

Física Numérica

Julio César Avila Torreblanca

14 de octubre del 2021

Tarea 5

1. Estudiando la evolución de la temperatura en una barra

- (a) Considere una barra cilíndrica de aluminio de longitud $L = 1$ m y un grosor ω colocada a lo largo del eje x . Dicha barra se encuentra aislada térmicamente a lo largo de su longitud, pero no en sus extremos. Inicialmente la barra se encuentra a una temperatura uniforme de $T_0 = 100$ °C, y los extremos se encuentran en contacto con una barra de hielo a 0 °C. El calor fluye a través de los extremos no aislados. ¿Cuál es la ecuación que modela este proceso?, ¿Qué tipo de ecuación es?, ¿Puede resolverla analíticamente?, ¿Qué métodos de solución conoce para este tipo de ecuaciones?, ¿Hay condiciones de frontera asociadas?

Solución.

La ecuación que relaciona el flujo del calor con el gradiente de temperatura T a través del material, es la llamada *Ecuación de calor*:

$$\frac{\partial T(\vec{x}, t)}{\partial t} = \frac{K}{C\rho} \nabla^2 T(\vec{x}, t) \quad (1)$$

Donde K es la conductividad térmica del material, C el calor específico y ρ su densidad. Además la temperatura T es función del tiempo t y del vector espacial \vec{x} . La ecuación de calor (1) es una ecuación diferencial parcial parabólica, cuyas variables independientes son temporales y espaciales. Notemos que la temperatura en la barra cilíndrica es uniforme, y el flujo de calor también lo es. Debido a esta simetría, es posible considerar nuestro problema en una sola dimensión, es decir, considerar la barra cilíndrica como un alambre en el eje x cuyos extremos se encuentran en $x = 0$ y $x = L$. De esta forma, la ecuación que describe el problema se reduce a:

$$\frac{\partial T(x, t)}{\partial t} = \kappa \frac{\partial^2 T(x, t)}{\partial x^2} \quad (2)$$

Donde hemos definido la constante:

$$\kappa = \frac{K}{C\rho}$$

Notemos que la ecuación (2) es fácil de resolver analíticamente si se utiliza separación de variables. Es decir, proponer una solución de la forma:

$$T(x, t) = X(x)F(t)$$

Por otra parte, la barra inicialmente tiene una temperatura uniforme $T_0 = 100$ °C. Por lo que nuestra primera condición de frontera es:

$$T(x, 0) = 100C$$

Por otro lado, en cualquier instante de tiempo posterior, los extremos de la barra que están en contacto con el hielo permanecerán a $0^\circ C$. Así nuestra segunda y tercer condición de frontera resultan:

$$T(0, t) = T(L, t) = 0C$$

- (b) ¿Cómo podría resolver numéricamente el problema? ¿Tiene algún algoritmo que podría ayudarlo?

Solución.

Para poder resolver la ecuación (2) de forma numérica se requiere de un algoritmo que permita calcular derivadas numéricamente. Nos apoyaremos del algoritmo de *Foward difference* y *Central difference* para calcular las derivadas parciales. Para la derivada respecto del tiempo, debido a que el fenómeno sucede para $t \geq 0$ usaremos aproximación por *Foward difference*. Es decir:

$$\frac{\partial T(x, t)}{\partial t} \approx \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t}$$

Por otro lado, para la derivada espacial tenemos mayor libertad ya que se conoce la temperatura a lo largo de todo el segmento. Aquí podemos usar *Central difference*. Aplicando este algoritmo dos veces, obtenemos la aproximación para la segunda derivada:

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

Sustituyendo estas aproximaciones en (2):

$$\frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} = \kappa \frac{T(x + \Delta x, t) + T(x - \Delta x, t) - 2T(x, t)}{(\Delta x)^2}$$

Reescribiendo:

$$T(x, t + \Delta t) = T(x, t) + \frac{\kappa \Delta t}{(\Delta x)^2} [T(x + \Delta x, t) + T(x - \Delta x, t) - 2T(x, t)] \quad (3)$$

Si definimos:

$$\eta = \frac{\kappa \Delta t}{(\Delta x)^2} = \frac{K \Delta t}{C \rho (\Delta x)^2}$$

Y discretizando a x, t de la forma:

$$x = i\Delta x \quad ; \quad t = j\Delta t$$

Además:

$$T(x, t) = T(i\Delta x, j\Delta t) = T_{ij}$$

Nuestra ecuación (3) se reescribe como:

$$T_{i,j+1} = T_{ij} + \eta [T_{i+1,j} + T_{i-1,j} - 2T_{ij}] \quad (4)$$

Esta relación de recurrencia permite conocer la temperatura en un instante de tiempo posterior $j+1$ para un punto espacial fijo i , a partir de conocer la temperatura en los puntos espaciales vecinos $i \pm 1$ en el instante de tiempo j y la temperatura para ese punto i en ese mismo tiempo j . Debido a la condición de frontera:

$$T(x, 0) = 100^{\circ}\text{C}, \quad \forall x$$

Es posible determinar la temperatura en cualquier instante de tiempo. Para generar un algoritmo que permita resolver esta ecuación diferencial de forma numérica, utilizaremos el programa visto en clase. El código es el siguiente:

```
In [ ]: from numpy import *; import matplotlib.pyplot as p
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
"""
    Nx = intervalo para la variable x [0,Nx]
    Nt = valores que consideraremos para t, pero más adelante
        tomaremos saltos en t para hacer más preciso el cálculo.
    Dx = valor de delta x
    Dt = valor de delta t
    K = conductividad térmica
    C = capacidad calorífica
    rho = densidad del material
    T = es una lista auxiliar que permitirá hacer cálculos para
        obtener las temperaturas a un tiempo posterior
    Tpl = es arreglo con las temperaturas para cada punto del espacio.
"""

Nx=100; Nt=20000; Dx=0.03; Dt=5.
K = 205.; C = 910.; rho = 2700.
T=zeros((Nx,2),float); Tpl=zeros((Nx,100),float)

print(" Trabajando")
for ix in range(1,Nx-1):
    T[ix,0]=100.0; # T condicion inicial

T[0,0]=0.0; T[0,1]=0. # Primero y ultimo T = 0
T[Nx-1,0]=0.0; T[Nx-1,1]=0.0

cons=K/(C*rho)*Dt/(Dx*Dx);

print(cons)
m=1 # Contador
for t in range(1,Nt):
    for ix in range(1,Nx-1):
        T[ix,1] = T[ix,0]+cons*(T[ix+1,0]+T[ix-1,0]-2.0*T[ix,0])
        if t%300==0 or t==1:
            for ix in range(1,Nx-1,2): Tpl[ix,m]=T[ix,1]
            print(m)
            m=m+1
        for ix in range(1,Nx-1): T[ix,0]=T[ix,1]

x=list(range(1,Nx-1,2))
y=list(range(1,30)) #modificar para el tiempo
X,Y=p.meshgrid(x,y)

def functz(Tpl):
    z=Tpl[X,Y]
    return z

Z=functz(Tpl)
fig=p.figure() # Crea la figura
ax=Axes3D(fig)
ax.plot_wireframe(X,Y,Z,color='b')
#ax.plot_surface(X, Y, Z, cmap=cm.coolwarm)
ax.set_xlabel('Posicion $x$')
ax.set_ylabel('tiempo $t$')
ax.set_zlabel('Temperatura $T$')
p.show()
print("Game over")
```

Figura 1: Algoritmo para resolver numéricamente la ecuación de Calor.

