



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



Resolución de EDOs y EDPs

Métodos Numéricos

MEC270 - 2022

Departamento de Ing. Mecánica UTFSM

Martín Achondo Mercado

Rol: 201860005-9

Catalina Santibáñez Mercado

Rol: 201804108-4

Profesor: Romain Jean-Michel Gers

24 de Mayo 2022

Resumen

En el presente trabajo se intentó resolver 2 problemas reales utilizando métodos numéricos de elaboración propia. El primero corresponde al atractor de Lorenz que se modela con un sistema de ecuaciones diferenciales de orden 1. Para variaciones del número de Rayleigh, se pudo obtener las gráficas de las variables que simulan la intensidad del movimiento y las diferencias de temperaturas entre corrientes, dadas por: (x, y, z) . De esta manera, se pudo evidenciar la formación de espirales en torno a los puntos críticos del sistema y su dependencia del número de Rayleigh. Para el segundo problema se utilizaron 3 esquemas de discretización para resolver la ecuación de calor en una pared. La pared se simula con variación sinusoidal en el extremo externo y condición convectiva en el interior. Con esto, se evaluó la posibilidad de situar un aislante y su mejor ubicación para reducir la transferencia de calor. Como resultado se obtuvo que la mejor posición es en la pared interna dado que existe un alto de gradiente inicial producto de la condición inicial. De todas formas, esta conclusión es válida para el caso transiente dado que se obtuvo que en términos estacionarios, o cuasi-estacionarios, su posición no es relevante.

Índice

1. Introducción	4
1.1. Problema 1: Atractor de Lorenz	4
1.2. Problema 2: EDP Ecuación de Calor	5
2. Metodología	6
2.1. Problema 1: EDO	6
2.1.1. Sistema de Lorenz	6
2.1.2. Sistema dinámico caótico	9
2.1.3. Runge-Kutta 4	10
2.2. Problema 2: EDP	12
2.2.1. Discretización	16
2.2.2. Planteamiento de Resultados	22
3. Resultados	23
3.1. Problema 1: Atractor de Lorenz	23
3.2. Problema 2: Ecuación de Calor	39
3.2.1. Transferencia de Calor en Estado Transiente	39
3.2.2. Transferencia de Calor en Estado Estacionario	44
3.2.3. Condición temperatura fija	50
3.2.4. Caso Sin Aislante	51
3.2.5. Tiempos de Cálculo	52
4. Análisis	53
4.1. Problema 1: EDOs Sistema de Lorenz	53
4.2. Problema 2: EDP Calor	55
5. Conclusión	57
6. Referencias	58
7. Anexos	59
7.1. Códigos Pregunta 1	59
7.1.1. Código Principal Pregunta 1	59
7.2. Códigos Pregunta 2	62
7.2.1. Código Principal Pregunta 2	62
7.2.2. Función Euler Explícito	65

7.2.3.	Función Euler Implícito	66
7.2.4.	Función Crank Nicolson	69
7.2.5.	Función Cálculo del Calor con Método del Trapecio	72
7.2.6.	Función Distribución Temperatura Estacionaria	73
7.2.7.	Algoritmo de Thomas	73

1. Introducción

En este trabajo se solicita resolver y analizar el comportamiento de dos problemas. El primero corresponde a un sistema de ecuación diferencial de orden 1 que modela el atractor de Lorenz y que será resuelto utilizando el método de Runge-Kutta. El segundo problema se solicita resolver una ecuación diferencial parcial que modela la transferencia de calor con el objetivo de decidir la instalación de un aislante en un muro de una casa.

1.1. Problema 1: Atractor de Lorenz

En el siguiente problema se nos presenta un sistema de ecuaciones de Lorenz, el cual es un ejemplo fundamental de un sistema de ecuación diferencial de orden 1, tridimensional, no lineal, que tiene un comportamiento caótico para algunos valores de sus parámetros.

$$\begin{cases} \frac{dx(t)}{dt} = \text{Pr}(y(t) - x(t)) \\ \frac{dy(t)}{dt} = \text{Ra } x(t) - y(t) - x(t) z(t) \\ \frac{dz(t)}{dt} = x(t) y(t) - \beta z(t) \end{cases} \quad (1)$$

donde Pr es el número de Prandtl y Ra se llama el número de Rayleigh. Las variables dinámicas x , y y z representan el estado del sistema a cada instante t :

- $x(t)$ es proporcional al intensidad del movimiento de convección.
- $y(t)$ es proporcional a la diferencia de temperatura entre las corrientes ascendentes y descendentes.
- $z(t)$ es proporcional a la diferencia entre el perfil vertical de temperatura y un perfil vertical de temperatura lineal.

Se solicita del sistema de lorenz:

- Definir: ¿Que es un sistema dinámico caótico? ¿ Qué es un atractor extraño?
- Escribir un programa que permite simular el comportamiento del sistema de Lorenz para una condición inicial y los parámetros de control Pr , Ra y β usando un esquema de Runge-Kutta de orden 4.
- Manteniendo los valores $\text{Pr} = 10$, $\beta = 8/3$, y variando el número de Rayleigh Ra , se debe:
 - a) Graficar las soluciones obtenidas en el espacio (x,y,z) para $\text{Ra} = 0$; 5, $\text{Ra} = 10$ y $\text{Ra} = 28$.
 - b) Dibujar $(x; t)$, $(y; t)$ y $(z; t)$ haciendo variar lentamente Ra entre 0 y 30.

1.2. Problema 2: EDP Ecuación de Calor

Para el segundo problema, se pide decidir si es mejor colocar un aislante por el lado interior o exterior del muro de una casa. De esta manera se evaluará la transferencia de calor para mantener el interior de la casa a temperatura constante de T_∞ . La cara externa de la pared sufrirá variaciones sinusoidales simulando el día y la noche. Una imagen esquemática se presenta a continuación: Para la evaluación y decisión de la mejor configuración, se calculará el calor

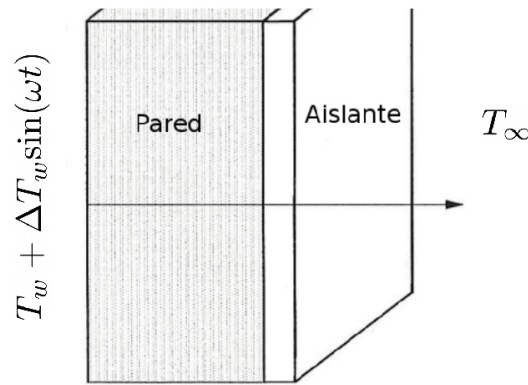


Figura 1: Imagen esquemática de la pared a modelar

total transferido desde el interior de la casa en un plazo de 24 horas para el estado transiente y el cuasi-estacionario que se pueda alcanzar.

Se simulará la pared de concreto con propiedades:

Espesor	$\ell_{conc} = 0.45 \text{ m}$
Difusividad Térmica	$\alpha_{conc} = 10^{-6} \text{ m}^2/\text{s}$
Conductividad Térmica	$\lambda_{conc} = 2 \text{ W/mK}$

Para el aislante, se simulará de madera con propiedades:

Espesor	$\ell_{mad} = 0.05 \text{ m}$
Difusividad Térmica	$\alpha_{mad} = 10^{-7} \text{ m}^2/\text{s}$
Conductividad Térmica	$\lambda_{mad} = 0.1 \text{ W/mK}$

Junto a esto, para la zona externa se supondrá la temperatura de la pared con promedio de $T_w = 10^\circ\text{C}$ y variación sinusoidal de amplitud $\Delta T_w = 5^\circ\text{C}$ con periodo de 24 horas. Por último, la temperatura interior de la casa se asumirá constante de $T_\infty = 20^\circ\text{C}$ con coeficiente de convección $h = 4 \text{ W/m}^2\text{k}$

2. Metodología

2.1. Problema 1: EDO

2.1.1. Sistema de Lorenz

Este sistema de ecuaciones permite modelar los rollos de convección por temperatura que se producen en la atmósfera terrestre, pues la dinámica atmosférica estaba gobernada por complejas ecuaciones diferenciales que relacionaban ciertas variables con su variación puntual en el tiempo y en el espacio, y una gran cantidad de parámetros. De este trabajo se obtuvo un sistema de ecuación diferencial de orden 1, donde las derivadas en el tiempo de cada variable dependían de combinaciones lineales y no lineales de dichas variables y algunos parámetros numéricos.

$$\begin{cases} \frac{dx(t)}{dt} = \text{Pr}(y(t) - x(t)) \\ \frac{dy(t)}{dt} = \text{Ra } x(t) - y(t) - x(t) z(t) \\ \frac{dz(t)}{dt} = x(t) y(t) - \beta z(t) \end{cases} \quad (2)$$

donde Pr es el conocido número de Prandtl, Ra es el número de Rayleigh relativo y β está relacionado con el tamaño físico del sistema y corresponde a un número adimensional.

- La variable $x(t)$ es proporcional a la intensidad de las corrientes convectivas, luego es una representación de la intensidad del movimiento convectivo. Si $x > 0$ el flujo de convección se produce en sentido horario, mientras que si $x < 0$ se produce en sentido antihorario.
- La variable $y(t)$ representa la diferencia de temperaturas entre los flujos de convección ascendentes y descendentes. Esta función es la que caracteriza la variación de temperatura en el fenómeno de convección que da origen a las celdas de Bénard.
- La variable $z(t)$ es la desviación de ΔT , como función lineal, desde el centro de la celda Bénard. Entonces, la función z es una representación de la variación global de temperatura de todas las moléculas del fluido desde T_0 hasta $T_0 + \Delta T$.

Propiedades del sistema de Lorenz:

El sistema de Lorenz se puede representar como:

$$\frac{dU}{dt} = F(U) \quad (3)$$

Con $U = (x, y, z)$

Existencia y unicidad: Las componentes de la función $F(x, y, z)$ y las derivadas parciales de esta, son funciones continuas en \mathbb{R}^3 , Por lo tanto tenemos garantizada la existencia y unicidad de soluciones para problemas de valor inicial con $t_0 \in \mathbb{R}$ y $x_0 \in \mathbb{R}^3$.

Simetría espacial: El sistema de Lorenz presenta una simetría natural que persiste para cualquier valor de los parámetros. Su significado radica en que si $(x(t), y(t), z(t))$ es una solución del sistema, entonces $(-x(t), -y(t), z(t))$ también lo es, pero es solo en las dos dimensiones.

Eje Z: Se denota como invariante, es decir que para todo punto del eje Z, la órbita que pasa por dicho punto permanece en este eje para todo instante $t \in \mathbb{R}$. Un punto importante es que las trayectorias que comienzan en este eje, permanecen en él y tienden hacia el origen cuando $t \rightarrow +\infty$.

Estabilidad: Para poder determinar la estabilidad de un punto crítico \bar{x} es suficiente con encontrar un entorno de dicho punto con la propiedad de que las órbitas que comienzan en dicho entorno permanezcan en él para todo instante positivo. Esto es equivalente a probar que el campo vectorial es, o bien tangente al contorno del entorno mencionado anteriormente, o bien apunta hacia el interior del mismo. Y, además, esto se cumple a medida que reducimos este entorno al punto crítico en cuestión.

Ahora bien, Si \bar{x} es un punto de equilibrio estable de un sistema autónomo no lineal y toda solución de dicho sistema converge a él cuando $t \rightarrow +\infty$, se dice entonces que \bar{x} es globalmente asintóticamente estable.

Existe un comportamiento importante para el parámetro Ra , pues las soluciones del sistema de Lorenz tenderán a cierto punto.

- Para $Ra < 1$, todas las soluciones del sistema de Lorenz tienden al origen. Por lo tanto tienen un comportamiento cualitativo similar entre las soluciones.
- Para $Ra = 1$, se pone de manifiesto la existencia de una bifurcación de Pitchfork.

Puntos de equilibrio: Estos corresponden a las soluciones constantes de sistema de Lorenz, por

lo que se iguala cada ecuación a cero. Dado esto, se obtienen que los puntos de equilibrio son:

$$P_1 = (0, 0, 0) \quad (4)$$

$$P_2 = (\sqrt{\beta(Ra - 1)}, \sqrt{\beta(Ra - 1)}, Ra - 1) \quad (5)$$

$$P_3 = (-\sqrt{\beta(Ra - 1)}, -\sqrt{\beta(Ra - 1)}, Ra - 1) \quad (6)$$

Para diferentes valores de Ra se tiene:

- Caso $Ra < 1$: El único punto de equilibrio que se presenta es el origen y es asintóticamente globalmente estable.
- Caso $Ra = 1$: De modo similar, el único punto de equilibrio que aparece es el origen.
- Caso $Ra > 1$: Se tienen los 3 puntos de equilibrio obtenidos.

Para P_1 (origen) se tiene una inestabilidad. Para los puntos P_2 y P_3 se debe considerar el punto crítico dado por:

$$Ra_{crítico} = \frac{Pr(Pr + \beta + 3)}{(Pr - \beta - 1)} \quad (7)$$

Pues este valor indicará la estabilidad del sistema de Lorenz dependiendo de Ra . Así, para valores entre $1 < Ra < Ra_{crítico}$ los puntos de equilibrio P_2 y P_3 son asintóticamente estables.

Para $Ra = Ra_{crítico}$ se tienen autovalores complejos (de la linealización de la matriz jacobiana en el punto crítico) que cruzan el eje imaginario dando lugar a una bifurcación de tipo Hopf, en la cual los puntos P_2 y P_3 pierden su estabilidad.

Para $Ra > Ra_{crítico}$ las raíces complejas del polinomio tienen parte real positiva. Por tanto, los puntos de equilibrio P_2 y P_3 son inestables.

Ante esto se entrega una tabla resumen de las propiedades del sistema de Lorenz para comprender los resultados gráficos de este problema.

Tabla 1: Resumen de propiedades

Ra	Puntos de equilibrio	Estabilidad	Bifurcación
$0 < Ra < 1$	Origen	Global. Asintótic. Estable	-
$Ra = 1$	Origen	Asintótic. Estable	Pitchfork
$1 < Ra < Ra_H$	Origen	Inestable	-
	P_2	Asintótic. Estable	-
	P_3	Asintótic. Estable	-
$Ra = Ra_H$	-	-	Hopf
$Ra > Ra_H$	Origen	Inestable	-
	P_2	Inestable	-
	P_3	Inestable	-

2.1.2. Sistema dinámico caótico

La teoría del caos es una rama de las matemáticas, la física y otras ciencias que estudian sistemas dinámicos que son muy sensibles a las variaciones en las condiciones iniciales, provocando un comportamiento impredecible en el tiempo.

Por lo tanto las soluciones del sistema de Lorenz son definitivamente ordenadas (siguen una espiral). Nunca se paran en un punto, ni se repiten, ni son periódicas. La representación gráfica se le conoce con el nombre Atractor de Lorenz.

Un atractor es una región del espacio de fases hacia la cual convergen las trayectorias posibles de un sistema, en general, los atractores poseen la forma de alguna figura geométrica conocida. El atractor de los sistemas estables es un punto, mientras que para los sistemas periódicos es un ciclo límite. El atractor de un sistema caótico recibe el nombre de extraño o caótico y está caracterizado por presentar un comportamiento caótico e impredecible. Este ocupa un subespacio de dimensión menor que el diagrama de fases. Los atractores extraños reaccionan de manera muy sensible a pequeñas variaciones de las condiciones iniciales. Los atractores anteriormente mencionados - un punto, un ciclo límite- son variedades, mientras que los atractores extraños no lo son.

2.1.3. Runge-Kutta 4

Los métodos de Runge-Kutta (RK) es utilizado para resolver sistemas de ecuaciones diferenciales ordinarias, especialmente en casos en que solución no puede hallarse por los métodos convencionales. Este método es más preciso y estable que el método de Euler dado que es de orden 4, entregando una estimación más precisa de la integral de la EDO.

Se define un intervalo de tiempo $T = [a, b]$, paso de tiempo h y condiciones iniciales (x_0, y_0, z_0) . Este método trabaja con reevaluaciones sucesivas n de la función f dada por:

$$\frac{dU}{dt} = f(x, y, z) \quad (8)$$

Con $U = (x, y, z)$

$$h = \frac{(b - a)}{n} \quad (9)$$

Para la cual entonces se genera la siguiente iteración, con h el paso en la variable t :

$$U(t + h) = U(t) + \int_t^{t+h} f(s, U) ds \quad (10)$$

Los métodos de Runge Kutta aproximan la integral planteada como:

$$\int_t^{t+h} f(s, U) ds \approx h \sum_{\ell=1}^m \gamma_{\ell} k_{\ell} \quad (11)$$

En donde γ corresponden a los pesos y k a las evaluaciones convenientes de f .

El método de RK-4 es la aplicación a 4 etapas, dadas por:

$$\begin{aligned} k_1^n &= f(x_n, y_n, z_n) & n &= 1, 2, 3 \dots n \\ k_2^n &= f\left(x_n + h \frac{k_1^n}{2}, y_n + h \frac{k_1^n}{2}, z_n + h \frac{k_1^n}{2}\right) \\ k_3^n &= f\left(x_n + h \frac{k_2^n}{2}, y_n + h \frac{k_2^n}{2}, z_n + h \frac{k_2^n}{2}\right) \\ k_4^n &= f\left(x_n + h k_3^n, y_n + h k_3^n, z_n + h k_3^n\right) \end{aligned} \quad (12)$$

Proceso sucesivo: Resultados del sistema.

$$x_{n+1} = x_n + \frac{h}{6}(k_1^n + 2k_2^n + 2k_3^n + k_4^n) \quad (13)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1^n + 2k_2^n + 2k_3^n + k_4^n) \quad (14)$$

$$z_{n+1} = z_n + \frac{h}{6}(k_1^n + 2k_2^n + 2k_3^n + k_4^n) \quad (15)$$

Se puede observar una diferencia al metodo de Euler: el algoritmo exige 4 veces más cálculos en cada paso y provocando que sea más lento. Además, los errores de redondeos se acumulan más rápidamente. Sin embargo, estos defectos son compensados por una mejor precisión.

2.2. Problema 2: EDP

En este problema se pide resolver la ecuación de calor en una pared compuesta por un muro y un aislante. Para simplificar el problema, se considerará una transferencia de calor unidimensional, propiedades de los materiales constantes y la no existencia de generación de energía.

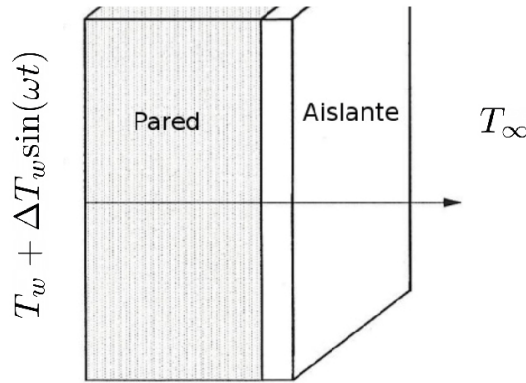


Figura 2: Representación de la pared a modelar

El problema, la ecuación con sus condiciones de borde e iniciales se puede plantear como:

$$\left\{ \begin{array}{ll} T(0, t) = T_w + \Delta T_w \sin(\omega t) & t > 0 \\ \frac{\partial T}{\partial t} = \alpha_1 \frac{\partial^2 T}{\partial x^2} & 0 < x < \ell_1, t > 0 \\ -\lambda_1 \frac{\partial T}{\partial x} = -\lambda_2 \frac{\partial T}{\partial x} & x = \ell_1, t > 0 \\ \frac{\partial T}{\partial t} = \alpha_2 \frac{\partial^2 T}{\partial x^2} & \ell_1 < x < \ell, t > 0 \\ -\lambda_2 \frac{\partial T}{\partial x} = h(T(\ell, t) - T_\infty) & x = \ell, t > 0 \end{array} \right. \quad (16)$$

Donde α_1 y λ_1 hacen referencia a la difusividad y conductividad térmica del material externo del muro (zona 1) y α_2 , λ_2 al interno (zona 2). Para este trabajo se probarán las dos configuraciones, aislante en la zona externa y en la zona interna y se intentará justificar cual es más conveniente.

Se nota que para el borde izquierdo se tiene una condición de Dirichlet para la temperatura, la cual es oscilante.

$$T(0, t) = T_w + \Delta T_w \sin(\omega t) \quad (17)$$

En la interfaz de los dos materiales existe una condición de Neumann para la igualdad de flujo

de calor.

$$-\lambda_1 \frac{\partial T}{\partial x} \Big|_{x=\ell_1} = -\lambda_2 \frac{\partial T}{\partial x} \Big|_{x=\ell_1} \quad (18)$$

Para el borde externo, se tendrá una condición de borde de Robin igualando el calor por conducción con el calor por convección. Se elige esta forma dado que es la más realista. De todas formas, en los resultados se comparará el error al utilizar una condición de Dirichlet en este borde fijando T_∞

$$-\lambda_2 \frac{\partial T}{\partial x} \Big|_{x=\ell} = h(T(\ell, t) - T_\infty) \quad (19)$$

Para poder encontrar una aproximación teórica de la solución y así validar el código, se intentará obtener el comportamiento de la distribución de temperatura. Dada la condición periódica de la pared externa, la solución $T(x, t)$ de la ecuación 16 tiene que estar compuesto como:

$$T(x, t) = T_{tr}(x, t) + T_{qs}(x, t) \quad (20)$$

En donde T_{tr} corresponde al término transiente y T_{qs} corresponde al término cuasi-estacionario. Para el término transiente, se espera que luego de un tiempo considerable, el término se anule.

$$\lim_{t \rightarrow \infty} T_{tr}(x, t) = 0 \quad (21)$$

Además, se nota que la solución cuasi-estacionaria se puede descomponer como:

$$T_{qs}(x, t) = T_s(x) + A(x) \sin(\omega t + \phi(x)) \quad (22)$$

Lo que significa que después de un tiempo considerable, la temperatura en cada punto de la pared va a oscilar con una amplitud dependiente de la posición alrededor de su temperatura promedio. Analíticamente es sencillo obtener la componente estacionaria T_s . Basta resolver la ecuación:

$$\frac{d^2 T_s}{dx^2} = 0 \quad (23)$$

Con las siguientes condiciones de borde e iniciales:

$$\begin{cases} T(0, t) = T_w + \Delta T_w \sin(\omega t) & t > 0 \\ -\lambda_1 \frac{dT_s}{dx} = -\lambda_2 \frac{dT_s}{dx} & x = \ell_1, t > 0 \\ -\lambda_2 \frac{dT_s}{dx} = h(T_s(\ell, t) - T_\infty) & x = \ell, t > 0 \end{cases} \quad (24)$$

La solución tiene la siguiente forma:

$$T_s(x) = \begin{cases} \frac{h\lambda_2(T_\infty - T_w)}{\varphi}x + T_w & 0 \leq x \leq \ell_1 \\ \frac{h\lambda_1(T_\infty - T_w)}{\varphi}x + \frac{h(\ell\lambda_1T_w - \ell_1T_\infty(\lambda_1 - \lambda_2)) + \lambda_1\lambda_2}{\varphi} & \ell_1 < x \leq \ell \end{cases} \quad (25)$$

En donde φ tiene el valor de:

$$\varphi = h(\ell\lambda_1 - \ell_1(\lambda_1 - \lambda_2)) + \lambda_1\lambda_2 \quad (26)$$

Se nota claramente que el perfil de temperatura estacionario es lineal. Además, la solución cuasi-estacionaria oscilará respecto a esta solución estacionaria.

