Modelización y simulación numérica de los fenómenos de transferencia de calor en una tarjeta electrónica de uso espacial

Mataix Caballero, Diego

Abstract

Del estudio de los fenómenos de transferencia de calor, tanto en términos de conducción como de radiación, surgen diversas opciones a la hora de modelizar un problema físico, cada una con un cierto nivel de complejidad. El estudio aquí desarrollado comprende el análisis de diversos modelos analíticos así como numéricos, para la resolución de la ecuación del calor para una tarjeta electrónica de uso espacial. Se realiza inicialmente una primera aproximación, considerando solo la transmisión de calor por conducción, pasando por la implementación de la radiación al problema, para finalmente desarrollar y resolver el modelo del problema térmico bidimensional a través del método de diferencias finitas. Finalmente, se comparan los resultados para determinar la eficacia de los métodos implementados en este estudio.

Palabras clave: Control térmico; Transferencia de calor; Conducción; Radiación; Simulación numérica; Tarjeta electrónica;

Índice

Índice de Figuras

Índice de Tablas

1. Introducción

El estudio consiste en la modelización y simulación numérica de los fenómenos de transferencia de calor en una tarjeta electrónica que consta de tres circuitos integrados (IC) que disipan 5 W cada uno. La geometría de la tarjeta está representada en las Figuras 1 y 2. Como se puede observar esta es simétrica en los ejes x e y. Además, la PCB tiene un recubrimiento de cobre por cada lado, que en una de las caras es continuo, y en la otra, donde van montados los IC, sólo ocupa el 10 % de la superficie.

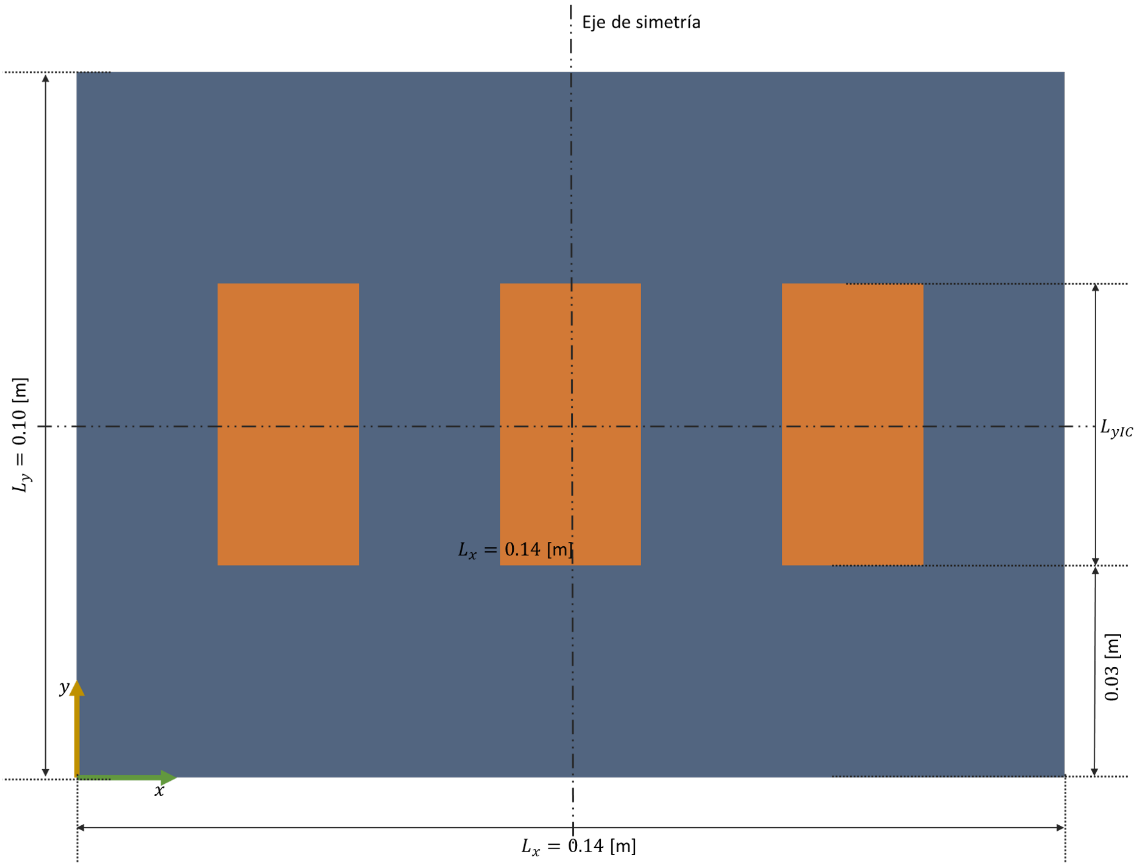

Figura 1: Representación esquemática de la tarjeta electrónica estudiada. Vista frontal. El color naranja representa los IC y el azul el PCB.

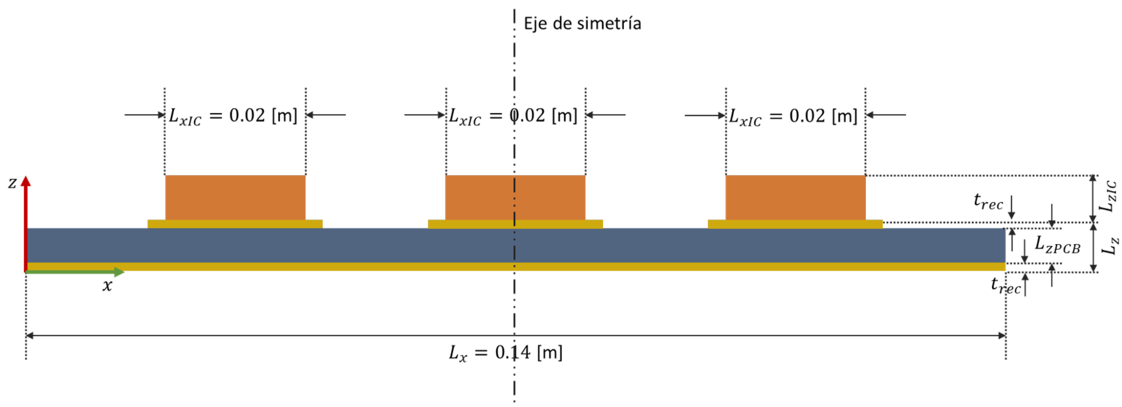

Figura 2: Representación esquemática de la tarjeta electrónica estudiada. Vista lateral. El color naranja representa los IC, el amarillo el recubrimiento de cobre y el azul el PCB.

Las Tablas 1, 2 y 3 muestran las propiedades de conductividad, densidad, capacidad térmica y geometría de los distintos componentes que componen la tarjeta electrónica.

Tabla 1: Propiedades del FR4.

Geometría	[m]
L_x	0.140
L_y	0.100
L_z	0.0015
Conductividad	[W/(m·K)]
k_{xy}	0.5
k_z	0.25
Densidad	[kg/m^3]
ρ	3000
Capacidad térmica	[J/(kg·K)]
c	1850

Tabla 2: Propiedades del cobre.

Geometría	[m]
L_x	0.140
L_y	0.100
L_z	0.0015
Conductividad	[W/(m·K)]
k	395
f (parte superior)	0.1
Densidad	[kg/m^3]
ρ	8260
Capacidad térmica	[J/(kg·K)]
c	385

Tabla 3: Propiedades de los circuitos integrados.

Geometría	[m]
L_x	0.020
L_y	0.040
L_z	0.003
Separación	0.020
Conductividad	[W/(m·K)]
k	50
Capacidad térmica	[J/K]
C	20
Potencia	[W]
$\dot{W}$	5

Por otro lado cabe destacar que se supondrá que los lados cortos, que se encuentran en las coordenadas $x = 0$ y $x = L_x$ (Figura 1), de la PCB tienen contacto térmico perfecto con paredes permanentemente a 298.15 K, y que los otros dos bordes están térmicamente aislados.

El estudio se realiza en primer lugar con un modelo muy simplificado, unidimensional y asumiendo disipación uniforme y considerando solo los fenómenos de transferencia de calor por conducción. A este modelo se le va incorporando mayor complejidad, pasando a considerar la disipación no uniforme y la transmisión de calor por radiación, para finalmente plantear y resolver el modelo térmico unidimensional y bidimensional de manera numérica a través del método de diferencias finitas (FDM). [1]

Aunque la implementación del método FDM es relativamente sencilla, puede presentar dificultades en casos en los que la geometría es compleja, por lo que una gran parte de los modelos de trasferencia de calor no se basan en el método de diferencias finitas. En cambio, se utilizan métodos como el de elementos finitos (FEM), o, para situaciones en las que los acoplamientos radiativos son complejos, se recurre al método de los elementos discretos con seguimiento estadístico de rayos. [2] Por lo que en áreas como la del control térmico espacial (CTE) se suele recurrir a programas tales como ESATAN. Aunque su funcionamiento es interesante, su aplicación está fuera de los objetivos de este estudio.

Este estudio recoge finalmente una comparativa entre los modelos estudiados, con el objetivo de determinar cómo de eficaces son los modelos unidimensionales respecto a los bidimensionales, así como comparar los resultados obtenidos de manera analítica y numérica.

2. Metodología

En este apartado se exponen las distintas metodologías empleadas para la resolución del problema, cada una contando con un nivel añadido de complejidad. Se comienza considerando sólo la transferencia de calor por conducción y que la tarjeta sólo evacua calor por los bordes. Para una primera aproximación se considera además que la disipación está uniformemente repartida en la PCB y los IC no influyen, y se prosigue el análisis con un modelo unidimensional en el que los IC llegan hasta los bordes aislados. Posteriormente se incluye el efecto de la transferencia de calor por radiación en la ecuación del calor, esto se hace primero a través de un análisis unidimensional, asumiendo disipación uniforme y linealizando las pérdidas radiativas, lo que permite determinar la solución analíticamente, seguido de la resolución del modelo FDM sin linealizar y con disipación no uniforme. Finalmente, se desarrolla y se resuelve el modelo térmico bidimensional del problema. Toda la metodología aquí descrita ha sido implementada a través de un programa de MATLAB, que se adjunta en los Anexos A y B.

2.1. Perfil de temperatura en la tarjeta electrónica considerando solo los fenómenos de conducción

En primer lugar se considera únicamente la transferencia de calor por conducción, que viene definida por la Ecuación 1:

$$\overrightarrow{q} = K\Delta T \left\{ \; conducción \quad \overrightarrow{q} \equiv -k\nabla T \quad \left(unidimensional: \dot{q} = -k\tfrac{\partial T}{\partial x}\right) \right. \quad , \tag{1}$$

donde $\overrightarrow{q}$ es la densidad de flujo de calor, K es el coeficiente de transmitancia, k es la conductividad térmica, ΔT es la diferencia de temperatura y $\frac{\partial T}{\partial x}$ es la derivada parcial de la temperatura respecto a la posición en el eje x. En este caso se considera la ecuación en su forma unidmensional así como que la tarjeta sólo evacua calor por los bordes.

2.1.1. Primera aproximación

En esta primera aproximación se pretende determinar la temperatura máxima que se alcanzaría si toda la disipación estuviese uniformemente repartida en la PCB y los IC no influyeran. Para ello se plantea la ecuación de la temperatura que resulta de esta forma:

$$T = a + b\,x - \frac{\phi\,x^2}{2\,k}, \tag{2}$$

donde ϕ es la disipación volumétrica, y a y b son los coeficientes que vienen dados por las Ecuaciones 3 y 4:

$$a = T_b, \tag{3}$$

$$b = \frac{Q}{A\,k}, \tag{4}$$

donde Q es la potencia disipada y A es el área transversal.

Para aplicar esta metodología al problema descrito en la Sección 1, se recurre al valor de la conductividad térmica efectiva, k_{eff}. Este parámetro nos permite simplificar el modelo, y viene dado por:

$$k_{eff} = \frac{\Sigma k_i A_i f_i}{A_{eff}} \tag{5}$$

donde k_i es el valor de la conductividad térmica efectiva, y A_i es el espesor de la capa.

2.1.2. Modelo unidimensional

Con el fin de obtener un modelo más cercano a la realidad se plantea un modelo unidimensional en el que los IC llegan hasta los bordes aislados, y se considera la conductividad térmica de los IC en en el límite $k_{IC} \to \infty$ y con la k_{IC} dada.

En este caso es necesario representar las discontinuidades de las propiedades de la tarjeta electrónica, por lo que se requiere determinar la conductividad térmica efectiva en los segmentos de la tarjeta donde se encuentran los IC, así como considerar el área en esas zonas, y que la disipación volumétrica, ϕ, es nula en los segmentos donde no hay circuitos. En la Figura 3 se muestra la división que se ha realizado de la tarjeta en 7 segmentos equiespaciados.

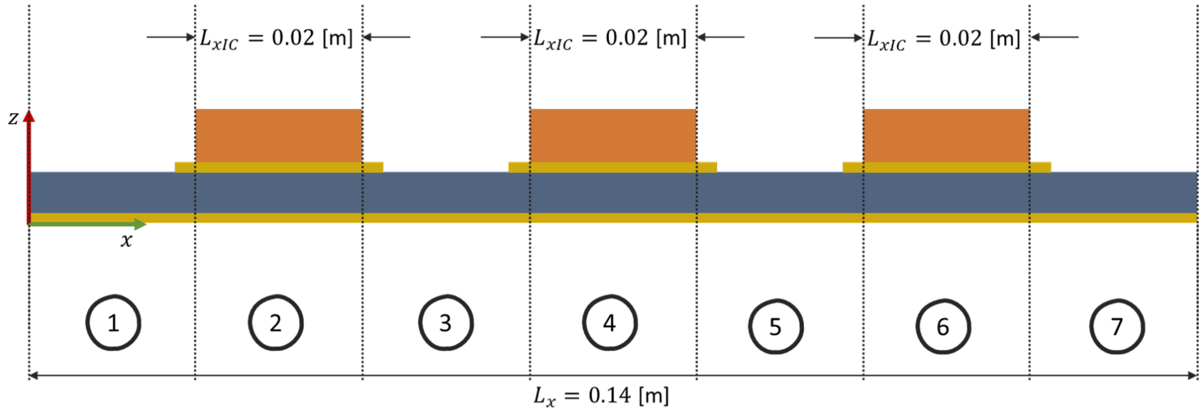

Figura 3: Modelo unidimensional de la tarjeta electrónica estudiada para representar sus discontinuidades.

Por lo tanto, es necesario plantear la Ecuación 2 para los tramos donde se encuentran los circuitos integrados, ya que los circuitos generan calor, y, para los tramos donde no se disipa calor, la Ecuación 2 pasar a ser lineal, por lo que tiene esta forma:

$$T = \frac{\mathrm{Q\,x}}{\mathrm{A\,k}} + a. \tag{6}$$

De este modo se obtiene un perfil de temperaturas en el que, para los segmentos donde no hay circuitos la temperatura sigue una recta, y en los segmentos donde se encuentran los IC la temperatura está descrita por una parábola. En el caso de $k_{IC} \to \infty$ la temperatura en todos los segmentos viene descrita por una recta.

