

Ecuaciones diferenciales parciales

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

17 de mayo de 2018

Outline

- 1 EDP Parabólicas
 - Introducción EDP Parabólicas
 - Ecuaciones de difusión y calor 1D
- 2 Métodos de diferencias finitas
- 3 Ejercicios
- 4 Problemas a resolver a cuenta de examen

Outline

- 1 EDP Parabólicas
- 2 Métodos de diferencias finitas
 - Tres enfoques de diferencias finitas
 - Método explícito de diferencias finitas
 - Algoritmo para el método FTCS
- 3 Ejercicios
- 4 Problemas a resolver a cuenta de examen

Outline

- 1 EDP Parabólicas
- 2 Métodos de diferencias finitas
- 3 Ejercicios
 - Ecuación de calor
 - Solución numérica
 - Resolviendo el problema
- 4 Problemas a resolver a cuenta de examen

Outline

- 1 EDP Parabólicas
- 2 Métodos de diferencias finitas
- 3 Ejercicios
- 4 Problemas a resolver a cuenta de examen
 - Problema 1
 - Problema 2
 - Problema 3

- 1 EDP Parabólicas
 - Introducción EDP Parabólicas
 - Ecuaciones de difusión y calor 1D
- 2 Métodos de diferencias finitas
- 3 Ejercicios
- 4 Problemas a resolver a cuenta de examen

Ecuación para el desarrollo

Como prototipo de una EDP de tipo parabólico, podemos utilizar la ecuación de difusión en 1D o la ecuación de calor.

Las técnicas de discretización aplicadas y el análisis de estabilidad correspondiente tienen validez general y son aplicables también a otras ecuaciones parabólicas.

Consideraciones importantes

A pesar del considerable número de soluciones analíticas disponibles para la ecuación de difusión, éstas se limitan a geometrías simples y coeficientes de difusión constantes.

Consideraciones importantes

Las condiciones de frontera manejables analíticamente son igualmente simples, sin embargo, las soluciones se expresan a menudo como series infinitas, que no son triviales para evaluar.

Consideraciones importantes

Para obtener soluciones de la ecuación de difusión que modelen de manera realista las situaciones prácticas, generalmente se necesita recurrir a algoritmos numéricos.

Consideraciones importantes

Básicamente, éstos implican restringir la solución a un conjunto discreto de puntos de malla y aproximar las derivadas por esquemas de diferencias finitas con relación a éstos.

Posteriormente, el enfoque numérico se reduce a resolver el sistema lineal resultante, cuyas incógnitas son los valores de la solución en los puntos de malla.

Ecuaciones de difusión y calor 1D

$$\frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} = 0 \quad (1)$$

$$\frac{\partial u}{\partial t} - K \frac{\partial^2 u}{\partial x^2} = f(x, t) \quad (2)$$

describen la evolución temporal y la distribución espacial de la concentración de un elemento difusor ($u(x, t) \equiv c(x, t)$) y la temperatura del sistema ($u(x, t) \equiv T(x, t)$), respectivamente.

Ecuaciones de difusión y calor 1D

El coeficiente de difusión D , relaciona el flujo difusivo J_{dif} , con el gradiente de concentración por la ley de Fick, mientras que la difusividad térmica K , relaciona el flujo de calor J_{calor} , con el gradiente de temperatura por la ley de Fourier:

$$J_{\text{dif}} = -D \frac{\partial c}{\partial x} \quad J_{\text{calor}} = -K \frac{\partial T}{\partial x}$$

La expresión anterior de la ecuación de calor también prevé una fuente de calor $f(x, t)$

Ecuaciones de difusión y calor 1D

A pesar de que ambas ecuaciones parecen estar espacialmente en 1D, de manera alternativa se puede asumir que modelan una geometría en un plano 3D, en una “losa”.

Ecuaciones de difusión y calor 1D

En la que existe una dependencia explícita sólo en la coordenada cartesiana perpendicular a la losa (x en las ecuaciones anteriores) y no hay dependencia de las otras dos coordenadas ($y - z$), a lo largo de las cuales la extensión del sistema se considera implícitamente infinita.

Condiciones iniciales y de frontera

Modelar una situación física en particular requiere, además de la EDP real, que se proporcionen las condiciones iniciales y de frontera.

Condiciones iniciales

Las condiciones iniciales especifican la solución sobre todo el dominio espacial en un momento inicial t_0 :

$$U(x, t_0) = u^0(x), \quad x \in [0, L] \quad (3)$$

Condiciones de frontera

Las CDF definen el comportamiento de la solución en la frontera para $t > t_0$.

Por simplicidad, consideraremos para un tiempo dado, las CDF de tipo Dirichlet, que implican valores de solución en las fronteras:

$$u(0, t) = u_0^0, \quad u(L, t) = u_L^0 \quad (4)$$

Discretización de la ED parabólicas

Hay varios métodos manuales para discretizar ecuaciones parabólicas.

No obstante, con ligeras modificaciones de los esquemas de discretización se tiene un impacto considerable en la estabilidad y exactitud de la evolución temporal de la solución.

1 EDP Parabólicas

2 Métodos de diferencias finitas

- Tres enfoques de diferencias finitas
- Método explícito de diferencias finitas
- Algoritmo para el método FTCS

3 Ejercicios

4 Problemas a resolver a cuenta de examen

Métodos de diferencias finitas

A continuación, abordamos tres enfoques de diferencias finitas:

- 1 El método explícito.

