Diferencias Finitas

Problemas estacionarios

Modelación computacional en las ciencias y las ingenierías como apoyo en el proceso enseñanza-aprendizaje (PAPIME-PE101019)

Instituto de Geofísica
Universidad Nacional Autónoma de México

Contenido

MODELO CONCEPTUAL

¿Qué es la transferencia de calor? [1]

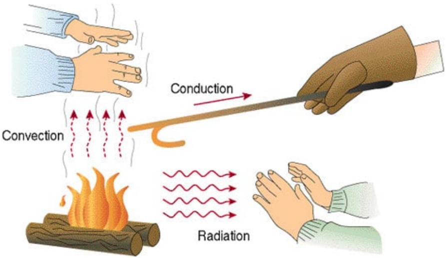
Cuando estudiamos termodinámica, aprendemos que la energía se puede transferir mediante interacciones de un sistema con el ambiente que lo rodea.

Estas interacciones se llaman *trabajo* y *calor*.

Sin embargo, en termodinámica se estudian los estados finales de los procesos, durante los que ocurren las interacciones, pero no se provee información de la naturaleza de las interacciones o de la velocidad a la que ocurren.

¿Qué es la transferencia de calor? [1]

Calor: forma de la energía que se puede transferir de un sistema a otro como resultado de la diferencia en la temperatura.
Transferencia de calor: determinación de las razones de esa transferencia de energía.



Conducción gradiente de temperaturas en un medio estacionario.

Convección ocurre entre una superficie y un fluido en movimiento cuando están a diferentes temperaturas.

Radiación calor emitido en forma de ondas electromagnéticas.

Conducción de calor

Conducción de calor [1]

Es posible cuantificar los procesos de transferencia de calor en términos de *ecuaciones de cambio*, que se pueden usar para calcular la cantidad de energía transferida por unidad de tiempo.

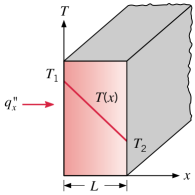
Dada una distribución de temperaturas $T(x)$, la **Ley de Fourier** para la conducción de calor es:

$$\boxed{q''_x = -\kappa \frac{dT}{dx}}$$

q''_x $(\mathrm{W/m^2})$ representa la velocidad con que se transfiere el calor en dirección x por unidad de área perpendicular a la dirección de transferencia. También se conoce como *flujo de calor* o *transferencia de calor por unidad de área*.

κ $(\mathrm{W/m{\cdot}K})$ es una propiedad del material que representa a la **conductividad térmica**.

Conducción de calor [1]



En la figura observamos la representación de un material con temperatura T_1 en una cara y T_2 en la otra. En este caso, se considera un estado estacionario (no depende del tiempo) y que la transferencia de calor es en una dimensión y en dirección x. Observamos que en este caso la $T(x)$ es una función lineal.

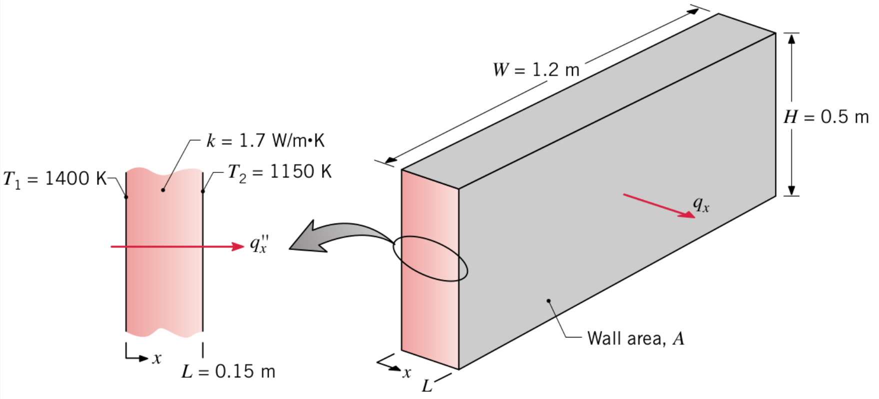
Entonces el gradiente de temperaturas se puede escribir como $dT/dx = (T_2 - T_1)/L$, por lo tanto, el flujo de calor sería:

$$q_x'' = -\kappa \frac{T_2 - T_1}{L} = \kappa \frac{\Delta T}{L}$$

donde $\Delta T = T_1 - T_2$. Finalmente, la **transferencia de calor por conducción** q_x (W), a través de una pared plana de área A, es: $q_x = q_x'' \cdot A$.