2.2. Transferencia de calor por radiación y conducción

Habiendo realizado un análisis de la transferencia de calor por conducción en el sistema, se desea implementar un modelo que incluya también la transmisión de calor por radiación. De esta forma, los modos de transmisión de calor (Ecuación 7) pasan a ser:

$$\vec{q} = K\Delta T \begin{cases} \textit{conducción} & \vec{q} \equiv -k\nabla T \quad \left(\textit{unidimensional}: \dot{q} = -k\frac{\partial T}{\partial x}, \dot{q} = -k\frac{\partial T}{\partial r}\right) \\ \textit{radiación} \quad \dot{q} & = \varepsilon F \sigma \left(T^4 - T_0^4\right) \left(\varepsilon < 1, F < 1, \sigma = 5{,}67 \cdot 10^{-8}\ \mathrm{W}/\left(\mathrm{m}^2 \cdot \mathrm{K}^4\right)\right) \end{cases}, \tag{7}$$

donde ε es la emisividad media y σ es la constante de Stefan–Boltzmann. Nótese que en la transferencia de calor por radiación las superficies se suponen isotermas y que el campo de temperatura en el interior solo se requiere resolver en problemas de conducción de calor.

2.2.1. Modelo con las pérdidas radiativas linealizadas

Se añade un nivel de dificultad al modelo, pasando a considerar que se transmite calor por radiación, con una emisividad media de 0,7 por el lado de los componentes, y de 0,5 por la cara opuesta, suponiendo la caja electrónica como un cuerpo negro a 318.15 K. En este caso, se implementa la radiación al modelo linealizando las pérdidas radiativas y considerando la disipación como uniforme, por lo que se resuelve el problema de manera analítica. Para ello, se procede desde la ecuación del calor:

$$\begin{aligned} &\rho A \tfrac{\partial T}{\partial t} = \dot{W}_{\text{dis}} + \dot{Q}_{cond} + \dot{Q}_{conv} + \dot{Q}_{\text{rad}} = \phi A \mathrm{d}x - kA\tfrac{\partial T}{\partial x} + kA(\tfrac{\partial T}{\partial x} + \tfrac{\partial^2 T}{\partial x^2}\mathrm{d}x) \\ &-h_p \mathrm{d}x\,(T - T_\infty) - \varepsilon p \mathrm{d}x \sigma \left(T^4 - T_\infty^4\right), \end{aligned} \tag{8}$$

donde ρ es la densidad, $\dot{W}_{\text{dis}}$ es la potencia disipada, $\dot{Q}_{\text{cond}}$ es el flujo de calor por conducción, $\dot{Q}_{\text{conv}}$ es el flujo de calor por convección, $\dot{Q}_{\text{rad}}$ es el flujo de calor por radiación, p es el perímetro bañado y T_∞ es la temperatura del cuerpo negro. Al no haber transmisión de calor por conducción, considerando solamente el régimen estacionario y operando, la Ecuación 8 pasa a ser:

$$0 = \phi A + kA\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} - \varepsilon p \sigma \left(T^4 - T_\infty^4\right). \tag{9}$$

Linealizando las pérdidas radiativas, se llega a la EDO:

$$0 = \phi A + kA\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} - 4\varepsilon p \sigma T_{avg}^3 \left(T - T_\infty\right), \tag{10}$$

donde T_{avg} es la temperatura media usada para linealizar las pérdidas radiativas. Para simplificar la Ecuación 10, se define el parámetro ξ:

$$\xi = 4\,T_{avg}^3\,\varepsilon\,p\,\sigma, \tag{11}$$

por lo que podemos plantear la EDO (Ecuación 10) de esta forma:

$$kA\frac{\mathrm{d}^2(T - T_\infty - \frac{\phi A}{\xi})}{\mathrm{d}x^2} = \xi(T - T_\infty - \frac{\phi A}{\xi}). \tag{12}$$

Al encontrarnos ante una EDO homogénea, la solución es de la forma:

$$(T - T_\infty - \frac{\phi A}{\xi}) = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x}, \tag{13}$$

donde λ_1 y λ_2 son los autovalores y c_1 y c_2 son los coeficientes de la solución. Operando, se llega a que los autovalores están definidos por:

$$\lambda = \sqrt{\frac{\xi}{A\,k}}. \tag{14}$$

Finalmente, se aplican las condiciones de contorno para obtener los coeficientes c_1 y c_2. Cabe destacar que se emplea el sistema de coordenadas definido en la Figura 2. Las condiciones que se imponen son las de temperatura del borde fija a 298.15 K, y que en el punto $L_x/2$ la derivada de la temperatura respecto a la coordenada x es igual a 0, ya que ahí se encuentra el máximo de la parábola que define el perfil de temperaturas. De este modo se consiguen unas expresiones para los coeficientes de la Ecuación 13:

$$c_2 = T_b - T_\infty - \frac{A\,\phi}{\xi\,\left(\mathrm{e}^{-L_x\,\lambda} + 1\right)}, \tag{15}$$

$$c_1 = c_2\,\mathrm{e}^{-L_x\,\lambda}. \tag{16}$$

Por lo que la solución general está definida por:

$$T = c_1\,\mathrm{e}^{\sqrt{\frac{\xi}{A\,k}}\,x} + c_2\,\mathrm{e}^{-\sqrt{\frac{\xi}{A\,k}}\,x} + T_\infty + \frac{A\,\phi}{\xi}. \tag{17}$$

2.2.2. Modelo numérico unidimensional con disipación no uniforme

Con el fin de resolver el caso anterior pero sin linealizar las pérdidas radiativas y considerando la disipación como no uniforme, se recurre al uso de métodos numéricos, en específico se emplea el método de diferencias finitas (FDM), transformando el problema continuo en un problema discreto. Aunque suele discretizarse la ecuación del calor en su forma diferencial, en este estudio se emplea la ecuación en su forma integral. De esta forma, se sustituyen la derivadas espaciales por incrementos centrados, y la derivada temporal por el incremento hacia adelante.

Se define el problema en su caso unidimensional, lo que permite validar el modelo con los resultados analíticos obtenidos en apartados anteriores. Por lo tanto, es necesario discretizar la solución $T(x)$ en $N+1$ puntos espaciales (T_i), a lo largo del eje x de la tarjeta electrónica, obteniéndose una malla espacial equiespaciada que empieza y acaba en los bordes de la tarjeta, pues queremos saber las temperaturas superficiales. Al usarse los valores de las variables en el instante anterior, que es conocido, el método se considera explícito. Por lo tanto a partir del 2° principio de la Termodinámica, se puede demostrar que el método explícito requiere de una discretización temporal fina, mucho mayor que la espacial. En términos del número de Fourier se requiere que este sea menor a 0.5: $Fo < \frac{1}{2}$ [2].

La ecuación en su forma de FDM para un tramo pasa a ser:

$$T_i^{j+1} = T_i^j + DtrcA_i \left[kALapla_i^j + \phi_i^j A_i - phDT_i^j \right], \tag{18}$$

donde los subíndices i determinan la posición espacial y los superíndices j, definen el instante de tiempo. De este modo T_i^j es el término de la temperatura en el punto espacial i y en el instante de tiempo $j+1$ (instante actual que se desea calcular) y T_i^j es el término de la temperatura en el punto espacial i y en el instante de tiempo j (instante anterior). El procedimiento consiste en imponer una distribución incial conocida de temperaturas, $T_i^{j=1}$ (en este caso se considera que toda la tarjeta se encuentra a la temperatura de los bordes cortos, 298.15 K), y se calcula la nueva distribución en cada instante posterior. En los nodos que se encuentran en los extremos, en $i = 1$ e $i = N+1$, se aplican las condiciones de contorno conocidas.

Con el fin de hacer la ecuación 18 más manejable, se definen 3 macro-variables: el vector $DtrcA_i$, y las matrices $kALapla_i^j$ y $phDT_i^j$, que representan los términos debidos a la conducción y la convección y radiación respectivamente. Su definición se muestra en las Ecuaciones 19, 20 y 21:

$$DtrcA_i \equiv \frac{\Delta t}{\rho_i c_i A_i}, \tag{19}$$

$$kALapla_i^j \equiv k_{i+} A_{i+} \frac{T_{i+1}^j - T_i^j}{(\Delta x)^2} - k_{i-} A_{i-} \frac{T_i^j - T_{i-1}^j}{(\Delta x)^2}, \tag{20}$$

$$phDT_i^j \equiv p_i \left[h_i \left(T_i^j - T_\infty \right) + \varepsilon_i \sigma \left(T_i^{j4} - T_\infty^4 \right) - E_i \right], \tag{21}$$

donde los términos h_i y E_i son nulos, y k_{i+}, A_{i+}, k_{i-} y A_{i-} son la media entre los términos i e $i+1$ en los primeros dos casos, y entre los términos i e $i-1$ en los otros dos casos. Cabe destacar, que la tarjeta electrónica no cuenta con propiedades uniformes, por lo que se tendrán que representar las discontinuidades de las propiedades como la conductividad efectiva, la capacidad térmica, el área, el volumen y la disipación volumétrica en forma vectorial.

Con el objetivo de validar el modelo numérico de diferencias finitas, previo a realizar el estudio incluyendo la radiación, se emplea el modelo unidimensional considerando solo fenómenos de conducción, para los casos de disipación uniforme y no uniforme. De este modo se comparan los resultados con los obtenidos con la metodología descrita en la Sección 2.1. Si los resultados concuerdan entre sí, se considera que el modelo está bien planteado y se prosigue resolviendo el problema térmico unidimensional, para el caso de régimen transitorio y estacionario.

2.3. Modelo térmico bidimensional

De manera similar al caso anterior, se resuelve el problema térmico bidimensional estacionario. Este modelo ya no asume que los IC llegan hasta los bordes aislados, por lo que es necesario representarlos de manera bidimensional, y redefinir las discontinuidades de las propiedades descritas en la Sección 2.2.2. En este caso contamos con una matriz espacio-temporal de temperaturas que tiene la forma $T(1 : M + 1, 1 : N_y + 1, 1 : N_x + 1)$, siendo N_x y N_y los números de tramos en la discretización espacial en los ejes x e y respectivamente. Además, en los extremos térmicamente aislados, se requiere actualizar en cada iteración la temperatura en el borde.

Respecto al mallado del problema, la Figura 4, muestra claramente la principal dificultad que se encuentra a la hora de implementarse. Ya que no en todos los casos el número de elementos y de nodos hará que estos últimos coincidan exactamente donde se encuentran las discontinuidades. Por lo tanto, es necesario interpolar, en esos puntos, los valores de las propiedades de conductividad efectiva, capacidad térmica, disipación volumétrica, etc. De esta forma se evita tener que realizar una discretización espacial muy fina que requiera a su vez de una discretización temporal aún mayor, que resulte en unos requisitos de esfuerzo computacional y memoria que estarían fuera del alcance de este estudio.

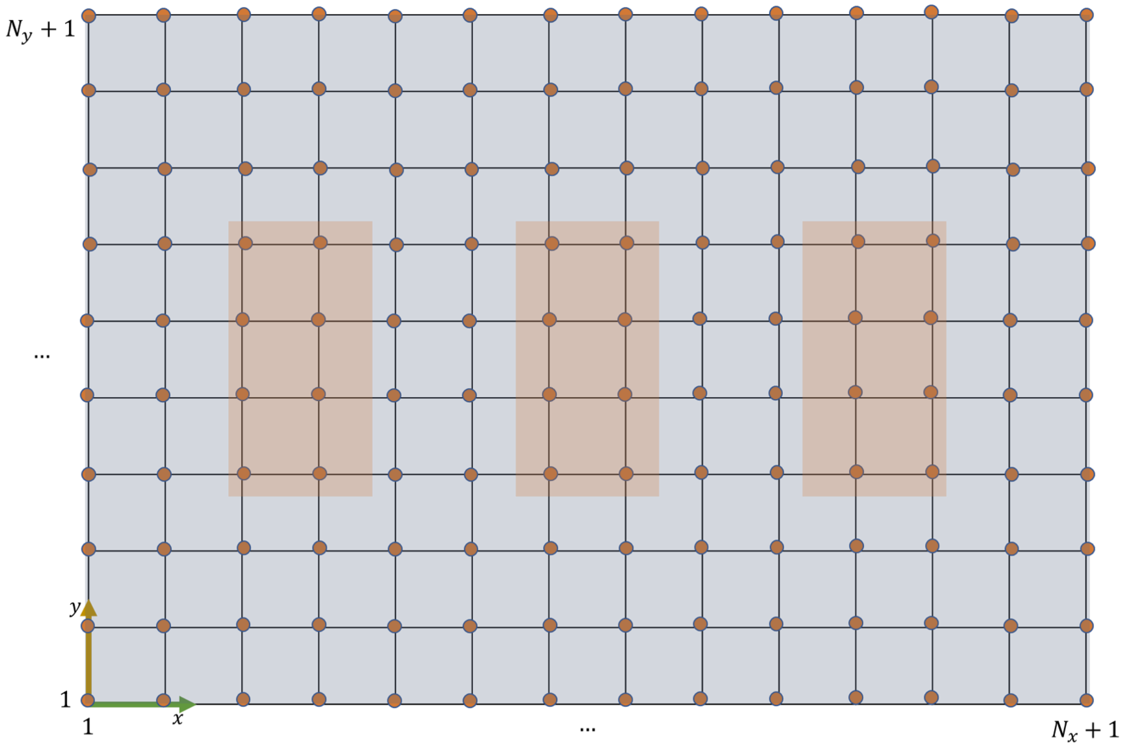

Figura 4: Mallado del modelo bidimensional de la tarjeta electrónica estudiada.

En el caso del problema térmico bidimensional, la Ecuación 18 pasa a ser:

$$T_{j,i}^{k+1} = T_{j,i}^{k} + Dtrcz_{j,i}\left[kzLaplx_{j,i}^{k} + kzLaply_{j,i}^{k} + \phi_{j,i}^{k} z_{j,i}^{k} - hDT_{j,i}^{k}\right], \tag{22}$$

donde se han definido 4 macro-variables para hacer más asequible la expresión. Estas 4 variables son: el vector $Dtrcz_{j,i}$, las matrices $kzLaplx_{j,i}^{k}$ y $kzLaply_{j,i}^{k}$ que representan los términos debidos a la conducción en el eje x e y, y la matriz $hDT_{j,i}^{k}$ que contiene los términos debidos a la radiación. Su definición se muestra en las Ecuaciones 19, 20 y 21:

$$Dtrcz_{j,i} \equiv \frac{\Delta t}{\rho_{j,i} c_{j,i} z_{j,i}}, \tag{23}$$

$$kzLaplx_{j,i}^{k} = k_{j,i+} z_{j,i+} \frac{T_{j,i+1}^{k} - T_{j,i}^{k}}{(\Delta x)^2} - k_{j,i-} z_{j,i-} \frac{T_{j,i}^{k} - T_{j,i-1}^{k}}{(\Delta x)^2}, \tag{24}$$

$$kzLaply_{j,i}^{k} = k_{j+,i} z_{j+,i} \frac{T_{j+1,i}^{k} - T_{j,i}^{k}}{(\Delta y)^2} - k_{j-,i} z_{j-,i} \frac{T_{j,i}^{k} - T_{j-1,i}^{k}}{(\Delta y)^2}, \tag{25}$$

$$hDT_{j,i}^{k} = (\varepsilon_{ji,up} + \varepsilon_{ji,dw})\sigma\left(T_{j,i}^{k4} - T_{\infty}^{4}\right), \tag{26}$$

donde se han excluido los términos debidos a los fenómenos de conducción, y, a diferencia del caso unidimensional, se emplean los términos $z_{j,i}$ referentes a el espesor en cada punto de la tarjeta.