Métodos de diferencias finitas

A continuación, abordamos tres enfoques de diferencias finitas:

- 1 El método explícito.
- 2 El método implícito.

Métodos de diferencias finitas

A continuación, abordamos tres enfoques de diferencias finitas:

- 1 El método explícito.
- 2 El método implícito.
- 3 El método de Crank-Nicolson.

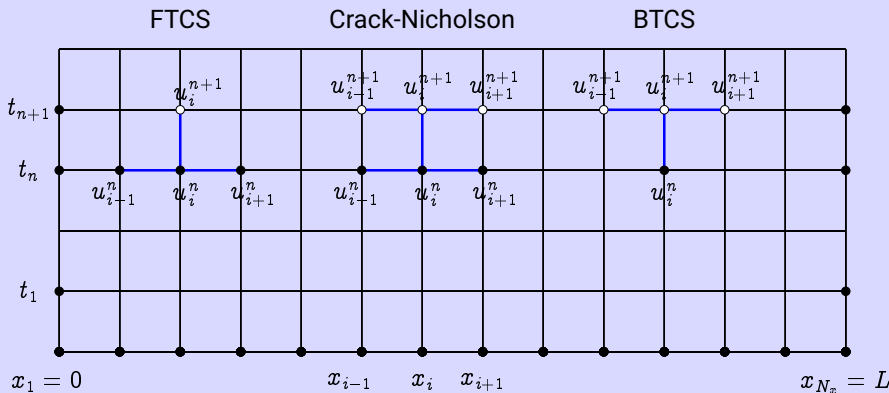


Figura (1): Configuración de nodos espaciales y temporales para tres métodos de discretización de EDP parabólicas.

Método explícito de diferencias finitas

Comenzaremos la discretización para la ecuación de difusión (1) junto con las CDF (3) y (4), haciendo una malla espacio-temporal regular, caracterizada por los nodos:

$$x_i = (i - 1) h_x, \quad i = 1, 2, \dots, N_x \quad (5)$$

$$t_n = n h_t, \quad n = 0, 1, 2, \dots \quad (6)$$

donde h_t es el paso temporal y h_x representa el paso entre los N_x nodos espaciales.

Método explícito de diferencias finitas

El espaciamiento de los nodos está dado por

$$h_x = \frac{L}{N_x - 1} \quad (7)$$

Método explícito de diferencias finitas

La primera derivada con respecto al tiempo $\partial u / \partial t$ en el nodo espacio - temporal (x_i, t_n) puede obtenerse de una manera directa a partir de la aproximación lineal de la serie de Taylor con respecto a t para la constante $x = x_i$

$$u_i^{n+1} = u_i^n + h_t \left(\frac{\partial u}{\partial t} \right)_{i,n} + O(h_t^2)$$

donde $u_i^n \equiv u(x_i, t_n)$

Método explícito de diferencias finitas

Separando la derivada temporal, se obtiene el esquema de diferencias hacia adelante:

$$\left(\frac{\partial u}{\partial t}\right)_{i,n} = \frac{u_i^{n+1} - u_i^n}{h_t} + O(h_t) \quad (8)$$

Método explícito de diferencias finitas

El carácter “hacia adelante” se da por la presencia de la solución propagada de t^{n+1} en la expresión de la derivada en t^n , mientras que el hecho de que el esquema mantenga sólo el primer orden exacto en h_t se debe a la división implícita por h_t .

Método explícito de diferencias finitas

Para la segunda derivada espacial en el nodo espacio - temporal (x_i, t_n) , podemos usar el esquema de diferencias centrales:

$$\left(\frac{\partial^2 u}{\partial x^2} \right)_{i,n} = \frac{u_{i+1}^n - 2 u_i^n + u_{i-1}^n}{h_x^2} + O(h_x^2) \quad (9)$$

Que involucra sólo la información de t_n y de los puntos espaciales ubicados simétricamente alrededor del nodo donde se está calculando la derivada.

Método FTCS

Sustituyendo las expresiones de diferencias finitas (8) y (9) en la ecuación de difusión (1), se obtiene el *método espacio - temporal hacia adelante* (**Forward Time Central Space FTCS**)

$$\frac{u_i^{n+1} - u_i^n}{h_t} = D \frac{u_{i+1}^n - 2 u_i^n + u_{i-1}^n}{h_x^2} \quad (10)$$

Método FTCS

Sustituyendo las expresiones de diferencias finitas (8) y (9) en la ecuación de difusión (1), se obtiene el *método espacio - temporal hacia adelante* (**Forward Time Central Space FTCS**)

$$\frac{u_i^{n+1} - u_i^n}{h_t} = D \frac{u_{i+1}^n - 2 u_i^n + u_{i-1}^n}{h_x^2} \quad (10)$$

que proporciona una aproximación del orden $O(h_x^2 + h_t)$ para la ecuación de difusión en el nodo espacio - temporal (x_i, t_n)

Método FTCS

Por lo que podemos expresar la solución propagada en el siguiente paso temporal t_{n+1} para cada punto interior x_i de la malla espacial, sólo en términos de valores previos de tiempo anterior t_n :

$$u_i^{n+1} = \lambda u_{i-1}^n + (1-2\lambda)u_i^n + \lambda u_{i+1}^n \quad i = 2, 3, \dots, N_x-1 \quad (11)$$

donde

$$\lambda = \frac{D h_t}{h_x^2} \quad (12)$$

En cuanto a los valores de la solución en la frontera, están fijos por las condiciones de Dirichlet (4) a lo largo de la propagación y no requieren ningún tratamiento particular:

$$u_1^{n+1} = u_1^n = u_0^0, \quad u_{N_x}^{n+1} = u_{N_x}^n = u_L^0 \quad (13)$$

Puesto que en cada paso del tiempo, los componentes de la solución propagada pueden expresarse independientemente de la ec. (11), exclusivamente basada en los datos del paso de tiempo previo.