Ejemplo: Horno industrial [1]

La pared de un horno industrial está construida de ladrillos de arcilla refractaria cuya conductividad térmica es de 1.7 W/m·K. En estado estacionario, durante su operación, la temperatura interna tiene una temperatura de 1400 K, mientras que la externa 1150 K. Si la pared es de 0.5 m × 1.2 m, ¿cuál es la razón de pérdida de calor (q_x) a través de la pared?



Ejemplo: Horno industrial [1]

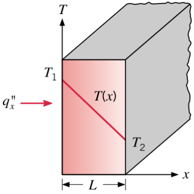
Tomando en cuenta la dirección del flujo de calor, es posible calcular q''_x usando la Ley de Fourier:

$$q''_x = \kappa \frac{\Delta T}{L} = 1.7\mathrm{W/m \cdot K} \times \frac{1400\mathrm{K} - 1150\mathrm{K}}{0.15\mathrm{m}} = 2833\mathrm{W/m}^2$$

Para conocer la pérdida de calor a través de la pared (transferencia de calor por conducción) multiplicamos q''_x por el área de la pared:

$$q_x = q''_x \cdot A = 2833\mathrm{W/m}^2 \times (0.5\mathrm{m} \times 1.2\mathrm{m}) = \boxed{1700\mathrm{W}}.$$

Supongamos ahora, que deseamos conocer la distribución de temperaturas al interior de la pared del horno.

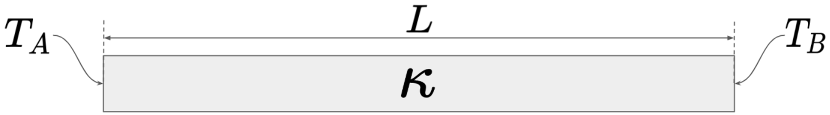


Para ello debemos plantear varias hipótesis y obtener un modelo matemático para $T(x)$.

La solución de ese modelo matemático, analítica y/o numérica, darán la distribución de temperaturas buscada.

Conducción de calor en 1D

Para fijar ideas, vamos a estudiar la transferencia de calor en un dominio como el que se muestra en la siguiente figura:



- Consideramos que no hay de flujo de calor en las paredes horizontales (son adiabáticas).
- Tenemos temperaturas T_A y T_B fijas en los extremos.
- No se consideran fuentes ni sumideros.
- κ representa la conductividad térmica.
- En este problema se desea calcular la temperatura T al interior del dominio.
- Como primera aproximación, el dominio se considera unidimensional.

MODELO MATEMÁTICO

Ecuación "general" de transferencia de calor (véase [5]):

$$\boxed{c_p\rho\frac{\partial T}{\partial t} + c_p\rho\frac{\partial}{\partial x_j}(u_jT) - \frac{\partial}{\partial x_j}\left(\kappa\frac{\partial T}{\partial x_j}\right) = S}$$

(índices repetidos se suman)

donde se define lo siguiente:

Símbolo		Unidades
	Parámetros físicos	
c_p	Capacidad calorífica específica.	[J / Kg ^{o}K]
ρ	Densidad.	[Kg / m^3]
κ	Conductividad térmica.	[W / m ^{o}K]
S	Ganancia (fuente) o pérdida (sumidero) de calor	[J/m^3 s]
$\alpha = \dfrac{\kappa}{c_p\rho}$	Difusividad térmica.	[m^2/s]
	Variables independientes	
x_j	Coordenadas cartesianas: $(x_1, x_2, x_3) \equiv (x, y, z)$.	[m]
t	Tiempo.	[s]
	Variables dependientes	
T	Temperatura.	[^{o}K]
u_j	Componentes de la velocidad: $(u_1, u_2, u_3) \equiv (u_x, u_y, u_z)$.	[m/s]