3. Resultados

Una vez realizado el estudio, se presentan los resultados obtenidos para los modelos descritos en la Sección 2, con el objetivo de comparar los perfiles de temperaturas y las temperaturas máximas que se consiguen con cada modelo.

3.1. Perfil de temperatura en la tarjeta electrónica considerando solo los fenómenos de conducción

En esta sección se exponen los resultados de los modelos analíticos descritos en las Secciones 2.1.1 y 2.1.2, que consideran únicamente la transmisión de calor por conducción.

3.1.1. Primera aproximación

En la primera aproximación se obtiene una distribución de temperaturas de forma parabólica, que presenta un máximo en el centro del eje x de la tarjeta, como se muestra en la Figura 5.

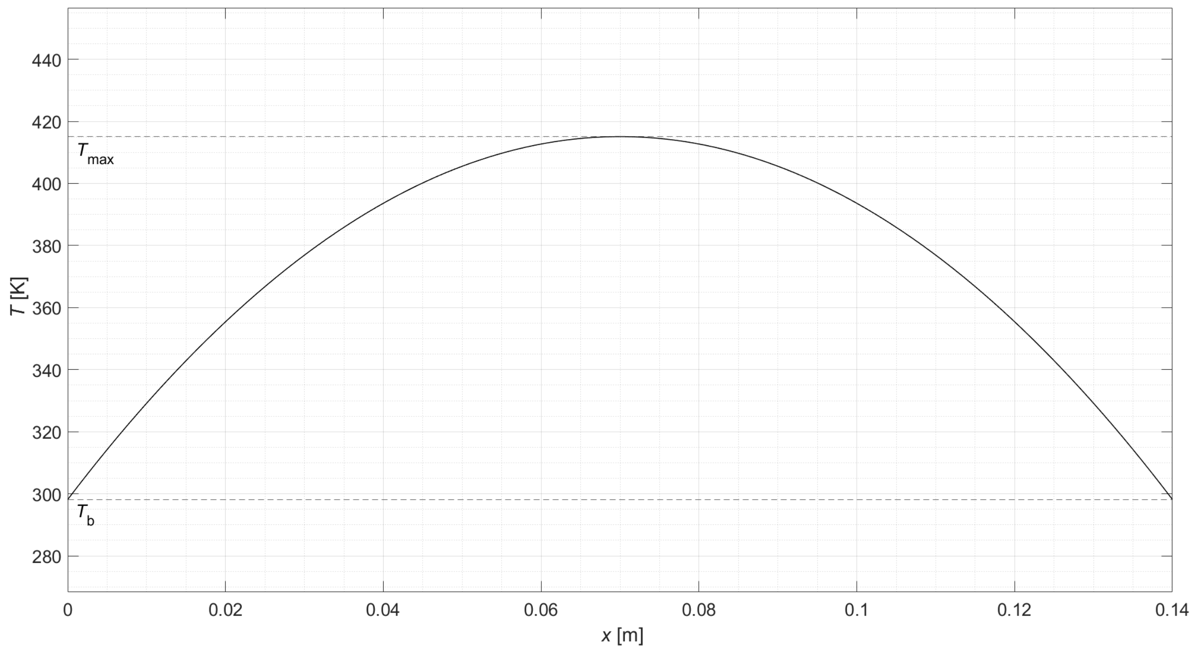

Figura 5: Perfil de temperaturas en régimen estacionario asumiendo solo los fenómenos de conducción y disipación uniforme.

Con las simplificaciones efectuadas para obtener esta solución a través de un método analítico, se llega a que la temperatura máxima que se alcanza, $T_{\max}$, es de 415.1 K.

3.1.2. Modelo unidimensional

Dividiendo la tarjeta en 7 segmentos equiespaciados con el fin de representar las discontinuidades y la no uniformidad de la disipación, se llega a una mejor aproximación donde la $T_{\max}$ alcanzada es significativamente menor.

Realizando este análisis, y considerando que en el límite $k_{IC} \to \infty$, se llega a un perfil de temperaturas compuesto por 7 rectas, en las que para los segmentos donde se encuentran los circuitos integrados la temperatura es constante, como se muestra en la Figura 6. Donde k_{IC} es la conductividad térmica en los tramos que cuentan con IC. En este caso, se obtiene una $T_{\max}$ de 383.8 K.

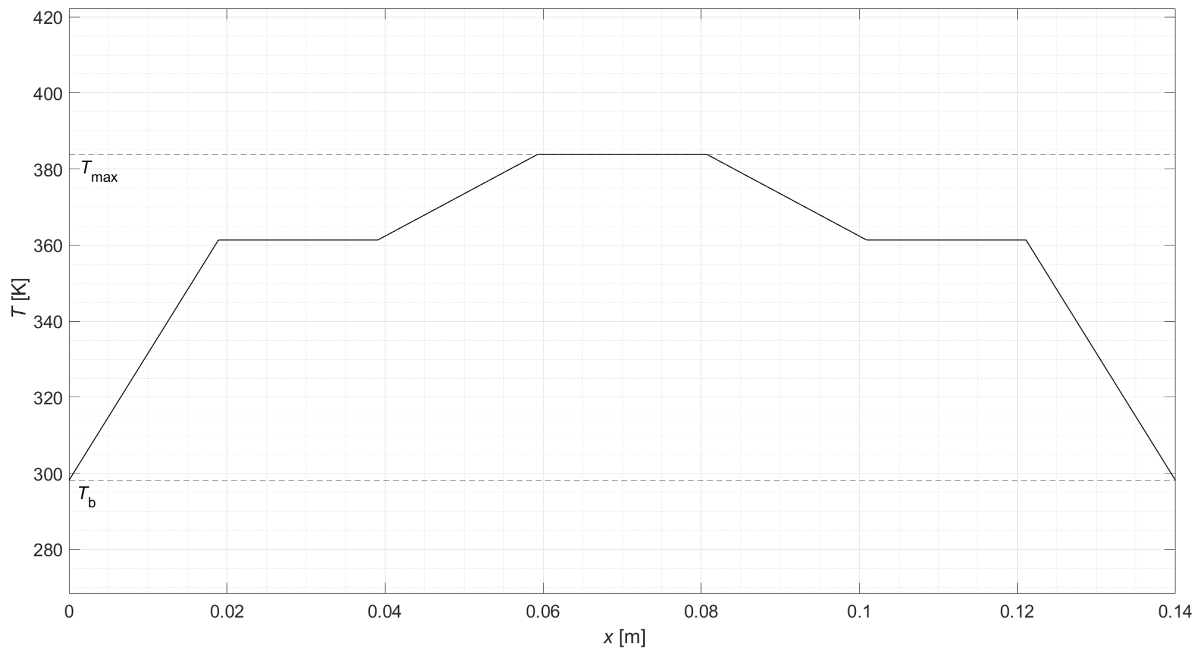

Figura 6: Perfil de temperaturas en régimen estacionario asumiendo solo los fenómenos de conducción y disipación no uniforme, en el límite $k_{IC} \to \infty$.

Por otro lado, considerando la k_{IC} dada, se obtiene un perfil de temperaturas en el que, para los segmentos donde no hay circuitos la temperatura sigue una recta, y en los segmentos donde se encuentran los IC la temperatura está descrita por una parábola (Figura 7). De este modo, se obtiene una $T_{\max}$ de 390.4 K, casi 7 K superior a la del caso de $k_{IC} \to \infty$, como es de esperar.

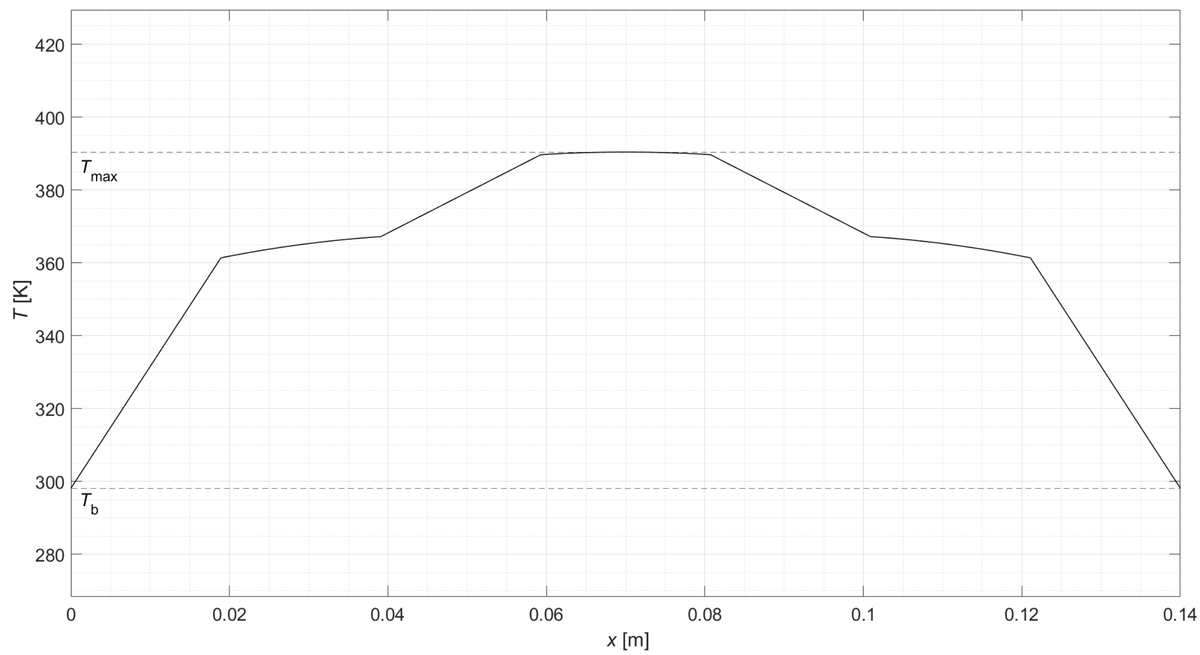

Figura 7: Perfil de temperaturas en régimen estacionario asumiendo solo los fenómenos de conducción y disipación no uniforme, considerando la k_{IC} dada.

Al ser una mejor aproximación al problema real, de aquí en adelante se considera solo la solución en la que se emplea la k_{IC} dada.

3.2. Transferencia de calor por radiación y conducción

En esta sección se exponen los resultados obtenidos con la metodología descrita en la Sección 2.2, es decir, se consideran los modos de transmisión de calor por conducción y por radiación.

3.2.1. Modelo con las pérdidas radiativas linealizadas

Linealizando las pérdidas y considerando disipación uniforme (Sección 2.2.1), se obtiene una EDO que, al resolverla, se llega a una expresión que describe el perfil de temperaturas en la tarjeta electrónica.

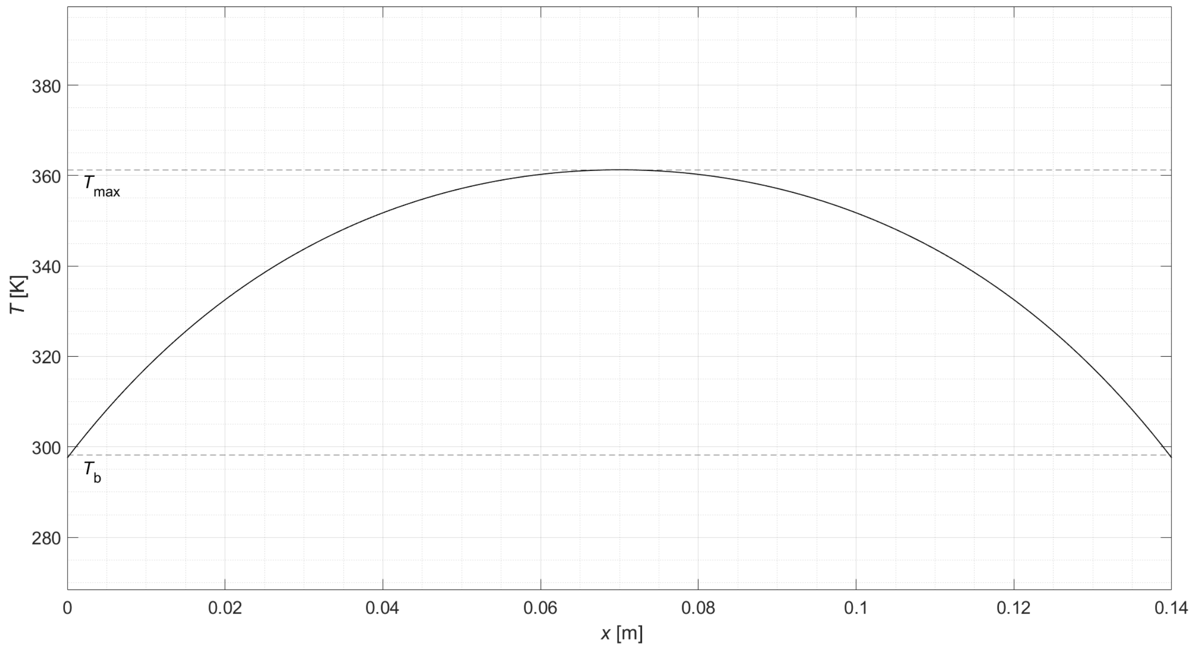

Figura 8: Perfil de temperaturas en régimen estacionario asumiendo disipación uniforme y linealizando las pérdidas radiativas.

En la Figura 8 se observa como, al haber otro modo de transmisión de calor, la temperatura máxima alcanzada, $T_{\max}$, es significativamente menor que la obtenida en la Sección 3.1.1, siendo de 361.3 K, casi 54 K menor que en el otro caso estudiado de disipación uniforme.

3.2.2. Modelo numérico unidimensional con disipación no uniforme

Con el fin de obtener un resultado para el que no sea necesario aplicar simplificaciones tales como la disipación uniforme o la linealización de las pérdidas radiativas, se plantea un modelo de diferencias finitas siguiendo la metodología descrita en la Sección 2.2.2.

En primer lugar se muestran los mapas unidimensionales de las propiedades de la tarjeta electrónica en la Figura 9. Estos mapas representan las discontinuidades de la conductividad efectiva, la capacidad térmica, el área, el volumen y la disipación volumétrica en forma vectorial. En la Figura 9 se aprecian claramente los distintos segmentos en los que se divide la PCB.