Proceso de propagación

Se dice que el **método FTCS es explícito** si consideramos el proceso de propagación, al re-escribir la ec. (11) a la ec. (13) en notación matricial

$$\mathbf{u}^{n+1} = \mathbf{B} \cdot \mathbf{u}^n, \quad n = 1, 2, \dots \quad (14)$$

Proceso de propagación

La matriz de propagación \mathbf{B} es tridiagonal y el vector \mathbf{u}^n recoge los valores de solución de todos los puntos de malla espacial en el paso de tiempo t_n :

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & & & 0 \\ \lambda & 1 - 2\lambda & \lambda & & \\ & \ddots & \ddots & \ddots & \\ & & \lambda & 1 - 2\lambda & \lambda \\ 0 & & & 0 & 1 \end{bmatrix}, \quad \mathbf{u}^n = \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N_x-1}^n \\ u_{N_x}^n \end{bmatrix} \quad (15)$$

Método FTCS

Cada componente de la solución propagada \mathbf{u}^{n+1} de manera individual, se obtiene de la multiplicación de la matriz \mathbf{B} con la solución previa del vector \mathbf{u}^n .

Problema de difusión en 1D

Consideremos la ecuación de difusión en 1D:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad x \in [0, L] \quad t > 0 \quad (16)$$

con una constante de difusión sujeta a las CDF de Dirichlet:

$$u(0, t) = u(L, t) = 0 \quad t > 0 \quad (17)$$

Solución inicial y exacta

Consideremos una solución inicial del tipo

$$u(x, 0) = \sin\left(\frac{\pi x}{L}\right) \quad x \in [0, L] \quad (18)$$

Solución inicial y exacta

Consideremos una solución inicial del tipo

$$u(x, 0) = \sin\left(\frac{\pi x}{L}\right) \quad x \in [0, L] \quad (18)$$

La solución exacta que satisface este problema es:

$$u(x, t) = \exp\left(\frac{-\pi D t}{L^2}\right) \sin\left(\frac{\pi x}{L}\right) \quad (19)$$

Otros valores

Consideramos $D = 0.1$ y $L = 1$, la solución se propaga al valor $t_{\max} = 6.0$.

El paso temporal es de $h_t = 1.25 \times 10^{-2}$, el paso espacial es de $h_x = 0.05$, por lo que el parámetro de discretización (ec. 12) es $\lambda = 0.5$

Algoritmo para el método FTCS

La rutina **PropagFTCS** resuelve un problema con CDF de Cauchy para la ecuación de difusión (1), propagando la solución inicial $u_0[]$ durante el intervalo de tiempo ht de acuerdo con las ecs. (11) - (13) obtenidas aplicando el esquema explícito de discretización FTCS.

Algoritmo para el método FTCS

El coeficiente de difusión constante, el tamaño del paso espacial y el número de nodos se reciben a través de los argumentos D , h y n , respectivamente.

La solución propagada se devuelve por medio de la matriz u .

Algoritmo para el método FTCS

Se supone que la secuencia de manejo en el programa principal realiza cualquier procesamiento adicional, sobre todo para liberar la matriz $u[]$ para un nuevo paso de propagación guardando su contenido en $u0[]$.

Algoritmo para el método FTCS

El código primero inicializa la solución $u_0[]$ mediante una llamada a la función **Init** y luego controla la propagación dentro de un bucle temporal mediante llamadas repetidas a **PropagFTCS**, cada una seguida necesariamente por una transferencia hacia atrás del contenido de $u[]$ a $u_0[]$.

Propagador explícito FTCS

Código (1): Propagador explícito FTCS para la ecuación de difusión

```
1 def PropagFTCS(u0, u, nx, D, hx, ht):
2
3     lam = D*ht/(hx*hx)
4     lam1 = 1e0 - 2e0*lam
5
6     u[1] = u0[1]; u[nx] = u0[nx]
7     for i in range(2, nx):
8         u[i] = lam*u0[i-1]+lam1*u0[i]+
9         lam*u0[i+1]
10
11     return u
```

Programa principal I

Código (2): Programa principal para el ejercicio

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def Init(u, x, nx):
5     global L
6     for i in range(1, nx+1):
7         u[i] = np.sin(np.pi * x[i]/L)
8
9 D      = 0.1e0
10 L      = 1e0
11 nx     = 21
```


Programa principal II

```
12 tmax = 6.0e0
13 ht    = 1.25e-2
14
15 u0 = [0]*(nx+1); u = [0]*(nx+1)
16
17 x = [0]*(nx+1)
18
19 hx = L/(nx-1)
20
21 for i in range(1, nx+1):
22     x[i] = (i-1)*hx
23
24 Init(u0, x, nx)
```

Programa principal III