Dado que estamos interesados en la conducción de calor estacionaria eliminamos el término temporal y el término de advección de la ecuación general:

$$\cancel{c_p\rho\frac{\partial T}{\partial t}} + \cancel{c_p\rho\frac{\partial}{\partial x_j}(u_j T)} - \frac{\partial}{\partial x_j}\left(\kappa\frac{\partial T}{\partial x_j}\right) = S$$

Entonces, el modelo matemático para este problema en 1D y con $\kappa = constante$ es:

$$\boxed{\begin{aligned} -\kappa\frac{d^2T}{dx^2} &= S \\ T(x=0) &= T_A \\ T(x=L) &= T_B \end{aligned}} \qquad (1)$$

Obsérvese que se tienen condiciones de tipo Dirichlet: la variable dependiente, T, está dada en las fronteras. Estas condiciones también se conocen como de primer tipo.

MODELO NUMÉRICO

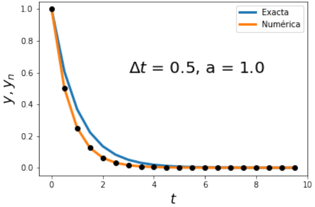
Discretización Proceso de transferir funciones continuas, modelos, variables y ecuaciones a sus contrapartes *discretas*.

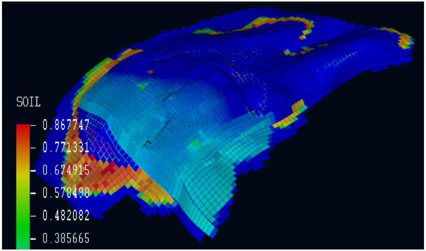
- La discretización es el primer paso en un modelo numérico cuyo objetivo es generar las contrapartes *discretas* de manera adecuada, para que posteriormente sean evaluadas numéricamente mediante algoritmos que se implementan en una computadora.

- La RAE define **Discreto**: 5. *adj. Mat.* Dicho de una magnitud: Que toma valores distintos y separados. Por ejemplo: *la sucesión de los números enteros es discreta, pero la temperatura no.*

Por ejemplo

En la gráfica de la derecha, la línea azul representa una función continua, mientras que los puntos negros, conectados con una línea naranja, representan una discretización que intenta aproximar a la función.



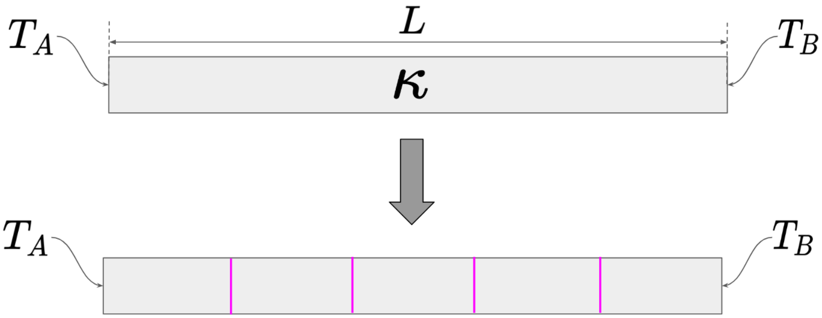


En la figura de la izquierda, se observa la discretización de un yacimiento petrolero en varias celdas o elementos.

Discretización del dominio

Discretización del dominio

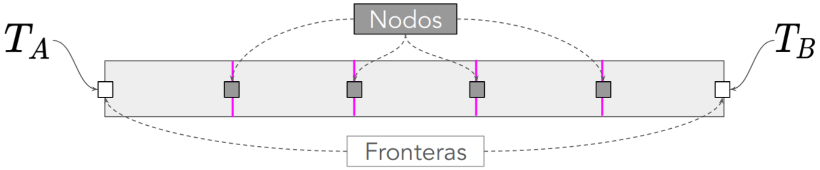
El primer paso en nuestro modelo numérico es discretizar el dominio:



Como se observa en la figura, el dominio se ha partido en varios *subdominios*.

Discretización del dominio

Una vez hecha la partición del dominio, es importante identificar los lugares del dominio donde se va a calcular la temperatura.



En la figura, se han identificado los *nodos* en el interior del dominio (cuadros grises).