Donde se ha definido a $\Delta x = 0.03$ y $\Delta t = 5.0$. Además, se han usado los siguientes valores para el aluminio:

$$K = 205 \text{ W/(m} \cdot \text{K)}, \quad C = 910 \text{ J/(kg K)}, \quad \rho = 2700 \text{ kg/m}^3 \quad (5)$$

- (c) Ya que tenga un programa que le ayude a resolver numéricamente el problema, varíe los pasos en el tiempo y en el espacio en sus cálculos para obtener soluciones que sean estables y varíen suavemente tanto en tiempo como en espacio.

Solución.

En el algoritmo usado se tuvo que variar las variables N_x , N_t , Δx y Δt , además de ajustar el tamaño de los arreglos T_{pl} y y de acuerdo a los valores ingresados. Los valores encontrados para generar una solución estable y suave en todo el espacio fueron los siguientes:

$$N_x = 100, \quad N_t = 20000, \quad \Delta x = 0.03, \quad \Delta t = 5.0 \quad (6)$$

En la figura (1) se aprecia la asignación de estos valores al inicio del código. Con esto, se logró obtener un gráfico 3D para analizar el cambio de temperatura a lo largo de toda la barra en distintos instantes de tiempo. El gráfico generado fue el siguiente:

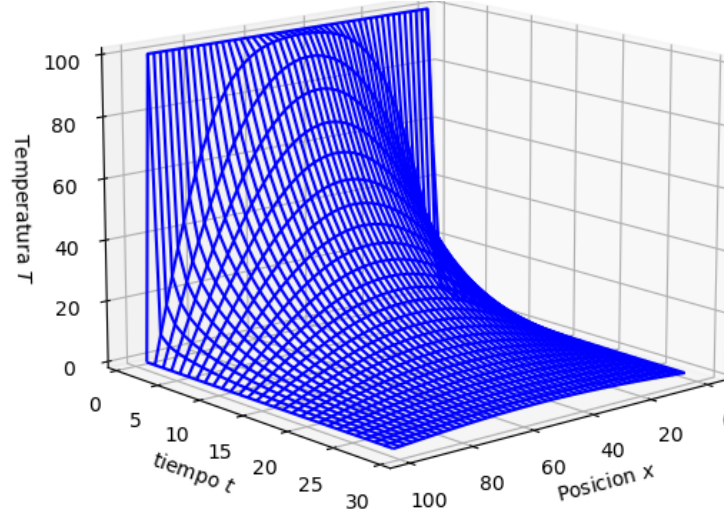


Figura 2: Temperatura en lo largo de barra de Aluminio a distintos tiempos.

Se puede apreciar que en $t = 0$ la temperatura a lo largo de la barra era $T = 100^\circ\text{C}$. Después de ello, fue disminuyendo suavemente hasta que alcanzar el equilibrio térmico en toda la barra.

- (d) Pruebe qué pasa cuando la condición de estabilidad de Neumann/Courant no se satisface.

Solución.

La condición de Neumann/Courant para este problema es la siguiente:

$$\eta = \frac{K\Delta t}{C\rho(\Delta x)^2} < \frac{1}{2} \quad (7)$$

Sustituyendo los valores considerados en (5) y (6), tenemos que el valor considerado en el inciso (c) fue:

$$\eta = \frac{(205)(5.0)}{(910)(2700)(0.03)^2} \approx 0.46 < \frac{1}{2}$$

Como se cumplió la condición de Neumann/Courant, la solución obtenida fue estable. Veamos que valor asignar a $\frac{\Delta t}{(\Delta x)^2}$ para no cumplir esta condición. Despejando de (7) encontramos que:

$$\frac{\Delta t}{(\Delta x)^2} < \frac{C\rho}{2K} \approx 5993 \quad (8)$$

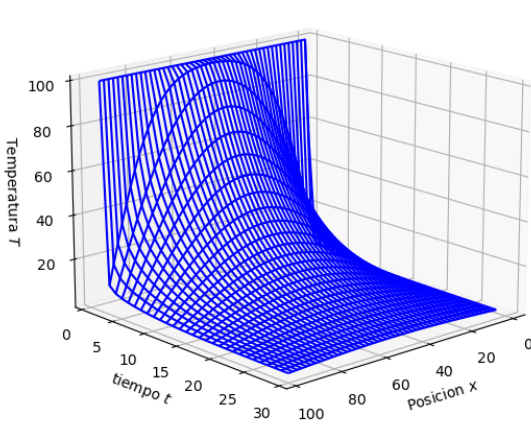
Así:

$$\Delta t < 5993(\Delta x)^2$$

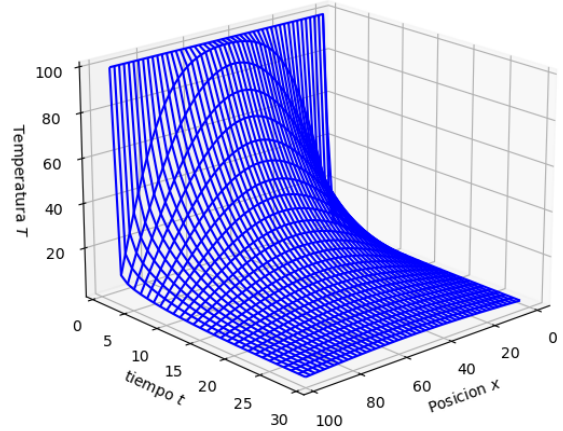
Si tomamos a $\Delta x = 0.03$ entonces:

$$\Delta t < 5.39$$

Así que, esta es la condición que debe cumplir Δt para tener estabilidad en la solución. A continuación presentamos varios gráficos donde se vario el valor de Δt para observar el comportamiento de nuestra solución:

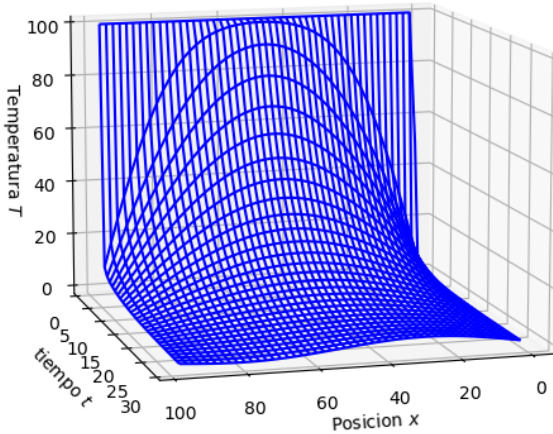


(a) Gráfico con $\Delta t = 5.30$

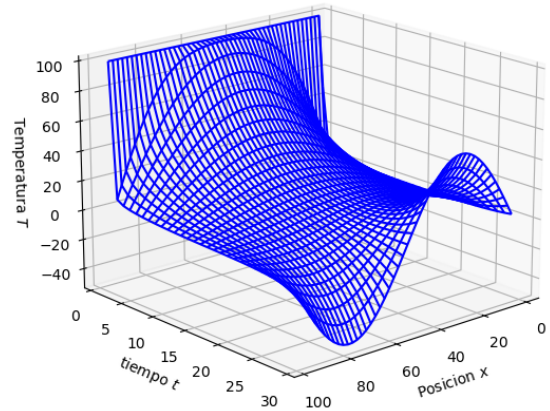


(b) Gráfico con $\Delta t = 5.39$

Figura 3



(a) Gráfico con $\Delta t = 5.40$



(b) Gráfico con $\Delta t = 5.401$

Figura 4

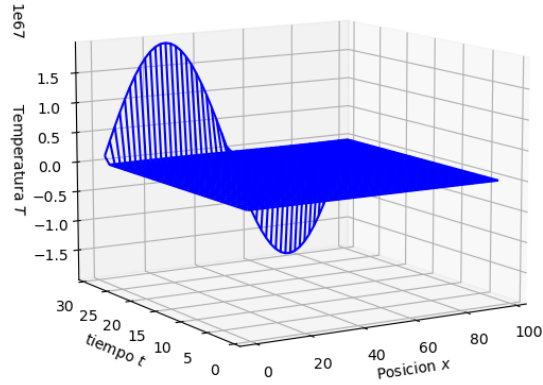


Figura 5: Gráfico con $\Delta t = 5.45$

Podemos ver que para $\Delta t = 5.30$ y $\Delta t = 5.39$ la solución sigue comportándose suavemente. Pero en $\Delta t = 5.40$ se comienza a ver una pequeña perturbación en $t = 30$ que no es de esperarse. Luego en $\Delta t = 5.401$, se aprecia mucho mejor que ya nuestra solución no es estable, pues la gráfica se comporta de forma anormal. Por último, cuando $\Delta t = 5.45$ la solución es completamente inestable. La gráfica resultante no dice nada respecto al problema, esto debido a que en los cálculos hechos por la computadora se acerca al overflow y comienza a tener errores con la interpretación de los resultados.

(e) Revise que sus resultados concuerden con las condiciones de frontera establecidas.

Solución.

Nuevamente analicemos la solución dada en el inciso (c) vista desde otro ángulo, donde hemos tomado los valores (5) y (6):

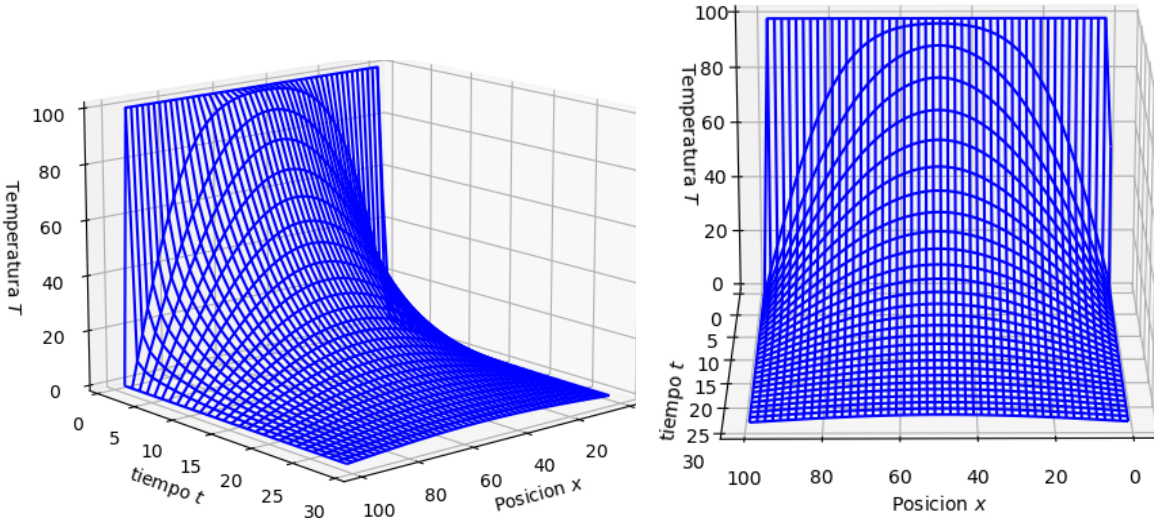


Figura 6: Solución numérica a la ecuación de calor para una barra de aluminio vista desde distintos ángulos.

Recordemos nuestras condiciones de frontera:

$$T(0, t) = 0 \quad (9)$$

$$T(L, t) = 0 \quad (10)$$

$$T(x, 0) = 100 \quad (11)$$

Analicemos los puntos en $x = 0$ y $x = L$ en la figura (6). Veamos que las líneas que conectan estos puntos siempre se mantienen en el nivel $T = 0$ para cualquier t . Por lo tanto, las condiciones (9) y (10) se cumplen. Por otro lado, cuando $t = 0$ y $x \in (0, L)$, vemos que la línea que une esos puntos se mantiene a nivel $T = 100$. De esa forma la tercera condición de frontera (11) también se cumple.

(f) ¿Alcanza su solución el equilibrio? ¿Es estable?

Solución.

De la figura (6) vemos que conforme t aumenta, la temperatura en la barra de aluminio disminuye. Lo cual tiene sentido, ya que en todo momento los extremos de la barra están en contacto con las barras de hielo a $T = 0$. De esa forma, al transcurrir el tiempo se alcanzará el equilibrio térmico. Por lo que existirá un t_f tal que $T(x, t_f) = 0$. En efecto, analizando nuestra solución vemos que para los últimos valores de t en la gráfica, la temperatura en la barra es muy baja.