Para justificar que la solución oscilatoria es en efecto solución de la ecuación 16, basta remplazar T_{qs} , así se obtiene:

$$\frac{\partial T_{osc}}{\partial t} = \alpha \frac{\partial^2 T_{osc}}{\partial x^2} \quad (27)$$

En donde:

$$T_{osc}(x, t) = A(x) \sin(\omega t + \phi(x)) \quad (28)$$

Escribiendo este componente oscilatorio como:

$$A(x) \sin(\omega t + \phi(x)) = \text{Im} (A(x)e^{i\phi(x)}e^{i\omega t}) \quad (29)$$

Con el siguiente cambio de variables:

$$U(x) = A(x)e^{i\phi(x)} \quad (30)$$

Y reemplazando en 16, se obtiene:

$$\frac{d^2 U}{dx^2} - \frac{i\omega}{\alpha} U = 0 \quad (31)$$

La cual en efecto tiene una solución dada por:

$$U(x) = C_1 e^{-\sqrt{\omega/2\alpha}(1+i)x} + C_2 e^{\sqrt{\omega/2\alpha}(1+i)x} \quad (32)$$

La función U entrega información sobre la amplitud de la oscilación en cada posición y su desfase. No es de interés obtenerla ya que el problema se resolverá numéricamente. Lo importante

es percatarse que la componente oscilante existe:

$$T_{osc}(x, t) = \text{Im} (U(x)e^{i\omega t}) \quad (33)$$

Y resuelve la ecuación diferencial 16.

Con la solución estacionaria obtenida y el conocimiento de la solución oscilante se podrá validar el código elaborado.

Respecto al flujo de calor, se nota que es de interés el flujo en la cara interna para evaluar cuanto calor pierde el ambiente interior al muro. El flujo entonces tendrá la siguiente forma:

$$\dot{q}_{qs} = -\lambda_2 \frac{\partial T_{qs}}{\partial x} \Big|_{x=\ell} \quad (34)$$

En donde:

$$\frac{\partial T_{qs}}{\partial x} = T'_s(x) + A'(x) \sin(\omega t + \phi(x)) + A(x) \phi'(x) \cos(\omega t + \phi(x)) \quad (35)$$

Dado que el flujo de calor es oscilatorio, será de interés el calor total transferido en un periodo (Δt) cuando se alcanza el estado cuasi-estacionario.

$$q_{qs} = \int_{\Delta t_p} \dot{q}_{qs} dt = \int_{\Delta t_p} -\lambda_2 \frac{\partial T_{qs}}{\partial x} \Big|_{x=\ell} dt \quad (36)$$

De donde se nota que la integral sobre un periodo del término oscilante es nulo. Así, el calor total queda como:

$$q_{qs} = -\lambda_2 \int_{\Delta t_p} T'_s(\ell) dt \quad (37)$$

Y dado que se conoce el perfil estacionario, se puede evaluar directamente el calor transferido en un periodo.

$$q_{qs} = -\lambda_2 \frac{h\lambda_1(T_\infty - T_w)}{h(\ell\lambda_1 - \ell_1(\lambda_1 - \lambda_2)) + \lambda_1\lambda_2} \Delta t_p \quad (38)$$

Con este resultado se podrá plantear una segunda evaluación al código cuando alcanza el estado cuasi-estacionario. Notar en todo caso que con un simple cambio de variables se puede demostrar que este calor total es independiente de la posición del aislante. Esto se debe a se alcanza este estado cuasi-estacionario y además, la “resistencia térmica” es la misma independiente de la configuración.

2.2.1. Discretización

La ecuación 16 se resolverá numéricamente utilizando 3 esquemas de discretización para obtener la distribución de temperatura a lo largo del muro para distintos instantes de tiempo por el método de diferencias finitas. Así, se podrá comparar el orden de convergencia de cada esquema. Para ambas zonas, 1 y 2, se utilizará el mismo espaciamiento espacial Δx para mantener el orden de convergencia de cada esquema. La cantidad de nodos entonces será:

$$N = \frac{\ell}{\Delta x} + 1 \quad (39)$$

De esta manera cada nodo $j = 1, 2, 3, \dots, k-1, k, k+1, \dots, N-1, N$. Es importante elegir bien el espaciamiento espacial para que calce un nodo en la interfaz. Este nodo se representará como $j = k$. El paso temporal será Δt , por lo que la cantidad de pasos temporales contando el instante inicial será de:

$$N_t = \frac{t_f}{\Delta t} + 1 \quad (40)$$

Un número adimensional importante que tiene relación con la discretización del problema es el número de Fourier, que puede ser definido para cada zona $i = 1, 2$ como:

$$\sigma_i = \alpha_i \frac{\Delta t}{\Delta x^2} \quad (41)$$

Con esto, una representación de la discretización espacial y temporal a realizar se visualiza esquemáticamente en la siguiente imagen.

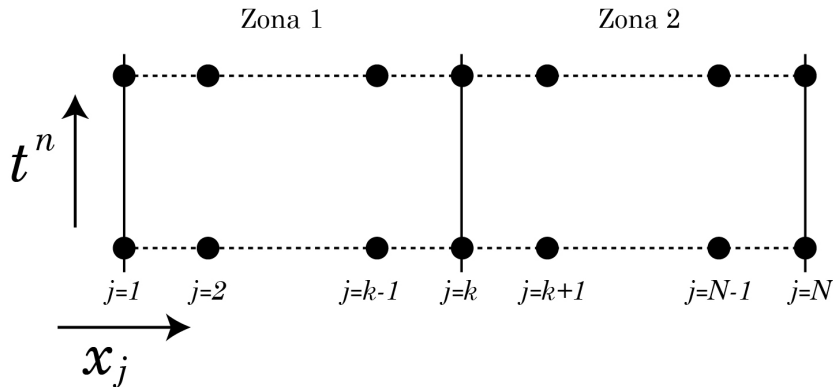


Figura 3: Esquema de discretización

Para todos los esquemas a presentar, se discretizará la segunda derivada de la temperatura como:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{j-1} - 2T_j + T_{j+1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \quad (42)$$

Las condiciones de borde se discretizarán con esquemas de orden 1 y 2 (las derivadas).

- Borde en pared exterior (Dirichlet), $j = 1$

$$T_1^n = T_w + \Delta T_w \sin(\omega t^n) \quad (43)$$

- Interfaz entre los dos materiales (Neumann), $j = k$

- Esquema orden 1:

$$\lambda_1 \frac{T_k^n - T_{k-1}^n}{\Delta x} = \lambda_2 \frac{T_{k+1}^n - T_k^n}{\Delta x} \quad (44)$$

- Esquema orden 2:

$$\lambda_1 \frac{T_{k-2}^n - 4T_{k-1}^n + 3T_k^n}{2\Delta x} = \lambda_2 \frac{-3T_k^n + 4T_{k+1}^n - T_{k+2}^n}{2\Delta x} \quad (45)$$

- Borde interior (Robin), $j = N$

- Esquema orden 1:

$$-\lambda_2 \frac{T_N^n - T_{N-1}^n}{\Delta x} = h(T_N^n - T_\infty) \quad (46)$$

- Esquema orden 2:

$$-\lambda_2 \frac{T_{N-2}^n - 4T_{N-1}^n + 3T_N^n}{2\Delta x} = h(T_N^n - T_\infty) \quad (47)$$

La condición inicial será de fijar la misma temperatura en toda la pared.

$$T_j^0 = T_\infty \quad 1 \leq j \leq N \quad (48)$$

Ahora se detallarán los esquemas de discretización de la ecuación de calor y los esquemas que se utilizarán en sus condiciones de borde.

Esquema Euler Explícito Discretiza la ecuación de calor utilizando los valores en el paso anterior. Así, la ecuación queda como:

$$\frac{T^{n+1} - T^n}{\Delta t} = f(t^n, T^n) \quad (49)$$

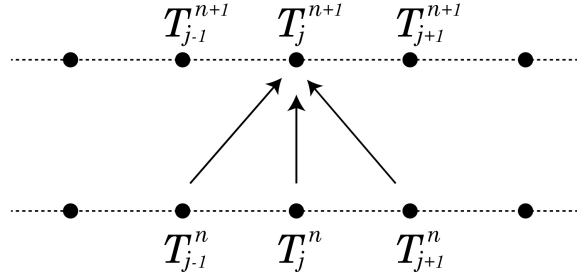


Figura 4: Esquema de discretización Euler Explícito

Así, la ecuación queda como:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha_i \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{\Delta x^2} \quad (50)$$

Despejando T^{n+1} y utilizando las condiciones de borde de orden 1, se forman las siguientes ecuaciones para cada nodo j . Con esto, el método queda con orden $\mathcal{O}(\Delta x, \Delta t)$, primer orden en tiempo y espacio.

$$\begin{cases} T_1^{n+1} = T_w + \Delta T_w \sin(\omega t^{n+1}) & j = 1 \\ T_j^{n+1} = T_j^n + \sigma_1(T_{j-1}^n - 2T_j^n + T_{j+1}^n) & 1 < j < k \\ T_j^{n+1} = T_j^n + \sigma_2(T_{j-1}^n - 2T_j^n + T_{j+1}^n) & k < j < N \\ T_k^{n+1} = \frac{\lambda_1 T_{k-1}^{n+1} + \lambda_2 T_{k+1}^{n+1}}{\lambda_1 + \lambda_2} & j = k \\ T_N^{n+1} = \frac{\lambda_2 T_{N-1}^{n+1} + \Delta x h T_\infty}{\Delta x h \lambda_2} & j = N \end{cases} \quad (51)$$

Esquema Euler Implícito Discretiza la ecuación de calor utilizando los valores en el nodo del paso anterior y los cercanos en el paso actual. Así, la ecuación queda como:

$$\frac{T^{n+1} - T^n}{\Delta t} = f(t^{n+1}, T^{n+1}) \quad (52)$$

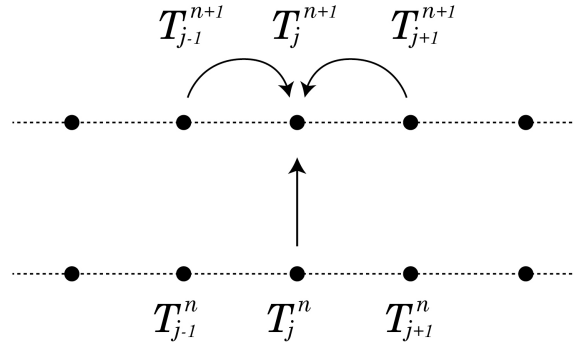


Figura 5: Esquema de discretización Euler Implícito

Así, la ecuación queda como:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha_i \frac{T_{j-1}^{n+1} - 2T_j^{n+1} + T_{j+1}^{n+1}}{\Delta x^2} \quad (53)$$

Despejando T^{n+1} y utilizando las condiciones de borde de orden 2, se forman las siguientes ecuaciones para cada nodo j . Con esto, el método queda con orden $\mathcal{O}(\Delta x^2, \Delta t)$, segundo orden en espacio y primero en el tiempo.

$$\begin{cases} T_1^{n+1} = T_w + \Delta T_w \sin(\omega t^{n+1}) & j = 1 \\ -T_{j-1}^{n+1} + \left(2 + \frac{1}{\sigma_1}\right) T_j^{n+1} - T_{j+1}^{n+1} = \frac{1}{\sigma_1} T_j^n & 1 < j < k \\ \lambda_1 T_{k-2}^{n+1} - 4\lambda_1 T_{k-1}^{n+1} + 3(\lambda_1 + \lambda_2) T_k^{n+1} - 4\lambda_2 T_{k+1}^{n+1} + \lambda_2 T_{k+2}^{n+1} = 0 & j = k \\ -T_{j-1}^{n+1} + \left(2 + \frac{1}{\sigma_2}\right) T_j^{n+1} - T_{j+1}^{n+1} = \frac{1}{\sigma_2} T_j^n & k < j < N \\ \theta T_{N-2}^{n+1} - 4\theta T_{N-1}^{n+1} + (3\theta + h) T_N^{n+1} = h T_\infty & j = N \end{cases} \quad (54)$$

Notar que se utiliza $\theta = \frac{\lambda_2}{2\Delta x}$. El sistema de ecuaciones que se forma es pentadiagonal y será resuelto con el algoritmo de Thomas para sistemas tridiagonales por bloques.

Esquema Euler Crank Nicolson Discretiza la ecuación de calor utilizando los valores del paso anterior y actual. Así, la ecuación queda como:

$$\frac{T^{n+1} - T^n}{\Delta t} = \frac{1}{2} (f(t^{n+1}, T^{n+1}) + f(t^n, T^n)) \quad (55)$$

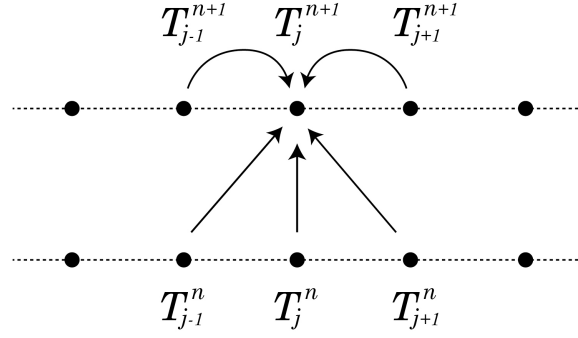


Figura 6: Esquema de discretización Crank Nicolson

Así, la ecuación queda como:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{\alpha_i}{2} \left(\frac{T_{j-1}^{n+1} - 2T_j^{n+1} + T_{j+1}^{n+1}}{\Delta x^2} + \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{\Delta x^2} \right) \quad (56)$$

Despejando T^{n+1} y utilizando las condiciones de borde de orden 2, se forman las siguientes ecuaciones para cada nodo j . Con esto, el método queda con orden $\mathcal{O}(\Delta x^2, \Delta t^2)$, segundo orden en espacio y tiempo.

$$\begin{cases} T_1^{n+1} = T_w + \Delta T_w \sin(\omega t^{n+1}) & j = 1 \\ -T_{j-1}^{n+1} + \left(2 + \frac{2}{\sigma_1}\right) T_j^{n+1} - T_{j+1}^{n+1} = T_{j-1}^n + \left(2 - \frac{2}{\sigma_1}\right) T_j^n + T_{j+1}^n & 1 < j < k \\ \lambda_1 T_{k-2}^{n+1} - 4\lambda_1 T_{k-1}^{n+1} + 3(\lambda_1 + \lambda_2) T_k^{n+1} - 4\lambda_2 T_{k+1}^{n+1} + \lambda_2 T_{k+2}^{n+1} = 0 & j = k \\ -T_{j-1}^{n+1} + \left(2 + \frac{2}{\sigma_2}\right) T_j^{n+1} - T_{j+1}^{n+1} = T_{j-1}^n + \left(2 - \frac{2}{\sigma_2}\right) T_j^n + T_{j+1}^n & k < j < N \\ \theta T_{N-2}^{n+1} - 4\theta T_{N-1}^{n+1} + (3\theta + h) T_N^{n+1} = h T_\infty & j = N \end{cases} \quad (57)$$

El sistema de ecuaciones que se forma es pentadiagonal y será resuelto con el algoritmo de Thomas para sistemas tridiagonales por bloques.

Para el **flujo de calor**, dado que es de interés la transferencia en la pared interna, se puede aprovechar de la condición de borde utilizada para calcularlo como:

$$\dot{q}_N^n = h(T_N^n - T_\infty) \quad (58)$$

De todas formas, la comparación se realizará con el calor total transferido en un periodo. Dado esto, es de relevancia calcular la integral:

$$q_N = \int \dot{q}_N dt \quad (59)$$

Esta integral se aproximará utilizando el método del trapecio el cual tiene orden $\mathcal{O}(\Delta t^3)$. Así, para un periodo completo, el calor total transferido en el borde interno queda como:

$$q_N = \frac{\Delta t}{2} \sum_n q_N^n + q_N^{n+1} \quad (60)$$

Algoritmo de Thomas Dado que en los esquemas de Euler Implícito y Crank Nicolson presentados aparece un sistema pentadiagonal por las aproximaciones de segundo orden en las condiciones de borde, es importante utilizar un método eficiente para resolverlo. Dado esto, se utilizará el algoritmo de Thomas para sistemas tridiagonales por bloques. Esquemáticamente, se está resolviendo el sistema de M ecuaciones matriciales:

$$A_i X_{i-1} + C_i X_i + B_i X_{i+1} = F_i \quad (61)$$

En donde A, B, C son matrices de 3×3 y X, F vectores de 3 componentes. El algoritmo de resolución tiene dos pasos. Primero, se calcula la matriz Λ y el vector β :

$$\begin{cases} \Lambda_1 = G_1 B_1 \\ \Lambda_i = G_i B_i \quad i = 2, \dots, M-1 \end{cases} \quad (62)$$

Y también:

$$\begin{cases} \beta_1 = G_1 F_1 \\ \beta_i = G_i (F_i - A_i \Lambda_{i-1}) \quad i = 2, \dots, M \end{cases} \quad (63)$$

En donde $G_1 = C_1^{-1}$ y $G_i = (C_i - A_i \Lambda_i)^{-1}$. Con esto, se puede encontrar el vector solución X con la iteración:

$$\begin{cases} X_M = \beta_M \\ X_i = \beta_i - \Lambda_i X_{i+1} \quad i = M-1, \dots, 1 \end{cases} \quad (64)$$

2.2.2. Planteamiento de Resultados

En las simulaciones para pasos de tiempo de 1 y 10 segundos y espaciamientos en la posición de 0.02 y 0.005 metros. Estos pasos serán utilizados en los 3 esquemas presentados anteriormente para modelar el perfil de temperatura en el estado transiente y el cuasi-estacionario. De esta forma se podrá comparar la convergencia de los esquemas con su costo computacional para así evaluar la conveniencia de utilizar uno o el otro.

Además, se calculará el calor total perdido por el interior en estas dos condiciones (transiente y cuasi-estacionario). Por último, se compararán los resultados con la teoría presentada al comienzo para validar los resultados y así intentar encontrar la amplitud del término oscilante en la temperatura. Todo esto para verificar si es más conveniente situar el aislante dentro o fuera de la pared. Además, se compararán los resultados con el caso si no existe aislante.

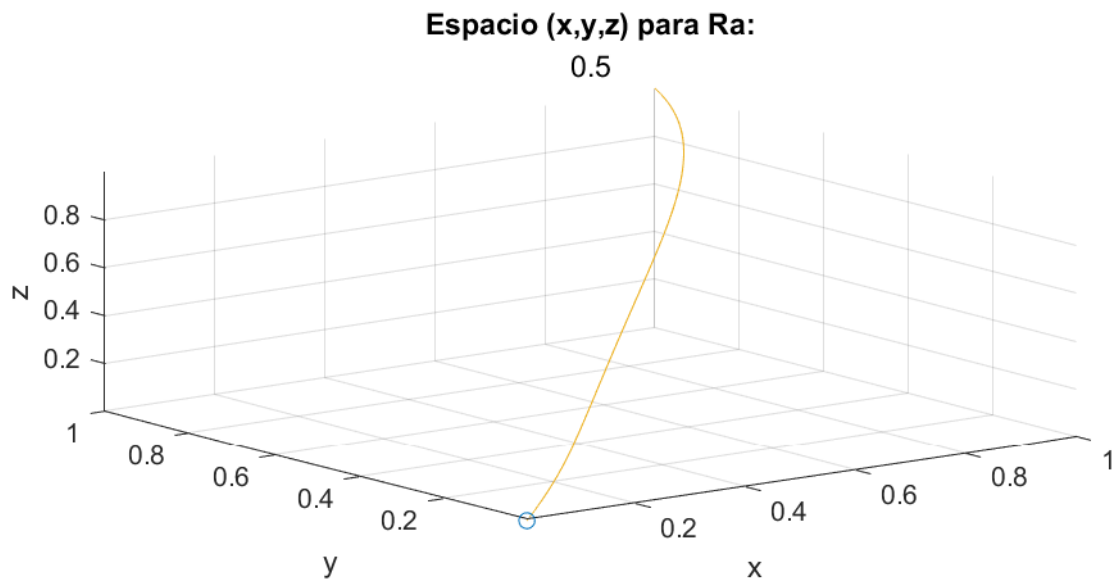
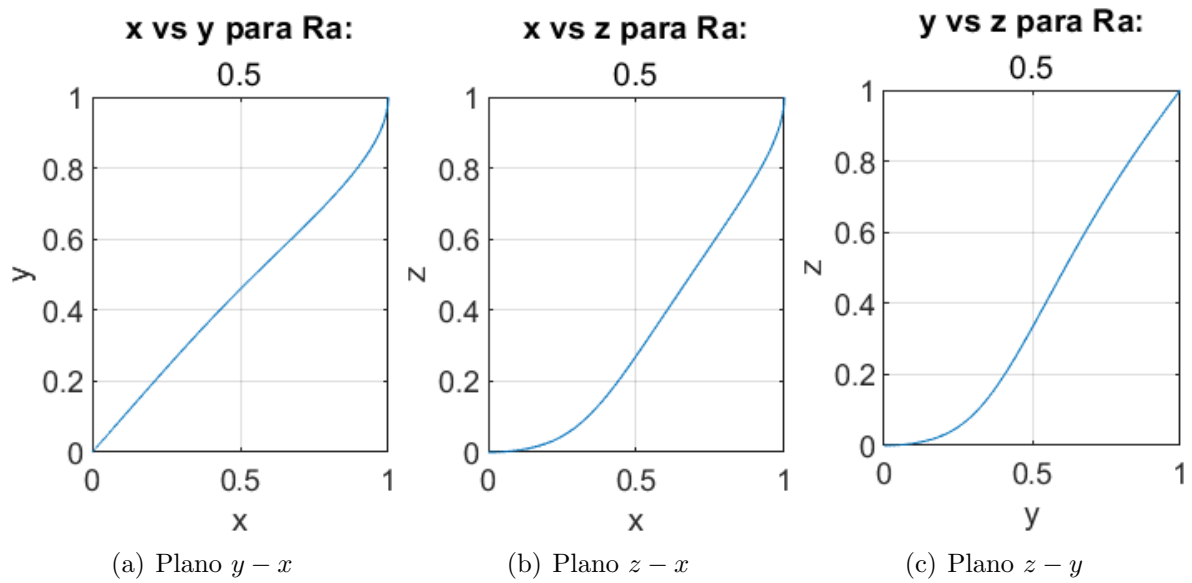
3. Resultados

3.1. Problema 1: Atractor de Lorenz

Como fue mencionado el sistema de Lorenz permite modelar los rollos de convección que se producen en la atmósfera terrestre, para comprender el comportamiento se realiza un programa en MATLAB, donde se utilizó el método de Runge-Kutta de orden 4 para la resolución del sistema de Lorenz presentado.