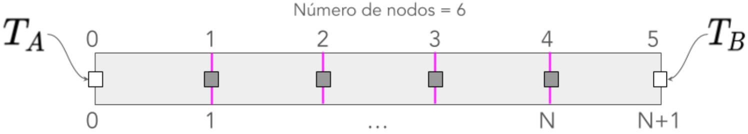
- Es en estos nodos donde se hará el cálculo.

También se identifican dos nodos especiales, los cuales están en los extremos del dominio (cuadros blancos).

- En esos nodos se imponen las condiciones de frontera.

Discretización del dominio

El siguiente paso es numerar los nodos:



Observaciones:

- La numeración la comenzamos en 0.
- Los nodos donde se va a calcular la temperatura T van de 1 a $N = 4$. Se dice que se tienen 4 grados de libertad, es decir, 4 incógnitas que se deben calcular.
- Las fronteras están identificadas en: $i = 0$ e $i = 5 (= N + 1)$.
- La discretización del domino genera lo que se conoce como *malla del dominio*, la cual define las coordenadas de los nodos y en varios casos también conectividades.

Discretización de las ecuaciones

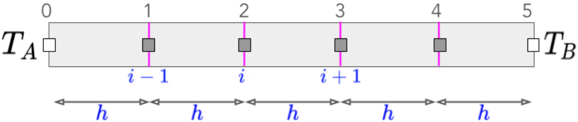
Discretización de las ecuaciones

Recordemos que el modelo matemático consta de la siguiente ecuación:

$$-\kappa \frac{d^2T}{dx^2} = S$$

Discretizamos la ecuación usando diferencias finitas de segundo orden:

- Consideramos un nodo i de la malla, junto con sus vecinos $i+1$ e $i-1$:



  Observe que todas las celdas son de la misma longitud h: la malla es *estructurada* y *uniforme*.

- La aproximación de la derivada se escribe como sigue:

$$\boxed{\left.\frac{d^2T}{dx^2}\right|_i = \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} + \mathcal{O}(h^2)} \tag{2}$$

Discretización de las ecuaciones

Ahora sustituimos la ecuación (2) en (1) para obtener:

$$-\kappa_i \left(\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} \right) = S_i \tag{3}$$

Esta ecuación representa la conducción de calor en el nodo i, y hace uso de sus vecinos $i+1$ e $i-1$. En este ecuación, tanto κ_i como S_i representan la conductividad térmica y la fuente en el nodo i,

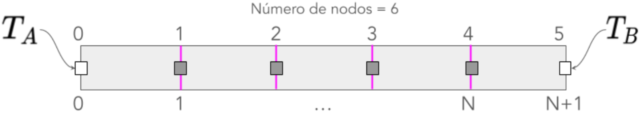
Podemos reescribir la ecuación (3) como sigue:

$$\boxed{-r_i T_{i-1} + 2r_i T_i - r_i T_{i+1} = S_i} \tag{4}$$

donde $r_i = \dfrac{\kappa_i}{h^2}$.

Discretización de las ecuaciones

En el caso que estamos estudiando, necesitamos calcular la temperatura en los nodos $i = 1, 2, 3, 4$, que son los nodos internos.



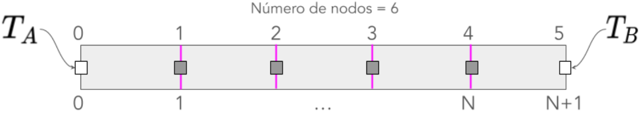
Debemos escribir una ecuación para cada uno de esos nodos:

$$
\begin{aligned}
-r_1T_0 + 2r_1T_1 - r_1T_2 &= S_1 \\
-r_2T_1 + 2r_2T_2 - r_2T_3 &= S_2 \\
-r_3T_2 + 2r_3T_3 - r_3T_4 &= S_3 \\
-r_4T_3 + 2r_4T_4 - r_4T_5 &= S_4
\end{aligned}
$$

Este es un sistema lineal que debemos resolver para obtener la temperatura en cada uno de los nodos internos.