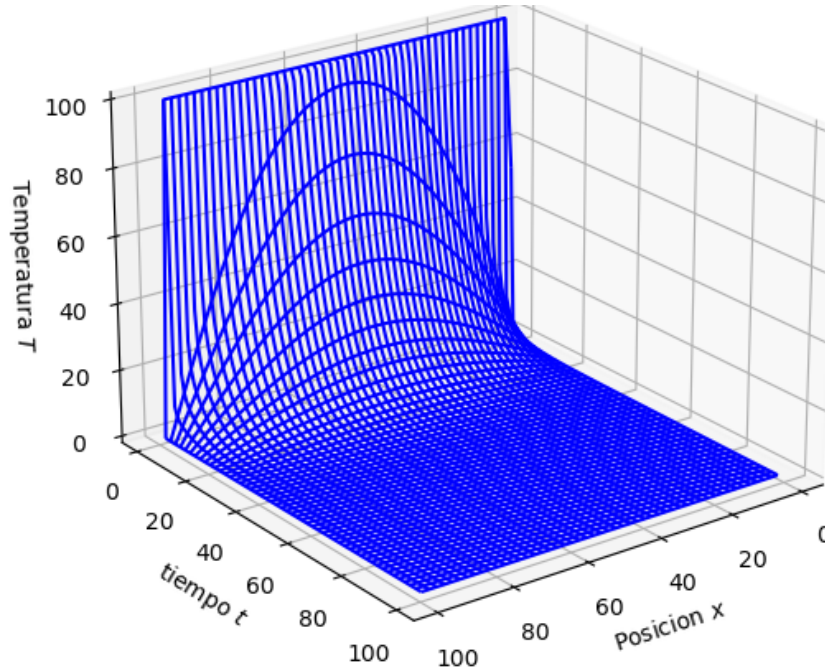


Figura 7: Solución estable. Se aprecia que llega un punto en el eje t donde se alcanza el equilibrio térmico

Así que si se proyecta más el eje t como en la gráfica anterior, la barra de aluminio estará en equilibrio a $T = 0$ y con ello nuestra solución es estable.

(g) Compare la solución numérica con la solución analítica. Conviene realizar una gráfica 3D de la temperatura vs posición vs tiempo e incluir las isoterms (contornos de temperatura constante).

Solución.

Aquí utilizaremos el siguiente comando para generar las isotermas:

```
ax.plot_surface(X, Y, Z, cmap=cm.coolwarm)
```

Por otro lado, la solución analítica está dada por:

$$T(x, t) = \sum_{n=1,3,\dots}^{\infty} \frac{4T_0}{n\pi} \sin(k_n x) e^{\frac{-k_n^2 K}{C\rho} t} \quad (12)$$

Donde $k_n = \frac{n\pi}{L}$. Para nuestro problema $T_0 = 100$ y $L = 1m$. Si recortamos la serie y consideramos los primeros 1000 términos obtenemos:

$$T(x, t) \approx \sum_{n=1,3,\dots}^{1000} \frac{400}{n\pi} \sin(k_n x) e^{\frac{-k_n^2 K}{C\rho} t} \quad (13)$$

Con base en esta aproximación, graficaremos la solución analítica a partir del siguiente algoritmo que tiene como base el algoritmo creado para la solución numérica.

```
In [ ]: import numpy as np
import matplotlib.pyplot as p
from mpl_toolkits.mplot3d import Axes3D

def T(x, t): #Función que evalúa los 1000 términos de la sol analítica
    n = 1000; K = 205.0
    C = 910.0; rho = 2700.0
    suma = 0
    c = K/(rho * C)
    if x == 1. or x == 0.: return 0
    elif t == 1: return 100
    else:
        for i in range(1, n, 2):
            suma += 400/(i*pi) * sin(i * pi * x) * exp(-(i*pi)**2 * t*c)
    return suma

Tpl = np.zeros((100, 100), float)
m = 1
prueba = np.zeros((100), float)
x = np.linspace(0, 1, 100)

for t in range(1, 20000):
    if (t%300 == 0) or (t==1):
        for ix in range(1, 100):
            Tpl[ix, m] = T(x[ix], t)
        print(m)
        m += 1

x = list(range(1, 100))
y = list(range(1, 30))

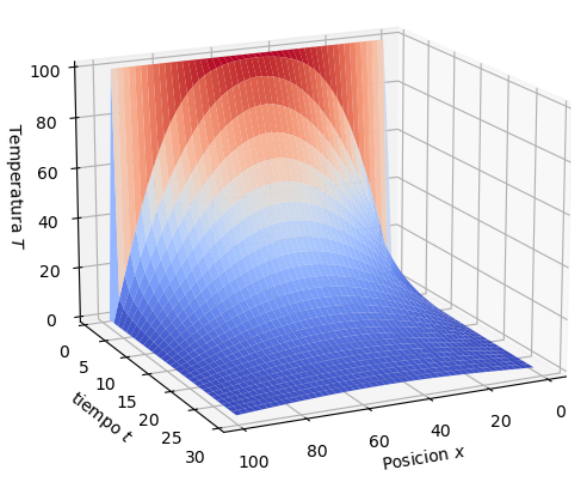
X, Y = p.meshgrid(x, y)

def functz(Tpl):
    z = Tpl[X, Y]
    return z

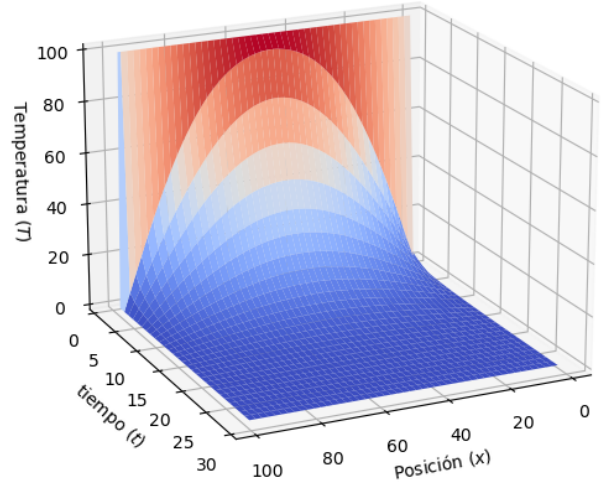
Z = functz(Tpl)
fig = p.figure()
ax = Axes3D(fig)
#ax.plot_wireframe(X, Y, Z, color="b")
ax.plot_surface(X, Y, Z, cmap=cm.coolwarm)
ax.set_xlabel('Posición $(x)$')
ax.set_ylabel('tiempo $(t)$')
ax.set_zlabel('Temperatura $(T)$')
p.show()
```

Figura 8: Algoritmo para graficar la solución analítica a la ecuación de Calor.