Para visualizar este sistema se tomaron los siguientes parámetros: $Pr = 10$, $\beta = 8/3$, valor inicial $(1,1,1)$, intervalo de tiempo de $[0,100]$, número de iteraciones $n = 10000$, paso de tiempo $h = 0.01$. Ahora bien, el comportamiento depende del número de Rayleigh. Por lo mismo este problema se divide en dos partes.

La primera es graficar el sistema dinámico manteniendo fijo Pr y β , pero variando el valor de $Ra = [0.5, 10, 28]$, permitiendo visualizar el comportamiento del sistema en el espacio.

Figura 7: Espacio (x,y,z) para $Ra=0.5$ Figura 8: Variación de x, y, z en planos para $Ra=0.5$

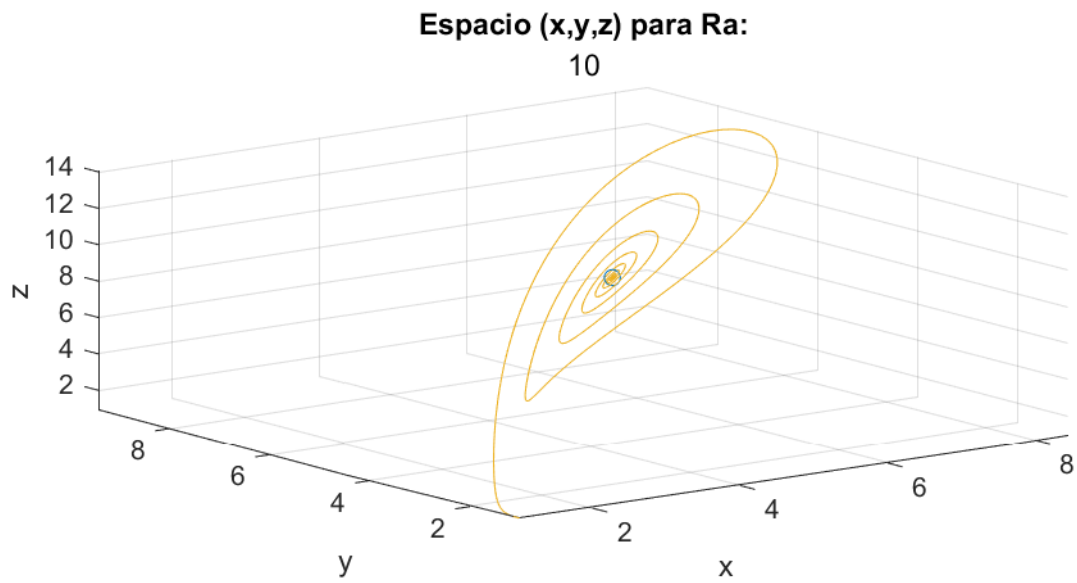
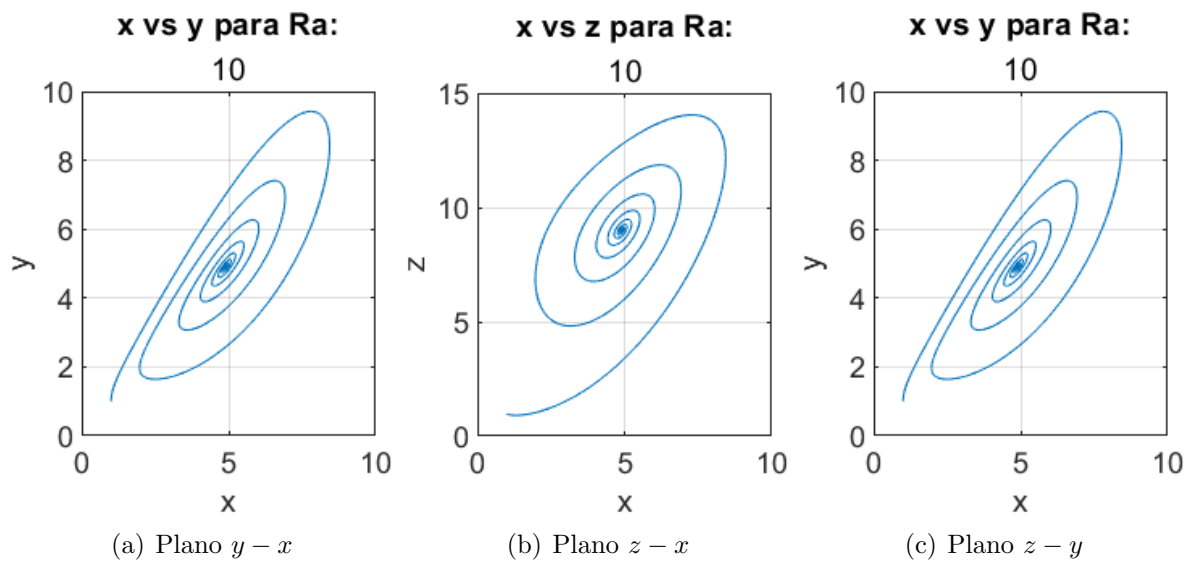


Figura 9: Espacio (x,y,z) para Ra=10

Figura 10: Variación de x, y, z en planos para Ra=10

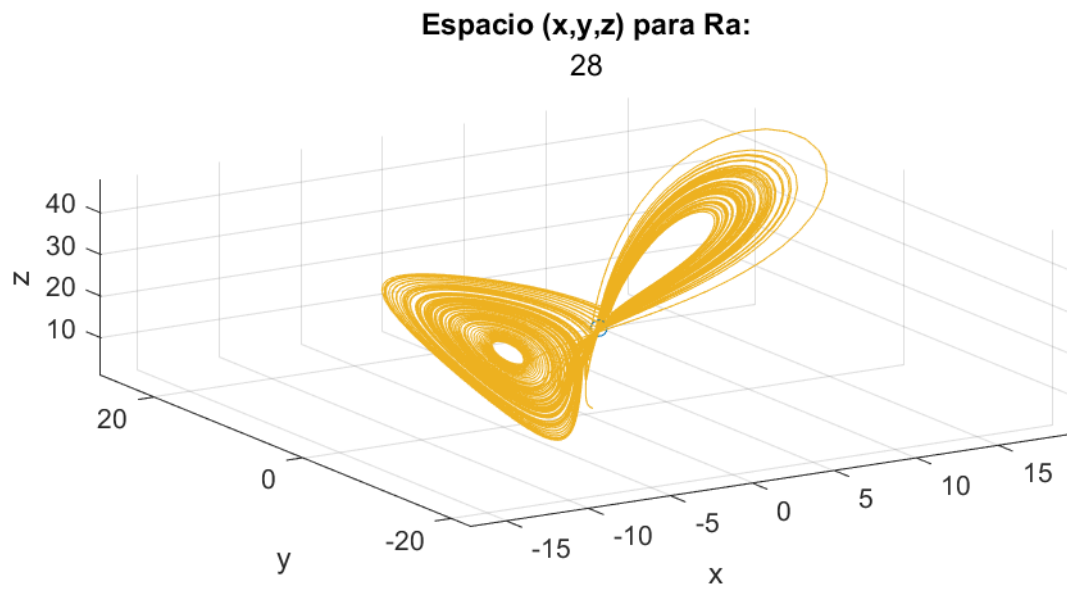
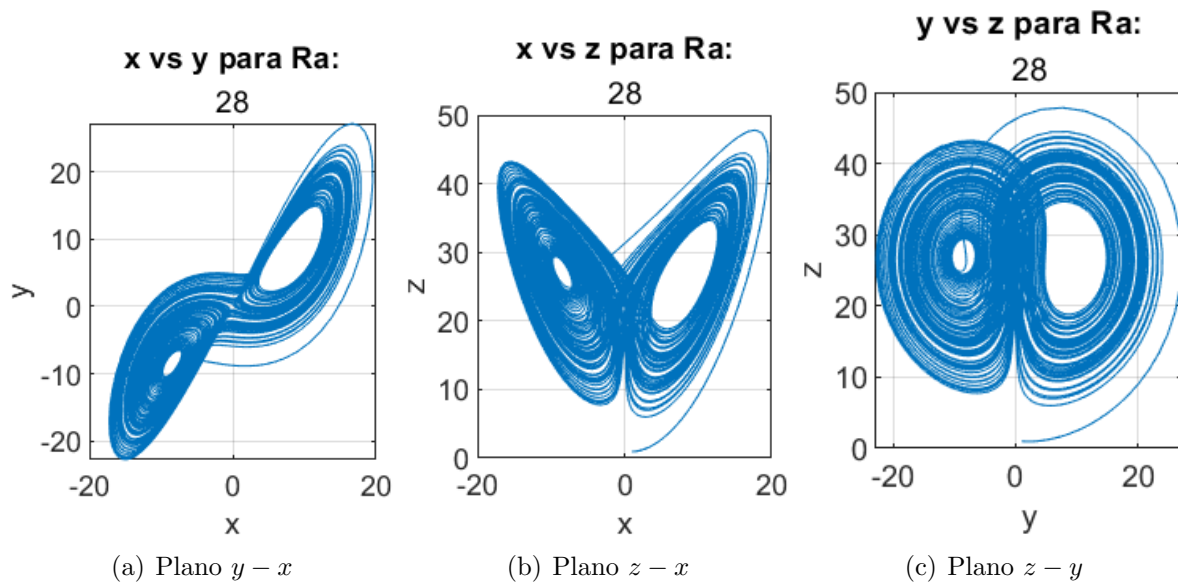


Figura 11: Espacio (x,y,z) para Ra=28

Figura 12: Variación de x, y, z en planos para Ra=28

Por otro lado se gráfica (x, t) , (y, t) y (z, t) haciendo variar lentamente Ra entre 0 y 30. En este caso se utiliza los valores $Ra=[0,2,4,6,8,10,12,14,16,18,20,22,24,26,28]$ y se divide en dos secciones de gráficos para que sea mas visible el comportamiento por dimensiones.

Se presentan los gráficos (x, t) variando Ra .

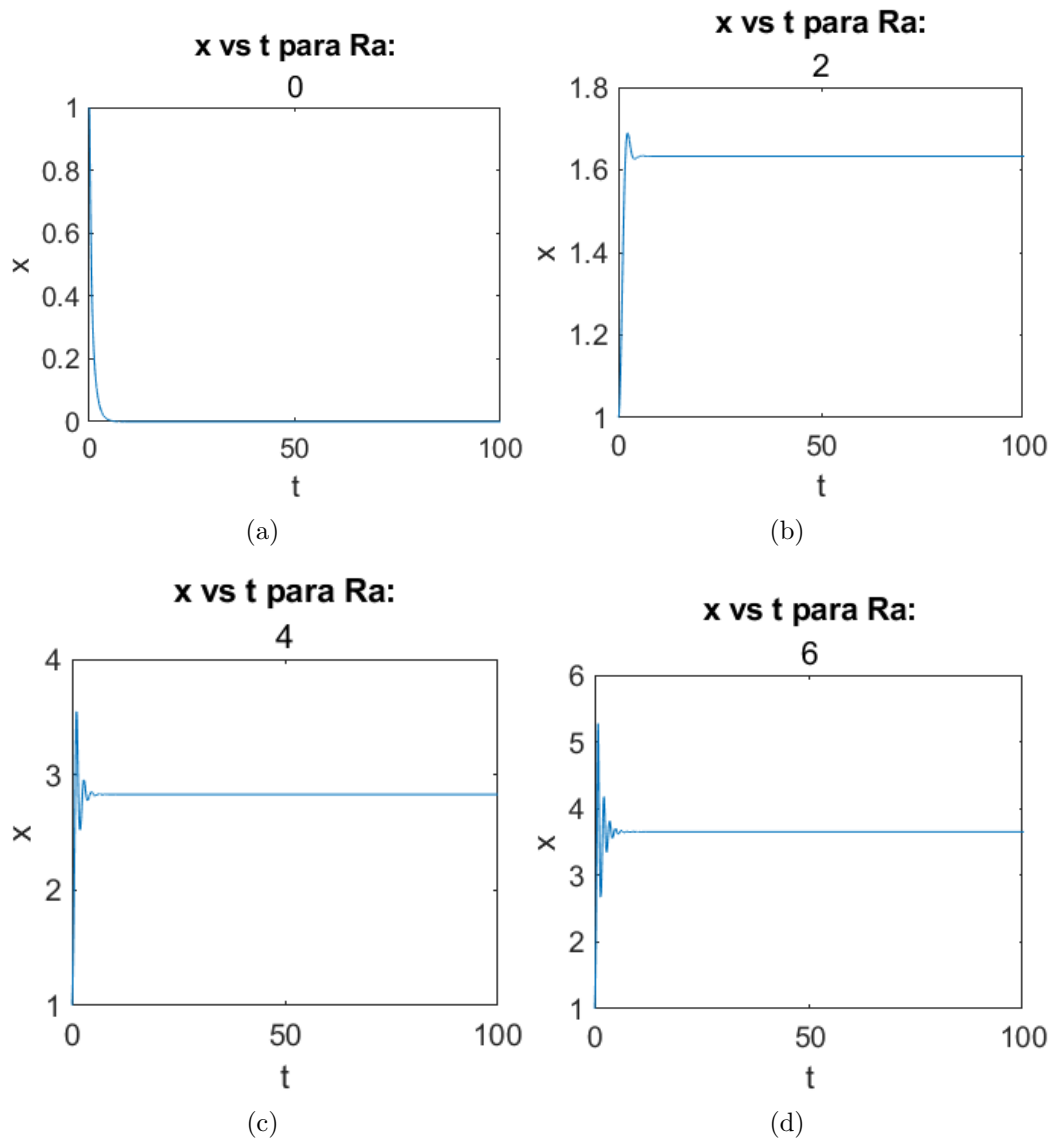
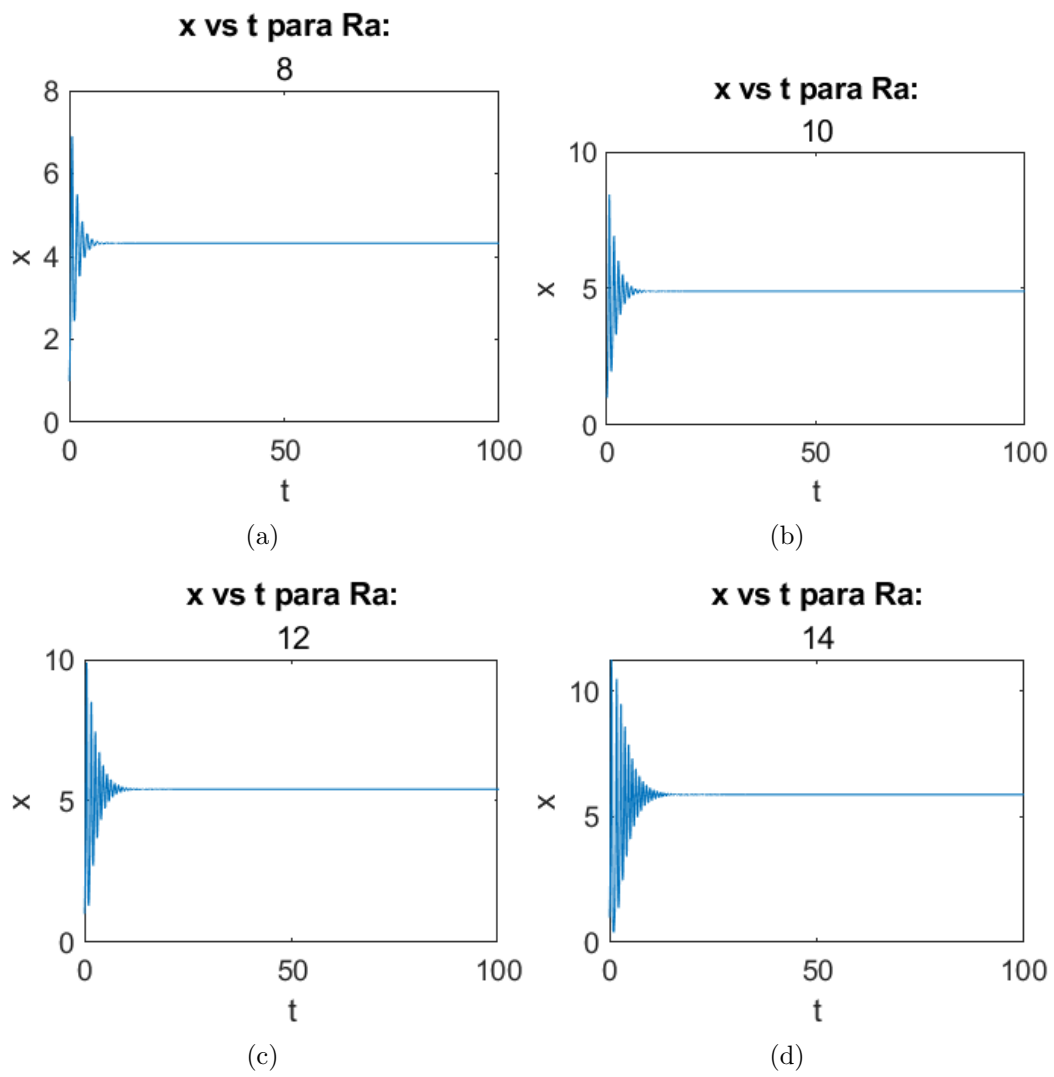
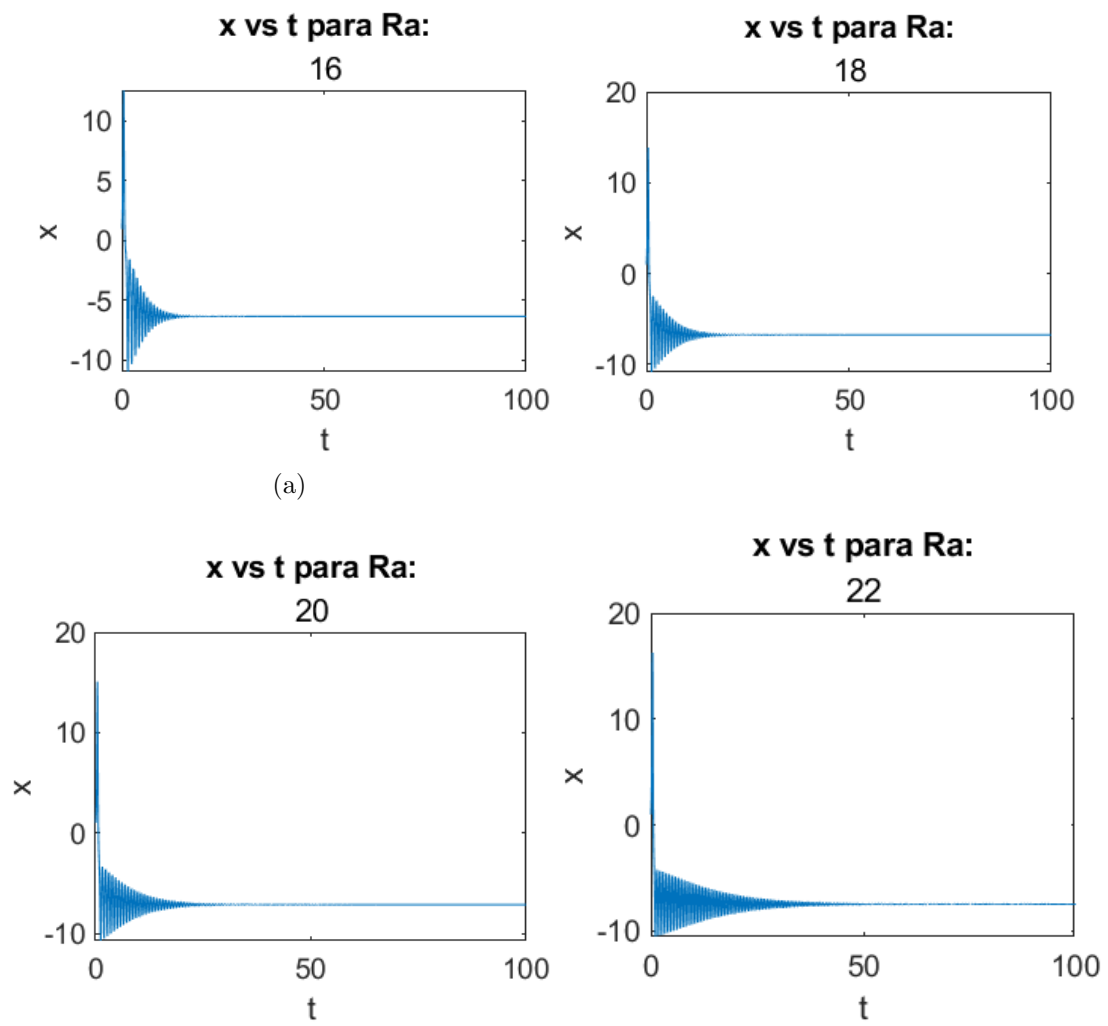
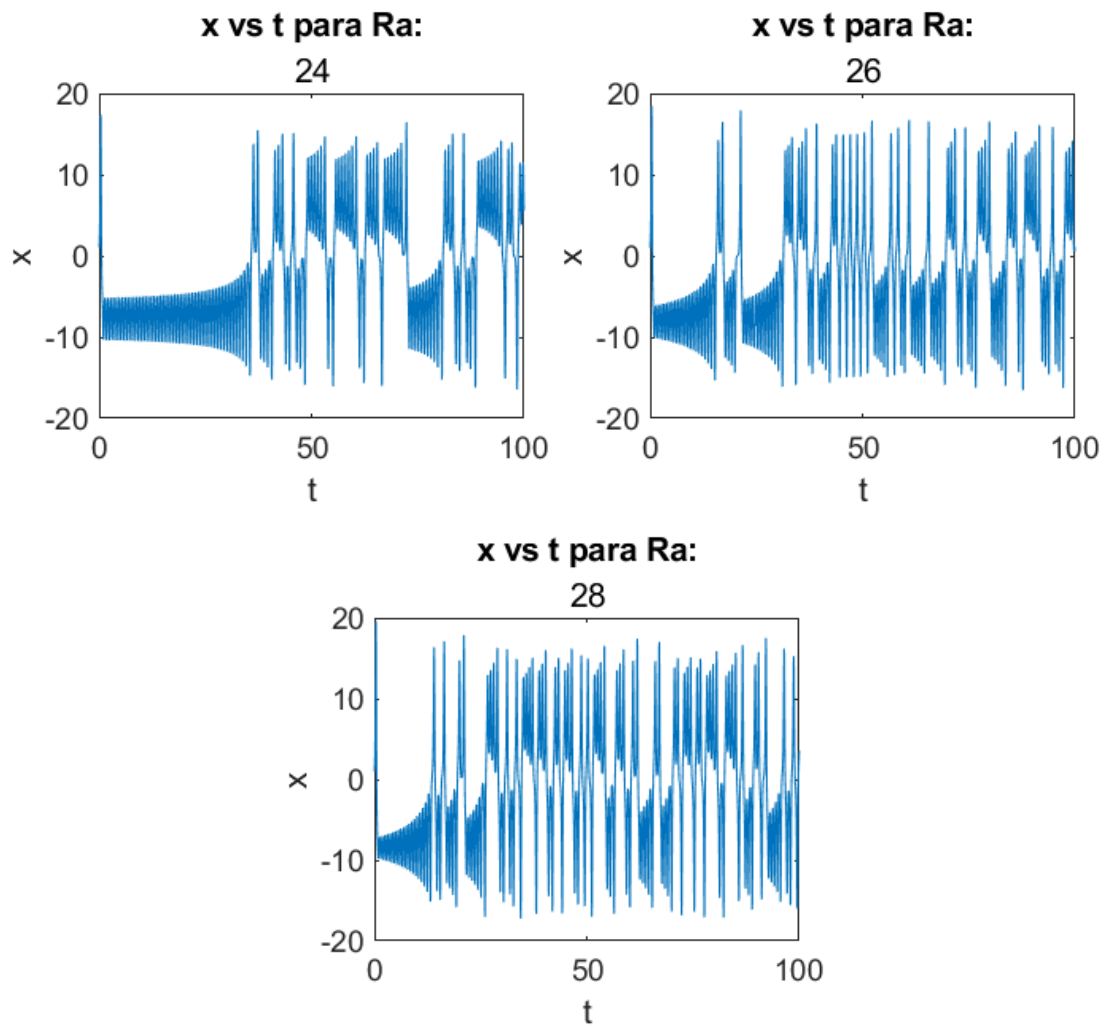


Figura 13: $x(t)$ para $Ra=0,2,4,6$

Figura 14: $x(t)$ para $Ra=8,10,12,14$

Figura 15: $x(t)$ para $Ra=16,18,20,22$

Figura 16: $x(t)$ para $Ra=24,26,28$

Se presentan los gráficos (y,t) variando Ra .