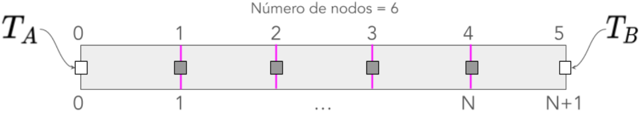
Discretización de las ecuaciones

Para incorporar las **condiciones de frontera** debemos modificar las ecuaciones para los nodos en los extremos: $i = 1$ e $i = 4$



Discretización de las ecuaciones

Para incorporar las **condiciones de frontera** debemos modificar las ecuaciones para los nodos en los extremos: $i = 1$ e $i = 4$



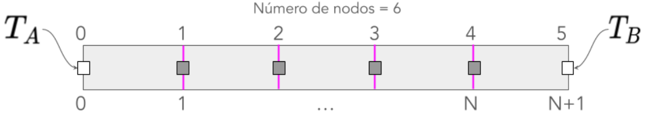
Para $i = 1$ tenemos $T_0 = T_A$

$$
\begin{aligned}
-r_1 T_0 + 2r_1 T_1 - r_1 T_2 &= S_1 \\
-r_1 \boxed{T_A} + 2r_1 T_1 - r_1 T_2 &= S_1
\end{aligned}
$$

$$
\boxed{2r_1 T_1 - r_1 T_2 = S_1 + r_1 T_A}
$$

Discretización de las ecuaciones

Para incorporar las **condiciones de frontera** debemos modificar las ecuaciones para los nodos en los extremos: $i = 1$ e $i = 4$



Para $i = 1$ tenemos $T_0 = T_A$

$$
\begin{aligned}
-r_1 T_0 + 2r_1 T_1 - r_1 T_2 &= S_1 \\
-r_1 \boxed{T_A} + 2r_1 T_1 - r_1 T_2 &= S_1
\end{aligned}
$$

$$\boxed{2r_1 T_1 - r_1 T_2 = S_1 + r_1 T_A}$$

Para $i = N = 4$ tenemos $T_{N+1} = T_B$

$$
\begin{aligned}
-r_N T_{N-1} + 2r_N T_N - r_N T_{N+1} &= S_N \\
-r_N T_{N-1} + 2r_N T_N - r_N \boxed{T_B} &= S_N
\end{aligned}
$$

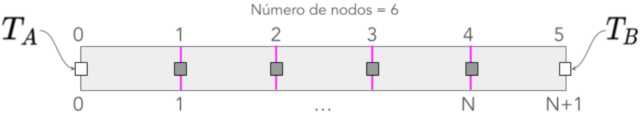
$$\boxed{-r_4 T_3 + 2r_4 T_4 = S_4 + r_4 T_B}$$

Estas dos últimas ecuaciones incorporan las condiciones de frontera de este problema.

Discretización de las ecuaciones

El sistema de ecuaciones, con las condiciones de frontera incorporadas, para los nodos $i = 1, 2, 3, 4$, se escribe como sigue:

$$
\begin{aligned}
2r_1T_1 - r_1T_2 &= S_1 + r_1T_A \\
-r_2T_1 + 2r_2T_2 - r_2T_3 &= S_2 \\
-r_3T_2 + 2r_3T_3 - r_3T_4 &= S_3 \\
-r_4T_3 + 2r_4T_4 &= S_4 + r_4T_B
\end{aligned}
$$



Discretización de las ecuaciones

Podemos ahora escribir, en forma de un sistema lineal, las ecuaciones del problema a resolver, para cualquier N y para $\kappa = constante$:

$$\underbrace{\begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & -1 & 2 \end{bmatrix}}_{\underline{\underline{\mathbf{A}}}_{N\times N}} \underbrace{\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{N-1} \\ T_N \end{bmatrix}}_{\mathbf{T}_N} = \underbrace{\frac{1}{r}\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_{N-1} \\ S_N \end{bmatrix} + \begin{bmatrix} T_A \\ 0 \\ 0 \\ \vdots \\ 0 \\ T_B \end{bmatrix}}_{\mathbf{b}_N}$$

donde $r = \dfrac{\kappa}{h^2}$

MODELO COMPUTACIONAL

Modelo Computacional

El objetivo del modelo computacional es escribir programas de cómputo que permitan obtener las soluciones numéricas del problema descrito en el Modelo Conceptual.