Con base en los algoritmos de las figuras (1) y (8) y la implementación del comando para generar las isotermas tenemos las siguientes gráficas para la solución numérica y analítica.



(a) Solución numérica



(b) Solución analítica

Figura 9

Se puede apreciar que ambas soluciones son muy parecidas entre sí. Existen pequeñas diferencias como en la curva que se genera en la parte posterior. Esta diferencia es debido a los errores que tienen los algoritmos de Forward difference y Central difference en el cálculo de la solución numérica. A pesar de ello, ambas soluciones son muy parecidas entre sí.

(h) ¿Qué pasa si el aluminio es sustituido por un mal conductor térmico como madera?

Solución.

Para esta parte consideraremos la madera de Pino. Para este material se tiene que:

$$K = 0.12 \text{ W}/(\text{m} \cdot \text{K}), \quad C = 2510 \text{ J}/(\text{kg K}), \quad \rho = 640 \text{ kg}/\text{m}^3 \quad (14)$$

Por lo que nuestra condición de estabilidad resulta:

$$\frac{\Delta t}{(\Delta x)^2} < \frac{C\rho}{2K} \approx 6.69 \times 10^6$$

Así para $\Delta x = 0.03$ tenemos que:

$$\Delta t < 6026$$

Aquí tenemos mayor rango de estabilidad. El valor que se consideró fue $\Delta t = 5$, obteniendo la siguiente solución donde se ha tenido que extender la cantidad de puntos a ser evaluados para apreciar mejor la solución a la ecuación de calor para el mal conductor térmico.

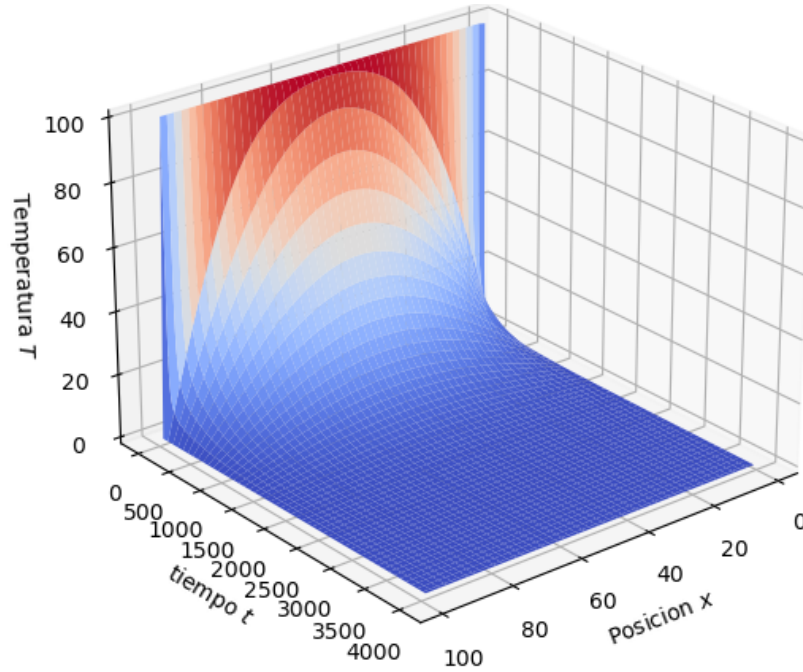


Figura 10: Solución estable a la ecuación de calor para madera de Pino

Como era de esperarse, al ser un mal conductor térmico la madera, la temperatura dentro de la barra de madera disminuye muy lentamente. Sin embargo, llegará un punto donde se alcance el equilibrio térmico. Pero para ello transcurrirá mucho tiempo.

2. La vibración de una cuerda

Este problema pretende estudiar las oscilaciones de una cuerda. Considere una cuerda de longitud L y densidad $\rho(x)$ por unidad de longitud, atada en ambos extremos y bajo una tensión $T(x)$. Suponga que el desplazamiento relativo de la cuerda respecto a su posición de equilibrio $\frac{y(x,t)}{L}$ es pequeño y que la pendiente de la cuerda $\frac{\partial y}{\partial x}$ también es pequeña.

- (a) Considere una sección infinitesimal de la cuerda, note que la diferencia en las componentes de las tensiones en x y $x + \Delta x$ tiene por resultado una fuerza restauradora. Demuestre que al aplicar las leyes de Newton a esta sección obtenemos la ecuación de onda:

$$\frac{dT(x)}{dx} \frac{\partial y(x,t)}{\partial x} + T(x) \frac{\partial^2 y(x,t)}{\partial x^2} = \rho(x) \frac{\partial^2 y(x,t)}{\partial t^2}$$

Solución.

Se sabe que:

$$\sum \vec{F}(x,y) = m\vec{a}(x,y)$$

Proyectando en el eje y tenemos:

$$\sum F_y = m \frac{\partial^2 y(x,t)}{\partial t^2}$$

Notemos que para el segmento infinitesimal de cuerda tenemos que:

$$\rho(x) = \frac{m}{\Delta x} \Rightarrow m = \rho(x)\Delta x$$

Por otro lado, si consideramos las tensiones del diagrama en los puntos x y $x + \Delta x$, debido a que la tensión no es constante tenemos que:

$$\sum F_y = (T(x') \sin \theta)|_{x'=x+\Delta x} - (T(x') \sin \theta)|_{x'=x}$$

El signo menos se debe a que las fuerzas van en sentidos contrarios. Luego, debido a que θ es muy pequeño y la pendiente también lo es, podemos hacer:

$$\sin \theta \approx \tan \theta \approx \frac{\partial y(x, t)}{\partial x}$$

Por lo que:

$$\begin{aligned} \sum F_y &\approx \left(T(x') \frac{\partial y(x', t)}{\partial x} \right) \Big|_{x'=x+\Delta x} - \left(T(x') \frac{\partial y(x', t)}{\partial x} \right) \Big|_{x'=x} \\ &= T(x + \Delta x) \frac{\partial y(x + \Delta x, t)}{\partial x} - T(x) \frac{\partial y(x, t)}{\partial x} \\ &= T(x + \Delta x) \frac{\partial y(x + \Delta x, t)}{\partial x} - T(x) \frac{\partial y(x, t)}{\partial x} + T(x) \frac{\partial y(x + \Delta x, t)}{\partial x} - T(x) \frac{\partial y(x + \Delta x, t)}{\partial x} \\ &= \frac{\partial y(x + \Delta x, t)}{\partial x} [T(x + \Delta x) - T(x)] + T(x) \left[\frac{\partial y(x + \Delta x, t)}{\partial x} - \frac{\partial y(x, t)}{\partial x} \right] \end{aligned}$$