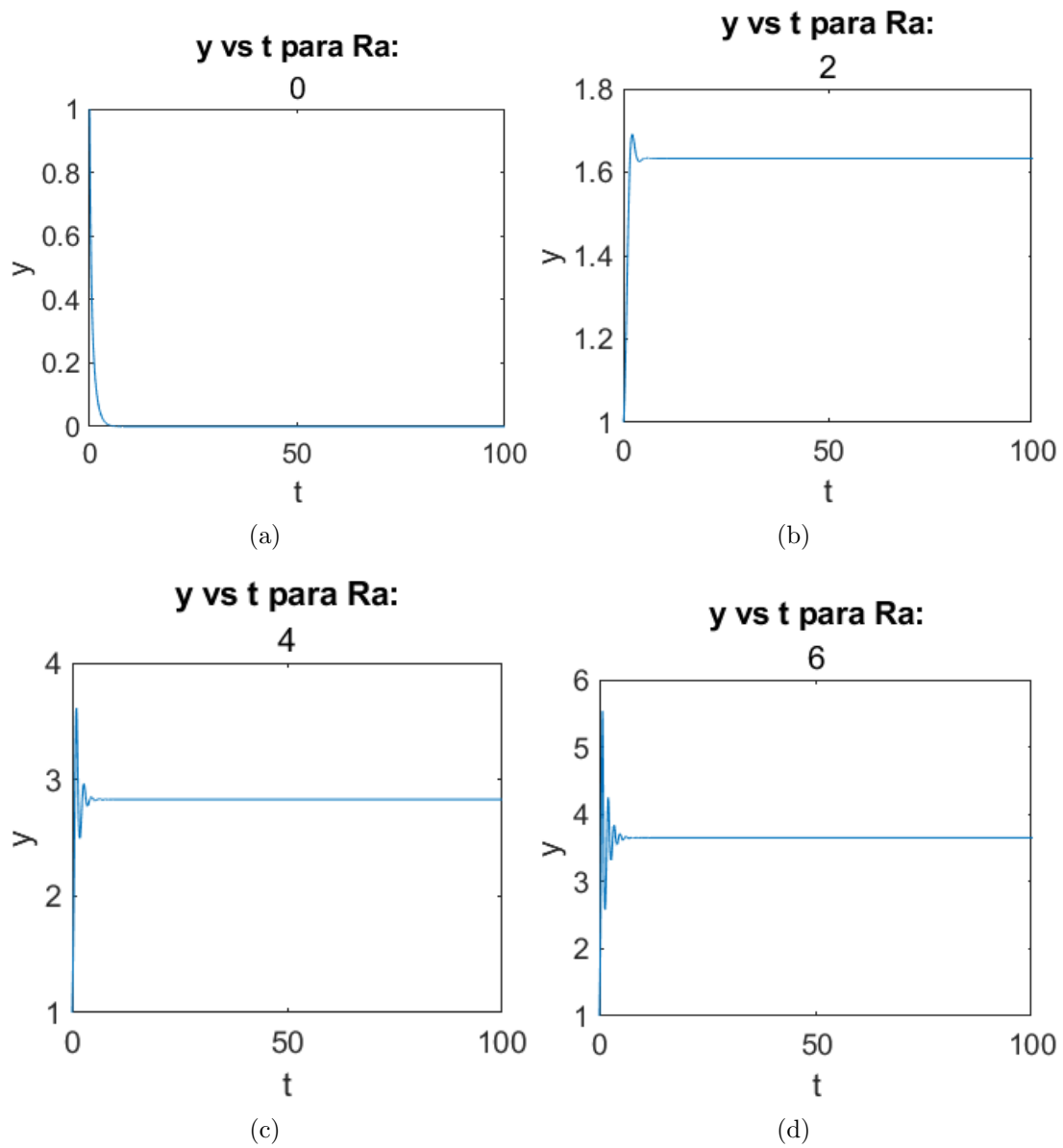
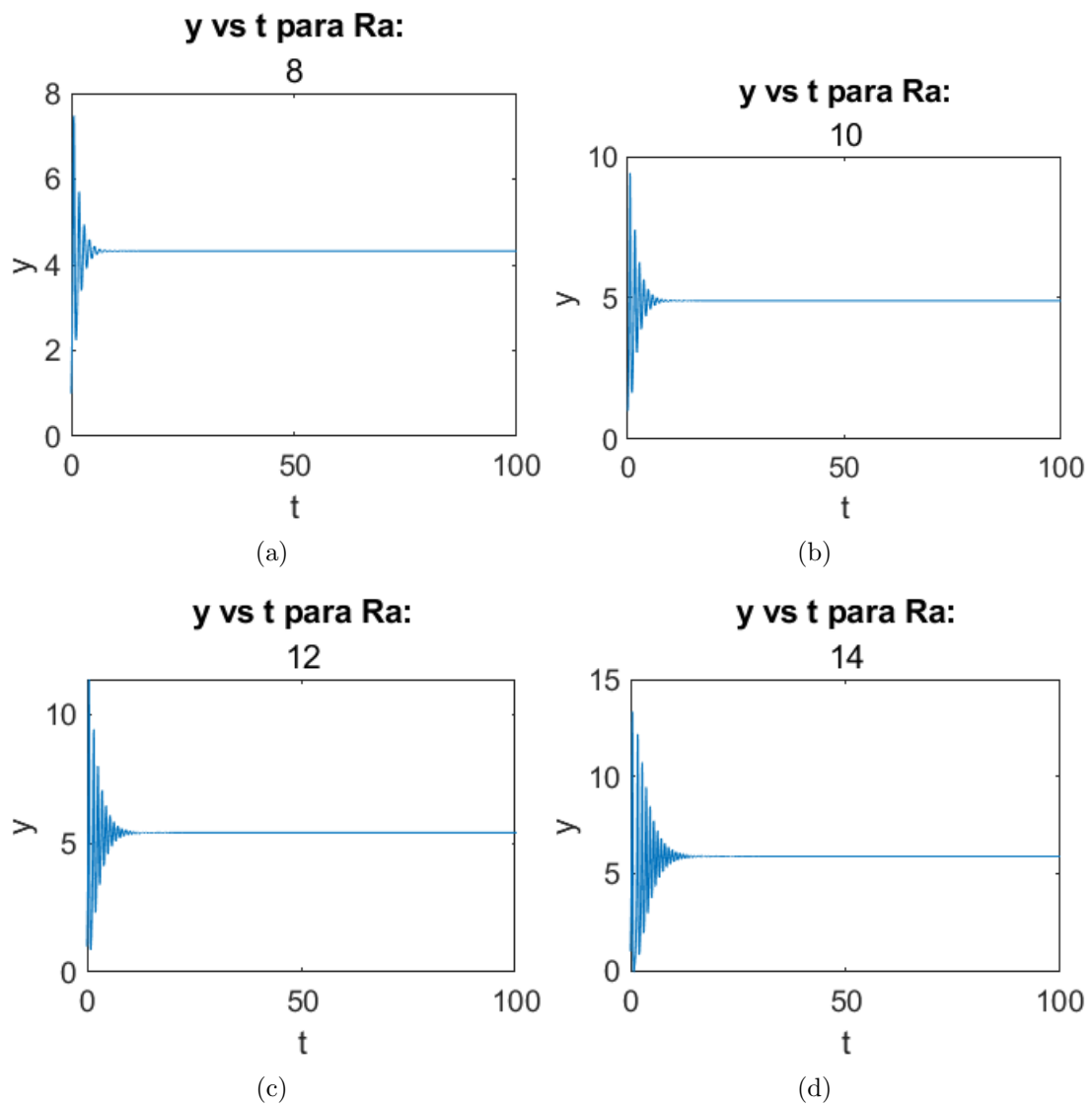
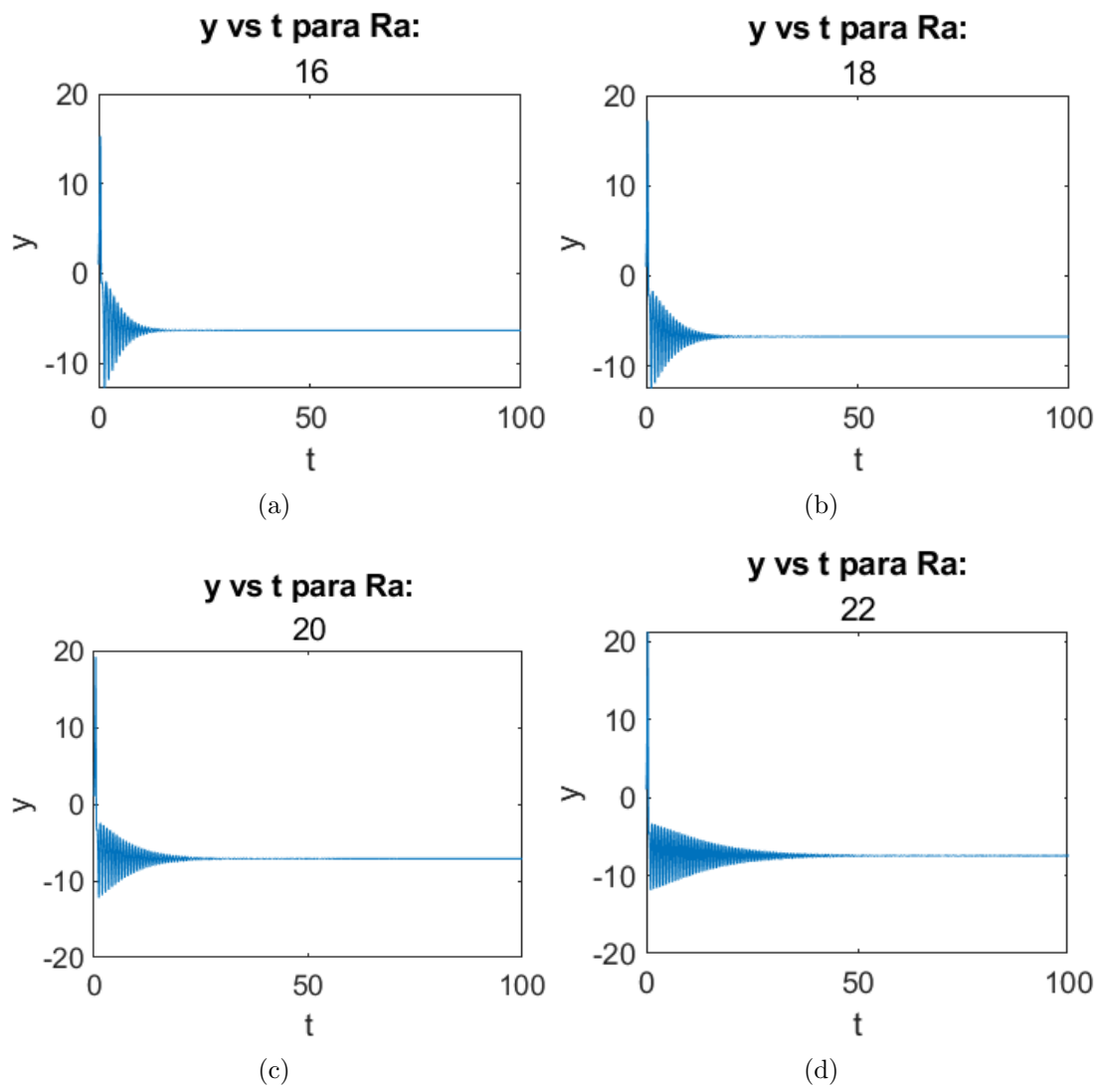
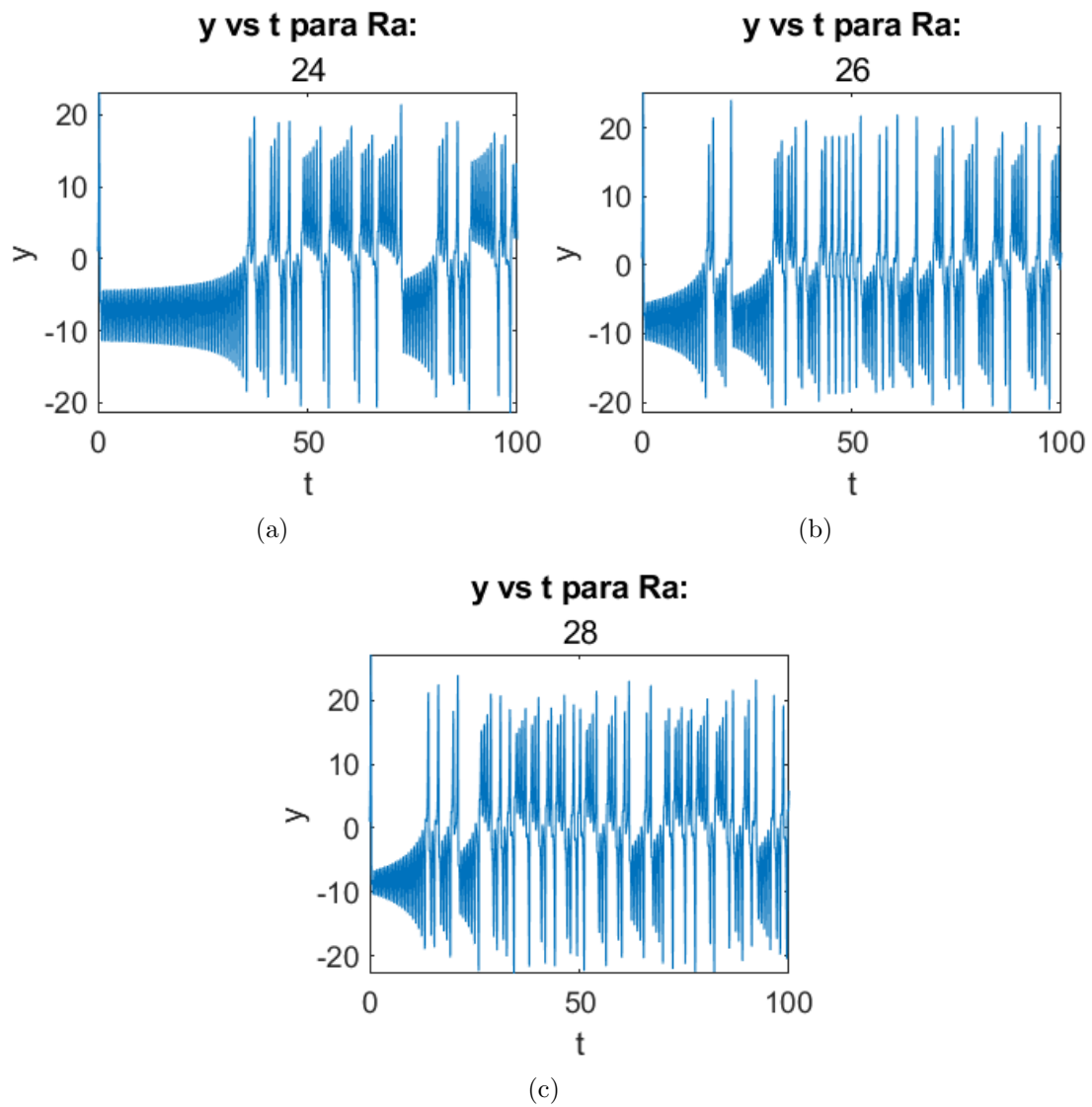


Figura 17: $y(t)$ para $Ra=0,2,4,6$

Figura 18: $y(t)$ para $Ra=8,10,12,14$

Figura 19: $y(t)$ para $Ra=16,18,20,22$

Figura 20: $y(t)$ para $Ra=24,26,28$

Se presentan los gráficos (z,t) variando Ra .