- En este caso se intenta obtener la temperatura T en los nodos interiores de la malla.
- Para ello, se debe encontrar la solución a un sistema del tipo:

$$\underline{\underline{\mathbf{A}}} \cdot \mathbf{T} = \mathbf{b}$$

  donde $\underline{\underline{\mathbf{A}}}$ es una matriz de $N \times N$ y $\mathbf{b}$ es un vector de tamaño N, ambos con coeficientes conocidos.
- $\mathbf{T}$ es un vector que almacenará la solución en los nodos internos.

Existen varias maneras de realizar esta implementación. A continuación mostraremos una muy sencilla.

Ejemplo 1: Modelo computacional sencillo

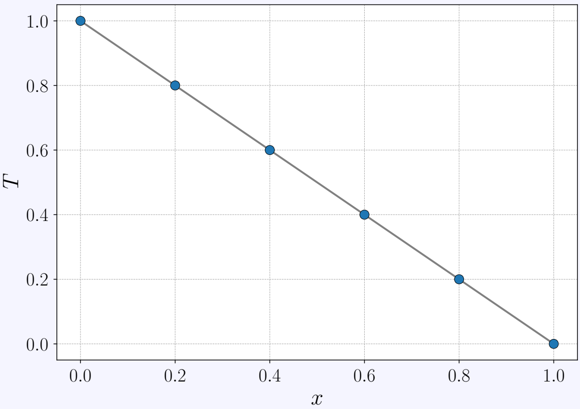
```python
import numpy as np
import matplotlib.pyplot as plt
def buildMatrix(N):
    # Matriz de ceros
    A = np.zeros((N,N))

    # Primer renglón
    A[0,0] = 2
    A[0,1] = -1
    # Renglones interiores
    for i in range(1,N-1):
        A[i,i] = 2
        A[i,i+1] = -1
        A[i,i-1] = -1
    # Último renglón
    A[N-1,N-2] = -1
    A[N-1,N-1] = 2

    return A

# Parámetros físicos
L = 1.0
TA = 1
TB = 0
k = 1.0
S = 0.0
# Parámetros numéricos
N = 4
h = L / (N+1)
r = k / h**2
```

```python
# Arreglo para almacenar la solución
T = np.zeros(N+2)
T[0]  = TA  # Frontera izquierda
T[-1] = TB  # Frontera derecha

# Lado derecho del sistema
b = np.zeros(N)
b[:] = S / r # Fuente o sumidero
b[0]  += T[0]   # Condición de frontera
b[-1] += T[-1]  # Condición de frontera

# Construcción de la matriz
A = buildMatrix(N)

# Solución del sistema lineal
T[1:N+1] = np.linalg.solve(A,b)

# Impresión y graficación de la solución
print(T)
x = np.linspace(0, L, N+2)
plt.figure(figsize=(10,7))
plt.plot(x, T, c='grey', lw=2.0)
plt.scatter(x, T, edgecolor='k', zorder= 10)
plt.xlabel('$x$')
plt.ylabel('$T$')
plt.grid()
plt.show()
```

Ejemplo 1: Modelo computacional sencillo

Salida del código anterior:

```
T =  [1.  0.8 0.6 0.4 0.2 0. ]
```



Contenido

Ejercicio 2.

1. Abrir la notebook `E02_ConduccionEst.ipynb`, del repositorio `Mixbaal`, y escribir el código del ejemplo 1 como sigue:

   A. Celda 1: La función `buildMatrix(N)`.
   B. Celda 2: Los parámetros físicos y numéricos.
   C. Celda 3: El código que resuelve el problema y grafica la solución.

   Ejecutar las tres celdas en orden y reproducir la salida del ejemplo 1.

2. El problema definido en (1) tiene la siguiente solución analítica:

$$T(x) = \boxed{\left(\frac{T_B - T_A}{L} + \frac{S}{2\kappa}(L-x)\right)x + T_A}$$

   En la misma notebook del punto 1, agregar

   A. El código para calcular la solución analítica:

```
def solExact(x, TA, TB, k, L, S):
    """
    Cálculo de la solución exacta.
    """
    return ((TB-TA)/L+(S/(2*k))*(L-x))*x+TA
```

Ejercicio 2.