Donde hemos sumado y restado un término para poder obtener el resultado anterior. De esta forma, se sigue que:

$$\begin{aligned} \sum F_y &= m \frac{\partial^2 y(x, t)}{\partial t^2} \\ \Rightarrow \frac{\partial y(x + \Delta x, t)}{\partial x} [T(x + \Delta x) - T(x)] + T(x) \left[\frac{\partial y(x + \Delta x, t)}{\partial x} - \frac{\partial y(x, t)}{\partial x} \right] &= \rho(x) \Delta x \frac{\partial^2 y(x, t)}{\partial t^2} \\ \Rightarrow \frac{\partial y(x + \Delta x, t)}{\partial x} \left[\frac{T(x + \Delta x) - T(x)}{\Delta x} \right] + T(x) \left[\frac{\frac{\partial y(x + \Delta x, t)}{\partial x} - \frac{\partial y(x, t)}{\partial x}}{\Delta x} \right] &= \rho(x) \frac{\partial^2 y(x, t)}{\partial t^2} \end{aligned}$$

Tomando el límite $\Delta x \rightarrow 0$ y aplicando las definiciones de derivada total y parcial obtenemos:

$$\frac{\partial y(x, t)}{\partial x} \frac{dT(x)}{dx} + T(x) \frac{\partial^2 y(x, t)}{\partial x^2} = \rho(x) \frac{\partial^2 y(x, t)}{\partial t^2} \quad (15)$$

Esta ecuación corresponde con la **ecuación de onda para una cuerda densidad y tensión variable**, que es lo que se pide en este inciso.

(b) ¿Qué condiciones son necesarias para obtener la ecuación de onda estándar?

$$\frac{\partial^2 y(x, t)}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y(x, t)}{\partial t^2}, \quad c = \sqrt{\frac{T}{\rho}} \quad (16)$$

Solución.

Para poder obtener la ecuación de onda estándar, basta considerar a la densidad ρ y la tensión T constantes en la deducción del ejercicio anterior. Tomando estas consideraciones en la ecuación (15) tenemos que:

$$\frac{\partial y(x, t)}{\partial x} \frac{dT}{dx} + T \frac{\partial^2 y(x, t)}{\partial x^2} = \rho \frac{\partial^2 y(x, t)}{\partial t^2}$$

$$\begin{aligned}\Rightarrow T \frac{\partial^2 y(x, t)}{\partial x^2} &= \rho \frac{\partial^2 y(x, t)}{\partial t^2} \\ \frac{\partial^2 y(x, t)}{\partial x^2} &= \frac{\rho}{T} \frac{\partial^2 y(x, t)}{\partial t^2} \\ \frac{\partial^2 y(x, t)}{\partial x^2} &= \frac{1}{c^2} \frac{\partial^2 y(x, t)}{\partial t^2}\end{aligned}$$

(c) ¿Qué condiciones deben cumplirse para obtener una única solución a esta EDP de segundo orden?

Solución.

Debido a que las derivadas parciales son de segundo orden tanto en la variable x como en la variable t , debemos conocer 4 condiciones, pues al realizar separación de variables se obtienen dos soluciones para la variable x y dos para t .

Las condiciones a conocer son las iniciales y de frontera. Por ejemplo, para las condiciones iniciales, uno puede considerar a la cuerda en reposo y después que en un punto de la cuerda se aplique una fuerza perturbativa. Por lo que en $t = 0$, la velocidad será cero (al estar en reposo) y tendrá un desplazamiento inicial (ej. la mitad de la cuerda con pendiente positiva y la otra mitad con la misma pendiente desplazada pero negativa formando un triángulo).

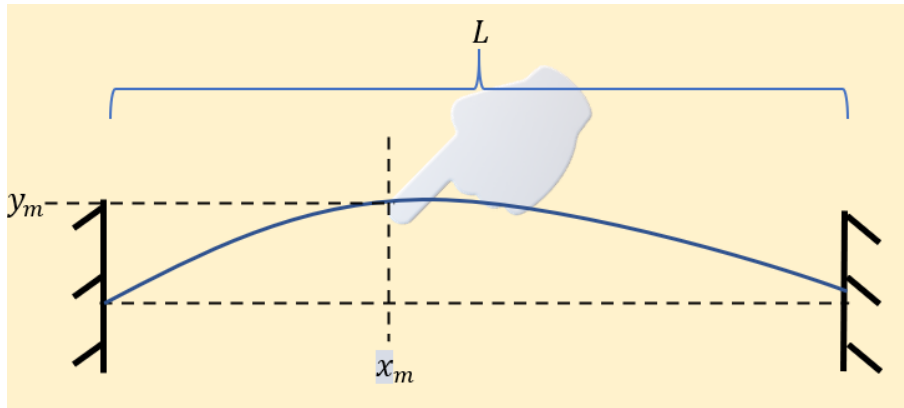


Figura 11: Cuerda fija en sus extremos en reposo ($t = 0$) siendo jalada en un punto para generar una perturbación. Condición inicial.

Así las condiciones iniciales considerando las ecuaciones de las rectas de la figura (11) y la velocidad inicial igual a cero:

$$y(x, t = 0) = \begin{cases} \frac{y_m}{x_m} x & \text{si } 0 \leq x \leq x_m \\ \frac{y_m}{L-x_m} (L - x) & \text{si } x_m < x \leq L \end{cases}$$

$$\frac{\partial y(x, t = 0)}{\partial t} = 0$$

Por otro lado, como la cuerda está sujeta de ambos extremos, las condiciones de frontera son:

$$y(x = 0, t) = y(x = L, t) = 0$$

- (d) Utilice una malla de longitud Δt en el tiempo y Δx en el espacio para generar una solución numérica.

$$y(x, t) = y(i\Delta x, j\Delta t) = y_{ij}$$

Solución.

Al igual como se realizó en el ejercicio 1, discretizaremos nuestras variables de la siguiente forma:

$$x = i\Delta x \quad ; \quad t = j\Delta t$$

Así es posible obtener una malla para nuestra solución numérica.

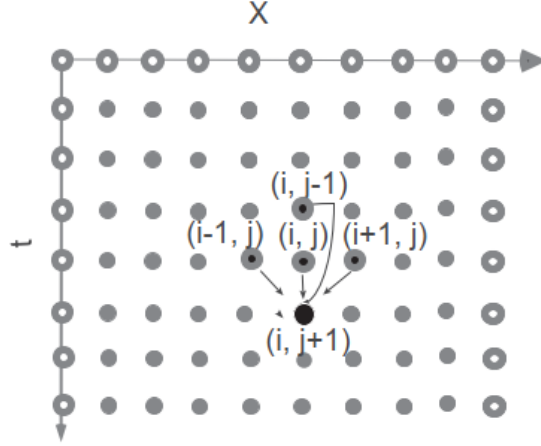


Figura 12: Malla discretizada

En el siguiente inciso obtendremos la ecuación de onda en diferencias.