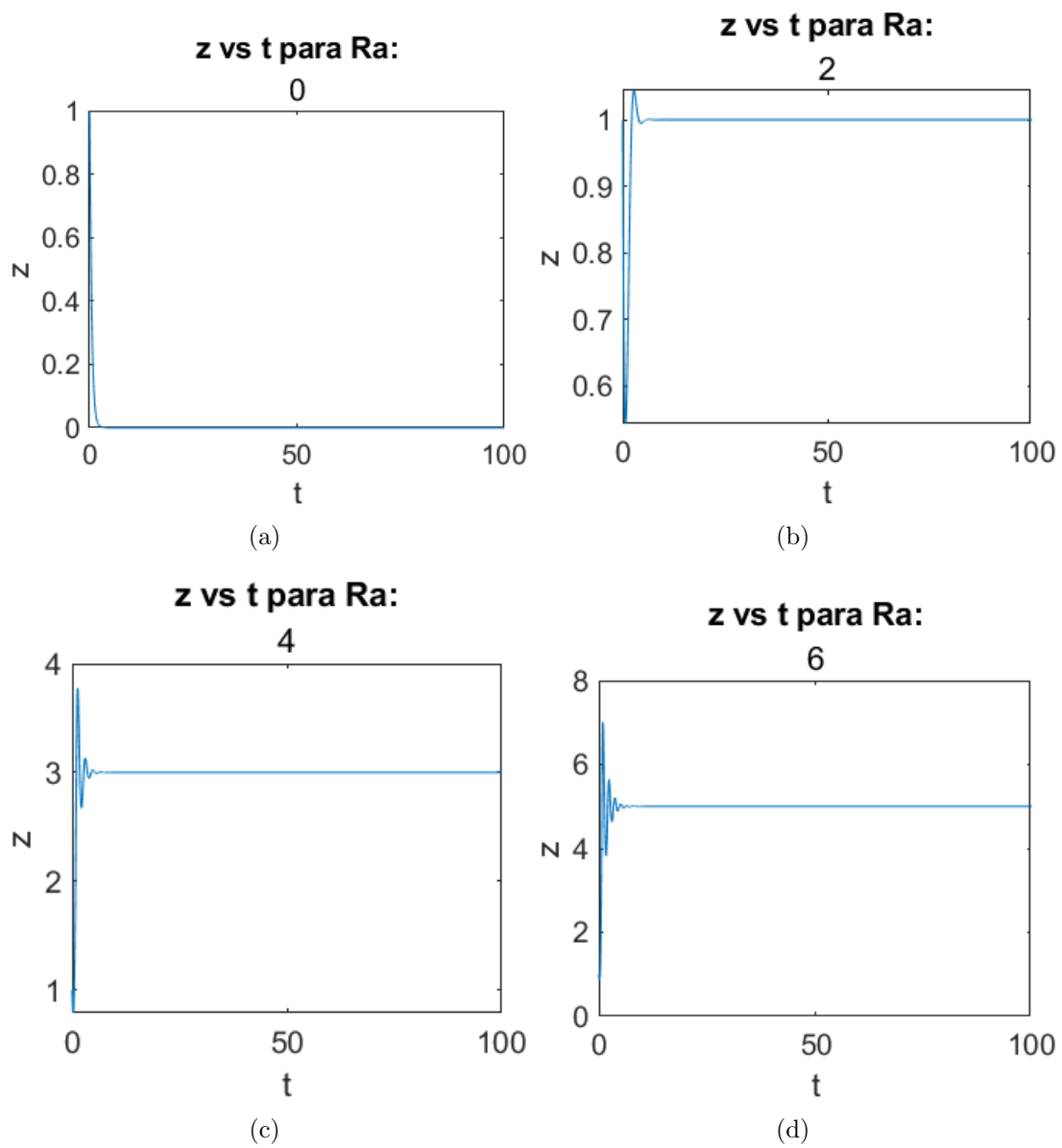
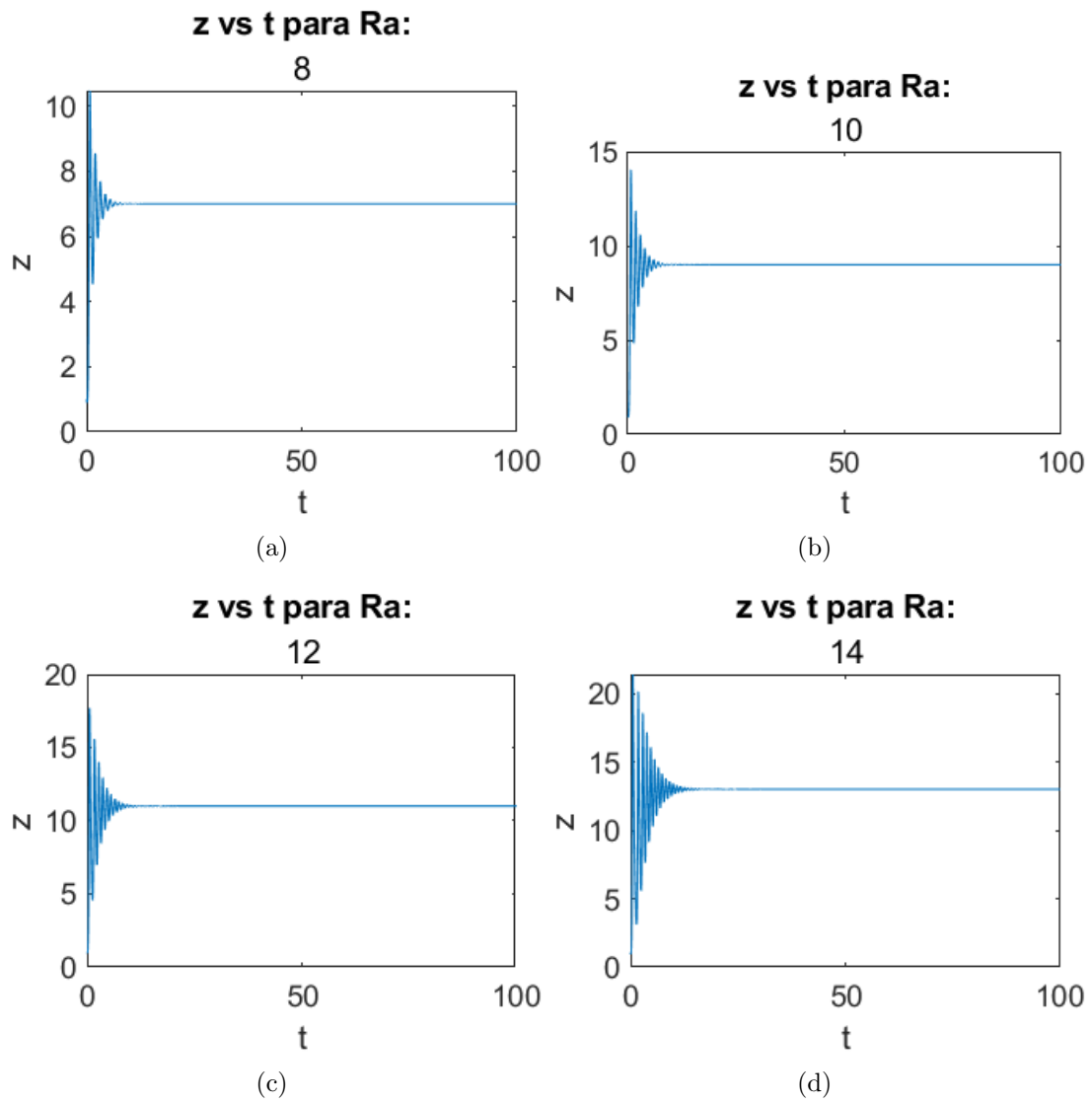
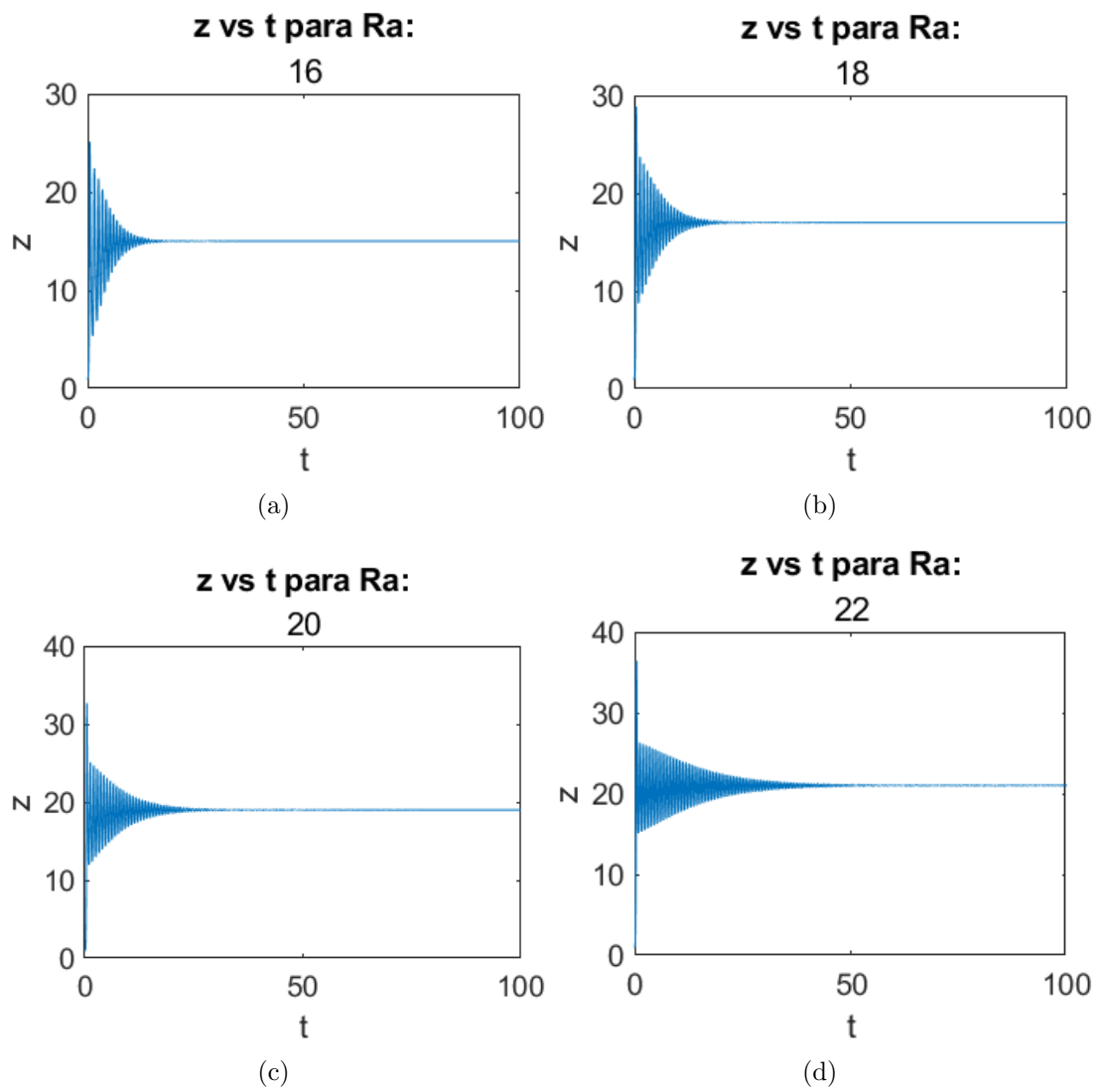
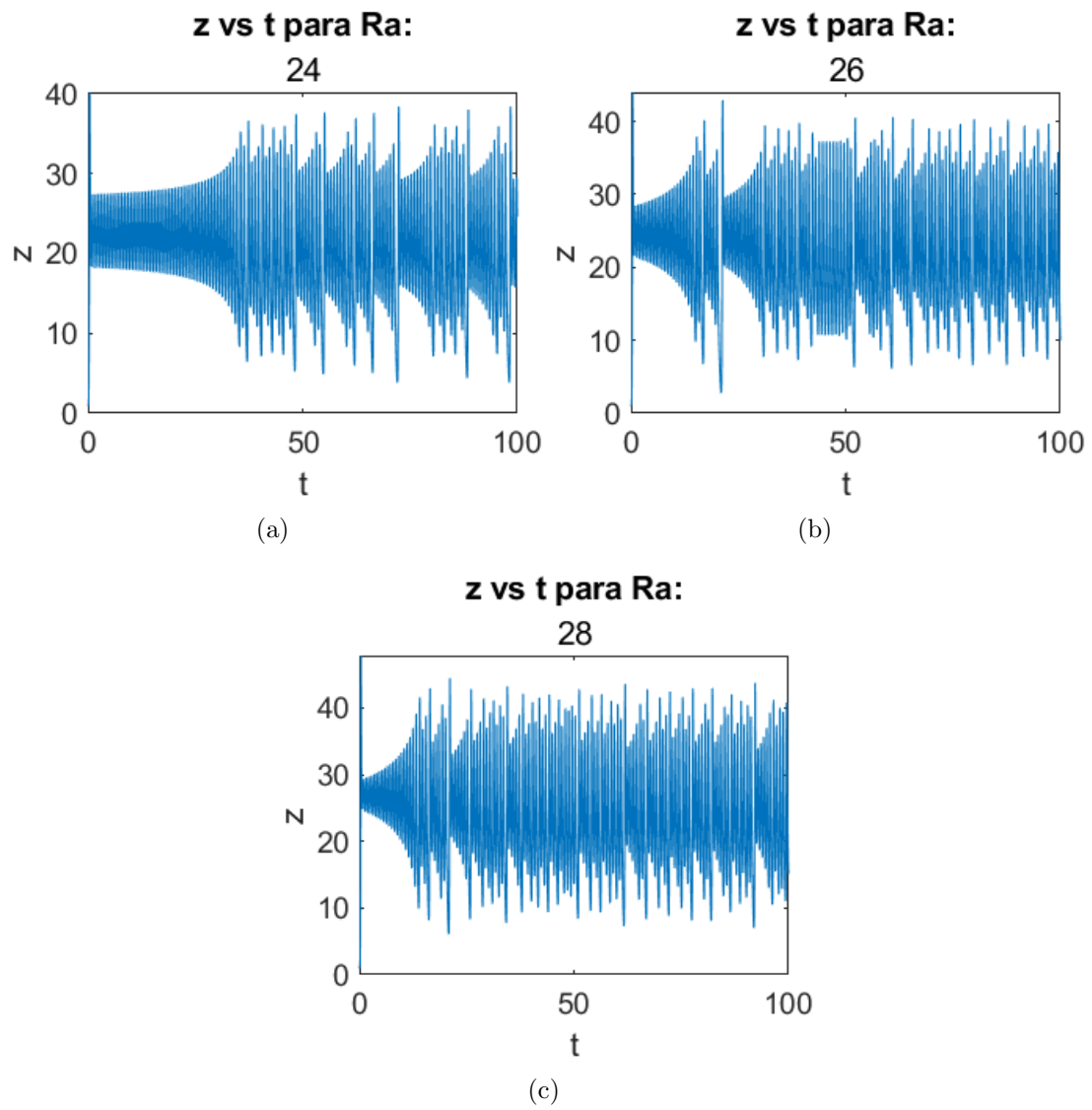


Figura 21: $z(t)$ para $Ra=0,2,4,6$

Figura 22: $z(t)$ para $Ra=8,10,12,14$

Figura 23: $z(t)$ para $Ra=16,18,20,22$

Figura 24: $z(t)$ para $Ra=24,26,28$

3.2. Problema 2: Ecuación de Calor

Se presentan los resultados para la modelación de la transferencia de calor en el muro. Se presentarán las distribuciones de temperatura para las configuraciones de aislante interior, exterior, sin aislante y condición de temperatura fija. En todas las gráficas se presentará la distribución para el caso estacionario como comparación. Por último, se presentan los gráficos de la distribución de temperatura y flujo de calor en la pared interior junto al calor total transferido en 24 horas.

3.2.1. Transferencia de Calor en Estado Transiente

Variación esquema de Discretización Aislante en Interior

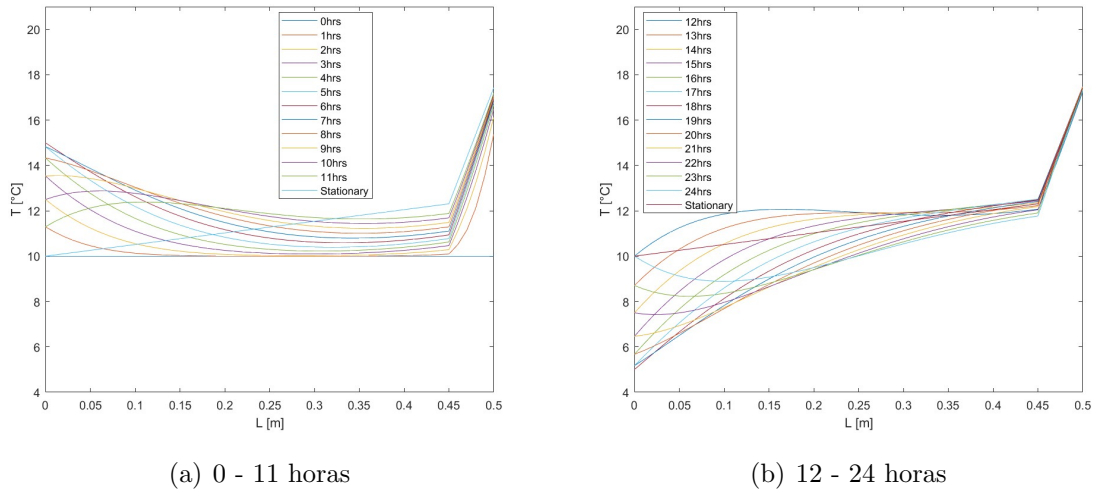
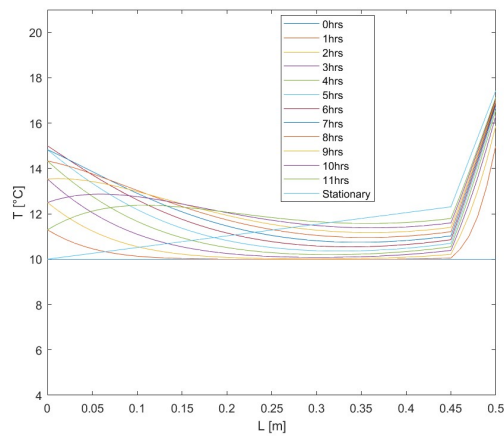
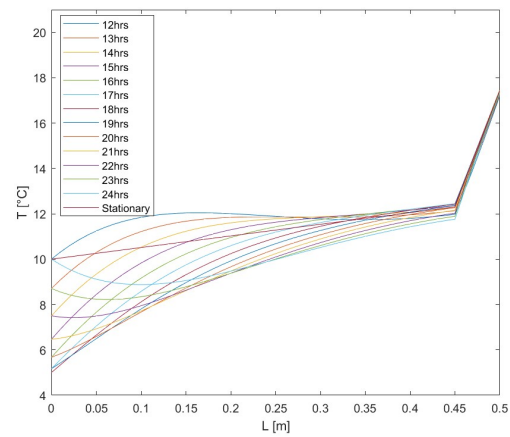


Figura 25: Distribución de temperatura para las primeras 24 horas con aislante en cara interior con Euler Explícito con $\Delta x = 0.005$ y $\Delta t = 1$

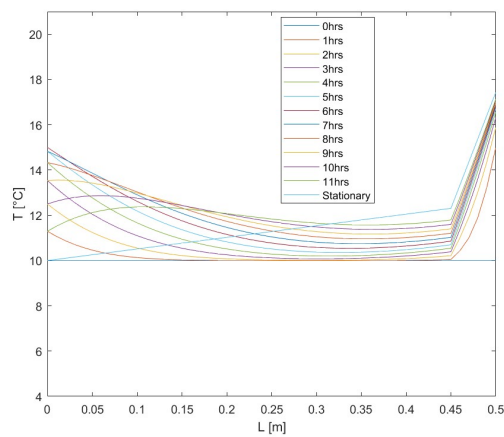


(a) 0 - 11 horas

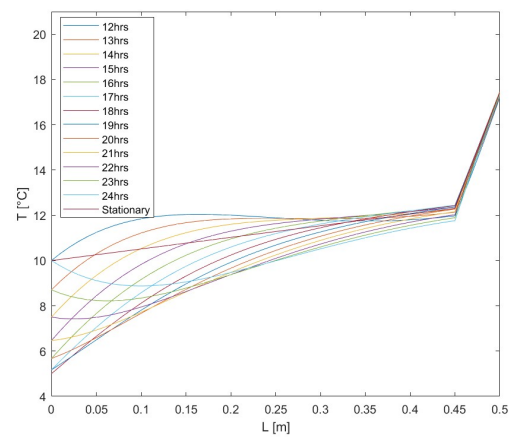


(b) 12 - 24 horas

Figura 26: Distribución de temperatura para las primeras 24 horas con aislante en cara interior con Euler Implícito $\Delta x = 0.01$ y $\Delta t = 5$



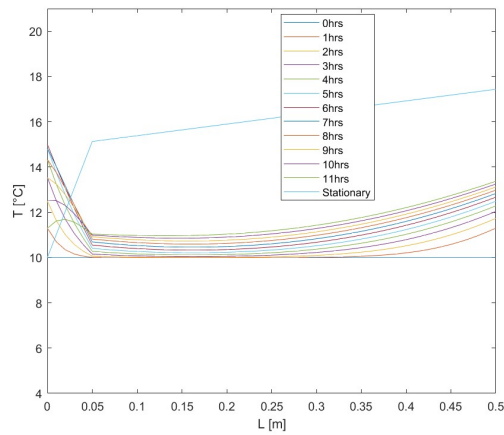
(a) 0 - 11 horas



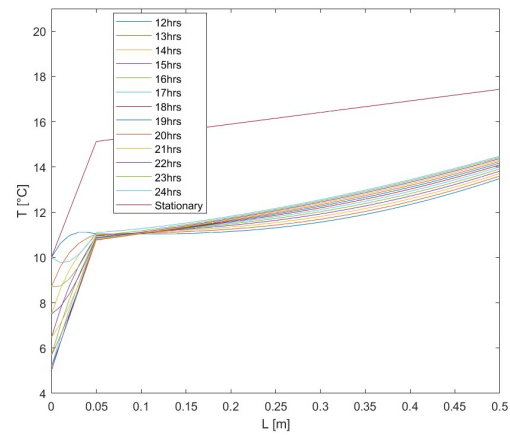
(b) 12 - 24 horas

Figura 27: Distribución de temperatura para las primeras 24 horas con aislante en cara interior con Crank Nicolson $\Delta x = 0.01$ y $\Delta t = 10$

Variación esquema de Discretización Aislante en Exterior

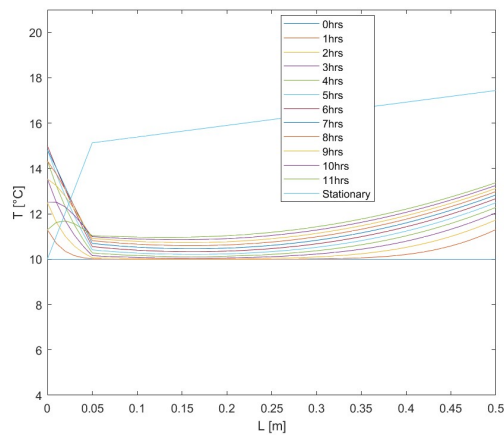


(a) 0 - 11 horas

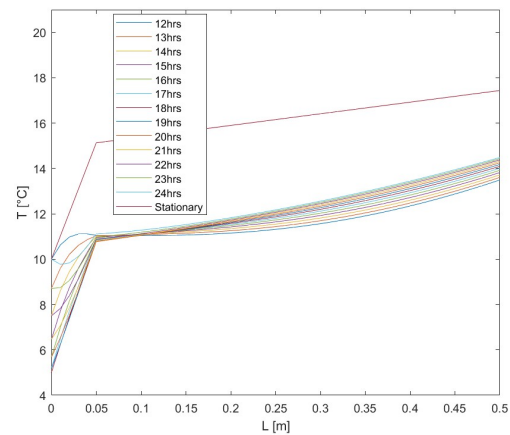


(b) 12 - 24 horas

Figura 28: Distribución de temperatura para las primeras 24 horas con aislante en cara exterior con Euler Explícito con $\Delta x = 0.005$ y $\Delta t = 1$

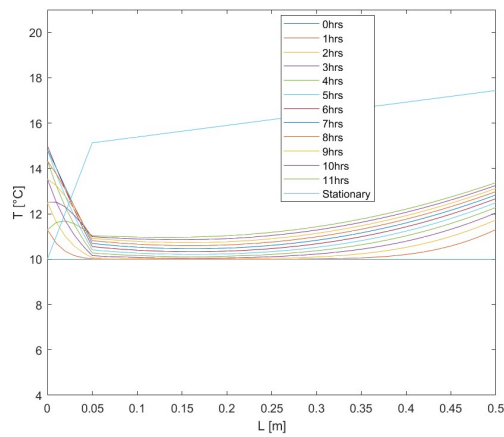


(a) 0 - 11 horas

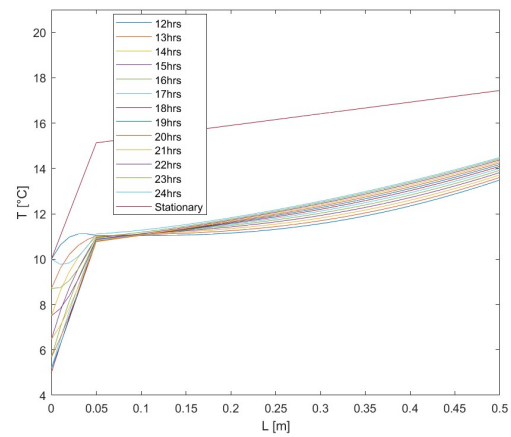


(b) 12 - 24 horas

Figura 29: Distribución de temperatura para las primeras 24 horas con aislante en cara exterior con Euler Implícito $\Delta x = 0.01$ y $\Delta t = 5$



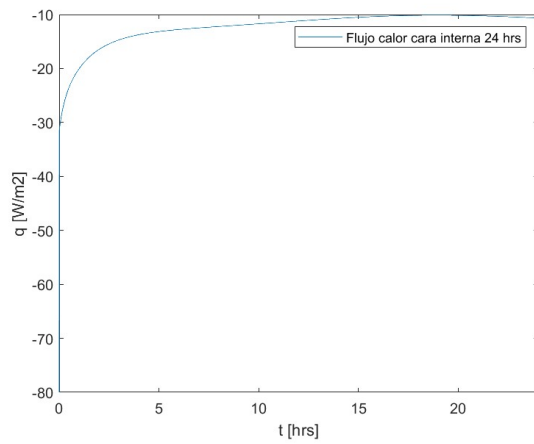
(a) 0 - 11 horas



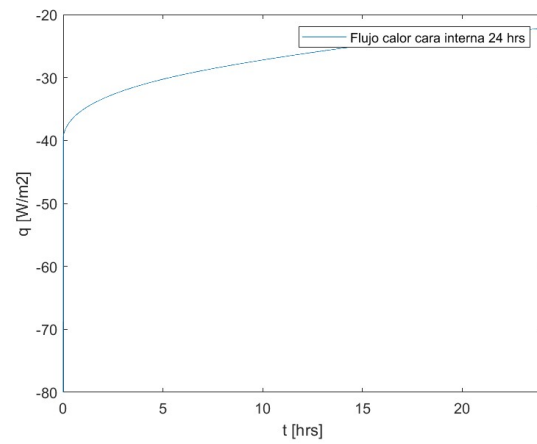
(b) 12 - 24 horas

Figura 30: Distribución de temperatura para las primeras 24 horas con aislante en cara exterior con Crank Nicolson $\Delta x = 0.01$ y $\Delta t = 10$

Flujo de Calor y Calor Total en Cara Interna



(a) Aislante interno



(b) Aislante externo

Figura 31: Flujo de Calor en la cara interna en las primeras 24 horas para ambas configuraciones

Tabla 2: Tabla para calor total transferido en las primeras 24 horas con aislante interior

Esquemas	Euler Explícito	Euler Implícito	Crank Nicolson
q_N [KJ/m ²]	1034	1035	1035

Tabla 3: Tabla para calor total transferido en las primeras 24 horas con aislante exterior

Esquemas	Euler Explícito	Euler Implícito	Crank Nicolson
q_N [KJ/m ²]	2321	2322	2322

3.2.2. Transferencia de Calor en Estado Estacionario

Se modelan para las 24 horas próximas a 10 días.