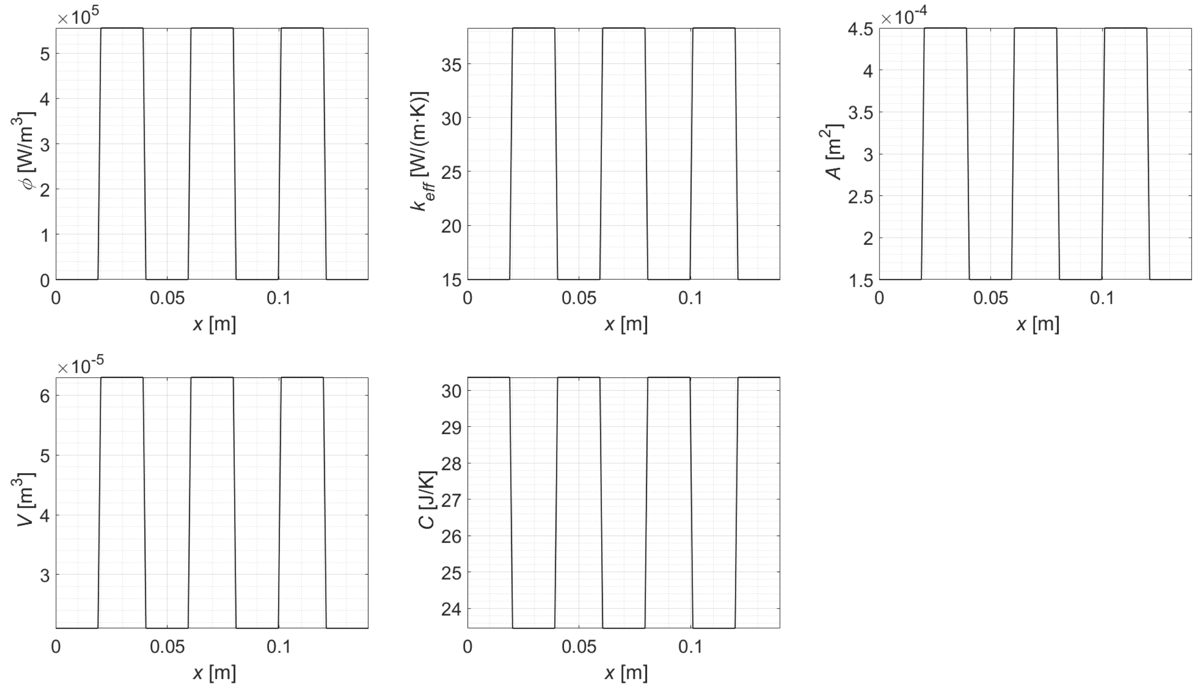

Figura 9: De izquierda a derecha y de arriba a abajo: mapa de disipación volumétrica, ϕ [W/m^3], de conductividad térmica longitudinal, k_{eff} [W/(m·K)], de área transversal, A [m^2], de volumen, V [m^3], y de capacidad térmica, C [J/K].

Haciendo uso de estos vectores se pretende resolver el problema térmico unidimensional primeramente sin considerar las pérdidas radiativas, con el fin devalidar el modelo numérico antes de obtener resultados nuevos. Los resultados se comparan con los obtenidos en la Sección 3.1.2. En la Figura 10 se muestra el perfil de temperaturas en régimen estacionario asumiendo disipación no uniforme para la resolución numérica y analítica del problema planteado en la Sección 2.1 y 2.2. Se puede observar que los resultados que no incluyen pérdidas radiativas, para la resolución analítica y numérica son muy similares.

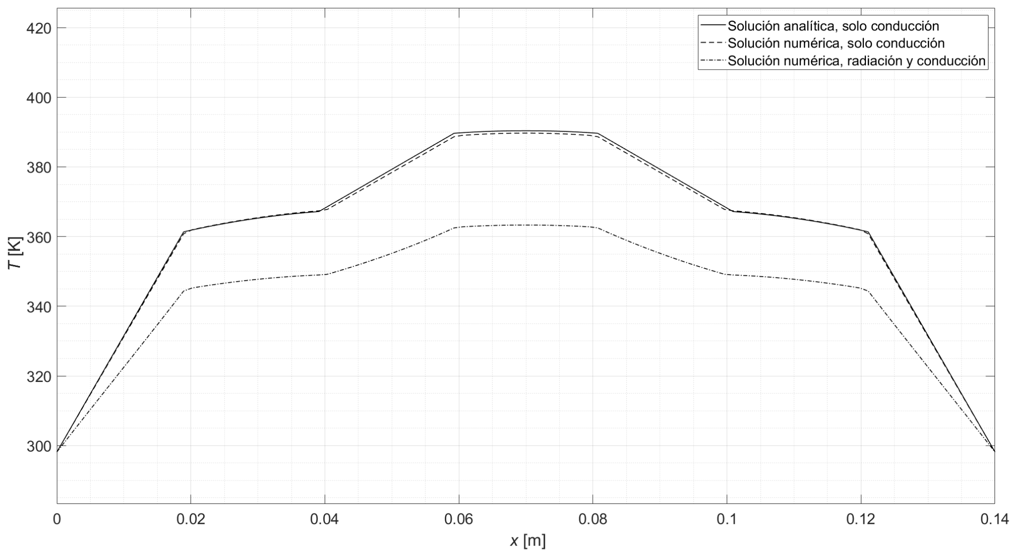

Figura 10: Perfil de temperaturas en régimen estacionario asumiendo disipación no uniforme para la resolución numérica y analítica del problema planteado en la Sección 2.1 y 2.2.

En la Figura 11 se muestran en cambio, los resultados obtenidos con el método analítico y numérico para el problema que se plantea en la Sección 2.1.1, también se muestran los resultados de la Sección 3.2.1, es decir el modelo analítico con las pérdidas radiativas linealizadas para comparar los dos modelos de disipación uniforme. Cabe destacar que, de nuevo, el método numérico se presenta muy similar al analítico.

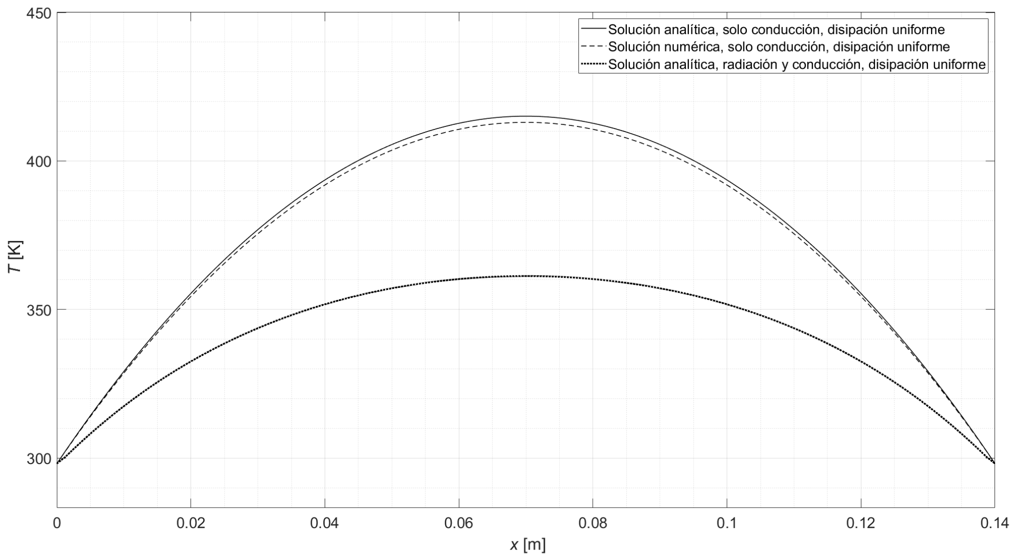

Figura 11: Perfil de temperaturas en régimen estacionario asumiendo disipación uniforme para la resolución numérica y analítica del problema planteado en la Sección 2.1 y 2.2.

La Tabla 4 recoge las diferencias obtenidas entre el método analítico y numérico para los modelos que no consideran las pérdidas radiativas. Al contar el modelo numérico con un error inferior al 1 % en todos los casos, se puede considerar que el modelo numérico desarrollado es válido, por lo que se pasa a determinar el resultado considerando también la transmisión de calor por radiación.

Tabla 4: Comparativa entre los resultados obtenidos con el método analítico y numérico para los modelos que no consideran las pérdidas radiativas

Modelo	Temperatura [K]		Error
de Disipación	Resolución Analítica	Resolución Numérica	(%)
Uniforme	415.1	413.0	0.5060
No-uniforme	390.4	389.7	0.1790

Los resultados obtenidos con el modelo térmico unidimensional completo se muestran en la Figura 12. En la parte de arriba de la imagen se muestra la temperatura de la tarjeta electrónica en distintos puntos respecto al tiempo, por lo que se observa claramente la transición de régimen transitorio a estacionario. Abajo se muestra el perfil de temperaturas $T(x,t)$ del modelo térmico unidimensional, respecto a la distancia del eje x, a distintos puntos temporales.

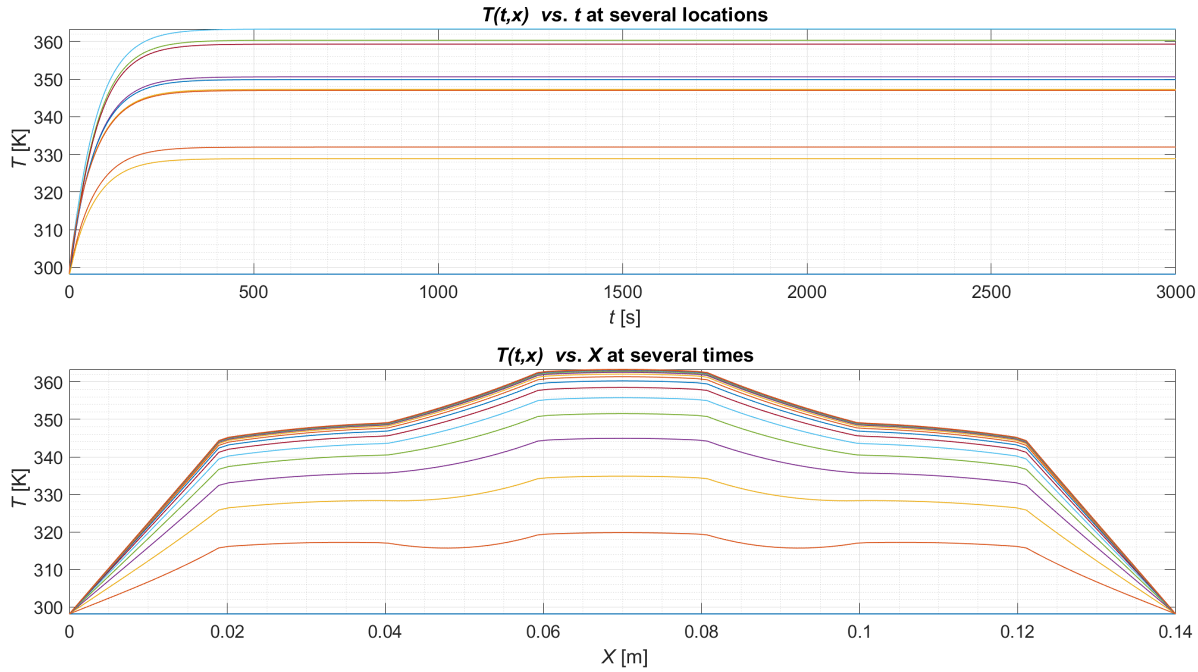

Figura 12: Arriba se muestra la temperatura de la tarjeta electrónica en distintos puntos respecto al tiempo, se observa el cambio de régimen transitorio a estacionario. Abajo se muestra el perfil de temperaturas del modelo térmico unidimensional, respecto a la distancia del eje x, a distintos puntos temporales.

La temperatura máxima alcanzada con este modelo es de 363.3 K, muy similar a la obtenida con el resultado analítico linealizando las pérdidas radiativas en la Sección 3.2.1.

3.3. Modelo térmico bidimensional

Se presentan en este apartado los resultados obtenidos para el modelo numérico del problema térmico bidimensional, para el que se ha considerado la geometría mostrada en la Sección 1 y se ha seguido la metodología descrita en la Sección 2.3.

En primer lugar, en la Figura 13 se muestran los mapas bidimensionales de disipación volumétrica, $\phi(x,y)$, de conductividad térmica longitudinal, $k_{eff}(x,y)$, de espesor, $z(x,y)$, de volumen, $V(x,y)$, y de capacidad térmica,

$C(x, y)$. De esta forma en la Figura 13 se aprecian claramente los distintos segmentos en los que se divide la PCB, así como las zonas donde ha sido necesario implementar una interpolación para tener en cuenta que los nodos de la malla no caen exactamente donde se encuentran las discontinuidades.

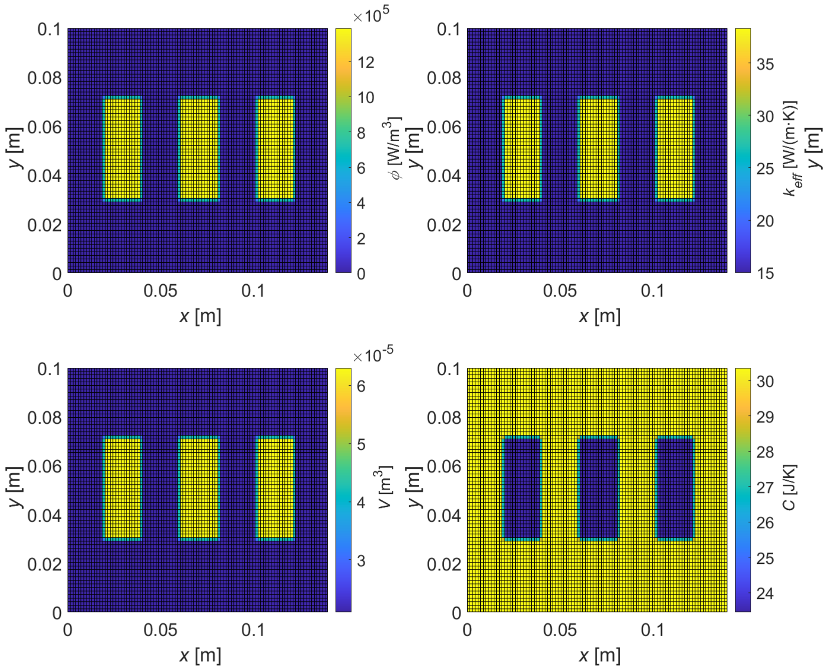

Figura 13: De izquierda a derecha y de arriba a abajo: mapas de disipación volumétrica, $\phi(x, y)$ [W/m^3], de conductividad térmica longitudinal, $k_{eff}(x, y)$ [W/(m·K)], de espesor, $z(x, y)$ [m^2], de volumen, $V(x, y)$ [m^3], y de capacidad térmica, $C(x, y)$ [J/K].

Haciendo uso de estos mapas, se pretende resolver el problema térmico bidimensional, con el fin de obtener un mapa de temperaturas sobre la tarjeta electrónica de la forma $T(x, y)$ en el régimen estacionario, así como mostrar la evolución de la temperatura en el régimen transitorio.

