura sobre la piel. Este problema fue abordado en dos etapas. En una primera instancia se modeló el proceso de transferencia de calor en tejidos vivos con el cual se plantea y posteriormente se resuelve el PD mediante el FEM. En una segunda instancia se plantea el PI el cual resolvemos mediante el método del gradiente o de máximo descenso.

Los resultados obtenidos para el problema directo fueron muy buenos por lo que podemos apreciar en los gráficos de temperatura de la sección 5.5 y en base a otros experimentos realizados durante el desarrollo, la temperatura se amolda a la ubicación y radio del tumor. Todo esto gracias al estudio hecho sobre la ecuación de calor de Pennes [8].

Una de las principales novedades de este trabajo, era implementar y aplicar el método de elementos finitos en la resolución del problema directo. Comparando con la resolución numérica de este problema utilizando diferencias finitas realizada en [10], podemos concluir que nuestro método numérico nos proporciona una resolución igual o aun más precisa.

Con respecto al problema inverso, de acuerdo a los resultados obtenidos en los experimentos numéricos (ver capítulo 5), podemos concluir que se logró el objetivo inicial ya que en los cuatro casos de estudio el tumor aproximado es muy similar a la localización y tamaño del tumor real. Los cuatro casos se corrieron con 800 iteraciones y podemos apreciar que el error alcanzado es bajo y vemos que va entre $4 * 10^{-6}$ el más alto y 0 el más bajo, por lo tanto la precisión es buena. La calidad de la aproximación depende de la ubicación y del radio del tumor real. Este es un tema que lo discutimos en la siguiente sección.

También concluimos que la plataforma elegida para el desarrollo del software, la cual es Python con el paquete numpy, ofreció un entorno adecuado para la implementación de la solución propuesta y su desempeño al momento de la ejecución fue muy bueno.

A continuación exponemos las dificultades y limitaciones con las que nos encontramos. Con diferentes casos de estudio que ejecutamos notamos que a medida que el tumor se encuentra más cerca de los límites del dominio y su radio es más pequeño, la aproximación que se consigue tiene un error mayor. Con respecto al problema directo un objetivo menos prioritario era analizar la posibilidad de paralelizar la solución, ya que hallar la solución a este problema implica la solución de un sistema de ecuación lineales mediante matrices. Los resultados no fueron los esperados y la API que usamos para paralelizar nos dio algunos problemas. Otra limitación del trabajo es que, a pesar que el algoritmo es derivado, no tenemos un conjunto de test definido el cual nos permita verificar rápidamente algunos casos.

6.2. Trabajo futuro

La siguiente es una lista de ideas para trabajo futuro con respecto a los temas visto en esta tesis y a los problemas que surgieron.

1. Investigar qué tipo de sistema de ecuaciones es el construido en esta tesis y cuál es el mejor algoritmo para paralelizar. Resolver si hay forma de que sea más eficiente mediante paralelización.
2. Mejorar la elección del paso α^k siguiendo las ideas introducidas por Barzilai y Borwein [15], conocido como método espectral. O investigar algún otro método para calcularlo.
3. Ya que no sabemos si la solución es única. Investigar cómo podría acotarse el conjunto de tumores posibles. Una idea es tener más mediciones sobre la piel en distintos ángulos.
4. Conseguir termografías y hacer casos de estudio sin datos artificiales.
5. Implementar un conjunto de test para el paquete el cual nos garantice casos de uso esenciales. De esta forma es mas rápido verificar en caso de una modificación.
6. Proponer un diseño orientado a objetos en caso de ser necesario.

Bibliografía

- [1] Bowman H.F., Estimation of tissue blood flow, in: A. Shitzer, R.C. Eberhart (Eds.), Heat Transfer in Medicine and Biology, vol I, Plenum Press, NY, 1985.
- [2] Chato J.C., Measurement of thermal properties of biological materials, in: A. Shitzer, R.C. Eberhart (Eds.), Heat Transfer in Medicine and Biology, vol I, Plenum Press, NY, 1985, pp.167-173.
- [3] Chen M.M., Pedersen C.O. and Chato J.C, On the feasibility of obtaining three dimensional information from thermographic measurements, ASME Journal of Biomechanical Engineering, 99 (1977), pp.58-64.
- [4] Lawson R.N., Implications of surface temperatures in the diagnosis of breast cancer, Canadian Medical Association Journal, vol. 75 (1956), pp.309-310.
- [5] Lawson R.N. and Chughtai M.S., Breast cancer and body temperatures, Canadian Medical Association Journal, vol. 88 (1963), pp.68-70.
- [6] Miyakawa M. and Bolomey J.C.(Eds), Non-Invasive Thermometry of the Human Body, CRC Press, Boca Raton, 1996.
- [7] Santa Cruz G.A., Bertotti J., Marín J., González S.J., Gossio S., Alvarez D., Roth B.M.C., Menéndez P., Pereira M.D., Albero M., Cubau L., Orellano P. and Liberman S.J., Dynamic infrared imaging of cutaneous melanoma and normal skin in patients treated with BNCT, Vol. 67 (2009), s54-s58.
- [8] Pennes H., Analysis of tissue and arterial blood temperature in the resting human forearm, J. Appl. Physiol. 1 (1948), pp.93-122.
- [9] Haberman R., Elementary applied partial differential equations with Fourier series and boundary value problems, second edition.
- [10] Agnelli J.P. and Turner C.V., Estimación de parámetros y clasificación de datos : aplicaciones biomédicas. Universidad Nacional de Córdoba. Facultad de Matemática, Astronomía y Física, 2011.
- [11] Evans L.C., Partial Differential Equations, American Mathematical Society, 1998.
- [12] Agnelli J.P., Padra C. and Turner C.V., Shape optimization for tumor tumor location Enviado: Computers and Mathematics with Applications, 2010.

- [13] Keller J.B., Inverse Problems, The American Mathematical Monthly, Vol. 83 (1976), No.2, pp. 107-118
- [14] Arias A. and Durango A., Ingeniería y Arquitectura del Software: 2ª Edición, IT Campus Academy, Jan 11, 2016, ISBN-13: 978-1523365487
- [15] Barzilai J. and Borwein J.M., Two point step size gradient methods, IMA Journal of Numerical Analysis, Vol. 8 (1988), Number 1, pp. 141-148.