Variación esquema de Discretización Aislante en Interior

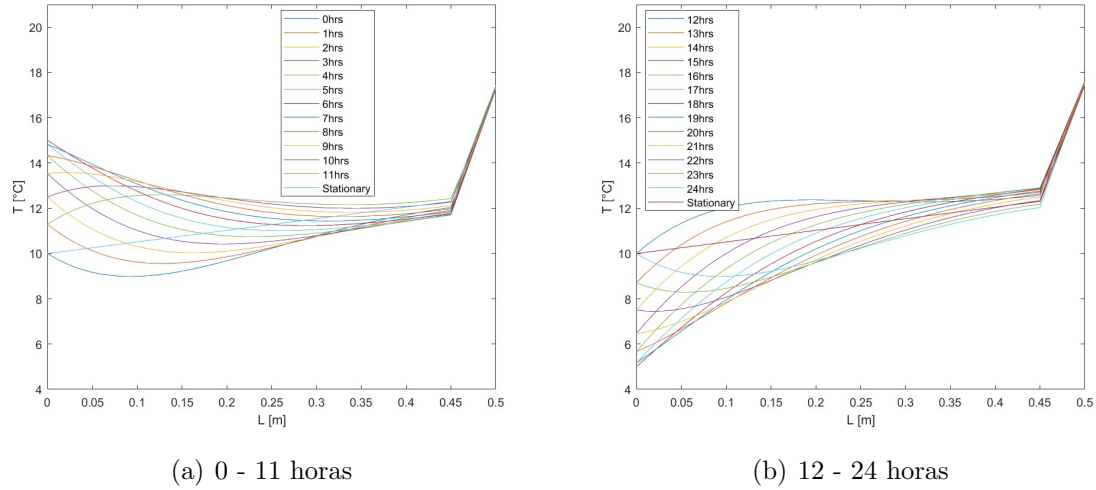


Figura 32: Distribución de temperatura para las próximas 24 horas pasados 10 días con aislante en cara interior con Euler Explícito con $\Delta x = 0.005$ y $\Delta t = 1$

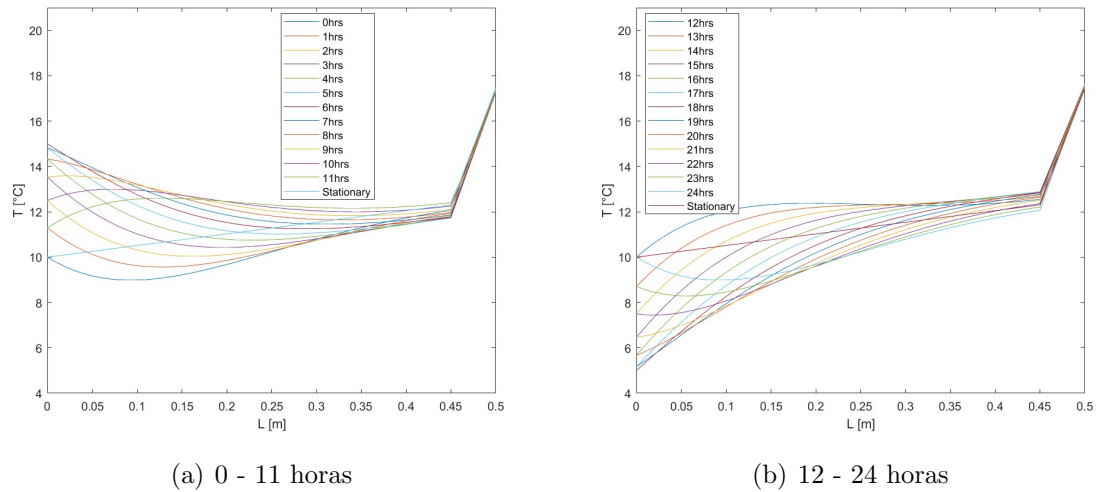
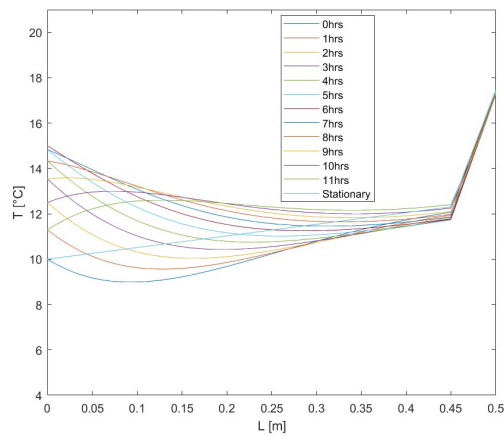
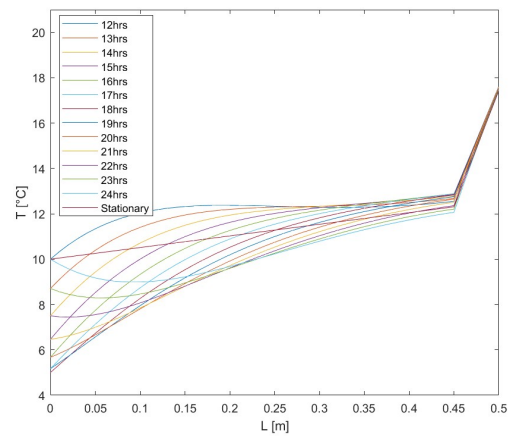


Figura 33: Distribución de temperatura para las próximas 24 horas pasados 10 días con aislante en cara interior con Euler Implícito $\Delta x = 0.01$ y $\Delta t = 5$



(a) 0 - 11 horas



(b) 12 - 24 horas

Figura 34: Distribución de temperatura para las próximas 24 horas pasados 10 días con aislante en cara interior con Crank Nicolson $\Delta x = 0.01$ y $\Delta t = 10$

Variación esquema de Discretización Aislante en Exterior

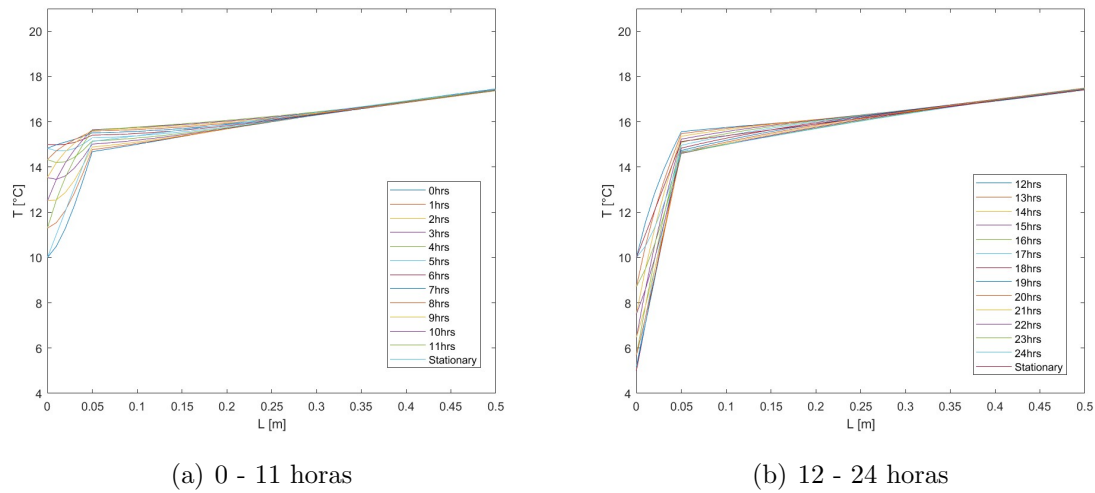


Figura 35: Distribución de temperatura para las próximas 24 horas a 10 días con aislante en cara exterior con Euler Explícito con $\Delta x = 0.005$ y $\Delta t = 1$

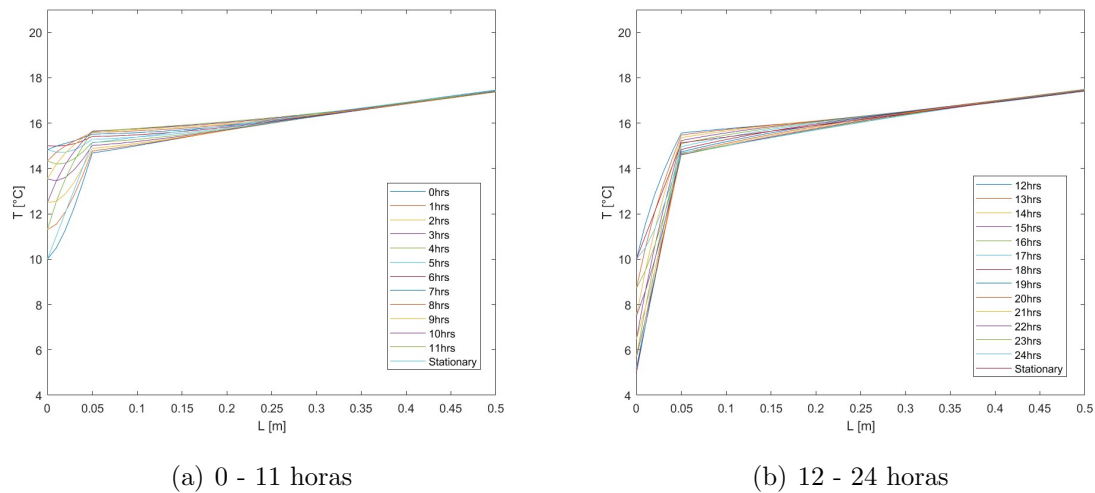
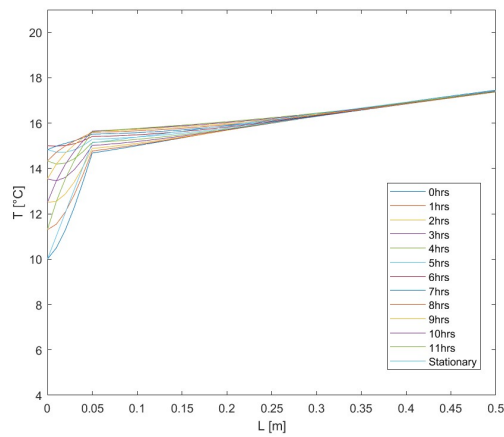
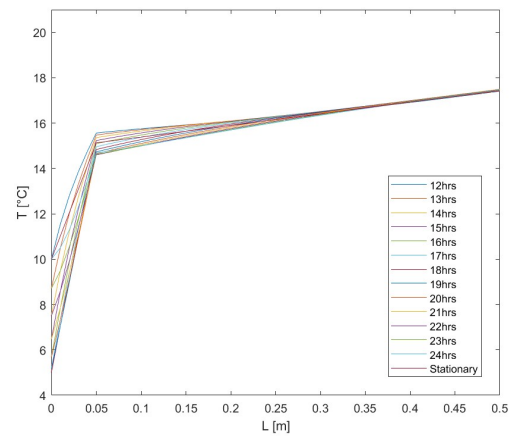


Figura 36: Distribución de temperatura para las próximas 24 horas a 10 días con aislante en cara exterior con Euler Implícito $\Delta x = 0.01$ y $\Delta t = 5$



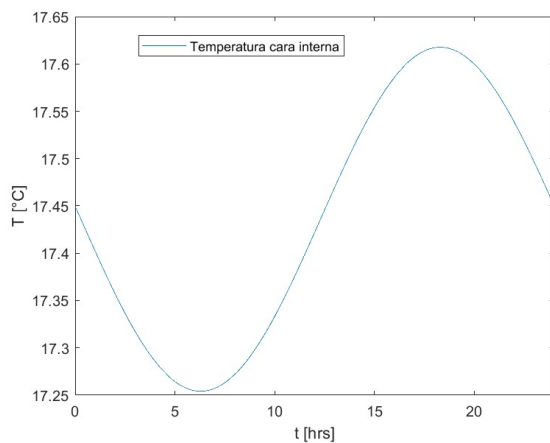
(a) 0 - 11 horas



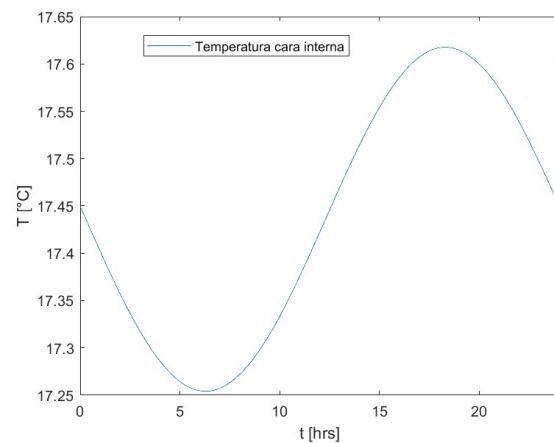
(b) 12 - 24 horas

Figura 37: Distribución de temperatura para las próximas 24 horas a 10 días con aislante en cara exterior con Crank Nicolson $\Delta x = 0.01$ y $\Delta t = 10$

Temperatura en cara externa en 24 horas a 10 días en estado estacionario



(a) Aislante interno



(b) Aislante externo

Figura 38: Temperatura en la cara interna en las próximas 24 horas a 10 días para ambas configuraciones

Flujo de Calor y Calor Total en Cara Interna

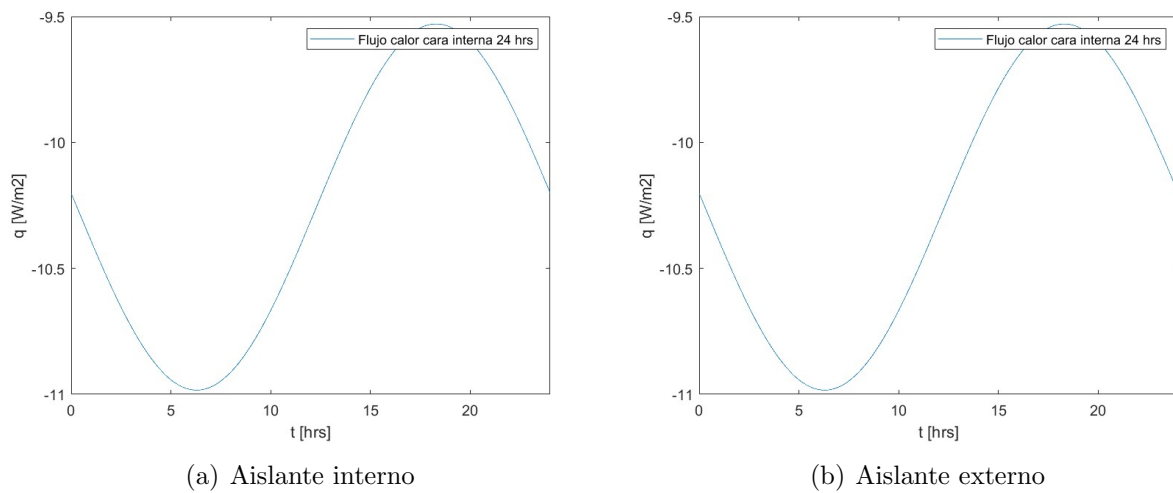


Figura 39: Flujo de Calor en la cara interna en las próximas 24 horas a 10 días para ambas configuraciones

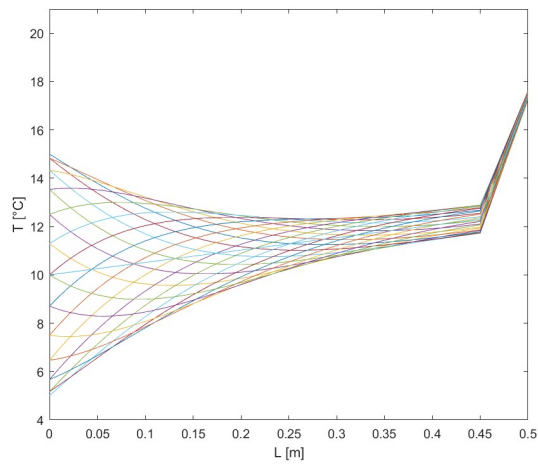
Tabla 4: Tabla para calor total transferido en las próximas 24 horas a 10 días con aislante interior

Esquemas	Euler Explícito	Euler Implícito	Crank Nicolson	Teórico
q_N [KJ/m²]	886.16	886.17	886.117	886.17

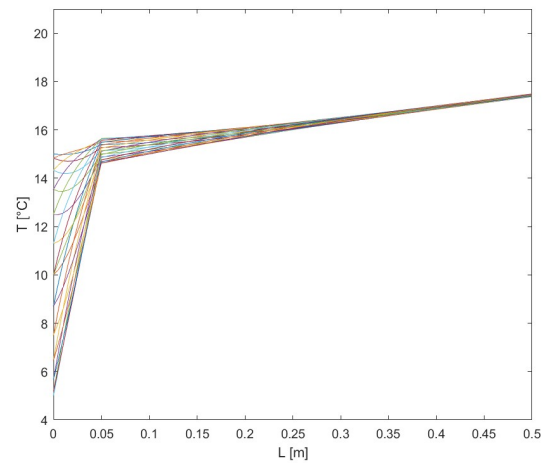
Tabla 5: Tabla para calor total transferido en las próximas 24 horas a 10 días con aislante exterior

Esquemas	Euler Explícito	Euler Implícito	Crank Nicolson	Teórico
q_N [KJ/m²]	887.22	886.17	886.17	886.17

Variación de Temperatura en 1 día completo



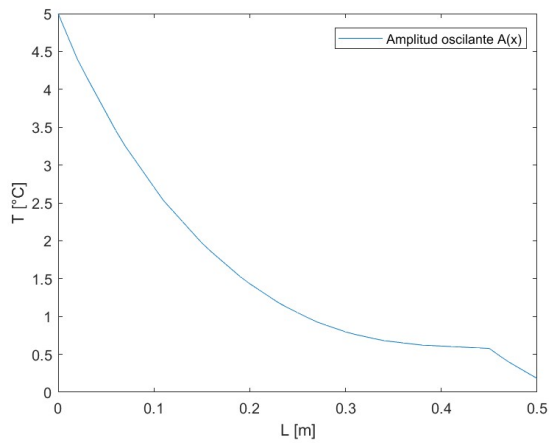
(a) Aislante interno



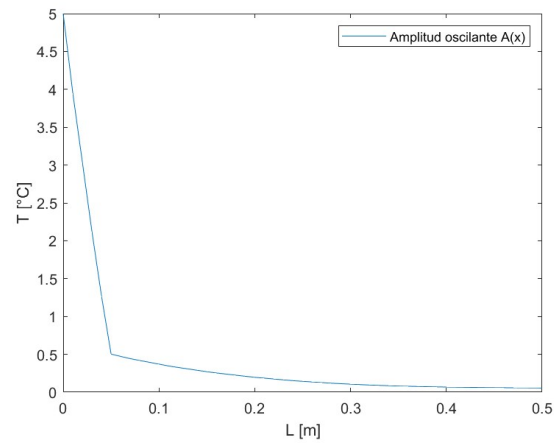
(b) Aislante externo

Figura 40: Variación de Temperatura en 1 día completo pasado 10 días

Amplitud de oscilación para estado estacionario



(a) Aislante interno



(b) Aislante externo

Figura 41: Amplitud de oscilación $A(x)$ para la temperatura

3.2.3. Condición temperatura fija

Variación al utilizar condición de Temperatura fija en la pared interior

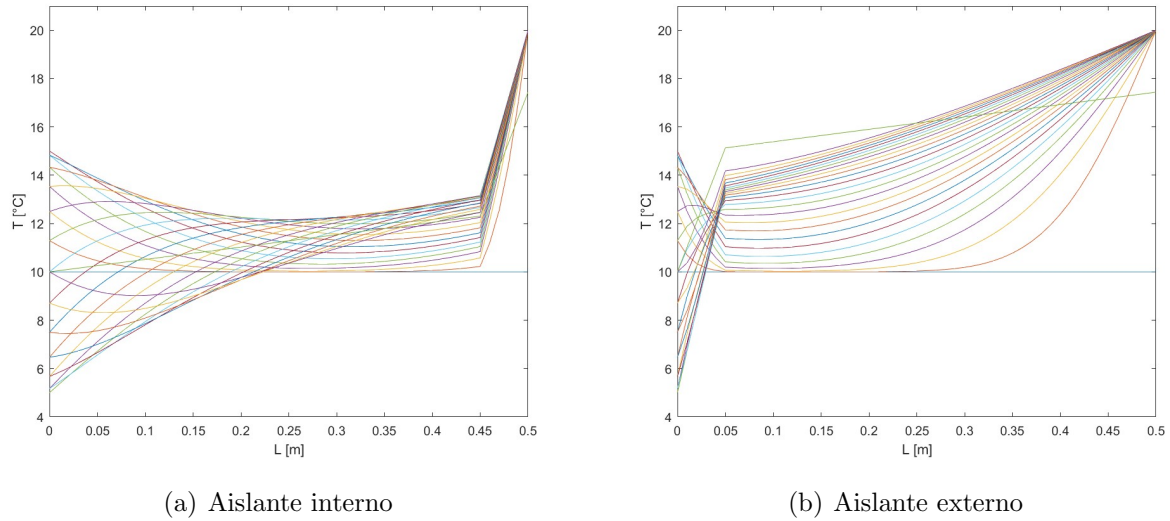


Figura 42: Variación de Temperatura en las primeras 24 horas (estado transiente)

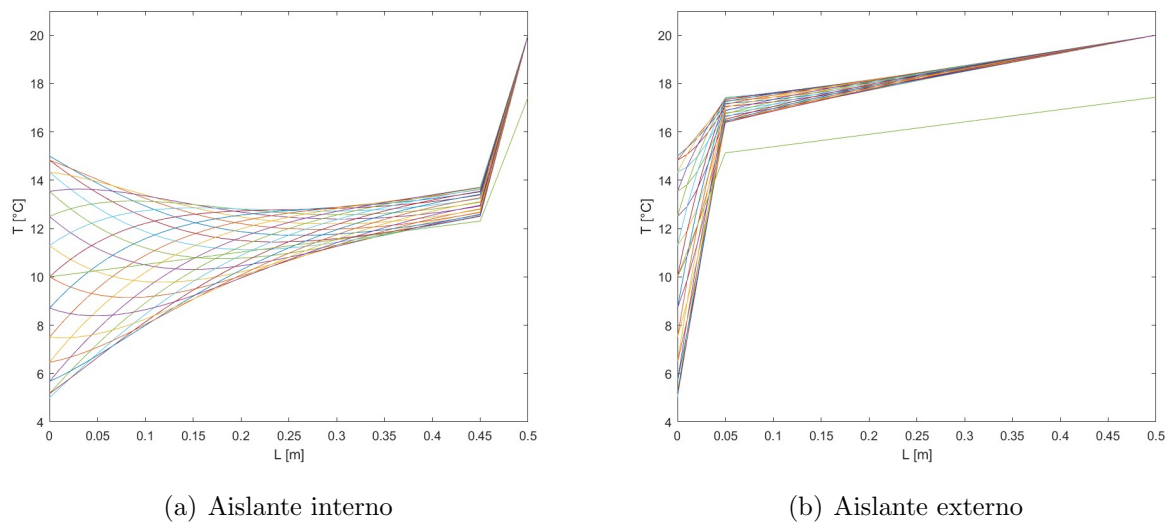


Figura 43: Variación de Temperatura en 24 horas (estado estacionario)

Tabla 6: Tabla para calor total transferido en las próximas 24 horas a 10 días para temperatura fija