- (e) Exprese las segundas derivadas de la EDP en términos de diferencias finitas, demuestre que esto resulta en la ecuación de onda en diferencias:

$$\frac{y_{ij+1} + y_{ij-1} - 2y_{ij}}{c^2(\Delta t)^2} = \frac{y_{i+1j} + y_{i-1j} - 2y_{ij}}{(\Delta x)^2}$$

Solución.

Retomemos la ecuación de onda:

$$\frac{\partial^2 y(x, t)}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y(x, t)}{\partial t^2} \quad (17)$$

Usando Central difference para el cálculo de ambas derivadas obtenemos:

$$\begin{aligned} \frac{\partial^2 y}{\partial t^2} &\approx \frac{\frac{\partial y}{\partial t} \Big|_{t+\frac{\Delta t}{2}} - \frac{\partial y}{\partial t} \Big|_{t-\frac{\Delta t}{2}}}{\Delta t} \\ &= \frac{\frac{y(x, t+\Delta t) - y(x, t)}{\Delta t} - \frac{y(x, t) - y(x, t-\Delta t)}{\Delta t}}{\Delta t} \\ &= \frac{y(x, t+\Delta t) + y(x, t-\Delta t) - 2y(x, t)}{(\Delta t)^2} \\ &= \frac{y_{ij+1} + y_{ij-1} - 2y_{ij}}{(\Delta t)^2} \end{aligned}$$

Similarmente:

$$\begin{aligned}
\frac{\partial^2 y}{\partial x^2} &\approx \frac{\left. \frac{\partial y}{\partial x} \right|_{x+\frac{\Delta x}{2}} - \left. \frac{\partial y}{\partial x} \right|_{x-\frac{\Delta x}{2}}}{\Delta x} \\
&= \frac{\frac{y(x+\Delta x, t) - y(x, t)}{\Delta x} - \frac{y(x, t) - y(x-\Delta x, t)}{\Delta x}}{\Delta x} \\
&= \frac{y(x+\Delta x, t) + y(x-\Delta x, t) - 2y(x, t)}{(\Delta x)^2} \\
&= \frac{y_{i+1j} + y_{i-1j} - 2y_{ij}}{(\Delta x)^2}
\end{aligned}$$

Así de (17) claramente se obtiene lo solicitado:

$$\frac{y_{i+1j} + y_{i-1j} - 2y_{ij}}{(\Delta x)^2} \approx \frac{y_{ij+1} + y_{ij-1} - 2y_{ij}}{c^2(\Delta t)^2} \quad (18)$$

(f) Demuestre que el algoritmo anterior puede reescribirse como:

$$y_{ij+1} = 2y_{ij} - y_{ij-1} + \frac{c^2}{\Delta x^2} [y_{i+1j} + y_{i-1j} - 2y_{ij}] \quad (19)$$

donde $c' = \frac{\Delta x}{\Delta t}$ es la velocidad de la malla, es decir, la razón numérica de los parámetros.

Solución.

Retomando el resultado del ejercicio anterior, al despejar de la ecuación (18) tenemos:

$$\begin{aligned}
y_{ij+1} + y_{ij-1} - 2y_{ij} &= c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 [y_{i+1j} + y_{i-1j} - 2y_{ij}] \\
\Rightarrow y_{ij+1} &= 2y_{ij} - y_{ij-1} + \frac{c^2}{\Delta x^2} [y_{i+1j} + y_{i-1j} - 2y_{ij}]
\end{aligned}$$

(g) ¿Cómo entran las condiciones iniciales y las condiciones de frontera?

Solución.

Retomemos las condiciones iniciales y de frontera:

$$y(x, t = 0) = \begin{cases} \frac{y_m}{x_m} x & \text{si } 0 \leq x \leq x_m \\ \frac{y_m}{L-x_m} (L-x) & \text{si } x_m < x \leq L \end{cases}$$

$$\begin{aligned}
\frac{\partial y(x, t = 0)}{\partial t} &= 0 \\
y(x = 0, t) &= y(x = L, t) = 0
\end{aligned}$$

Usando la notación discretizada, estas condiciones se pueden reescribir como:

$$y_{1j} = y_{Lj} = 0 \quad (20)$$

$$y_{i1} = \begin{cases} \frac{y_m}{x_m} x & \text{si } 1 \leq i \leq m \\ \frac{y_m}{L-x_m} (L-x) & \text{si } m < i \leq L \end{cases} \quad (21)$$

Donde hemos asumido que $i = 1$ corresponde a $x = 0$, $i = L \rightarrow x = L$ y $j = 1 \rightarrow t = 0$. Esta última condición permite saber la solución en todo el espacio para $t = 0$.

Por otro lado, la condición de la derivada usando Central difference con un salto de Δt resulta:

$$\frac{\partial y(x, 0)}{\partial t} \approx \frac{y(x, \Delta t) - y(x, -\Delta t)}{2\Delta t} = 0$$

Por lo que, si asumimos que $j = 0$ corresponde a $t = -\Delta t$, $j = 1$ corresponde a $t = 0$ y $j = 2$ a $t = \Delta t$, obtenemos que:

$$\frac{y_{i2} - y_{i0}}{2\Delta t} = 0$$

Como $\Delta t \neq 0$:

$$y_{i2} = y_{i0} \quad (22)$$

Así del inciso (f), usando este resultado en la relación de recurrencia obtenemos:

$$y_{i2} = y_{i1} + \frac{c^2}{2c^2} [y_{i+1,1} + y_{i-1,1} - 2y_{i1}] \quad (23)$$

Esta ecuación usa la solución en todo el espacio para $t = 0$ para propagar hacia adelante en el tiempo con un salto de Δt .

De esta forma, al conocer la solución en todo el espacio en $t = 0$ con (21), propagar esta solución a un tiempo posterior con (23) y continuar desplazando la solución a tiempos posteriores con (19), es posible obtener la solución a lo largo de toda la cuerda en el tiempo que se quiera.

(h) La condición de Courant para la estabilidad de la solución es que:

$$\frac{c}{c'} \leq 1$$

¿Qué significa en términos de los pasos?

Solución.