```
25
26 t = 0e0
27
28 while (t < tmax):
29     t += ht
30     PropagFTCS(u0, u, nx, D, hx, ht)
31     for i in range(1, nx):
32         u0[i] = u[i]
33
34 f = np.exp(-np.pi*np.pi*D*t/(L*L))
35
36 exacta = []
37
```

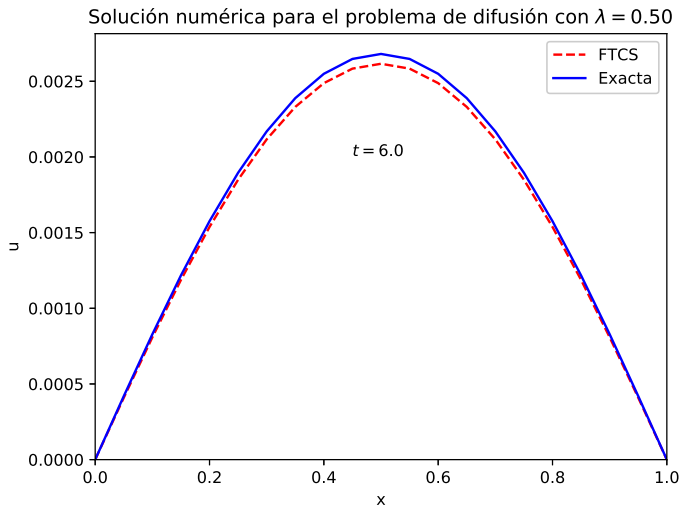
Programa principal IV

```
38 for i in range(nx+1):
39     exacta.append(f * np.sin(np.pi*x[
        i]/L))
40
41 plt.plot(x, u, 'r', label='FTCS', ls
        ='dashed')
42 plt.plot(x, exacta, 'b', label='
        Exacta')
43 plt.title('Solucion numerica para el
        problema de difusion con  $\lambda$ 
        =0.50')
44 plt.xlabel('x')
45 plt.ylabel('u')
```

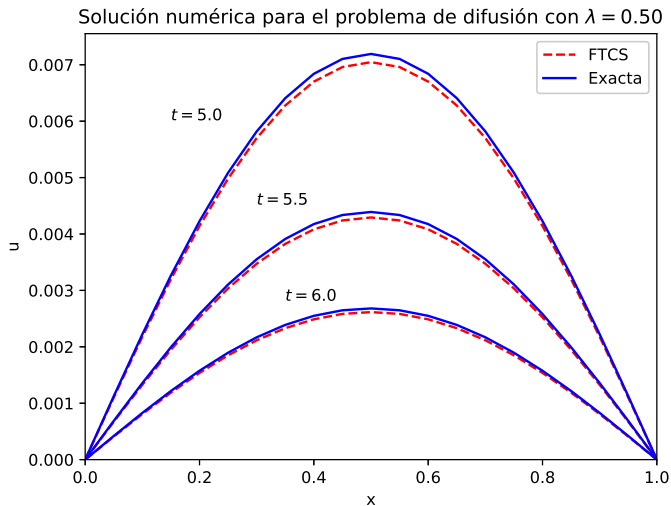
Programa principal V