- Agregar la siguiente función para el cálculo de la solución numérica:

```
def solNum(L, N, k, S, A, b, T, etiqueta):
    h = L / (N+1)
    r = k / h**2

    # Lado derecho del sistema
    b = np.zeros(N)
    b[:] = S / r # Fuente o sumidero
    b[0]  += T[0]   # Condición de frontera
    b[-1] += T[-1]  # Condición de frontera

    # Solución del sistema lineal
    T[1:N+1] = np.linalg.solve(A,b)

    # Impresión y graficación de la solución
    x = np.linspace(0, L, N+2)

    # Construcción de la etiqueta de cada gráfica
    if etiqueta == 'L':
        etiqueta = '$L$ = {:3.2f}'.format(L)
    elif etiqueta == 'k':
        etiqueta = '$\kappa$ = {:3.2f}'.format(k)
    elif etiqueta == 'S':
        etiqueta = '$S$ = {:3.2f}'.format(S)

    # Se grafican los puntos de la solución
    plt.scatter(x, T, edgecolor='k', s=50, zorder= 10, label=etiqueta)
```

Ejercicio 2.

ⓒ Agregar la siguiente función para la graficación:

```
def plotSol(title, filename):
    plt.suptitle('Conducción estacionaria',fontsize=24,y=0.94,va='center_baseline')
    plt.title(title, fontsize=20, color='blue')
    plt.ylabel('$T$')
    plt.xlabel('$x$')
    plt.legend(loc='center left', bbox_to_anchor=(1, 0.5), fontsize=12)
    plt.grid()
    plt.savefig(filename)
    plt.show()
```

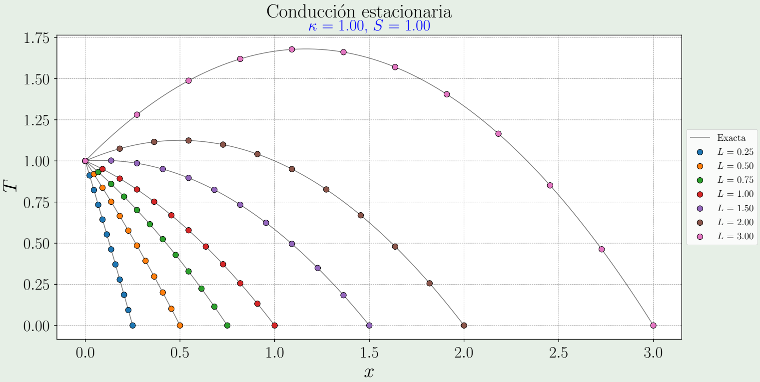
3. Agregar el siguiente código para variar la longitud del dominio (L):

```
# Parámetros físicos
l = [0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 3.0]
TA = 1.0
TB = 0.0
k = 1.0
S = 1.0
# Parámetros numéricos
N = 10

# Arreglo para almacenar la solución
T = np.zeros(N+2)
T[0]  = TA  # Frontera izquierda
T[-1] = TB  # Frontera derecha
```

```
# Construcción de la matriz
A = buildMatrix(N)

for L in l:
    solNumerica(L, N, k, S, A, b, T, 'L')
    xe = np.linspace(0,L,100)
    plt.plot(xe,
             solExacta(xe, TA, TB, k, L, S),
             'k-', lw=1.0, alpha=0.5)

plotSol('$\kappa$ = {:3.2f}'.format(k) +
        '$S$ = {:3.2f}'.format(S),
        'nombre_archivo.pdf')
```

Ejercicio 2.

Al ejecutar el código anterior deberías obtener algo similar a lo siguiente:

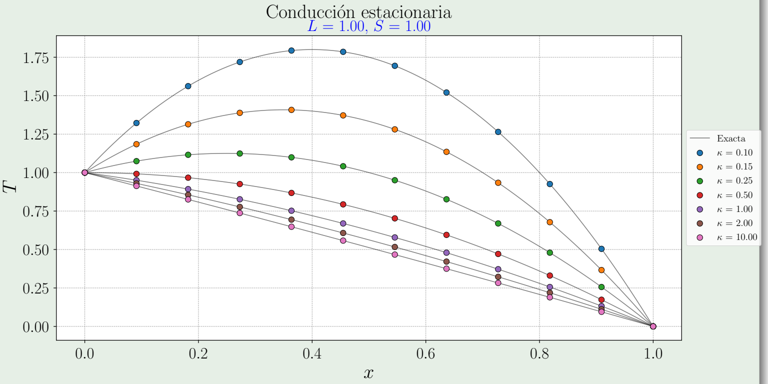


Explique el comportamiento de este resultado en términos matemáticos y físicos de la conducción de calor.