En la Figura 14 se muestra el mapa de temperaturas $T(x, y)$ en el régimen estacionario. Se pueden observar las zonas donde se presentan discontinuidades, siendo los segmentos de la malla donde se encuentran los IC claramente reconocibles, encontrándose el punto de $T_{\max}$ en el centro de la tarjeta electrónica. La temperara máxima alcanzada con el modelo bidimensional aquí descrito es de $T_{\max}(x, y, t) = 381{,}3$ K

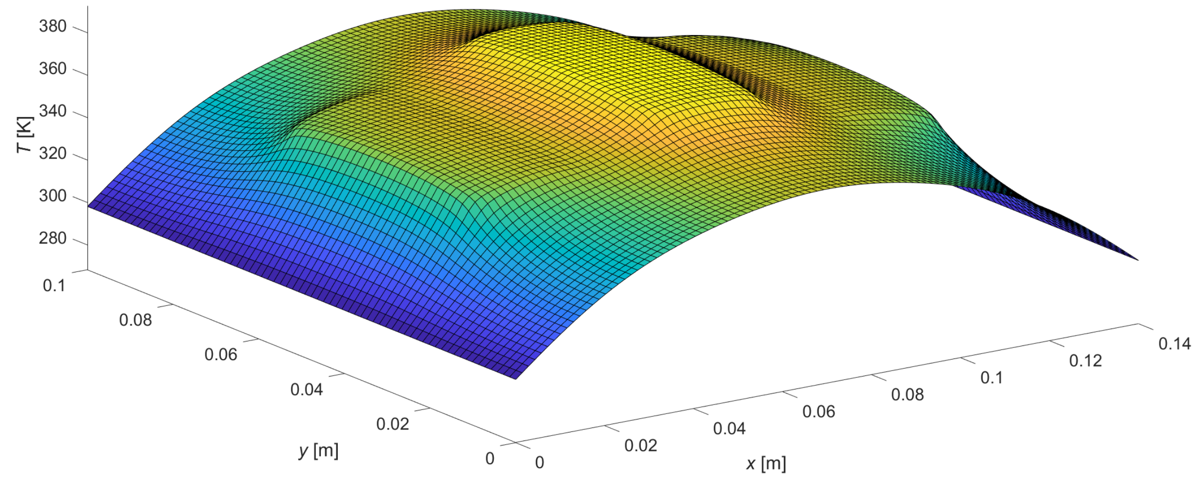

Figura 14: Mapa de temperaturas $T(x, y)$ en el régimen estacionario.

En la Figura 15, se muestra el mapa de temperaturas $T(x, y)$ en el régimen estacionario, visto en los ejes T-x. Este perfil es ya reconocible, al ser similar al que se ha obtenido en apartados anteriores al considerar la disipación como no uniforme. También se observan claramente las condiciones de contorno en los bordes.

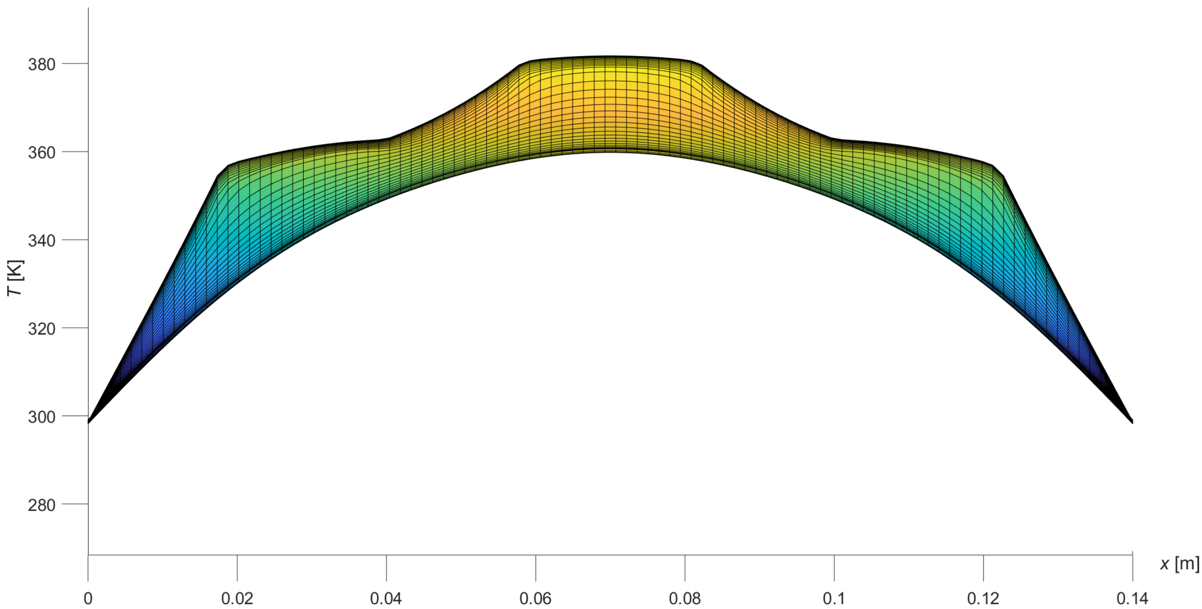

Figura 15: Mapa de temperaturas $T(x, y)$ en el régimen estacionario. Vista frontal.

En la Figura 16 se puede observar el mapa de temperaturas $T(x, y)$ en el régimen estacionario, visto en los ejes T-y. Se puede reconocer claramente la condición adiabática impuesta en los bordes que se encuentran térmicamente aislados. En la Figura 14 se observa como en estos bordes el perfil de temperaturas forma una parábola.

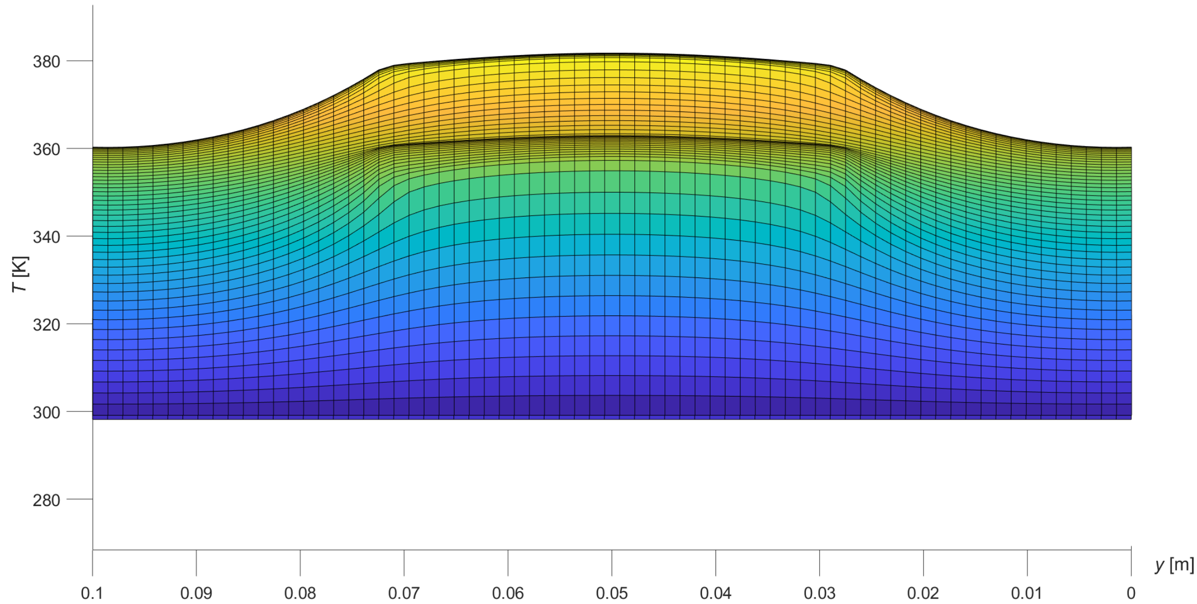

Figura 16: Mapa de temperaturas $T(x, y)$ en el régimen estacionario. Vista lateral.

En la Figura 17 se representa la distribución de temperaturas en el régimen estacionario utilizando un mapa de contornos donde cada color indica un rango de temperaturas entre las que se encuentra esa parte de la superficie de la tarjeta electrónica. Se observa claramente la simetría que caracteriza a la geometría estudiada, así como que los bordes en contacto térmico perfecto con las paredes son las zonas a menor temperatura de toda la tarjeta, aumentando la temperatura a medida que se acercan las coordenadas al centro, donde presentan un máximo.

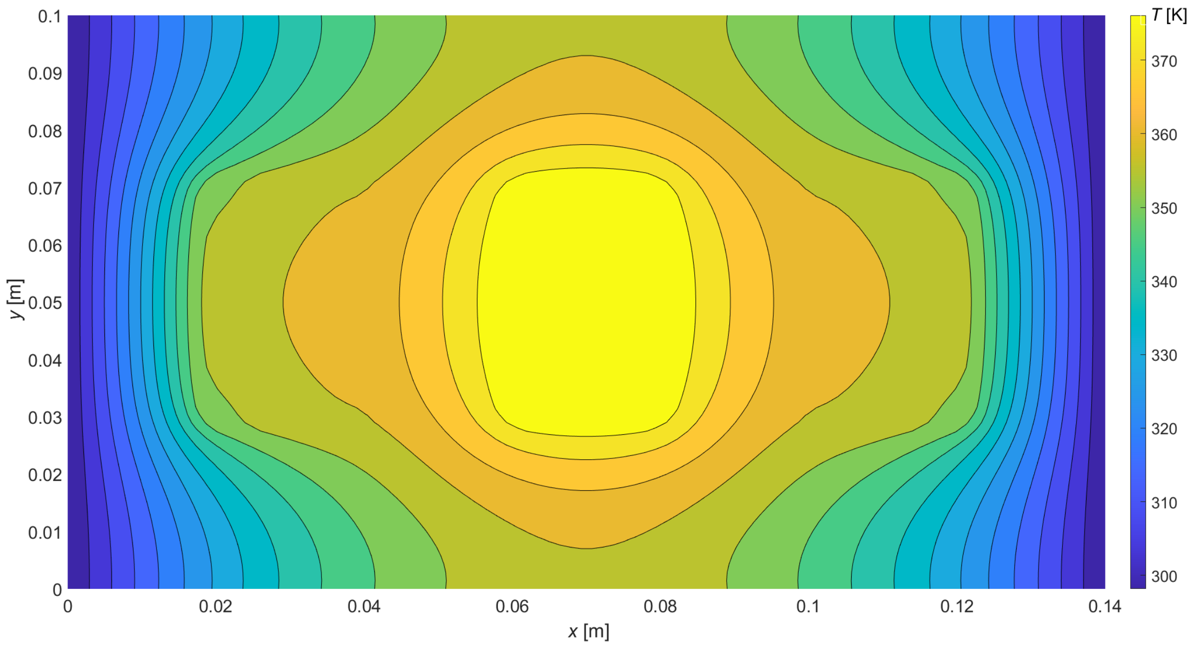

Figura 17: Contornos del mapa de temperaturas $T(x, y)$ en el régimen estacionario.

Finalmente, en la Figura 18 se muestra la evolución de la temperatura $T(x, y, t)$ en seis puntos temporales, pasando del régimen transitorio hasta el estacionario. También se ha producido una animación de este proceso, que puede consultarse en esta referencia [3].

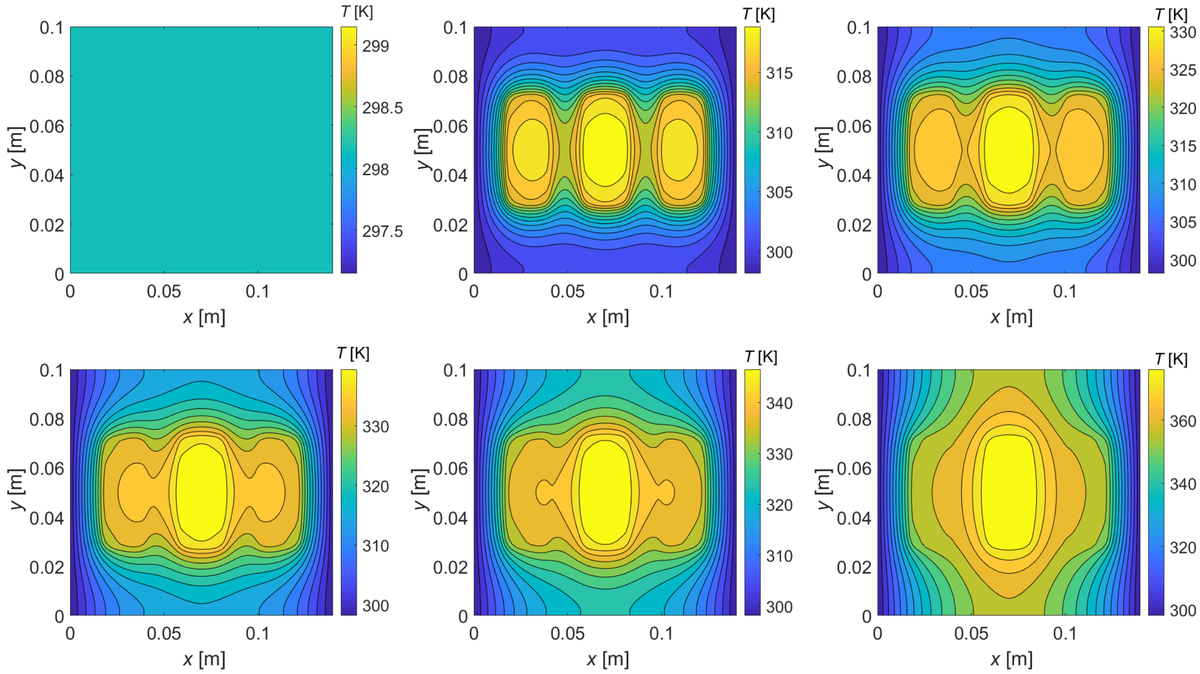

Figura 18: Mapa de temperaturas $T(x, y, t)$ en el régimen transitorio y estacionario. De izquierda a derecha y de arriba a abajo: instantes de tiempo distintos para el mapa de temperaturas, desde el momento inicial, pasando por el régimen transitorio hasta llegar en la última imagen al régimen estacionario.

3.4. Observaciones de los resultados

Una vez presentados los datos obtenidos por cada modelo, se pretende analizar las diferencias entre los modelos y los métodos empleados, así como comentar brevemente el efecto que tiene la densidad de la malla en el resultado obtenido para el modelo numérico bidimensional de diferencias finitas.

3.4.1. Resumen de los resultados

En este apartado se pretende comparar los perfiles de temperaturas y las temperaturas máximas que se consiguen con cada modelo, así como las diferencias que presentan las resoluciones de manera numérica y analítica.

Respecto a los modelos de solo conducción (Sección 3.1), en la Figura 19, se muestran los perfiles de temperaturas en régimen estacionario para los casos de disipación uniforme y no uniforme, así como para la resolución numérica y analítica del problema. Se puede observar como las soluciones analíticas y numéricas son prácticamente idénticas, siendo el resultado analítico ligeramente mayor al numérico. Se puede concluir que ambos métodos resuelven el problema de manera eficaz. Aunque implementar el método numérico resulta más laborioso, una vez planteado es relativamente sencillo incorporar otros fenómenos al modelo como es la transmisión de calor por convección o radiación.