```
46 plt.legend(loc=1)
47 plt.xlim([0, 1])
48 plt.ylim(ymin=0)
49 plt.show()
```

Gráfica del perfil espacial



Gráfica de los perfiles a tres tiempos



Inestabilidad en el método FTCS

La solución de la ecuación de difusión (1) por el método explícito FTCS es propensa a una fenomenología numérica particular, siendo confrontada bajo ciertas condiciones con una inestabilidad numérica severa.

Inestabilidad en el método FTCS

La inestabilidad implica que, en lugar de perfiles espaciales suaves y de buen comportamiento, la propagación desarrolla oscilaciones espúreas, que crecen exponencialmente en el tiempo, distorsionando la solución de manera irrecuperable.

Inestabilidad en el método FTCS

Este comportamiento crítico se debe al creciente dominio de los errores de redondeo y siempre surge cuando el tamaño del paso de tiempo excede un cierto límite superior, que se correlaciona con el tamaño de paso espacial empleado.

Cambio en los valores del paso temporal

Usaremos ahora el valor de $h_t = 1.30 \times 10^{-2}$,
manteniendo el mismo paso espacial de
 $h_x = 0.05$.

Cambio en los valores del paso temporal

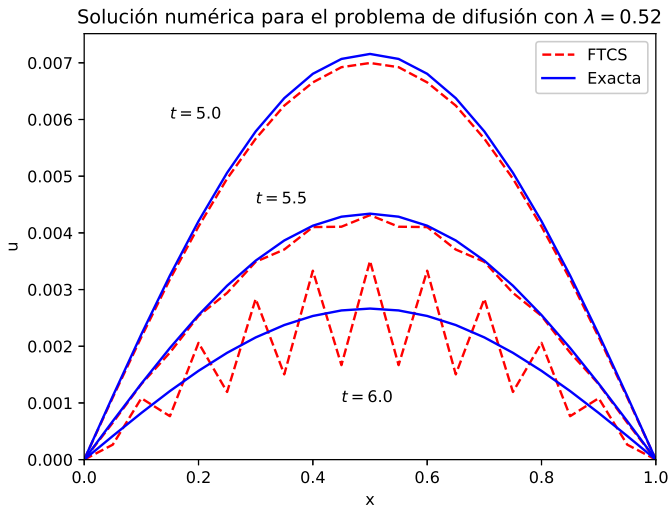
Usaremos ahora el valor de $h_t = 1.30 \times 10^{-2}$,
manteniendo el mismo paso espacial de
 $h_x = 0.05$.

El parámetro de discretización ahora será
 $\lambda = 0.52$, ligeramente mayor a $\lambda = 0.50$.

Cambio en los valores del paso temporal

Ocupando el mismo algoritmo de solución, ejecutamos el código y a continuación veremos el comportamiento de la solución.

Perfiles de solución



Inestabilidad para un valor de λ

Mientras que en $t = 5.0$, las soluciones obtenidas con los dos pasos de tiempo son apenas perceptibles, las inestabilidades comienzan a desarrollarse en $t = 5.5$, y dominan completamente la solución en $t = 6.0$.

Inestabilidad para un valor de λ

Por lo tanto, un aumento aparentemente marginal del paso de tiempo h_t genera un tremendo cambio cualitativo en el comportamiento de la solución.

Inestabilidad para un valor de λ

Entonces con $\lambda = 1/2$ se tiene un valor crítico, que separa dos dominios distintos de propagación estable ($\lambda \leq 1/2$) e inestable ($\lambda > 1/2$), respectivamente.

Criterio de inestabilidad

Revisa la guía de apoyo con el tema de análisis de estabilidad de Von Neumann, en donde se desarrolla y discuten ciertas características y propiedades de las soluciones para las EDP con los esquemas mostrados.

- 1 EDP Parabólicas
- 2 Métodos de diferencias finitas
- 3 Ejercicios
 - Ecuación de calor
 - Solución numérica
 - Resolviendo el problema
- 4 Problemas a resolver a cuenta de examen

Problema de inicio

Ecuación de calor

Tenemos una barra de aluminio con una longitud $L = 1\text{m}$ y de diámetro w (vista desde el eje x). La barra está aislada en su perímetro, excepto en los extremos.

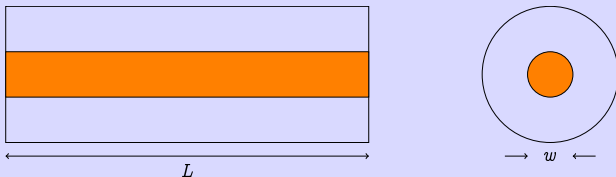


Figura (2): La barra está rodeada de un aislante.

Condiciones del problema

Al inicio, la barra tiene una temperatura uniforme de 100°C y los extremos de la misma están en contacto con agua helada a 0°C . El calor fluye hacia los extremos que no están dentro del aislante.

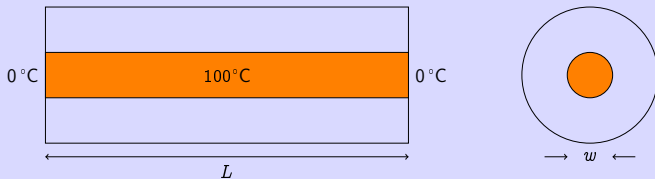


Figura (3): Condiciones de temperatura dentro y en los extremos de la barra.

¿Qué tenemos que hacer?

El problema a resolver es el siguiente:

Calcular cómo varía la temperatura a lo largo de la barra conforme transcurre el tiempo.

Ecuación de calor

¿Cómo es el flujo de calor de una región caliente a una región fría?

Ecuación de calor

Expresando el fenómeno en términos matemáticos: decimos que la razón de cambio de flujo de calor H a través de un material, es proporcional al gradiente de temperatura T en el material.

$$H = -K \nabla T(x, t) \quad (20)$$

donde K es la conductividad térmica del material.

Ecuación de calor

La cantidad total de calor $Q(t)$ en el material y para cualquier momento, es proporcional a la integral de la temperatura sobre del volumen del material:

$$Q(t) = \int C \rho(x) T(x, t) dx \quad (21)$$

Donde C es el calor específico del material y ρ es la densidad del material.

Dado que la energía se conserva, la razón de decremento de Q con el tiempo debe de ser igual a la cantidad de calor fluyendo fuera del material.

Ecuación de calor

Después de este balance de energía, aplicamos el teorema de la divergencia, y por tanto la ecuación de calor resulta:

$$\frac{\partial T(x, t)}{\partial t} = \frac{K}{C \rho} \nabla^2 T(x, t) \quad (22)$$