Sustituyendo tenemos que:

$$\frac{c}{c'} = \frac{\frac{\sqrt{T}}{\rho}}{\frac{\Delta x}{\Delta t}} \leq 1$$

Así:

$$\frac{\Delta t}{\Delta x} \leq \sqrt{\frac{\rho}{T}} \quad (24)$$

Esta ecuación nos dice que entre más pequeños sean los pasos en el tiempo tendremos una mejor solución, a menos que también el tiempo sea pequeño. Por lo que hay que tomar pasos en el tiempo grandes para lograr una mejor estabilidad.

- (i) Escriba un programa que implemente la fórmula anterior, grafique el movimiento de la cuerda, o mejor aún, produzca una animación.

Solución.

En esta parte optaremos por realizar un programa que genere una animación de la cuerda. Para ello se apoyará de la librería `matplotlib.animation`. El algoritmo es el siguiente:

```

In [ ]: from os import linesep
        from numpy import *
        import numpy as np, matplotlib.pyplot as plt
        import matplotlib.animation as animation

        rho = 0.05 #Densidad en kg/m
        ten = 10. #Tension en Newtons
        c = sqrt(ten/rho)
        c1 = 2*c
        ratio = c*c/(c1*c1)
        print(ratio)
        xi = np.zeros((101, 3), float)
        k = range(0, 101)

        def Initialize():
            for i in range(0, 81):
                xi[i, 0] = 0.00125 * i
            for i in range(81, 101):
                xi[i, 0] = 0.1 - 0.005*(i-80)

        def animate(num):
            for i in range(1, 100):
                xi[i, 2] = 2* xi[i, 1] - xi[i, 0] + ratio*(xi[i + 1, 1]
                                                            + xi[i-1, 1] - 2*xi[i, 1])

            line.set_data(k, xi[k, 2])
            for m in range(0, 101):
                xi[m, 0] = xi[m, 1]
                xi[m, 1] = xi[m, 2]
            return line

        Initialize()

        fig = plt.figure()
        ax = fig.add_subplot(111, autoscale_on=False, xlim=(0, 101),
                             ylim=(-0.15, 0.15))
        ax.grid()
        plt.title('Vibraciones')
        line , = ax.plot(k, xi[k,0] , lw=2)
        for i in range(1, 100):
            xi[i, 1] = xi[i, 0] + 0.5*ratio*(xi[i+1, 0] + xi[i-1, 0]
                                             - 2*xi[i, 0])
        ani = animation.FuncAnimation(fig, animate, 1)
        plt.show()

```

Figura 13: Algoritmo para generar una animación de la cuerda

La forma en como funciona el algoritmo, es que genera un arreglo de arreglos, para ir calculando los términos y_{ij-1} , y_{ij} y y_{ij+1} a partir de la relación de recurrencia (19). Para ello primero utiliza las otras condiciones (21) y (23). Aquí hemos considerado una cuerda de longitud 1 m, densidad 0.05 kg/m, y una tensión de 10 N. Generando así el siguiente resultado:

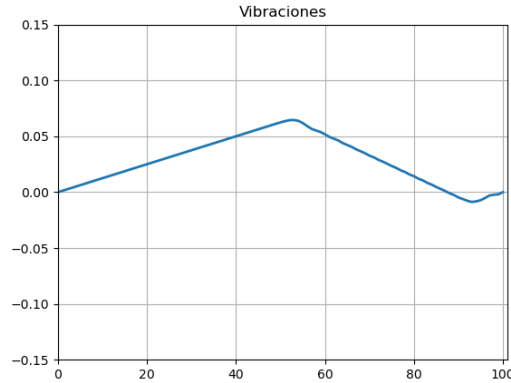


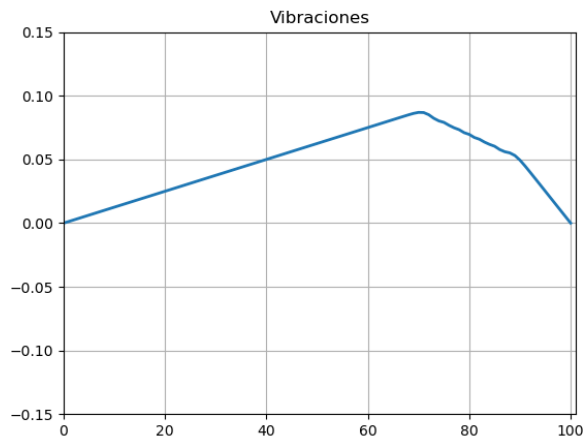
Figura 14: Comportamiento de la cuerda

Este gráfico se va moviendo con el paso del tiempo. Conforme transcurre el tiempo se desplaza la cuerda de forma que vemos las ondas generadas. En el eje x se tiene la longitud de la cuerda que va de 0 a 100 cm. En el eje y , es la altura que va tomando la onda en la cuerda. Se puede apreciar que en los extremos de la cuerda la altura es cero, lo cual cumple con las condiciones de frontera. Además, si se deja el gráfico durante mucho tiempo, se apreciará que va disminuyendo el movimiento hasta llegar nuevamente al reposo, como era de esperarse.

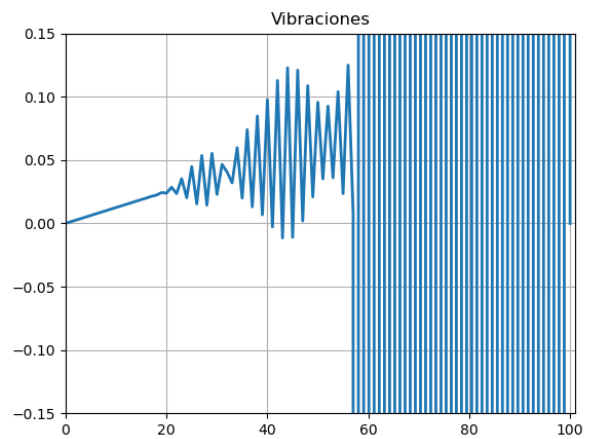
- (j) Cambie los pasos en el tiempo y el espacio en su simulación para que algunas veces satisfaga la condición de Courant y otras no.

Solución.

En nuestro algoritmo, hemos guardado la condición de Courant en la variable `ratio`. Así para tener estabilidad en nuestro programa es necesario que $\text{ratio} \leq 1$. Veamos como se ve nuestro resultado para distintos valores de `ratio`.



(a) Simulación estable para $\text{ratio} = 0.1$



(b) Simulación inestable para $\text{ratio} = 2.0$

Figura 15

Vemos que cuando se cumple la condición, nuestra solución es estable y se aprecia bien el comportamiento de la cuerda. Sin embargo, cuando no se cumple la condición, nuestra solución es completamente inestable. Esto se debe a que nuevamente la computadora empieza a acercarse al overflow en sus cálculos y los resultados divergen tal como se aprecia en la imagen.