Calor total	Asilante adentro	Aislante afuera
q_N [KJ/m ²]	1192	1192

3.2.4. Caso Sin Aislante

Variaciones para caso sin aislante

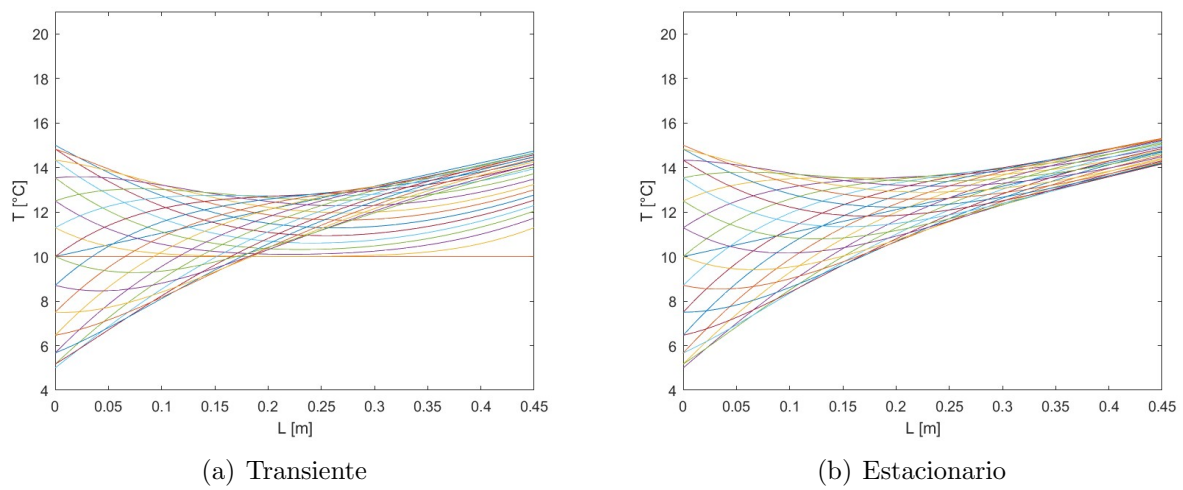


Figura 44: Variación de Temperatura en 24 horas (estado transiente y estacionario)

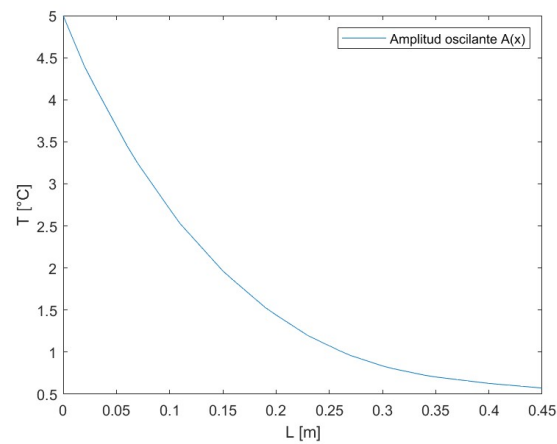
Figura 45: Amplitud $A(x)$ para caso sin aislante

Tabla 7: Tabla para calor total transferido en las próximas 24 en estado transiente y estacionario

Calor total	Transiente	Estacionario
q_N [KJ/m2]	2273	1818

3.2.5. Tiempos de Cálculo

Tiempo de Cálculo para distintos pasos de tiempo con $\Delta x = 0.01$

Tabla 8: Tiempos de cálculo en segundos para modelar 15 días con distintos pasos de tiempo

Método	$\Delta t = 15$	$\Delta t = 60$	$\Delta t = 120$	$\Delta t = 1200$	$\Delta t = 3600$
Euler Explícito	1.5	-	-	-	-
Euler Implícito	42	9	4.9	1.1	0.73
Crank Nicolson	43	8	5.2	1.1	0.73

4. Análisis

4.1. Problema 1: EDOs Sistema de Lorenz

El sistema de Lorenz permite modelar los roles de convección de la atmósfera terrestre, es decir, modelamos un sistema dinámico caótico, ya que este varía durante el tiempo y es sumamente sensible a los cambios de ciertas variables, produciendo que el sistema se comporte de forma impredecible durante el tiempo. Por lo mismo la visualización de este problema permite comprender ciertos factores que influyen en este comportamiento.

Se asume el número de Prandtl: $Pr=10$, el tamaño físico del sistema $\beta = 8/3$, un valor inicial de $(1,1,1)$ el intervalo de tiempo de 0 a 100 segundos, un número de iteraciones de $n = 10000$ y por lo tanto un paso de tiempo de $h = 0.01$. Ahora bien, el número de Rayleigh se varía para observar como se comporta el sistema dinámico.

Este sistema fue programado en MATLAB y los resultados obtenidos serán comparados por los resultados expuestos por Edward Lorenz en la revista Journal of the Atmospheric Sciences.

Como primera observación se tiene que una de las condiciones matemáticas para determinar un Ra crítico es mediante la ecuación 7, es decir que para estas condiciones asumidas tenemos un $Ra_{critico} = 24.74$, que corresponde al mismo valor propuesto por Edward Lorenz.

Si se analiza las gráficas podemos observar que su comportamiento va variando a medida que aumenta su Ra , pues llega a un punto de comportarse de forma caótica. Por lo mismo se toma un rango de análisis de Ra entre $1 < Ra < 30$.

Se puede observar que cuando Ra se encuentra entre $1 < Ra < Ra_{critico}$ se observa que la mayoría de las trayectorias comienzan cerca del origen (el cual es inestable), pero se mueven alejándose de él. Se puede decir que el origen genera dos tipos de atractores que corresponden a los puntos de equilibrio P_2 y P_3 , que fueron denominados por Lorenz como C y C' , los cuales son puntos asintóticamente estables y la trayectoria gira en forma de una espiral hacia adentro en sentido horario o antihorario. Por lo mismo este rango se le denomina régimen precaótico.

Ahora bien, cuando se llega al valor crítico, es decir, $Ra = Ra_{critico}$ se observa que se produce una bifurcación del tipo Hopf. Se puede ver que la trayectoria girará entorno a los puntos P_2 y P_3 , la cual tienen forma de órbitas periódicas, con un recorrido infinitamente largo comparado al rango anterior.

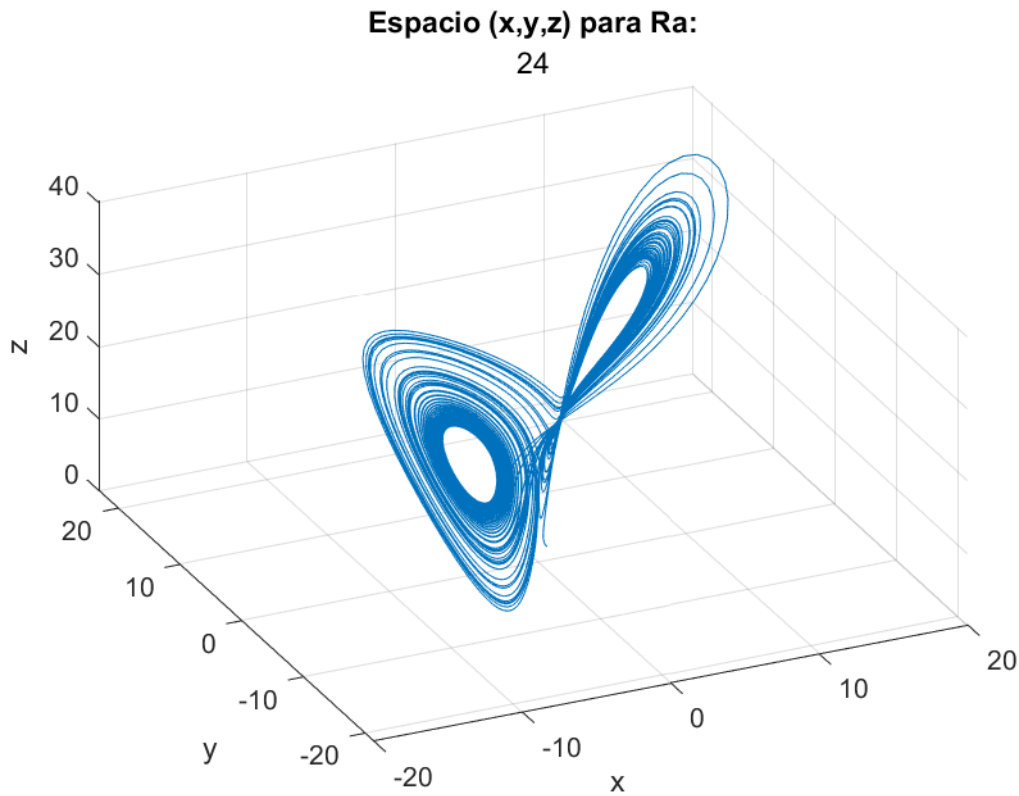


Figura 46: Espacio (x,y,z) para Ra=24

Cuando $Ra > R_{critico}$ se tiene que los puntos P_2 y P_3 . Se puede observar que estos comienzan a dejar de atraer como antes y se vuelven inestables, esto hace que el flujo de trayectorias que giran entorno a ellos deja de explosionar hacia dentro y sigue dando vueltas eternamente. Por lo tanto, si se analiza el gráfico de $Ra=28$, se puede observar que tiene el mismo comportamiento que el presentado por Edward Lorenz y que los puntos de atractor serán calculados por la ecuación 5 y 6, donde se obtiene que $P_2 = (6\sqrt{2}, 6\sqrt{2}, 27)$ y $P_3 = (-6\sqrt{2}, -6\sqrt{2}, 27)$, los cuales coinciden por los expuestos en el documento. A este fenómeno se le denomina régimen caótico.

Por lo tanto se comprueba que los resultados obtenidos por el código elaborado para el método de Runge Kutta de orden 4 corresponden a los resultados expuestos por Edwards Lorenz.

4.2. Problema 2: EDP Calor

Luego de tener las simulaciones realizadas, se nota como concuerdan con la teoría presentada al comienzo. Para ambas configuraciones la temperatura de la pared tiende a la distribución estacionaria. En estado transiente, pareciera que cuando se fija el aislante en la zona interna se tiende al estado estacionario más rápido que con la configuración con el aislante fuera. Esto se puede deber a la gran diferencia de temperatura que existe justo en los bordes del aislante. Se estima que pasado 10 días, ambas configuraciones alcanzan el estado estacionario. En la figura 40 presentadas se puede visualizar como los perfiles de temperatura oscilan en torno al estado estacionario presentado en la metodología en la ecuación 35, lo que valida los resultados. Además, se muestra como existe mayor oscilación para el caso de aislante interno. Esa oscilación es sinusoidal, la cual se ve reflejada en las figuras 38 y 39 para la temperatura y el flujo de calor en el borde interno. La amplitud de oscilación pudo ser obtenida satisfactoriamente en la figura 41, de donde se muestra como esta es mayor en la pared exterior, en donde existe la condición periódica. Notar que para la pared interior la amplitud no es nula pero si bastante baja.

Para el cálculo del calor transferido, se nota que la configuración con aislante interno es bastante mejor ya que transfiere un 55 % de calor menos que el caso de aislante externo. Esto se puede deber a que el caso de aislante interno alcanza más rápido el caso estacionario. Ahora en condiciones estacionarias (10 días), ambas configuraciones transfieren el mismo calor, lo que concuerda con la teoría. Este calor es de 886 [kJ/m²] comparado con 1034 [kJ/m²] y 2322 [kJ/m²] que corresponden a aislante interior y exterior en estado transiente respectivamente. Si se comparan los resultados para el caso sin aislante, se nota que en estado transiente situar el aislante afuera aumenta en un 2 % el calor total, mientras que situarlo adentro disminuye en un 54 %. Claramente confirma que en estado transiente es preferible situar el aislante en la zona interior. Ya en estado estacionario, ambas configuraciones transfieren un 55 % menos calor que para el caso sin aislante. Estos resultados demuestran que utilizar un aislante funciona para disminuir la transferencia de calor dado que se aumenta la “resistencia térmica”. Sin embargo, el donde situarlo no es trivial dado que en ciertos intervalos de tiempo, como las primeras 24 horas, puede aumentar el calor transferido. Como se dijo para este caso es preferible situar el aislante en la zona interior. Es importante destacar que esto se debe netamente a la condición inicial. Por esto, situarlo en la zona donde existe el mayor gradiente de temperatura tiene sentido. Si el perfil de temperatura inicial hubiera sido distinto, por ejemplo de $T_{\infty} = 20^{\circ}\text{C}$, el aislante debería haber estado en la cara externa. De todas formas, la diferencia solo ocurrirá en el estado transiente dado que a largo plazo (estacionario) se demostró que cualquier configuración terminará transfiriendo lo mismo.

Para los casos de temperatura fija en el borde interno, se nota como se alejan bastante de la realidad, generando bastantes diferencias para las distribuciones de temperatura respecto al caso con condición convectiva. Además, se alejan del perfil estacionario. Para el cálculo del calor transferido en estado estacionario, se estiman variaciones respecto al teórico de 38 % lo que es bastante si se necesita exactitud en el valor.

Respecto a los esquemas de discretización, los 3 métodos utilizados fueron satisfactorios para modelar la situación propuesta. Los esquemas fueron aplicados a un espaciamiento espacial de $\Delta x = 0.01$ y $\Delta x = 0.005$ y un paso temporal de 1, 5 y 10 segundos. En las figuras presentadas no se notan diferencias y en el cálculo de calor varían los valores en aproximadamente en un 0.0001 %, lo que en este escenario puede ser insignificante. No se quiso disminuir más el espaciamiento espacial para poder simular de buena manera el aislante, el cual es bastante pequeño comparado con el dominio del problema. La discretización realizada necesita un nodo justo en la interfaz y un Δx constante para no disminuir el orden del método. Esta puede ser una gran desventaja del método de diferencias finitas. Por último, se varió el paso temporal con valores de 15, 60, 120, 1200, 3600 segundos para los 3 esquemas. Para $\Delta t = 15$, se nota el gran costo computacional que tienen los esquemas de Euler Implícito y Crank Nicolson, dado que resuelven un sistema lineal en cada iteración. Es por esto que sus tiempos de cálculo son aprox. 28 veces el tiempo de Euler Explícito. A medida que se aumenta el paso temporal con el espaciamiento en la posición fija, los métodos implícitos empiezan a demorarse cada vez menos llegando a tiempos menores a 1 segundo y con perfiles de temperatura muy parecidos a los otros casos. Además, se nota como el método de Euler Explícito diverge al aumentar el paso temporal dada su estabilidad condicional dada por el número de Fourier. Con lo analizado se puede deslumbrar las ventajas de cada método y sus rangos de aplicación, en donde el Explícito permite cálculos bastante exactos y rápidos para tiempos totales bajos. En cambio los métodos implícitos pueden ser utilizados para simular grandes intervalos de tiempo con grandes pasos temporales sin perder mucha precisión. Por último, el costo computacional del método de Crank Nicolson y Euler Implícito son bastante similares y por ende, se prefiere el método de Crank Nicolson para simulaciones con altos pasos temporales dado su orden 2 en el tiempo. Cabe mencionar que los tiempos de cálculo para las simulaciones hubieran sido mayores si se hubiera trabajado con matrices llenas en comparación con el algoritmo de Thomas utilizado.

5. Conclusión

Mediante el análisis realizado para ambos problemas resueltos, se puede concluir que existe una gran variedad de métodos dependiendo del problema a resolver. Los métodos numéricos pueden ser útiles para resolver ecuaciones diferenciales ordinarias y/o parciales que modelan problemas físicos reales, como lo es por ejemplo el atractor de Lorenz y la ecuación de calor.

En el problema 1 se pudo simular el comportamiento del sistema de Lorenz exitosamente, donde se observó que el comportamiento de los rollos de convección variarán dependiendo del valor del número de Rayleigh, ya que existen puntos de atractor donde su efecto será mayor mientras aumenta el Ra . Por lo tanto mientras Ra se encuentre entre $1 < Ra < Ra_{crítico}$, existirá un comportamiento inestable para el origen y asintóticamente estable, con una trayectoria de giro en forma de espiral. Para $Ra = Ra_{crítico}$ existe una bifurcación, donde la trayectoria girará entorno a los puntos, pero en forma de órbitas periódicas que son infinitamente largas. Por último $Ra > Ra_{crítico}$ tenemos una disminución en la atracción y una inestabilidad. El método de RK-4 utilizado demuestra su gran convergencia para modelar incluso sistemas dinámicos caóticos, entregando resultados que concuerdan con la literatura referenciada.

Para el problema 2 se concluye que se pudo modelar satisfactoriamente la transferencia de calor en el muro. Con los distintos esquemas de discretización y parámetros numéricos se pudo concluir que el aislante sirve para reducir la transferencia de calor. Además, su posición no debe ser aleatoria para reducir el flujo de calor en el estado transiente hasta alcanzar el estacionario. Se infiere que este debe ir siempre en la posición con mayor gradiente de temperatura dada por la condición inicial. Para el caso descrito en este trabajo, para reducir el flujo de calor lo más conveniente es situar el aislante en la pared interna, logrando reducir casi un 55 % el calor total en el primer día respecto al caso sin aislante. Respecto a los esquemas, se pudo evidenciar la estabilidad condicional del Método de Euler Explícito, el cual divergió para grandes pasos de tiempo. Por otra parte, los métodos implícitos, Euler Implícito y Crank Nicolson lograron buenas precisiones utilizando incluso pasos de tiempo de 1 hora, permitiendo así modelar incluso semanas. De estos 2 últimos, se prefiere el Método de Crank Nicolson dado su similar costo computacional y su convergencia de segundo orden en el tiempo.

6. Referencias

- [1] Chapra, S; Canale, R.(2007). *Métodos numéricos para ingenieros*, 5ª Edición.
- [2] Gers, R. Apuntes de clases : *MEC270-Métodos numéricos en ingeniería mecánica*.
- [3] Llorente, J. (2017). *Tratamiento cualitativo del Sistema de Lorenz* Journal of the Atmospheric Sciences, Deterministic Nonperiodic Flow, 20, 130-141, 1963.
- [4] Quarteroni, A; Sacco, R; Saleri, F. (2000). *Numerical Mathematics*. Springer.
- [5] Cengel, Y. (2007). *Transferencia de Calor y Masa*. 3ª Edición.
- [6] Hancock, M (2006). *The 1-D Heat Equation*

7. Anexos

7.1. Códigos Pregunta 1

7.1.1. Código Principal Pregunta 1

```
1  clc
2  clear all
3  Pr=10;
4  beta=8/3;
5  valor=0:2:30;
6  valor1=[0.5,10,28];
7
8  t0=0;
9  tf=100;
10 n = 10000;
11 h=(tf-t0)/n;
12 t = 0:h:n*h-h;
13
14
15 %graficos
16
17 %for i=1:15
18 %     Ra=valor(i);
19 %     [x,y,z]=Kutta(Ra,Pr,beta,n,h);
20 %     figure(1)
21 %     nexttile
22 %     ploteo=plot(t,z)
23 %     xlabel('t')
24 %     ylabel('z')
25 %     title('z vs t para Ra:',Ra)
26 %end
27
28 %for i=1:3
29 %     Ra=valor1(i);
30 %     [x,y,z]=Kutta(Ra,Pr,beta,n,h);
31 %     figure(i)
32 %     nexttile
33 %     ploteo=plot3(x,y,z)
34 %     comet3(x,y,z)
35 %     xlabel('x')
36 %     ylabel('y')
37 %     zlabel('z')
38 %     title('Espacio (x,y,z) para Ra:',Ra)
39 %     grid on
40 %end
41
42 %for i=1:3
43 %     Ra=valor1(i);
```

```
44 % [x,y,z]=Kutta(Ra,Pr,beta,n,h);
45 % figure(i)
46 % nexttile
47 % subplot(2,3,1)
48 % plot(x,y)
49 % xlabel('x')
50 % ylabel('y')
51 % title('x vs y para Ra:',Ra)
52 % grid on
53 % subplot(2,3,2)
54 % plot(x,z)
55 % xlabel('x')
56 % ylabel('z')
57 % title('x vs z para Ra:',Ra)
58 % grid on
59 % subplot(2,3,3)
60 % plot(y,z)
61 % xlabel('y')
62 % ylabel('z')
63 % title('y vs z para Ra:',Ra)
64 % grid on
65 %end
66
67 figure(6)
68 [x,y,z]=Kutta(24,Pr,beta,n,h);
69 comet3(x,y,z)
70 xlabel('x')
71 ylabel('y')
72 zlabel('z')
73 title('Espacio (x,y,z) para Ra:',24)
74 grid on
75
76
77 %% Metodo de RK-4
78
79 function [x,y,z]=Kutta(Ra,Pr,beta,n,h)
80 x1=1;
81 y1=1;
82 z1=1;
83
84 x = zeros(n,1);
85 y = zeros(n,1);
86 z = zeros(n,1);
87
88 x(1)=x1;
89 y(1)=y1;
90 z(1)=z1;
91
```

```

92  for i=1:n-1
93      k1=[fun_dx(Pr,x(i),y(i)) fun_dy(Ra,x(i),y(i),z(i)) fun_dz(beta,y(i)
          ,x(i),z(i))];
94      k2=[fun_dx(Pr,x(i)+(h/2)*k1(1),y(i)+(h/2)*k1(2)) fun_dy(Ra,x(i)+(h
          /2)*k1(1),y(i)+(h/2)*k1(2),z(i)+(h/2)*k1(3)) fun_dz(beta,y(i)+(h
          /2)*k1(2),x(i)+(h/2)*k1(1),z(i)+(h/2)*k1(3))];
95      k3=[fun_dx(Pr,x(i)+(h/2)*k2(1),y(i)+(h/2)*k2(2)) fun_dy(Ra,x(i)+(h
          /2)*k2(1),y(i)+(h/2)*k2(2),z(i)+(h/2)*k2(3)) fun_dz(beta,y(i)+(h
          /2)*k2(2),x(i)+(h/2)*k2(1),z(i)+(h/2)*k2(3))];
96      k4=[fun_dx(Pr,x(i)+h*k3(1),y(i)+h*k3(2)) fun_dy(Ra,x(i)+h*k3(1),y(i)
          +h*k3(2),z(i)+h*k3(3)) fun_dz(beta,y(i)+h*k3(2),x(i)+h*k3(1),z(
          i)+(h/2)*k3(3))];
97      x(i+1)=x(i)+(h/6)*(k1(1)+2*k2(1)+2*k3(1)+k4(1));
98      y(i+1)=y(i)+(h/6)*(k1(2)+2*k2(2)+2*k3(2)+k4(2));
99      z(i+1)=z(i)+(h/6)*(k1(3)+2*k2(3)+2*k3(3)+k4(3));
100  end
101
102  x;
103  y;
104  z;
105  end
106
107  %funciones derivadas
108  function [u]=fun_dx(Pr,x,y)
109      u=Pr*(y-x);
110  end
111
112  function [v]=fun_dy(Ra,x,y,z)
113      v=Ra*x-y-x*z;
114  end
115  function [w]=fun_dz(beta,y,x,z)
116      w=y*x-beta*z;
117  end

```

7.2. Códigos Pregunta 2

Se presentan los códigos elaborados para la resolución de la ecuación de calor.