Suponemos que la densidad del material es constante.

Ecuación de calor

Tenemos una EDP de tipo parabólico con variables de posición y tiempo independientes.

Al especificar este tipo de problema, implica que no hay variación de la temperatura en las direcciones perpendiculares de la barra (y, z), por lo que sólo tenemos una coordenada espacial en el laplaciano:

$$\frac{\partial T(x, t)}{\partial t} = \frac{K}{C \rho} \frac{\partial^2 T(x, t)}{\partial x^2} \quad (23)$$

La temperatura inicial de la barra y las condiciones de frontera son:

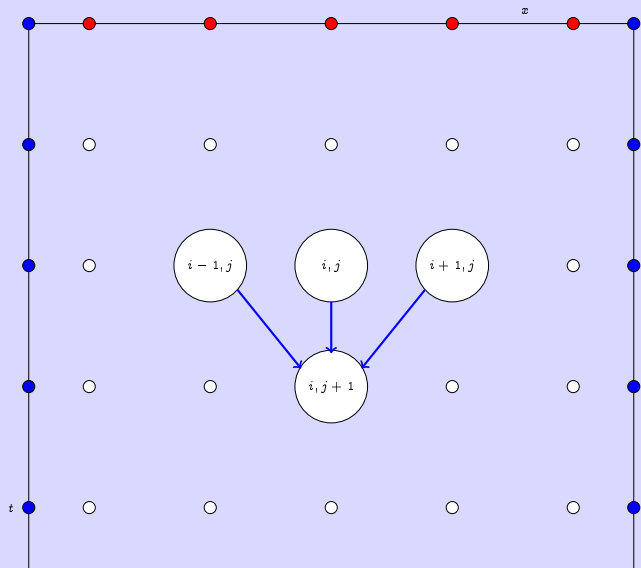
$$\begin{aligned} T(x, t = 0) &= 100^{\circ}\text{C} \\ T(x = 0, t) &= T(x = L, t) = 0^{\circ}\text{C} \end{aligned} \quad (24)$$

Solución numérica

Como se revisó con la ecuación de Laplace y con la de difusión, la solución numérica se basa en convertir una ecuación diferencial en una aproximación por diferencias finitas.

Se discretiza el espacio y el tiempo en una malla y se resuelven las soluciones en los nodos.

Malla para los cálculos



Los nodos horizontales de color rojo corresponden a los valores conocidos de la temperatura para el tiempo inicial, mientras que los nodos azules verticales corresponden a la temperatura fija en los extremos.

Con sólo la fila superior conocida, proponemos un algoritmo que avanza en el tiempo una fila a la vez, como en el salto de una rana, a este método se le conoce como **leapfrog**.

Desarrollo para la solución

Como suele ser el caso con EDP, el algoritmo se personaliza para la ecuación que se está resolviendo y para las restricciones impuestas por el conjunto particular de condiciones iniciales y de frontera.

Primera derivada para la solución

Con solo un renglón de datos para comenzar, usamos una aproximación por diferencias hacia adelante para la derivada de la temperatura con respecto al tiempo

$$\frac{\partial T(x, t)}{\partial t} \simeq \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} \quad (25)$$

Primera derivada para la solución

Debido a que conocemos la variación espacial de la temperatura a lo largo de toda la fila superior y en los lados izquierdo y derecho, estamos menos limitados con la derivada espacial que con la derivada del tiempo.

Segunda derivada para la solución

Usaremos una aproximación por diferencias central para la derivada espacial

$$\frac{\partial^2 T(x, t)}{\partial x^2} \simeq \frac{T(x + \Delta x, t) - T(x - \Delta x, t) - 2 T(x, t)}{(\Delta x)^2} \quad (26)$$

Ecuación por diferencias finitas

Al sustituir las aproximaciones en la ec. (23), se obtiene la ecuación de calor por diferencias:

$$\frac{K}{C \rho} \frac{T(x + \Delta x, t) - T(x - \Delta x, t) - 2 T(x, t)}{(\Delta x)^2} \quad (27)$$

Algoritmo discretizado

Reordenamos la ec. (27) en una forma en la que T puede avanzar en t :

$$T_{i,j+1} = T_{i,j} + \eta [T_{i+1,j} + T_{i-1,j} - 2 T_{i,j}], \quad \eta = \frac{K \Delta t}{C \rho \Delta x^2} \quad (28)$$

donde $x = i \Delta x$ y $t = j \Delta t$.

Naturaleza del algoritmo

Este algoritmo es **explícito** porque proporciona una solución en términos de valores conocidos de la temperatura.

Naturaleza del algoritmo

Si tratamos de resolver simultáneamente la temperatura en todos los sitios de la malla, entonces tendríamos un algoritmo **implícito** que requiere que resolvamos las ecuaciones que involucran valores desconocidos de la temperatura.

Naturaleza del algoritmo

Vemos que la temperatura en el nodo espacio-tiempo $(i, j + 1)$ se calcula a partir de los tres valores de temperatura en un tiempo anterior j y en los valores espaciales adyacentes $i \pm 1, i$.

Naturaleza del algoritmo

Comenzamos la solución en la fila superior, avanzando en el tiempo y manteniendo la temperatura a lo largo de los extremos fijos en 0°C .

Datos para el problema

Considera una barra de aluminio de longitud $L = 1\text{m}$, con las condiciones de frontera y condiciones iniciales:

$$\begin{aligned}T(x = 0, t) = T(x = L, t) &= 0^{\circ}\text{C} \\T(x, t = 0) &= 100^{\circ}\text{C}\end{aligned}\tag{29}$$

Datos para el problema

La conductividad térmica, el calor específico y la densidad del aluminio son, respectivamente:

$$\begin{aligned}K &= 237 \text{ W m}^{-1} \text{ K}^{-1} \\C &= 900 \text{ J kg}^{-1} \text{ K}^{-1} \\ \rho &= 2700 \text{ kg m}^{-3}\end{aligned}\tag{30}$$

Importando las librerías

Código (3): Llamada a las librerías

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import
   Axes3D
4 from matplotlib import cm
```