Ejercicio 2.

4. Modifica el código del punto 3 para hacer lo siguiente:

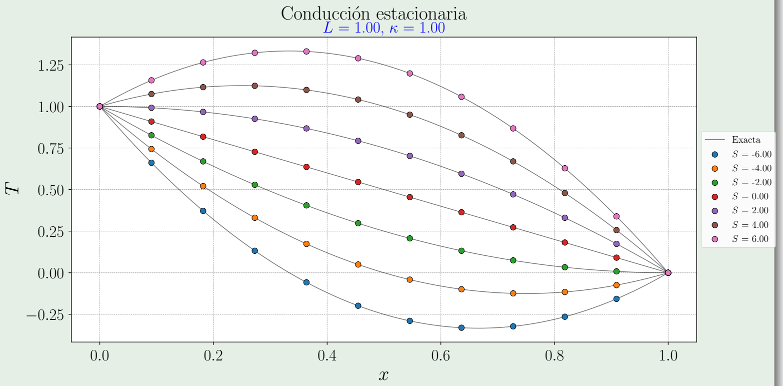
   A. $L = 1.0$, $S = 1.0$ y $\kappa =$ `[0.1, 0.15, 0.25, 0.5, 1.0, 2.0, 10]`. Reproduce la siguiente gráfica:



Ejercicio 2.

4. cont.

   B. $L = 1.0$, $\kappa = 1.0$ y $S =$ `[-6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 6.0]`. Reproduce la siguiente gráfica:



Ejercicio 2.

Nota: para obtener el mismo estilo de las gráficas de esta presentación, deberás usar el siguiente código y ponerlo en una celda al principio del notebook (no olvides ejecutarla).

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros para el estilo de las gráficas
plt.style.use('seaborn-paper')
params = {'figure.figsize' : (14,7),
          'text.usetex'    : True,
          'xtick.labelsize': 20,
          'ytick.labelsize': 20,
          'axes.labelsize' : 24,
          'axes.titlesize' : 24,
          'legend.fontsize': 24,
          'lines.linewidth': 3,
          'lines.markersize': 10,
          'grid.color'     : 'darkgray',
          'grid.linewidth' : 0.5,
          'grid.linestyle' : '--',
          'font.family': 'DejaVu Serif',
}
plt.rcParams.update(params)
```

OJO: la línea que contiene el código: `'text.usetex' : True` requiere de que tengas instalado LaTeXen tu computadora. Si no lo tienes, debes comentar esta línea (poner un # al principio de la línea).

Contenido

[1] Bergman, T.L. and Incropera, F.P. and DeWitt, D.P. and Lavine, A.S.,
Fundamentals of Heat and Mass Transfer,
Wiley, **2011**.

[2] R.J. Leveque,
Finite Difference Method for Ordinary and Partial Differential Equations: Steady State and Time-Dependent Problems ,
Society for Industrial and Applied Mathematics (SIAM), Philadelphia, **2007**.

[3] Y. Saad
Iterative Methods for Sparse Linear Systems.
PWS/ITP 1996.
Online: http://www-users.cs.umn.edu/~saad/books.html, **2000**

[4] Richard Burden and J. Douglas Faires
Numerical Analysis
Cengage Learning; 9 edition (August 9, **2010**)

[5] I. Herrera & G. F. Pinder,
Mathematical Modeling in Science and Engineering: An Axiomatic Approach,
John Wiley **2012**.

Contenido

Créditos

Dr. Luis M. de la Cruz Salas

Departamento de Recursos Naturales

Instituto de Geofísica

Universidad Nacional Autónoma de México

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE101019