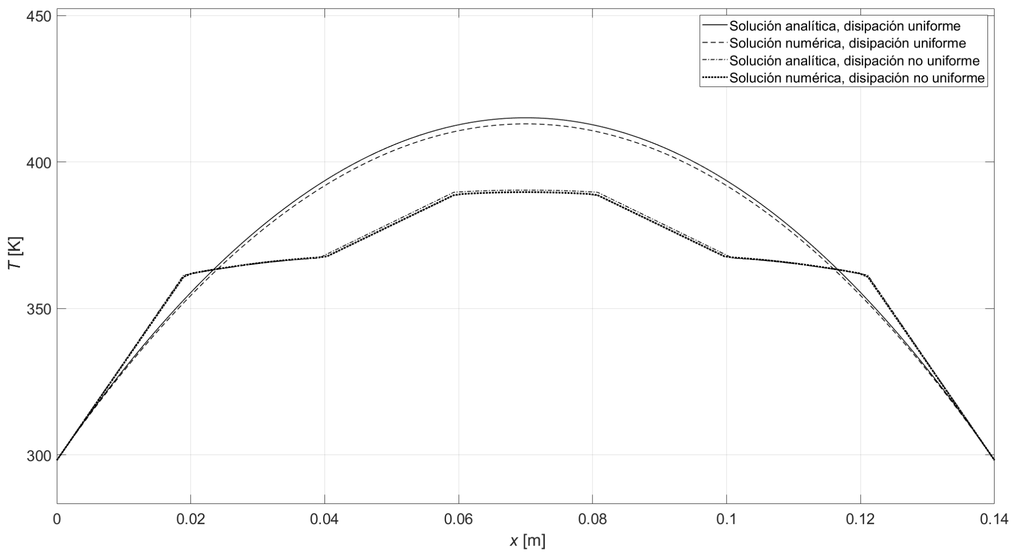

Figura 19: Perfiles de temperaturas en régimen estacionario para casos de disipación uniforme y no uniforme, así como para la resolución numérica y analítica del problema planteado en la Sección 2.1.

Respecto a los resultados obtenidos para los modelos completos que incluyen las pérdidas radiativas (Sección 3.2 y 3.3), en la Figura 20, se exponen los perfiles de temperaturas $T(x)$ en régimen estacionario para los casos de disipación uniforme y no uniforme, así como para la resolución numérica y analítica del problema. En el caso del modelo bidimensional, se muestra únicamente el perfil central de temperaturas. Nótese que en la Figura 20 se observa como el perfil de temperaturas del modelo bidimensional es significativamente mayor al del modelo unidimensional, siendo este último similar al resultado analítico obtenido en la Sección 3.2.1. Todas las curvas comparten el mismo punto de $T_{\max}$ que se encuentra en el centro de la tarjeta, como se ha mencionado previamente. De estos datos se puede concluir que el modelo unidimensional es claramente conservativo, y si se utiliza en lugar de un modelo bidimensional, esta diferencia en la temperatura máxima obtenida debe ser tenida bajo consideración.

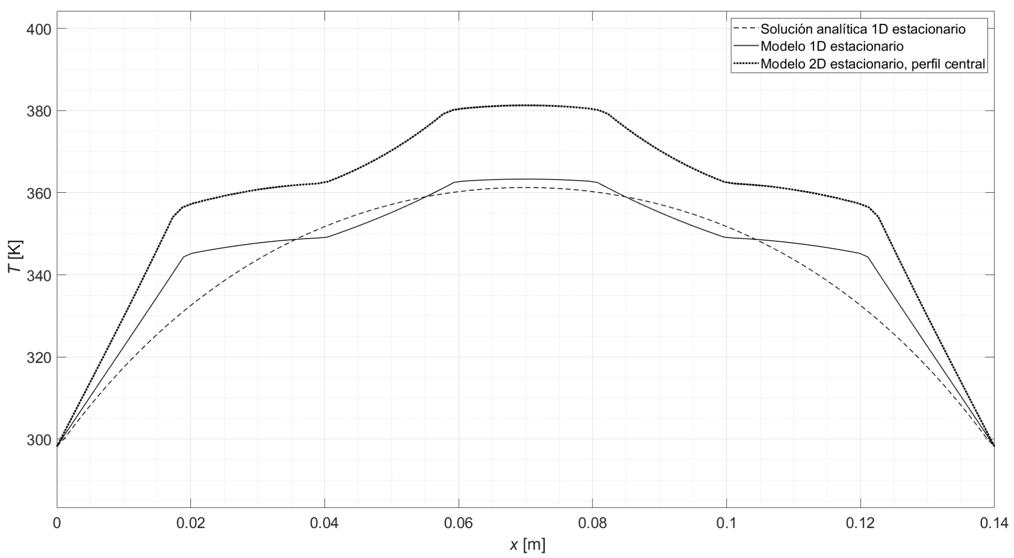

Figura 20: Perfiles de temperaturas en régimen estacionario para casos de disipación uniforme y no uniforme, así como para la resolución numérica y analítica del problema planteado en la Sección 2.2.

Comparando los modelos de disipación no uniforme, como se muestra en la Figura 21, se puede observar como el modelo unidimensional que solo considera la conducción produce, como es de esperar, el perfil de temperaturas mayor, encontrándose el modelo bidimensional completo entre este caso y el del modelo unidimensional que si que considera las pérdidas radiativas. Por lo tanto, el método descrito en la Sección 2.1.2, puede ser utilizado como una primera aproximación.

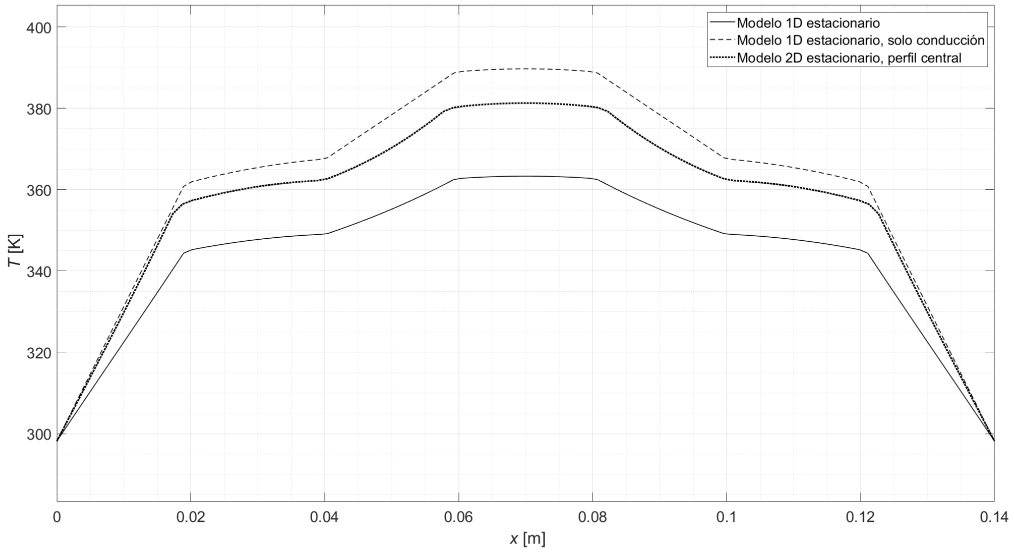

Figura 21: Perfiles de temperaturas en régimen estacionario asumiendo disipación no uniforme, para la resolución numérica y analítica del problema planteado en las Secciones 2.1, 2.2 y 2.3.

Finalmente, en la Tabla 5 se recogen las temperaturas máximas obtenidos con cada modelo estudiado.

Tabla 5: Comparativa entre los resultados obtenidos con el método analítico y numérico para todos los modelos estudiados.

Modos de transmisión de calor	Modelo	Temperatura Máxima [K]	
		Resolución Analítica	Resolución Numérica
Conducción	1D, Disipación uniforme	415.1	413.0
	1D, Disipación no-uniforme	390.4	389.7
Conducción y Radiación	1D, Disipación uniforme	361.3	361.0
	1D, Disipación no-uniforme	N/A	363.3
	2D, Disipación no-uniforme	N/A	381.3

3.4.2. Efecto de la densidad de la malla en el resultado de las simulaciones numéricas

Respecto al efecto de la densidad de la malla en el resultado obtenido al implementar el modelo numérico, cabe destacar que, como se comenta en la Sección 2.2.2, una discretización espacial final requiere de una discretización temporal que requiere de una gran cantidad de esfuerzo computacional. Por ello es de interés mantener la malla espacial con relativamente pocos elementos, con el fin de acelerar las simulaciones. En la Tabla 6 se muestra como varía la temperatura máxima obtenida para el modelo térmico bidimensional según la densidad de la malla escogida. Se puede observar que a partir de una densidad de la malla de 42 elementos en el eje x y 30 elementos en el eje y la solución empieza a converger. No se incluyen casos con un mayor número de subdivisiones ya que se supera la cantidad máxima de memoria computacional disponible.

Tabla 6: Comparativa entre los resultados obtenidos en el modelo térmico bidimensional estacionario para distintos niveles de discretización de los elementos espaciales.

Número de elementos espaciales (N_x, N_y)	Temperatura Máxima [K]
(14,10)	368.1
(42,30)	380.4
(70,50)	381.0
(98,70)	381.3

4. Conclusiones

En este estudio se han obtenido modelos unidimensionales con distintos niveles de complejidad, así como se ha desarrollado un modelo bidimensional del problema térmico. Además, se ha realizado una comparación entre los distintos modelos empleados. Para ilustrar la eficacia de los modelos considerados en el estudio, se recoge en la Tabla 5 una recopilación de todos los resultados obtenidos.

En conclusión, respecto a los modelos de solo conducción, se puede observar como las soluciones analíticas y numéricas son prácticamente idénticas, siendo el resultado analítico ligeramente mayor al numérico. Se puede concluir que ambos métodos resuelven el problema proporcionando una temperatura máxima mayor a la que proporciona el modelo bidimensional. Aún así, cabe destacar, que el modelo de disipación no uniforme sin incluir las pérdidas radiativas puede ser empleado como primera aproximación, ya que proporciona una temperatura máxima similar a la del modelo bidimensional.

Respecto al uso de métodos analíticos o numéricos, aunque implementar el método numérico resulta más laborioso, una vez planteado es relativamente sencillo incorporar otros fenómenos al modelo como es la transmisión de calor por convección o radiación. Por lo que su uso es recomendable si se desea hacer un estudio más allá de la primera aproximación. Aún así, el método analítico resulta especialmente útil a la hora de utilizarse para validar el modelo numérico implementado.

Respecto a los resultados obtenidos para los modelos completos que incluyen las pérdidas radiativas, los perfiles de temperaturas en el régimen estacionario para los casos de disipación uniforme y no uniforme, proporcionan resultados similares en el caso de los modelos unidimensionales estudiados. Por el contrario, el perfil de temperaturas obtenido con el modelo bidimensional es significativamente mayor al del modelo unidimensional. De estos datos se puede concluir que el modelo unidimensional es claramente conservativo.

Por último, comparando los modelos de disipación no uniforme, se puede concluir que el modelo unidimensional que solo considera la transmisión de calor por conducción produce, como es de esperar, el perfil de temperaturas mayor, situándose el modelo bidimensional completo entre este caso y el del modelo unidimensional que si que considera las pérdidas radiativas.

Referencias

[1] I. Martínez, HEAT CONDUCTION.
URL http://imartinez.etsiae.upm.es/$\sim$isidoro/bk3/c11/Heatconduction.pdf

[2] I. Martínez, Conducción. Modelos Numéricos. 0 1–3.

[3] Simulación del régimen transitorio y estacionario para el modelo numérico bidimensional de diferencias finitas. [link].
URL https://youtu.be/fppZ1LOmf0M

A. Anexo I

```
%% ****************************************************************************
%                    CONDUCCIN DE CALOR, SIMULACI N NUMRICA
% ______________________________________________________________________________

% Realizado por Diego Mataix Caballero.
%
% ADDITIONAL NOTES:
%
%---------------------------------------------------------------------
close all; clear all; clc;

%%Datos
%%%% PCB (de FR4) %%%%
dx = 140e-3;      % [m]
dy = 100e-3;      % [m]
dz = 1.5e-3;      % [m]
dz_pcb = 1.4e-3; % [m]
dz_cu = (dz - dz_pcb)/2; % [m]
A = dx*dy;          % [m^2]
Vol = dx*dy*dz;  % [m^3]
rho_FR4 = 2100;  % [kg/m^3]
    % capacidad termica:
c_FR4 = 700;        % [J / kg * K]
c_Cu = 390;         % [J / kg * K]
    % Recubrimiento de cobre del PCB por cada lado
t_rec = 50e-6;   % [m]      % en una de las caras es continuo, y en la otra
                            % ocupa solo el 10% de la superficie en la cual van
                            % montados tres IC
rho_Cu = 8960;   % [kg/m^3]
    % los lados cortos de la PCB tienen contacto termico con paredes a 25C
T_b = convtemp(25, 'C', 'K');
    % los otros dos bordes estan termicamente aislados.
    % para el FR4:
k_plano = 0.5;                % [W / ( m * K )]
k_traves = k_plano / 2;       % [W / ( m * K )]

%%%% IC %%%%
dy_ic = 40e-3;   % [m]
```

```
dx_ic = 20e-3;   % [m]
dz_ic = 3e-3;    % [m]
Vol_ic = dx_ic * dy_ic * dz_ic; % [m^3]
A_ic = (dz+dz_ic)*dy;           % [m^2]
    %dispando:
Q_ic = 5;                       % [W]
Q_ic_tot = 3 * Q_ic /2;         % [W] (symmetry)
    %conductividad termica:
k_ic = 50;          % [W / ( m * K )]
k_Cu = 395;         % [W / ( m * K )]
    %capacidad termica:
C_ic = 20;          % [J / K]
    %separaci n entre ICs
dist_ic = 20e-3; % [m]

%%%%Otros datos %%%%
%calor transmitido por radiacion:
emiss_comp = 0.7; %emisividad media por el lado de los componentes
emiss_cara = 0.5; %emisividad media por el lado de la cara opuesta
%caja electronica que se puede considerar negra
T_box = convtemp(45, 'C', 'K'); %T caja electronica (para apartado c)

%%Constantes

h = 6.6256e-34;   % [J * s]                       %Plank's constant
c_0 = 2.9979e8;   % [m / s]                       %velocidad de la luz
stefan_boltz = 5.67e-8; % [W / ( m^2 * K^4 )]     %Stefan-Boltzmann constant
T_sun = 5800;     % [K]
T_cbr = 2.7;      % [K]
```