7.2.1. Código Principal Pregunta 2

```
1
2 % Programa principal EDP Calor
3 % Caso convectivo en interior
4
5 format compact
6
7 % Modificacion aislante interior o exterior
8 aisl_in = true;
9
10 if aisl_in
11     lambda_1 = 2;
12     pC_1 = 2*10^6;
13     lambda_2 = 0.1;
14     pC_2 = 10^6;
15     L1 = 0.45;
16     L2 = 0.05;
17 else
18     lambda_2 = 2;
19     pC_2 = 2*10^6;
20     lambda_1 = 0.1;
21     pC_1 = 10^6;
22     L2 = 0.45;
23     L1 = 0.05;
24 end
25
26 alpha_1 = lambda_1/pC_1;
27 alpha_2 = lambda_2/pC_2;
28 h = 4;
29 prop = [alpha_1,alpha_2,lambda_1,lambda_2,h];
30
31 Tw = 10;
32 dTw = 5;
33 T_inf = 20;
34 prob = [Tw,dTw,T_inf];
35
36 % parametros numericos
37
38 dx = 0.01;    %% 0.005    0.01
39 dt = 10;     %1    10 15
40
41 t = 0;
42 hrs = 49;
43 hrsi = 25;
```

```

44 tf = hrs*3600;
45
46 N1 = floor(L1/dx);
47 N2 = floor(L2/dx);
48 N = N1 + N2 + 1;
49 Nt = hrs + 1;
50
51 Ti = ones(1,N)*Tw;
52 L = linspace(0,L1+L2,N);
53 k = N1 + 1;
54
55 % outputs para figuras en pared interior
56
57 dt2 = 30;
58 if dt2<dt
59     dt2 = dt;
60 end
61 Nt2f = hrs*3600/dt2;
62 Nt2i = (hrs-24)*3600/dt2+1;
63
64
65 %%%% EULER EXPLICITO %%%%
66
67 [T_out1,TN] = Euler_Explicito(Ti,N,Nt,t,k,dx,dt,tf,prob,prop,dt2,Nt2f,
    Nt2i);
68 f1 = figure(1);
69 plot_hrs(L,T_out1,hrsi,hrs)
70 hold on
71
72 [q,qj] = Heat_flux_int(TN,dt2,prob,prop);
73
74 T_out = Stationary(prob,prop,L1,L1+L2,k,N);
75 plot(L,T_out,'DisplayName','Stationary')
76 hold on
77 legend('Location','Best')
78 set(f1,'Position',[150 90 650 530]);
79
80 %%%% EULER IMPLICITO %%%%
81
82 [T_out2,TN] = Euler_Implicito(Ti,N,Nt,t,k,dx,dt,tf,prob,prop,dt2,Nt2f,
    Nt2i);
83 f2 = figure(2);
84 plot_hrs(L,T_out2,hrsi,hrs)
85 hold on
86
87 [q,qj] = Heat_flux_int(TN,dt2,prob,prop);
88
89 T_out = Stationary(prob,prop,L1,L1+L2,k,N);

```



```

90 plot(L,T_out,'DisplayName','Stationary')
91 hold on
92 legend('Location','Best')
93 set(f2,'Position',[150 90 650 530]);
94
95 %%%% CRANK NICOLSON %%%%
96
97
98 [T_out3,TN] = Crank_Nicolson(Ti,N,Nt,t,k,dx,dt,tf,prob,prop,dt2,Nt2f,
    Nt2i);
99 f3 = figure(3);
100 plot_hrs(L,T_out3,hrsi,hrs)
101 hold on
102
103 [q,qj] = Heat_flux_int(TN,dt2,prob,prop);
104
105 T_out = Stationary(prob,prop,L1,L1+L2,k,N);
106 plot(L,T_out,'DisplayName','Stationary')
107 hold on
108 legend('Location','Best')
109 set(f3,'Position',[150 90 650 530]);
110
111 %%%% GRAFICO CALOR EN CARA INTERIOR %%%%
112
113 figure(4)
114 t = linspace(0,24,length(qj));
115 plot(t,qj,'DisplayName','Flujo calor cara interna 24 hrs')
116 ylabel('q [W/m2]')
117 xlabel('t [hrs]')
118 xlim([0 24])
119 legend
120
121 %%%% GRAFICO TEMPERATURA EN INTERIOR %%%%
122
123 figure(5)
124 plot(t,TN,'DisplayName','Temperatura cara iterna')
125 ylabel('T [ C ]')
126 xlabel('t [hrs]')
127 xlim([0 24])
128 legend
129
130 %%%% AMPLITUD TMPERATURA OSCILANTE %%%%
131
132 figure(6)
133 Tsin = T_out3(hrs-24:hrs,:);
134 T_tot = zeros(1,N);
135 for j=1:N
136     xx = max(Tsin(:,j));

```

```

137     T_tot(j) = xx;
138 end
139 plot(L,T_tot-T_out)
140 xlabel('L [m]')
141 ylabel('T [ C ]')
142 legend('Amplitud oscilante A(x)')
143
144
145
146 %%%%%%%%%%% FUNCTIONS
147 %%%%%%%%%%%
148 % Funcion plots
149
150 function plot_hrs(L,T,ti,tf)
151     if ti==0
152         ti=1;
153         tf = tf + 1;
154     end
155     for j=ti:tf
156         %a = mod(j,24);
157         % if a==0
158         %     a=24
159         % elseif a==1
160         %     a=25
161         % end
162         plot(L,T(j,:), 'DisplayName', strcat(int2str(j-1), 'hrs'))
163         hold on
164     end
165     xlabel('L [m]')
166     ylabel('T [ C ]')
167     legend
168     ylim([4 21])
169 end

```

7.2.2. Función Euler Explícito

```

1
2
3 % Euler explícito
4
5 function [T_out,TN_out] = Euler_Explicito(Ti,N,Nt,t,k,dx,dt,tf,prob,
6     prop,dt2,Nt2f,Nt2i)
7
8     r = num2cell(prop);
9     [alpha_1,alpha_2,lambda_1,lambda_2,h] = deal(r{:});
10    r = num2cell(prob);
11    [Tw,dTw,T_inf] = deal(r{:});

```

```

12     T = zeros(1,N);
13     T_old = Ti;
14     T_out = zeros(Nt,N);
15     TN = zeros(1,Nt2f);
16     z = 1;
17     zz = 1;
18     T_out(z,:) = Ti(:);
19
20     while t<tf
21         t = t + dt;
22         for i=1:N
23             if i==1
24                 T(i) = Tw + dTw*sin((2*pi*t/3600)/(24));
25                 %T(i) = Tw;
26             elseif i>1 && i<k
27                 T(i) = T_old(i) + (dt/(dx^2))*(T_old(i+1)-2*T_old(i)+
28                     T_old(i-1))*alpha_1;
29             elseif i>k && i<N
30                 T(i) = T_old(i) + (dt/(dx^2))*(T_old(i+1)-2*T_old(i)+
31                     T_old(i-1))*alpha_2;
32             elseif i==N
33                 %T(i) = T_inf;
34                 T(i) = (lambda_2*T(i-1)/dx + h*T_inf)/(h+lambda_2/dx);
35             end
36         end
37         T(k) = (lambda_2*T(k+1)+lambda_1*T(k-1))/(lambda_1+lambda_2);
38         T_old(:) = T(:);
39         if mod(t,3600)==0
40             z = z + 1;
41             T_out(z,:) = T(:);
42         end
43         if mod(t,dt2) == 0
44             zz = zz + 1;
45             TN(zz) = T(N);
46         end
47     end
48     TN_out = TN(Nt2i:Nt2f);
49 end

```

7.2.3. Función Euler Implícito

```

1
2 % Euler implicito
3
4 function [T_out,TN_out] = Euler_Implicito(Ti,N,Nt,t,k,dx,dt,tf,prob,
5     prop,dt2,Nt2f,Nt2i)
6
7     r = num2cell(prop);
8     [alpha_1,alpha_2,lambda_1,lambda_2,h] = deal(r{:});

```

```

8   r = num2cell(prob);
9   [Tw,dTw,T_inf] = deal(r{:});
10
11   T_old(:) = Ti(:);
12   T_out = zeros(Nt,N);
13   TN = zeros(1,Nt2f);
14   z = 1;
15   zzz = 1;
16   T_out(z,:) = Ti(:);
17
18   sigma_1 = alpha_1*dt/(dx^2);
19   sigma_2 = alpha_2*dt/(dx^2);
20   theta = lambda_2/(2*dx);
21   zz = 0;
22   if mod(N,3) == 2
23       zz = 1;
24       M = floor(N/3) + zz;
25   elseif mod(N,3) == 0
26       M = floor(N/3);
27   end
28   kk = floor(k/3);
29   if mod(k,3)==0
30       kk = kk - 1;
31   end
32
33   A = zeros(M,3,3);
34   B = zeros(M,3,3);
35   C = zeros(M,3,3);
36   F = zeros(M,3);
37   for i=1:M
38       if i==1
39           C(i,:,:) = [1 0 0; -1 2+1/sigma_1 -1; 0 -1 2+1/sigma_1];
40           B(i,:,:) = [0 0 0; 0 0 0; -1 0 0];
41       elseif i<=kk
42           A(i,:,:) = [0 0 -1; 0 0 0; 0 0 0];
43           C(i,:,:) = [2+1/sigma_1 -1 0; -1 2+1/sigma_1 -1; 0 -1 2+1/
                        sigma_1];
44           B(i,:,:) = [0 0 0; 0 0 0; -1 0 0];
45       elseif i==kk+1
46           if mod(k,3)==0
47               A(i,:,:) = [0 0 -1; 0 0 0; 0 0 0];
48               C(i,:,:) = [2+1/sigma_1 -1 0; -1 2+1/sigma_1 -1;
                           lambda_1 -4*lambda_1 3*(lambda_1+lambda_2)];
49               B(i,:,:) = [0 0 0; 0 0 0; -4*lambda_2 lambda_2 0];
50           else
51               A(i,:,:) = [0 lambda_1 -4*lambda_1; 0 0 0; 0 0 0];
52               C(i,:,:) = [3*(lambda_1+lambda_2) -4*lambda_2 lambda_2;
                           -1 2+1/sigma_2 -1; 0 -1 2+1/sigma_2];

```

```

53         B(i,:,:) = [0 0 0; 0 0 0; -1 0 0];
54     end
55     elseif i>kk+1 && i<M
56         A(i,:,:) = [0 0 -1; 0 0 0; 0 0 0];
57         C(i,:,:) = [2+1/sigma_2 -1 0; -1 2+1/sigma_2 -1; 0 -1 2+1/
                    sigma_2];
58         B(i,:,:) = [0 0 0; 0 0 0; -1 0 0];
59     elseif i==M
60         if mod(N,3) == 2
61             A(i,:,:) = [0 0 -1; 0 0 theta; 0 0 0];
62             C(i,:,:) = [2+1/sigma_2 -1 0; -4*theta 3*theta+h 0; 0 0
                        1];
63         elseif mod(N,3) == 0
64             A(i,:,:) = [0 0 -1; 0 0 0; 0 0 0];
65             C(i,:,:) = [2+1/sigma_2 -1 0; -1 2+1/sigma_2 -1; theta
                        -4*theta 3*theta+h];
66         end
67     end
68 end
69
70 while t<tf
71     t = t + dt;
72     for i=1:M
73         if i==1
74             F(i,1) = Tw + dTw*sin((2*pi*t/3600)/(24));
75             F(i,2) = T_old(i*3-1)/sigma_1;
76             F(i,3) = T_old(i*3)/sigma_1;
77         elseif i<=kk
78             F(i,1) = T_old(i*3-2)/sigma_1;
79             F(i,2) = T_old(i*3-1)/sigma_1;
80             F(i,3) = T_old(i*3)/sigma_1;
81         elseif i==kk+1
82             if mod(k,3)==0
83                 F(i,1) = T_old(i*3-2)/sigma_1;
84                 F(i,2) = T_old(i*3-1)/sigma_1;
85                 F(i,3) = 0;
86             else
87                 F(i,1) = 0;
88                 F(i,2) = T_old(i*3-1)/sigma_2;
89                 F(i,3) = T_old(i*3)/sigma_2;
90             end
91         elseif i>kk+1 && i<M
92             F(i,1) = T_old(i*3-2)/sigma_2;
93             F(i,2) = T_old(i*3-1)/sigma_2;
94             F(i,3) = T_old(i*3)/sigma_2;
95         elseif i==M
96             if mod(N,3) == 2
97                 F(i,1) = T_old(i*3-2)/sigma_2;

```

```

98         F(i,2) = h*T_inf;
99         F(i,3) = 1;
100        elseif mod(N,3) == 0
101            F(i,1) = T_old(i*3-2)/sigma_2;
102            F(i,2) = T_old(i*3-1)/sigma_2;
103            F(i,3) = h*T_inf;
104        end
105    end
106 end
107 T_star = Thomas_algorithm(A,B,C,F,M);
108 T_old(:) = T_star(1:M*3-zz);
109 if mod(t,3600)==0
110     z = z + 1;
111     T_out(z,:) = T_old(:);
112 end
113 if mod(t,dt2) == 0
114     zzz = zzz + 1;
115     TN(zzz) = T_old(N);
116 end
117 end
118 TN_out = TN(Nt2i:Nt2f);
119 end

```

7.2.4. Función Crank Nicolson

```

1
2 % Crank Nicolson
3
4 function [T_out,TN_out] = Crank_Nicolson(Ti,N,Nt,t,k,dx,dt,tf,prob,prop
,dt2,Nt2f,Nt2i)
5
6     r = num2cell(prop);
7     [alpha_1,alpha_2,lambda_1,lambda_2,h] = deal(r{:});
8     r = num2cell(prob);
9     [Tw,dTw,T_inf] = deal(r{:});
10
11     T_old(:) = Ti(:);
12     T_out = zeros(Nt,N);
13     TN = zeros(1,Nt2f);
14     z = 1;
15     zzz = 1;
16     T_out(z,:) = Ti(:);
17
18     sigma_1 = alpha_1*dt/(dx^2);
19     sigma_2 = alpha_2*dt/(dx^2);
20     theta = lambda_2/(2*dx);
21     zz = 0;
22     if mod(N,3) == 2
23         zz = 1;

```

```

24     M = floor(N/3) + zz;
25     elseif mod(N,3) == 0
26         M = floor(N/3);
27     end
28     kk = floor(k/3);
29     if mod(k,3)==0
30         kk = kk - 1;
31     end
32
33     A = zeros(M,3,3);
34     B = zeros(M,3,3);
35     C = zeros(M,3,3);
36     F = zeros(M,3);
37     for i=1:M
38         if i==1
39             C(i,::) = [1 0 0; -1 2+2/sigma_1 -1; 0 -1 2+2/sigma_1];
40             B(i,::) = [0 0 0; 0 0 0; -1 0 0];
41         elseif i<=kk
42             A(i,::) = [0 0 -1; 0 0 0; 0 0 0];
43             C(i,::) = [2+2/sigma_1 -1 0; -1 2+2/sigma_1 -1; 0 -1 2+2/
sigma_1];
44             B(i,::) = [0 0 0; 0 0 0; -1 0 0];
45         elseif i==kk+1
46             if mod(k,3)==0
47                 A(i,::) = [0 0 -1; 0 0 0; 0 0 0];
48                 C(i,::) = [2+2/sigma_1 -1 0; -1 2+2/sigma_1 -1;
lambda_1 -4*lambda_1 3*(lambda_1+lambda_2)];
49                 B(i,::) = [0 0 0; 0 0 0; -4*lambda_2 lambda_2 0];
50             else
51                 A(i,::) = [0 lambda_1 -4*lambda_1; 0 0 0; 0 0 0];
52                 C(i,::) = [3*(lambda_1+lambda_2) -4*lambda_2 lambda_2;
-1 2+2/sigma_2 -1; 0 -1 2+2/sigma_2];
53                 B(i,::) = [0 0 0; 0 0 0; -1 0 0];
54             end
55         elseif i>kk+1 && i<M
56             A(i,::) = [0 0 -1; 0 0 0; 0 0 0];
57             C(i,::) = [2+2/sigma_2 -1 0; -1 2+2/sigma_2 -1; 0 -1 2+2/
sigma_2];
58             B(i,::) = [0 0 0; 0 0 0; -1 0 0];
59         elseif i==M
60             if mod(N,3) == 2
61                 A(i,::) = [0 0 -1; 0 0 theta; 0 0 0];
62                 C(i,::) = [2+2/sigma_2 -1 0; -4*theta 3*theta+h 0; 0 0
1];
63             elseif mod(N,3) == 0
64                 A(i,::) = [0 0 -1; 0 0 0; 0 0 0];
65                 C(i,::) = [2+2/sigma_2 -1 0; -1 2+2/sigma_2 -1; theta
-4*theta 3*theta+h];

```

```

66         end
67     end
68 end
69
70 while t<tf
71     t = t + dt;
72     for i=1:M
73         if i==1
74             F(i,1) = Tw + dTw*sin((2*pi*t/3600)/(24));
75             F(i,2) = T_old(i*3-2)+(2/sigma_1-2)*T_old(i*3-1)+T_old(i*3);
76             F(i,3) = T_old(i*3-1)+(2/sigma_1-2)*T_old(i*3)+T_old(i*3+1);
77         elseif i<=kk
78             F(i,1) = T_old(i*3-3)+(2/sigma_1-2)*T_old(i*3-2)+T_old(i*3-1);
79             F(i,2) = T_old(i*3-2)+(2/sigma_1-2)*T_old(i*3-1)+T_old(i*3);
80             F(i,3) = T_old(i*3-1)+(2/sigma_1-2)*T_old(i*3)+T_old(i*3+1);
81         elseif i==kk+1
82             if mod(k,3)==0
83                 F(i,1) = T_old(i*3-3)+(2/sigma_1-2)*T_old(i*3-2)+T_old(i*3-1);
84                 F(i,2) = T_old(i*3-2)+(2/sigma_1-2)*T_old(i*3-1)+T_old(i*3);
85                 F(i,3) = 0;
86             else
87                 F(i,1) = 0;
88                 F(i,2) = T_old(i*3-2)+(2/sigma_2-2)*T_old(i*3-1)+T_old(i*3);
89                 F(i,3) = T_old(i*3-1)+(2/sigma_2-2)*T_old(i*3)+T_old(i*3+1);
90             end
91         elseif i>kk+1 && i<M
92             F(i,1) = T_old(i*3-3)+(2/sigma_2-2)*T_old(i*3-2)+T_old(i*3-1);
93             F(i,2) = T_old(i*3-2)+(2/sigma_2-2)*T_old(i*3-1)+T_old(i*3);
94             F(i,3) = T_old(i*3-1)+(2/sigma_2-2)*T_old(i*3)+T_old(i*3+1);
95         elseif i==M
96             if mod(N,3) == 2
97                 F(i,1) = T_old(i*3-3)+(2/sigma_2-2)*T_old(i*3-2)+T_old(i*3-1);
98                 F(i,2) = h*T_inf;
99                 F(i,3) = 1;
100             elseif mod(N,3) == 0

```



```

101         F(i,1) = T_old(i*3-3)+(2/sigma_2-2)*T_old(i*3-2)+
            T_old(i*3-1);
102         F(i,2) = T_old(i*3-2)+(2/sigma_2-2)*T_old(i*3-1)+
            T_old(i*3);
103         F(i,3) = h*T_inf;
104     end
105 end
106 end
107 T_star = Thomas_algorithm(A,B,C,F,M);
108 T_old(:) = T_star(1:M*3-zz);
109 if mod(t,3600)==0
110     z = z + 1;
111     T_out(z,:) = T_old(:);
112 end
113 if mod(t,dt2) == 0
114     zzz = zzz + 1;
115     TN(zzz) = T_old(N);
116 end
117 end
118 TN_out = TN(Nt2i:Nt2f);
119 end

```

7.2.5. Función Cálculo del Calor con Método del Trapecio

```

1
2
3 % Calculo del flujo de calor en interior
4
5 function [q,qj] = Heat_flux_int(T,dt,prob,prop)
6
7     r = num2cell(prop);
8     [~,~,~,~,h] = deal(r{:});
9     r = num2cell(prob);
10    [~,~,T_inf] = deal(r{:});
11    qj(:) = h*(T(:)-T_inf);
12
13    q = Total_heat(qj,dt);
14
15    function [qt] = Total_heat(qj,dt)
16        sum = 0;
17        for j=1:length(qj)-1
18            sum = sum + qj(j) + qj(j+1);
19        end
20        qt = sum*dt/2;
21    end
22
23 end

```

7.2.6. Función Distribución Temperatura Estacionaria

```

1 % Perfil de temperatura estacionaria teorica
2
3 function [T_out] = Stationary(prob,prop,L1,L,k,N)
4
5     r = num2cell(prop);
6     [~,~,lambda_1,lambda_2,h] = deal(r{:});
7     r = num2cell(prob);
8     [Tw,~,T_inf] = deal(r{:});
9
10    x = linspace(0,L,N);
11
12    phi = h*(L*lambda_1-L1*(lambda_1-lambda_2))+lambda_1*lambda_2;
13    c1 = h*lambda_2*(T_inf-Tw)/phi;
14    c2 = h*lambda_1*(T_inf-Tw)/phi;
15    c3 = (h*(L*lambda_1*Tw-L1*(lambda_1-lambda_2)*T_inf)+lambda_1*
        lambda_2*Tw)/phi;
16
17    T_out = zeros(1,N);
18    T_out(1:k) = c1*x(1:k) + Tw;
19    T_out(k+1:N) = c2*x(k+1:N) + c3;
20
21 end

```

7.2.7. Algoritmo de Thomas

```

1
2 % Algoritmo de thomas por bloques
3
4 function [X_out] = Thomas(A,B,C,F,n)
5
6     psi_1 = zeros(n,3,3);
7     psi_2 = zeros(n,3);
8     X = zeros(n,3);
9     X_out = zeros(1,n*3);
10
11     psi_1(1,:,:) = (to_mat(C,1))\to_mat(B,1);
12     psi_2(1,:) = transpose((to_mat(C,1))\to_vec(F,1)');
13
14
15     for i=2:n-1
16         psi_1(i,:,:) = (to_mat(C,i)-to_mat(A,i)*to_mat(psi_1,i-1))\
            to_mat(B,i);
17         psi_2(i,:) = transpose((to_mat(C,i)-to_mat(A,i)*to_mat(psi_1,i-1))\
            transpose(to_vec(F,i)-transpose(to_mat(A,i)*to_vec(
            psi_2,i-1)')));
18     end
19     psi_2(n,:) = transpose((to_mat(C,n)-to_mat(A,n)*to_mat(psi_1,n-1))\
        transpose(to_vec(F,n)-transpose(to_mat(A,n)*to_vec(psi_2,n-1)'))

```

```
20     );
21     X(n,:) = to_vec(psi_2,n);
22     for ss=1:n-1
23         i = n-ss;
24         X(i,:) = to_vec(psi_2,i) - transpose(to_mat(psi_1,i)*to_vec(X,i
25             +1)');
26     end
27     for i=1:n
28         X_out(i*3-2) = X(i,1);
29         X_out(i*3-1) = X(i,2);
30         X_out(i*3) = X(i,3);
31     end
32
33
34     function [Z] = to_mat(X,q)
35         Z = zeros(3,3);
36         for l=1:3
37             for j=1:3
38                 Z(l,j) = X(q,l,j);
39             end
40         end
41     end
42
43     function [Z] = to_vec(X,q)
44         Z = zeros(1,3);
45         for l=1:3
46             Z(l) = X(q,l);
47         end
48     end
49
50 end
```