Declarando valores

Se declaran las constantes físicas del aluminio, así como dos los valores: Nx para los puntos en la barra y Nt para la evolución temporal, así como el incremento espacial $\Delta x = 0.01414$ y el incremento temporal $\Delta t = 1$.

Declarando valores I

Código (4): Declarando valores para el algoritmo

```
1 kappa = 210.  
2 sph = 900.  
3 rho = 2700.  
4  
5 cons = kappa / (sph*rho) *Dt / (Dx*Dx)  
6  
7 Nx = 101  
8 Nt = 3000  
9 Dx = 0.01414  
10 Dt = 1.
```

Arreglos para los cálculos I

Se generan los arreglos para almacenar los valores nuevos de temperatura y de tiempo:

Código (5): Arreglos para los cálculos de temperatura

```
1 T = np.zeros((Nx, 2), dtype=float)
2 Tpl = np.zeros((Nx, 31), dtype=
    float)
```


Condiciones iniciales y de frontera I

Se establecen las condiciones iniciales y de frontera:

Código (6): Se definen las condiciones iniciales y de frontera

```
1 for ix in range(1, Nx-1):  
2     T[ix, 0] = 100.0  
3  
4 T[0, 0] = 0.0  
5 T[0, 1] = 0.0  
6  
7 T[Nx-1, 0] = 0.0  
8 T[Nx-1, 1] = 0.0
```

Ciclo de iteración I

Código (7): Ciclo de iteración para calcular los nuevos valores de temperatura

```
1 m = 1
2
3 for t in range(1, Nt):
4     for ix in range(1, Nx-1):
5         T[ix, 1] = T[ix, 0] + cons * (T[
ix+1, 0] + T[ix-1, 0] - 2.*T[ix, 0])
6         if t % 100 == 0 or t == 1:
7             for ix in range(1, Nx-1, 2):
8                 Tpl[ix, m] = T[ix, 1]
9             print (m)
10            m = m+1
11    for ix in range(1, Nx-1):
```

Ciclo de iteración II

$$12 \quad \boxed{T[ix, 0] = T[ix, 1]}$$

Creando la malla para la solución I

Se crea la malla espacio-temporal y definimos una función que nos devuelve la temperatura

Código (8): Definición de la malla

```
1 x = list(range(1, Nx-1, 2))
2 y = list(range(1, 30))
3
4 X, Y = np.meshgrid(x, y)
5
6 def functz(Tpl):
7     z = Tpl[X, Y]
8     return z
9
```

Creando la malla para la solución II

```
10 | Z = functz(Tpl)
```

Graficando los resultados I

Hacemos la rutina para presentar los datos en una gráfica 3D con **matplotlib**

Código (9): Rutina para graficar los resultados

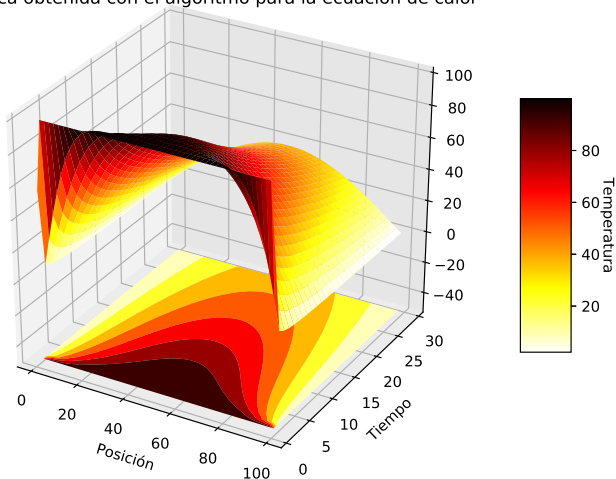
```
1 fig = plt.figure()
2 ax = Axes3D(fig)
3
4 surf = ax.plot_surface(X,Y,Z,rstride
    =1,cstride=1,linewidth=0.5,cmap=cm
    .hot_r)
5
6 ax.set_zlim(-50, 100)
7
```

Graficando los resultados II

```
8 cset = ax.contourf(X,Y,Z,zdir='z',  
    offset=-50,cmap=cm.hot_r)  
9 cbar = fig.colorbar(surf,shrink=0.5,  
    aspect=5)  
10 cbar.ax.set_ylabel('Temperatura',  
    rotation=270,labelpad=10)  
11 ax.set_xlabel('Posicion')  
12 ax.set_ylabel('Tiempo')  
13 plt.title('Grafica obtenida con el  
    algoritmo para la ecuacion de  
    calor')  
14 plt.show()
```

Gráfica obtenida

Gráfica obtenida con el algoritmo para la ecuación de calor



Problemas a resolver

Se hacen algunos cambios en el planteamiento del problema, pero el algoritmo que hay que usar es el mismo, hay que resolver los siguientes casos:

- 1 Distribución inicial de temperatura de forma senoidal: $\sin(\pi x / L)$

Problemas a resolver

Se hacen algunos cambios en el planteamiento del problema, pero el algoritmo que hay que usar es el mismo, hay que resolver los siguientes casos:

- 1 Distribución inicial de temperatura de forma senoidal: $\sin(\pi x / L)$
- 2 Dos barras en contacto cada una con diferente temperatura.

Problemas a resolver

Se hacen algunos cambios en el planteamiento del problema, pero el algoritmo que hay que usar es el mismo, hay que resolver los siguientes casos:

- 1 Distribución inicial de temperatura de forma senoidal: $\sin(\pi x / L)$
- 2 Dos barras en contacto cada una con diferente temperatura.
- 3 Modificación de la ecuación de calor para incluir un término y obtener la ley de enfriamiento de Newton.