B. Anexo II

```
%% ********************************************************************************
%                          CONDUCCIN DE CALOR, SIMULACI N NUMRICA
% ________________________________________________________________________________
% Realizado por Diego Mataix Caballero.
%
% ADDITIONAL NOTES:
% PCB de FR-4 =: 140 x 100 x 1.5 (dx * dy * dz)
% Recubrimiento de Cu de 50e-6 m
%       - en cara 1 : continuo
%       - en cara 2 : 90% FR-4, 10% Cu
% 3 IC, cada uno disipa 5W, con k_ic = 5 [W/(mK)], con c_ic = 20 [J/K]
%       - distribuidos uniformemente en la PCB, 20 mm de separacion
% PCB tiene contact termico perfecto con paredes permanentemente a 25C, los
% otros dos bordes est n t rmicamente aislados.
%-------------------------------------------------------------------------
close all; clear all; clc;
%-------------------------------------------------------------------------
%% Datos
Conduccion_NumSim_DATOS
%-------------------------------------------------------------------------
%% Choose exercise to run
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
choose = 'e';       % 'a', 'b', 'c', 'd' & 'e' %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------------------------------------------------------------
%% Define global parameters
%%% Define coefficients and some parameters %%%
phi = (3 * Q_ic) / Vol;                       % Volumetric dissipation [W/m^3]
l =     [t_rec dz_pcb t_rec];                 % Dimension Vector [m]
k_vect = [k_Cu k_plano (0.1*k_Cu+0.9*k_plano)];   % Conductivity Vector [W/(m K)] tercera capa es donde van los IC, cubierta solo al 10% de cobre
k_eff = effective(k_vect, l);                 % Effective Conductivity [W/(m K)]
L = dx/2;                                     % [m]
A = dy * dz;                                  % [m^2]
c_eff = (c_Cu*t_rec + c_FR4*dz_pcb + (0.1*c_Cu*t_rec)) / (t_rec + dz_pcb + 0.1*
```

```
    t_rec);  % Thermal Capacity [J / kg.K]
C_eff = c_eff * rho_FR4 * dx*dz*dy;     % Thermal Capacity [J / K]
%% %Define the parameters for the sections containing the IC %%%
l_ic =       [t_rec dz_pcb t_rec dz_ic];                % Dimension Vector [m]
k_vect_ic = [k_Cu k_plano (0.1*k_Cu+0.9*k_plano) k_ic];     % Conductivity Vector [W/(m K)] tercera capa es donde van los IC, cubierta solo al 10% de cobre
k_eff_ic = effective(k_vect_ic, l_ic);                 % Effective Conductivity [W/(m K)]
C_eff_ic = (C_eff *dz + C_ic*dz_ic)/ (dz + dz_ic);      % Thermal Capacity [J / K]
%% %Define the emissivity %%%
emiss_vect = [emiss_cara emiss_comp];
p_vect = [dx+dz dx+dz];
emiss = effective(emiss_vect, p_vect);
%% %Mesh %%%
m = 8e0;                                               % Spatial Subdivisions
M = 14*m;                                              % Total n of spatial subdivisions
x = linspace(0, dx, M);                                % Spatial Coordinates [m]
N = 6e5;                                               %# of time steps
tsim = 3000;                                           % Total simulation time [s]
%-----------------------------------------------------------------------------
switch(choose)
    %-----------------------------------------------------------------------------
    case 'a'
        %%Apartado A
        % Considerando que la tarjeta s lo evacua calor por los bordes,
        % determinar la temperatura m xima que se alcanzar a si toda la disipaci n
        % estuviese uniformemente repartida en la PCB y los IC no influyeran.
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        method = 2;          % 1: Only max Temp        2: Show all Temp profile
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        switch(method)
            case 1
```

```
                DT = 1/8 * ( phi * dx^2 / k_eff );            % Delta T [K]
                T_0 = T_b + DT                                % Max T [K]
                T_0_C = convtemp(T_0, 'K', 'C')               % Max T [C]
            case 2
                b = b_coef(Q_ic_tot, k_eff, dz*dy);           % [K*m]
                for i = 1:M
                    T(i) = temp_parb(T_b, b, (x(i)), phi, k_eff);  % [K]
                end
                T_0 = max(T)                                  % Max T [K]
                T_0_C = convtemp(T_0, 'K', 'C')               % Max T [Celsius]

                figure()
                myplot(x,T)
                hold on
                axis([0 dx T_b*0.9 max(T)*1.1])
                ylabel('{\it T} [K]')
                xlabel('{\it x} [m]');
                xline(L, '-.')
                yline(T_b, '--')
                yline(T_0, '--')
                hold off
        end
        % ---------------------------------------------------------------------------
        %%Apartado B
    case 'b'
        % Considerando que la tarjeta s lo evacua calor por los bordes, determinar
        % la temperatura m xima que se alcanzar a con un modelo unidimensional en el
        % que los IC llegaran hasta los bordes aislados, en el l mite k I C , y con la kIC dada.
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        method = 1;        % 1: Method @ k --> k_ic        2: Method @ k --> inf
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % L mites de cada tramo
```

```
    for i = 1:6
        lim(i) = 2*i*m;                                     % [m]
    end

    %%%Define the heat for each type of section: 1: w/o IC, 2 w/ IC %%%
    Q1 = Q_ic_tot;                                          % [W]
    Q2 = Q_ic_tot*(1/3);                                    % [W]

    %%%Coefficients for the first section %%%
    a = T_b;                                                % [K]
    b = b_coef(Q_ic_tot, k_eff, dz*dy);                     % [K*m]

    A_ic = (dz+dz_ic)*dy;                                   % [m^2]
    phi_ic = Q_ic / (0.02 * 0.1 * 0.0045);                  % Volumetric dissipation [W/m^3]

    %%%%%%%%%%CALCULATIONS %%%%%%%%%%%%
    %%%SECTION 1: From the PCB border to the start of the 1st IC %%%
    T = zeros(1, M);
    for i = 1: lim(1)
        T(i) = temp_lin(Q1, x(i), k_eff, A, a);
    end
    %%%SECTION 2: From the start of the 1st IC to the end of that IC %%%
    switch(method)
        case 1 %%%k --> k_ic %%%
            a = T(lim(1));                                  % [K]
            b = b_coef(Q1, k_eff_ic, A_ic);                 % [K*m]
            for i = lim(1):lim(2)
                T(i) = temp_parb(a, b, (x(i) - x(lim(1))), phi_ic, k_eff_ic);   % [K]
            end
        case 2 %%%k --> inf %%%
            for i = lim(1)+1:lim(2)
                T(i) = T(lim(1));                           % [K]
            end
    end
    %%%SECTION 3: From the end of the 1st IC to the start of the 2nd IC
    for i = (lim(2)+1):lim(3)
        T(i) = temp_lin(Q2, (x(i) - x(lim(2))), k_eff, A, T(lim(2)));   % [K]
    end
    %%%SECTION 4: From the start of the 2nd IC to the center of that IC
```

```
        switch(method)
            case 1 %%%k --> k_ic
                a = T(lim(3));                          % [K]
                b = b_coef(Q2, k_eff_ic, A_ic);         % [K*m]
                for i = (lim(3)+1): (M/2)
                    T(i) = temp_parb(a, b, (x(i) - x(lim(3))), phi_ic, k_eff_ic);     % [K]
                end

            case 2 %%%k --> inf
                for i = (lim(3)+1): (M/2)
                    T(i) = T(lim(3));                   % [K]
                end
        end
        %%%TAKE ADVANTAGE OF SYMMETRY %%%
        T(((M/2)+1):M) = T(M/2:-1:1);               % Mirror curve

        T_0 = max(T)                                % Max T [K]
        T_0_C = convtemp(T_0, 'K', 'C')             % Max T [Celsius]

        %%%PLOT TEMPERATURE PROFILE %%%
        figure()
        hold on
        myplot(x,T)
        axis([0 dx T_b*0.9 max(T)*1.1])
        ylabel('{\it T} [K]')
        xlabel('{\it x} [m]');
        xline(L, '-.')
        yline(T_b, '--')
        yline(T_0, '--')
        hold off
        %-------------------------------------------------------------------------

        %%Apartado C
        % Considerando que se transmite calor por radiaci n , con una emisividad media de 0,7
        % por el lado de los componentes, y de 0,5 por la cara opuesta, con una caja electr nica
        % que se puede suponer negra y a 45  C , determinar la temperatura m xima linealizando las
        % p rdidas radiativas y con disipaci n uniforme.
```

```
    case 'c'

        p = (2*dy);             % Perimeter [m]
        A = dy * dz;            % Area [m^2]
        T_avg = 375;            % Average Temperature [K] % from 'a' --> 375K; from 'b' --> 363K

        eta = 4*p * stefan_boltz * emiss * T_avg^3;          % Auxiliary function for simplifying the ODE
        lambda = sqrt( eta / ( (k_eff * A) ) )                                  % Eigenvalues of the ODE
        c2 = (T_b - T_box - ( (phi * A) / eta ) / (1+exp(-lambda *dx)) )        % Coef. of the ODE
        c1 = c2*exp(-lambda*dx)                                                 % Coef. of the ODE

        % Temperature profile, the expression is found by solving the ODE
        % and applying the BC.
        for i = 1:M
            T(i) = c1* exp(lambda * x(i) ) + c2*exp(-lambda * x(i) ) + T_box + ( (phi*A)/eta);   % [K]
        end

        T_0 = max(T)                                         % Max T [K]
        T_0_C = convtemp(T_0, 'K', 'C')                      % Max T [Celsius]

        %% %PLOT TEMPERATURE PROFILE %%%
        figure()
        hold on
        myplot(x,T)
        axis([0 dx T_b*0.9 max(T)*1.1])
        ylabel('{\it T} [K]')
        xlabel('{\it x} [m]');
        xline(L, '-.')
        yline(T_b, '--')
        yline(T_0, '--')
        hold off

        % ---------------------------------------------------------------------

        %%Apartado D
```

```
        % Resolver el caso anterior pero sin linealizar y con la disipaci n no uniforme.
    case 'd'
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        validate = 1;       % 1: Uniform dissipation    % 2: Non-uniform dissipation
        radiation = 1;      % 1: Include radiation      % 2: Do not include radiation
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        h=0;                % Convective coefficient [W/(m^2 K )] (NO CONVECTION)
        p = (2*dy);         % Radiative perimeter [m]
        if radiation == 2
            emiss = 0;
        end
        %
        %%%Limits for each segment %%%
        for i = 1:6
            lim(i) = 2*i*m;
        end

        %%%Definir los vectores para representar las discontinuidades %%%
        % Initialising
        phi = ones(1,M);
        k = ones(1,M);
        A_vect = ones(1,M);
        V = ones(1,M);
        C = ones(1,M);
        switch(validate)
            case 1
                for i = 1:M
                    phi(i) = (3 * Q_ic) / Vol ;
                    k(i) = k_eff;
                    A_vect(i)= A;
                    V(i) = dx * dz * dy;
                    C(i) =  C_eff;
                end

            case 2
```

```
        %%% NO IC %%%
        for i = 1:lim(1)
            phi(i) = 0;
            k(i) = k_eff;
            A_vect(i) = A;                              % [m^2]
            V(i) = dx * dz * dy;                        % [m^3]
            C(i) = C_eff;                               % [J / K]
        end
        %%% IC %%%
        for i = lim(1)+1:lim(2)
            phi(i) = Q_ic / (0.02 * 0.1 * 0.0045);
            k(i) = k_eff_ic;
            A_vect(i) = (dz+dz_ic)*dy;                  % [m^2]
            V(i) = dx * (dz+dz_ic) * dy;                % [m^3]
            C(i) = C_eff_ic;                            % [J / K]
        end
        %%% NO IC %%%
        for i = (lim(2)+1):lim(3)
            phi(i) = 0;
            k(i) = k_eff;
            A_vect(i) = A;                              % [m^2]
            V(i) = dx * dz * dy;                        % [m^3]
            C(i) = C_eff;                               % [J / K]
        end
        %%% IC %%%
        for i = (lim(3)+1): (M/2)
            phi(i) = Q_ic / (0.02 * 0.1 * 0.0045);
            k(i) = k_eff_ic;
            A_vect(i) = (dz+dz_ic)*dy;                  % [m^2]
            V(i) = dx * (dz+dz_ic) * dy;                % [m^3]
            C(i) = C_eff_ic;                            % [J / K]
        end
        %%% TAKE ADVANTAGE OF SYMMETRY
        phi(((M/2)+1):M) = phi((M/2):-1:1);             % Mirror vector phi
        k(((M/2)+1):M) = k(M/2:-1:1);                   % Mirror vector k
        A_vect(((M/2)+1):M) = A_vect(M/2:-1:1);         % Mirror vector A_vect
        V(((M/2)+1):M) = V(M/2:-1:1);                   % Mirror vector V
        C(((M/2)+1):M) = C(M/2:-1:1);                   % Mirror vector
```

```
C
end
%%Initialising:           %N time %M space
Dx=dx/M;                  %Element width
X=linspace(0,dx,M);       %Node position list (equispaced)
Dt=tsim/N;                %Time step (you might fix it instead of tsim)
t=linspace(0,tsim,N)';    %Time vector
DtrcA = ones(1,M);
kALapla = ones(N,M);
phDT = ones(N,M);
T=T_b*ones(N,M);          %Temperature-matrix (times from 1 to n, and positions from 1 to M+1)
%% %plot parameters for heat equation
subplot(2,3,1);myplot(X,phi); ylabel('{\it \phi} [W/{m^{3}}]'); xlabel('{\it x} [m]');
subplot(2,3,2);myplot(X,k); ylabel('{\it k_{eff}} [W/{(m K)}]'); xlabel('{\it x} [m]');
subplot(2,3,3);myplot(X,A_vect); ylabel('{\it A} [m^2]');  xlabel('{\it x} [m]');
subplot(2,3,4); myplot(X,V); ylabel('{\it V} [m^3]'); xlabel('{\it x} [m]');
subplot(2,3,5);myplot(X,C); ylabel('{\it C} [J/K]'); xlabel('{\it x} [m]');
%% %Check for stability of the explicit finite difference method %%
for i = 1:M
    Fo_vect(i)=k(i)/(C(i)/V(i))*Dt/(Dx*Dx);          %Fourier's number
    Bi_vect(i)=h*p*Dx/(k(i)*A_vect(i)/Dx);           %Biot's number
end