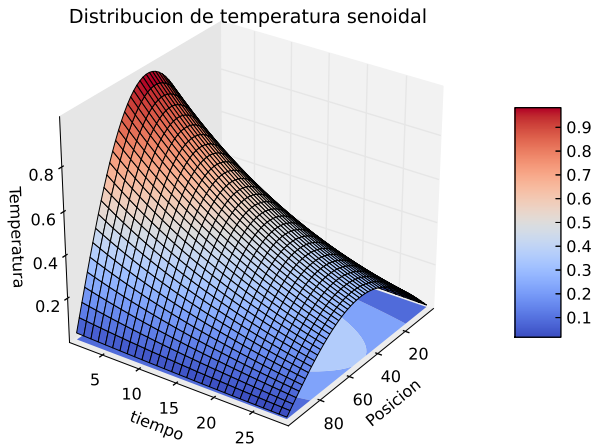
Problema 1

Distribución inicial de temperatura de forma senoidal: $\sin(\pi x / L)$

Utiliza las mismas constantes que en el primer ejemplo. Puedes comparar los resultados con la solución analítica:

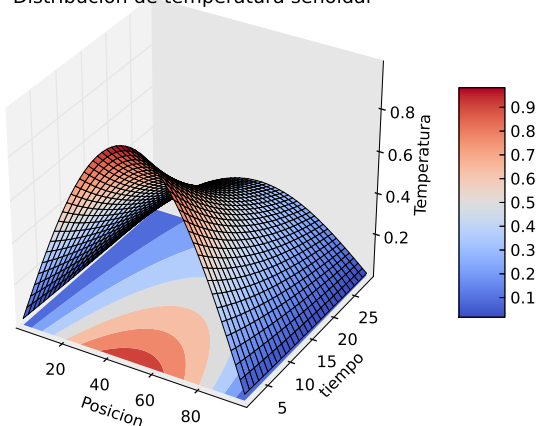
$$T(x, t) = \sin\left(\frac{\pi x}{L}\right) \exp\left(-\frac{\pi^2 R t}{L^2}\right), \quad R = \frac{K}{C \rho}$$

Solución del problema 1



Solución del problema 1 -rotada-

Distribucion de temperatura senoidal



Problema 2

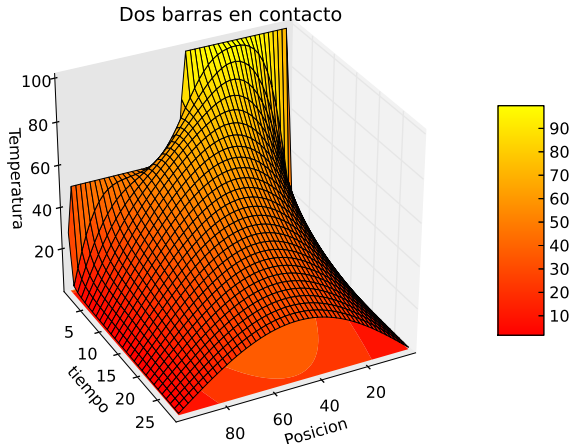
Tenemos dos barras del mismo material en contacto, cada una de ellas a diferente temperatura.

Problema 2

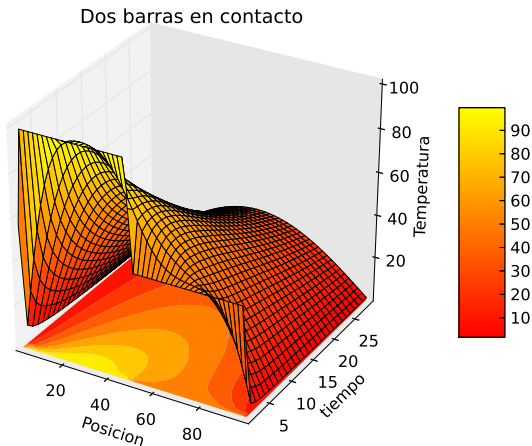
Las barras miden 50 cm de longitud. Una de ellas se mantiene a una temperatura de 100°C y la otra a 50°C , se ponen en contacto a lo largo de su eje, un extremo de cada barra se mantiene a 0°C .

Determina cómo varía la temperatura con respecto a la posición y al tiempo.

Dos barras con diferente temperatura en contacto



Dos barras en contacto -rotada-



Problema 3

Modificación de la ecuación de calor para incluir un término y obtener la ley de enfriamiento de Newton.

Problema 3

Imagina ahora que la barra que estaba aislada (el problema con el que comenzamos la clase), se deja en contacto con el ambiente que se encuentra a una temperatura T_e , tal que es diferente a la temperatura inicial de la barra.

Problema 3

La ley de enfriamiento de Newton nos dice que la razón de cambio de la temperatura debido a la radiación es:

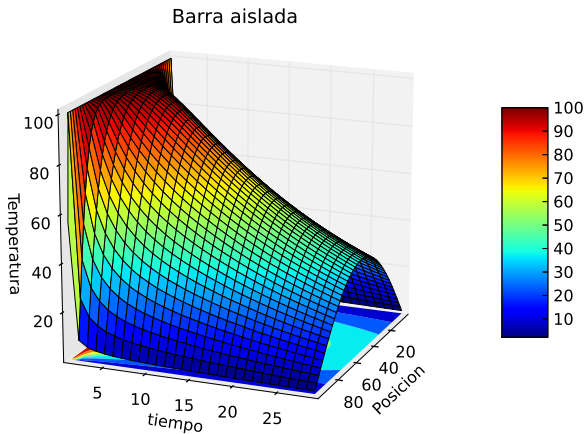
$$\frac{\partial T}{\partial t} = -h(T - T_e)$$

La ecuación de calor se modifica, quedando:

$$\frac{\partial T(x, t)}{\partial t} = \frac{K}{C \rho} \frac{\partial^2 T}{\partial x^2} - h T(x, t)$$

Ajusta el algoritmo y el programa para introducir el término de enfriamiento de Newton a lo largo de la barra. Compara el enfriamiento de esta barra con el ejemplo de la barra aislada.

Barra aislada



Ley de enfriamiento de Newton, con

$$T_e = 25^{\circ}\text{C}$$

Ley de enfriamiento de Newton