Fo = max(Fo_vect);
Bi = max(Bi_vect);
disp(['Stability requires 1-Fo*(2+Bi)<0. It actually is =',num2str(1-Fo*(2+Bi))])
if 1-Fo*(2+Bi)<0 disp('This is unstable; increase number of time steps'), end

%% %Temperature profile equation by means of finite elements methods %%%
j=1; T(j,:)=T_b;          %Initial temperature profile T(x,t)=0 (assumed uniform)
it=M; T(:,it)=T_b;
for j=2:N                 %Time advance
    i=1; T(j,i)=T_b;      %Left border (base) maintained at T_b
    for i=2:M-1           %Generic spatial nodes
```

```
                DtrcA(i) = (Dt/((C(i)/V(i))*A_vect(i)));
                kALapla(j,i) = (( ((k(i+1)+k(i))/2) * ((A_vect(i)+A_vect(i+1))/2) *...
                    (T(j-1,i+1)-T(j-1,i))- ((k(i)+k(i-1))/2) * ((A_vect(i)+A_vect(i-1))/2) *...
                    (T(j-1,i)-T(j-1,i-1)) )/Dx^2);
                phDT(j,i) = (p*(emiss*stefan_boltz*(T(j-1,i)^4 - T_box^4)));

                T(j,i)=T(j-1,i)+(DtrcA(i))*...
                    ((kALapla(j,i))+(phi(i)*A_vect(i)) - (phDT(j,i)) );
            end
            %Boundory condition in node 0:
            T(j,1)=T_b;         %if Troot is fixed
            %Boundory condition in node N:
            T(j,M)=T_b;       %if Troot is fixed
        end
        %%%PLOT TEMPERATURE PROFILE %%%
        T_0 =  max(T(N,:))                                  %Max T [K]
        T_0_C = convtemp(T_0, 'K', 'C')                     %Max T [Celsius]
        % Plot transitory
        subplot(2,1,1);myplot(t,T(:,1:M/10:M));xlabel('{\it t} [s]'),ylabel('{\it T} [K]');
        title('{\it T(t,x)} {\it vs}.{\it t} at several locations')
        subplot(2,1,2);myplot(X,T(1:N/100:N,:));xlabel('{\it X} [m]'),ylabel('{\it T} [K]');
        title('{\it T(t,x)} {\it vs}.{\it X} at several times')
        % Plot stationary
        figure()                                        % Plot 1D Stationary Temp profile
        hold on
        myplot( X, T(N,:) )
        axis([0 dx T_b*0.9 max(T_0)*1.1])
        ylabel('{\it T} [K]')
        xlabel('{\it x} [m]');
        xline(L, '-.')
        yline(T_b, '--')
        yline(T_0, '--')
        hold off
        % ---------------------------------------------------------------------
```

```
        %%Apartado E
```

```
        % Resolver el problema t rmico bidimensional estacionario y comparar el perfil
        % central de temperaturas con el del caso anterior.
    case 'e'
        phi = (Q_ic) / ((dz+dz_ic)*dy_ic*dx_ic);                              % Volumetric dissipation [W/m^3]
        %%%Define 2D mesh %%%
        m =7e0;                   % Spatial Subdivisions                  % 7e0 works
        Mx = 14*m;                % Total n of spatial subdivisions (x-direction)
        my = 7e0;                 % Spatial Subdivisions (y-direction)          % 7e0 works
        My = 10*my;               % Total n of spatial subdivisions (y-direction)
        N = 2.1e5;                %# of time steps                            % 1.9e5 works
        tsim = 1000;              % Total simulation time [s]             % 750 works
        %%%Initialise %%%
        Dx=dx/Mx;                 % Element width (x-direction)
        Dy=dy/My;                 % Element width (y-direction)
        X=linspace(0,dx,Mx);      % Node position list (equispaced) (x-direction)
        Y=linspace(0,dy,My);      % Node position list (equispaced) (y-direction)
        Dt=tsim/N;                % Time step (you might fix it instead of tsim)
        t=linspace(0,tsim,N)';    % Time vector
        % Initial temperature profile T(x,t)=T_b (assumed uniform)
        T = T_b*ones(Mx, My, N);
        %%%%%%Definir los vectores para representar las discontinuidades %%%
        % Initialising
        phi2d = ones(Mx,My);
        k_effxy = ones(Mx,My);
        z = ones(Mx,My);
        V2d = ones(Mx,My);
        C = ones(Mx,My);
        %%%Limits for each segment %%%
        for i = 1:6
            limx(i) = 2*i*m;
        end
        for i = 1:10
            limy(i) = 1*i*my;
        end
        %%%NO IC SEGMENTS %%%
        for i = 1:(Mx/2)
```

```
                for k = 1:limy(3)
                    phi2d(i,k) = 0;
                    k_effxy(i,k) = k_eff;
                    z(i,k) = dz;
                    V2d(i,k) = dx * dz * dy;
                    C(i,k) = C_eff;
                end
            end
            for i = 1:limx(1)
                for k = limy(3):(My/2)
                    phi2d(i,k) = 0;
                    k_effxy(i,k) = k_eff;
                    z(i,k) = dz;
                    V2d(i,k) = dx * dz * dy;
                    C(i,k) = C_eff;
                end
            end
            for i = limx(2)+1:limx(3)
                for k = limy(3):(My/2)
                    phi2d(i,k) = 0;
                    k_effxy(i,k) = k_eff;
                    z(i,k) = dz;
                    V2d(i,k) = dx * dz * dy;
                    C(i,k) = C_eff;
                end
            end
            %%%IC SEGMENTS %%%
            for i = limx(1)+1:limx(2)
                for k = limy(3)+1:(My/2)
                    phi2d(i,k) = phi;
                    k_effxy(i,k) = k_eff_ic;
                    z(i,k) = dz+dz_ic;
                    V2d(i,k) = dx * (dz+dz_ic) * dy;
                    C(i,k) = C_eff_ic;
                end
            end
            for i = limx(3)+1:(Mx/2)
                for k = limy(3)+1:(My/2)
                    phi2d(i,k) = phi;
                    k_effxy(i,k) = k_eff_ic;
                    z(i,k) = dz+dz_ic;
                    V2d(i,k) = dx * (dz+dz_ic) * dy;
```

```
                C(i,k) = C_eff_ic;
            end
        end
        %% %INTERPOLATE %%%%
        for j = 1:3
            for i =limx(j):limx(j)
                for k = limy(3):(My/2)+1
                    phi2d(i,k) = (phi2d(i-1,k)+phi2d(i+1,k))/2;
                    k_effxy(i,k) = (k_effxy(i-1,k)+k_effxy(i+1,k))/2;
                    z(i,k) = (z(i-1,k)+z(i+1,k))/2;
                    V2d(i,k) = (V2d(i-1,k)+V2d(i+1,k))/2;
                    C(i,k) = (C(i-1,k)+C(i+1,k))/2;
                end
            end
        end
        for i =limx(1):limx(2)
            for k = limy(3):limy(3)
                phi2d(i,k) = (phi2d(i,k-1)+phi2d(i,k+1))/2;
                k_effxy(i,k) = (k_effxy(i,k-1)+k_effxy(i,k+1))/2;
                z(i,k) = (z(i,k-1)+z(i,k+1))/2;
                V2d(i,k) = (V2d(i,k-1)+V2d(i,k+1))/2;
                C(i,k) = (C(i,k-1)+C(i,k+1))/2;
            end
        end
        for i =limx(3):(Mx/2)
            for k = limy(3):limy(3)
                phi2d(i,k) = (phi2d(i,k-1)+phi2d(i,k+1))/2;
                k_effxy(i,k) = (k_effxy(i,k-1)+k_effxy(i,k+1))/2;
                z(i,k) = (z(i,k-1)+z(i,k+1))/2;
                V2d(i,k) = (V2d(i,k-1)+V2d(i,k+1))/2;
                C(i,k) = (C(i,k-1)+C(i,k+1))/2;
            end
        end
        %% %% %TAKE ADVANTAGE OF SYMMETRY
        phi2d( ( ( (Mx/2)+1):Mx), 1:(My/2)) =   phi2d( ( ( (Mx/2)):-1:1), 1:(My/2)
            );
        phi2d( ( 1:(Mx)),((My/2)+1):My) =   phi2d( ( 1:(Mx)),((My/2)):-1:1);
        k_effxy( ( ( (Mx/2)+1):Mx), 1:(My/2)) =   k_effxy( ( ( (Mx/2)):-1:1), 1:(
            My/2));
        k_effxy( ( 1:(Mx)),((My/2)+1):My) =   k_effxy( ( 1:(Mx)),((My/2)):-1:1);
        z( ( ( (Mx/2)+1):Mx), 1:(My/2)) =   z( ( ( (Mx/2)):-1:1), 1:(My/2));
        z( ( 1:(Mx)),((My/2)+1):My) =   z( ( 1:(Mx)),((My/2)):-1:1);
```

```
V2d( ( ( (Mx/2)+1):Mx), 1:(My/2)) = V2d( ( ( (Mx/2)):-1:1), 1:(My/2));
V2d( ( 1:(Mx)),((My/2)+1):My) = V2d( ( 1:(Mx)),((My/2)):-1:1);
C( ( ( (Mx/2)+1):Mx), 1:(My/2)) = C( ( ( (Mx/2)):-1:1), 1:(My/2));
C( ( 1:(Mx)),((My/2)+1):My) = C( ( 1:(Mx)),((My/2)):-1:1);
%%%plot parameters for heat equation
subplot(2,3,1); pcolor(X,Y,phi2d'); colorbar; a = colorbar; a.Label.String = '{\it \phi} [W/{m^{3}}]'; xlabel('{\it x} [m]');ylabel('{\it y} [m]');set(gca,'FontSize',18);
subplot(2,3,2);pcolor(X,Y,k_effxy'); colorbar; a = colorbar; a.Label.String = '{\it k_{eff}} [W/{(m K)}]'; xlabel('{\it x} [m]');ylabel('{\it y} [m]');set(gca,'FontSize',18);
subplot(2,3,3);pcolor(X,Y,z'); colorbar; a = colorbar; a.Label.String = '{\it z} [m]'; xlabel('{\it x} [m]');ylabel('{\it y} [m]');set(gca,'FontSize',18);
subplot(2,3,4);pcolor(X,Y,V2d'); colorbar; a = colorbar; a.Label.String = '{\it V} [m^3]'; xlabel('{\it x} [m]');ylabel('{\it y} [m]');set(gca,'FontSize',18);
subplot(2,3,5);pcolor(X,Y,C'); colorbar; a = colorbar; a.Label.String = '{\it C} [J/K]'; xlabel('{\it x} [m]');ylabel('{\it y} [m]');set(gca,'FontSize',18);
%%%Check for stability of the explicit finite difference method %%
for i = 1:Mx
    for k = 1:My
        Fo_vectx(i,k)=k_effxy(i,k)/(C(i,k)/V2d(i,k))*Dt/(Dx*Dx);    % Fourier's number
        Fo_vecty(i,k)=k_effxy(i,k)/(C(i,k)/V2d(i,k))*Dt/(Dy*Dy);    % Fourier's number
    end
end
Fo = max(max(max(Fo_vectx)), max(max(Fo_vecty)));
disp(['Stability requires Fo<1/4. It actually is =',num2str(Fo)])
if Fo>(1/4), disp('This is unstable; increase number of time steps'), end
%%%%%%Bidimensional temperature profile equation by means of finite elements methods %%%
for k = 1:N
    for i = 2:Mx-1
        for j = 2:My-1
            T(i, j, k+1) = T(i, j, k) + (Dt*V2d(i,j))/(C(i,j)*z(i,j))*...
                ( ((k_effxy(i+1,j)+k_effxy(i, j))/2)*((z(i+1,j)+z(i, j))/2)*(T(i+1, j, k)-T(i, j, k))/(Dx^2)...
                -((k_effxy(i,j)+k_effxy(i-1,j))/2)*((z(i,j)+z(i-1,j))/2)*(T(i, j, k)-T(i-1, j, k))/(Dx^2)...
```

```
                    +((k_effxy(i,j)+k_effxy(i,j+1))/2)*((z(i,j)+z(i,j+1))/2)*(
                        T(i, j+1, k)-T(i, j, k))/(Dy^2)...
                    -((k_effxy(i,j)+k_effxy(i,j-1))/2)*((z(i,j)+z(i,j-1))/2)*(
                        T(i, j, k)-T(i, j-1, k))/(Dy^2)...
                    + phi2d(i,j)*z(i, j)+...
                    -((emiss_cara+emiss_comp)*stefan_boltz*(T(i, j, k)^4-T_box
                        ^4)) );
            end
        end
        for i = 2:Mx-1
            T(i, 1, k+1) = T(i, j-1, k);
            T(i, My, k+1) = T(i, j-1, k);
        end
    end
    T_0 =  max(max(max(T)))                                  %Max T [K]
    T_0_C = convtemp(T_0, 'K', 'C')                          %Max T [Celsius]
    %%%%% PLOT TEMPERATURE PROFILE %%%
    % Define temperature for easy plotting
    T_stat_plot = T(:,:,N);                                           % take
        stationary values
    T_stat_plot_central = T(:,(My/2),N);
    % Transitory simulation matrix
    %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    sim = 'n';       % 'y': Save an animation of transitory phase
    sim_frames = 600; % choose number of frames for the simulation
    %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    switch(sim)
        case 'y'
            for i = 1:sim_frames
                T_transit_PLOT(:,:,i) = T(:, :, round(N-(350*(i))+1));   % take
                    transitory values
            end
    end
    %%% PLOT stationary %%%
    my3dplot(X, Y, T_stat_plot, dx, dy, T_b, T_0) % Plot 2D Temp profile
    figure()                                      % Plot 2D central Temp
        profile
    hold on
```

```
        myplot(X, T_stat_plot_central)
        axis([0 dx T_b*0.9 max(T_stat_plot_central)*1.1])
        ylabel('{\it T} [K]'); xlabel('{\it x} [m]');
        xline(L, '-.'); yline(T_b, '--'); yline(T_0, '--')
        hold off
        %% %ANIMATE transitory %%%
        switch(sim)
            case 'y'
                % Contour plot Animation
                % Initialize video
                myVideo = VideoWriter('SimTransit5');   % open video file
                myVideo.FrameRate = 50;                  % can adjust this
                open(myVideo)
                for i = sim_frames:-1:1
                    figure()
                    clf
                    hold on
                    contourf(X,Y,T_transit_PLOT(:,:,i)',15)
                    xlabel('{\it x} [m]'); ylabel('{\it y} [m]'); set(gca,'FontSize',18);
                    colorbar
                    hold off
                    pause(0.01)                          % Pause and grab frame
                    frame = getframe(gcf);               % get frame
                    writeVideo(myVideo, frame);
                    pause(0.01)
                    close()
                end
                close all
                close(myVideo)
        end
                % ---------------------------------------------------------------------------
end
%          % ----------------------------------------------------------------------------
%%====== FUNCIONES ADICIONALES ====== %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Effective thermal conductivity k
function k_eff = effective(k_vect, l_vect)
k_eff = sum(k_vect.*l_vect)/sum(l_vect);
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parabolic temperature eq
function T = temp_parb(a, b, x, phi, k)
T = a + b * (x) - ( phi/(2*k) ) * (x)^2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 'b' parameter in parabolic temperature eq
function b = b_coef(Q, k, A)
b = Q / (k * A);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Linear termperature eq
function T = temp_lin(Q, x, k, A, a)
DT = Q * x / (k * A);
T = a + DT;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plotting function
function myplot(x, y)
plot(x,y, '-k','LineWidth',1)
box on
grid on
grid minor
axis tight
set(gca,'FontSize',18)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3D Plotting function
function my3dplot(x, y, z, dx, dy, ymin, ymax)
%Contour plot
figure()
hold on
contourf(x,y,z',15)
xlabel('{\it x} [m]')
ylabel('{\it y} [m]');
set(gca,'FontSize',18)
colorbar
hold off
% Surf plot
figure()
hold on
```

```
surf(x,y,z')
zlabel('{\it T} [K]')
xlabel('{\it x} [m]')
ylabel('{\it y} [m]');
set(gca,'FontSize',18)
axis([0 dx 0 dy ymin*0.9 ymax*1.03])
hold off
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```