Ecuaciones diferenciales parciales

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

25 de mayo de 2018

Outline

Outline

Ecuación de onda

El prototipo clásico de la EDP de tipo hiperbólico es la ecuación de onda, que en 1D tiene la expresión

$$\frac{\partial^2 u}{\partial t} = c^2 \, \frac{\partial^2 u}{\partial x^2} \tag{1}$$

donde c es la velocidad de fase de la onda.

Ecuación de onda

Aunque la ecuación anterior espacialmente está en 1D, se puede suponer que modela una geometría planar 3D a modo de losa, con dependencia explícita de la coordenada cartesiana transversal a la losa, es decir, x, y no depende de las otras dos coordenadas, y y z , a lo largo de los cuales, la extensión del sistema puede considerarse virtualmente infinita.

Ecuación de onda

En dicha geometría, una onda plana es una onda de frecuencia constante, cuyos frentes de onda (superficies de fase constante) son planos paralelos infinitos (y, z) de amplitud constante, normal a la dirección de propagación, x.

Ecuación de onda

Para modelar una situación física concreta, la ecuación de onda necesita completarse con condiciones iniciales y de frontera.

Ecuación de onda

Las condiciones iniciales especifican la solución y su derivada de tiempo de primer orden sobre todo el dominio espacial en algún momento inicial t_0:

$$u(x, t_0) = u^0(x), \qquad x \in [-x_{\text{max}}, x_{\text{max}}] \tag{2}$$

$$\frac{\partial u}{\partial t}(x, t_0) = v^{0(x)} \tag{3}$$

Condiciones de frontera

Por simplicidad nuevamente, consideraremos la CDF de tipo Dirichlet, que implican valores constantes en las fronteras:

$$u(x_{\text{min}}, t) = u^0_{x_{\text{min}}}, \qquad t > t_0 \tag{4}$$

$$u(x_{\text{max}}, t) = u^0_{x_{\text{max}}} \tag{5}$$

Espaciamiento en la malla

Se hará la aproximación al problema ecs. (1) - (5) mediante el esquema de diferencias finitas, ocupando un espaciamiento regular en la malla:

$$x_i = x_{\mathrm{min}} + (i-1)\,h_x, \qquad i = 1, 2, \ldots, N_x \tag{6}$$

$$t_n = n\,h_t, \qquad n = 0, 1, \ldots, \tag{7}$$

donde h_t representa el paso temporal, N_x es el número de nodos en el eje espacial de la malla.

Tamaño del paso espacial

El tamaño del paso espacial está definido por:

$$h_x = (x_{\text{max}} - x_{\text{min}})/(N_x - 1) \tag{8}$$

Desarrollo de la segunda derivada

Para la segunda derivada con respecto a la posición en el nodo (x_i, t_n), usaremos la fórmula de diferencias centrales para la derivada de segundo orden:

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

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

La segunda derivada con respecto a x

Se requiere sólo la información de los puntos en el paso temporal t_n y los puntos vecinos del punto requerido.

La segunda derivada con respecto a t

La derivada temporal se aproxima usando una fórmula similar

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

La ecuación de onda

Aplicando los esquemas anteriores, se tiene una aproximación a la ecuación de onda (1) para el nodo espacio - temporal (x_i, t_n):

$$\frac{u_i^{n+1} - 2\,u_i^n + u_i^{n-1}}{h_t^2} = c^2\,\frac{u_{i+1}^n - 2\,u_i^n + u_{i-1}^n}{h_x^2} \tag{9}$$

El error asociado es del orden de $O(h_x^2 + h_t^2)$.

Propagación de la solución

Usando la siguiente notación

$$\lambda = \left(\frac{c\, h_t}{h_x} \right)^2 \tag{10}$$

podemos expresar la solución propagada al tiempo t_{n+1} para los puntos interiores x_i de la malla, en términos de los dos pasos temporales anteriores, t_n y t_{n-1}:

Propagación de la solución

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

Las CDF

En cuanto a las CDF, en virtud de las condiciones de Dirichlet, éstas permanecen constantes durante toda la propagación, y esto se expresa simplemente por

$$u_1^{n+1} = u_1^n, \qquad u_{N_x}^{n+1} = u_{N_x}^n \tag{12}$$

El sistema matricial

En notación matricial, el sistema (11) - (12), tiene la forma:

$$\mathbf{u}^{n+1} = \mathbf{B} \cdot \mathbf{u}^n - \mathbf{u}^{n-1}, \qquad n = 0, 1, 2, \ldots, \tag{13}$$

El sistema matricial

Donde

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & & & 0 \\ \lambda & 1-2\lambda & \lambda & & \\ & \ddots & \ddots & \ddots & \\ & & \lambda & 1-2\lambda & \lambda \\ 0 & & & 0 & 1 \end{bmatrix}, \quad \mathbf{u}^n = \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N_x-1}^n \\ u_{N_x}^n \end{bmatrix} \tag{14}$$

la matriz de propagación B mantiene una estructura tridiagonal.

Naturaleza explícita del método

Dado que los componentes de la solución propagada $\mathbf{u}^{n+1}$ se pueden expresar de forma independiente, se dice que *el método es explícito* y es similar al método explícito FTCS para la ecuación de difusión.

Consideraciones del método

A pesar de la similitud formal, existe, sin embargo, una diferencia no trivial entre los métodos explícitos para EDP parabólicas e hiperbólicas, a saber, la aparición en la segunda solución anterior $\mathbf{u}^{n-1}$, a lo largo de la anterior, $\mathbf{u}^{n}$.

Consideraciones del método

Una vez que comenzó la propagación, este aspecto se maneja fácilmente mediante el almacenamiento continuo de las dos soluciones más recientes.

Consideraciones del método

Una vez que comenzó la propagación, este aspecto se maneja fácilmente mediante el almacenamiento continuo de las dos soluciones más recientes.

Un aspecto un tanto delicado se refiere al inicio de la propagación.

Consideraciones del método

De hecho, junto con la solución inicial u_0, es necesario proporcionar la primera solución propagada, $\mathbf{u}^1$, en lugar de la primera derivada $\partial \mathbf{u}^0 / \partial t$

Consideraciones del método

Una forma efectiva de hacerlo, manteniendo la precisión general de segundo orden del método, comienza con la serie Taylor de los componentes de la solución con respecto al tiempo:

$$u_i^{n+1} = u_i^n + h_t \left(\frac{\partial u}{\partial t}\right)_{i,n} + \frac{1}{2} h_t^2 \left(\frac{\partial^2 u}{\partial t^2}\right)_{i,n} + O(h_t^3)$$

Consideraciones del método

En particular, para el primer paso de propagación, se tiene que

$$u_i^1 = u_i^0 + h_t \left(\frac{\partial u}{\partial t}\right)_{i,0} + \frac{1}{2} h_t^2 \left(\frac{\partial^2 u}{\partial t^2}\right)_{i,0} + O(h_t^3)$$

Consideraciones del método

Mientras que la primera derivada está disponible desde la condición inicial (3), la segunda derivada puede ser reemplazada con la segunda derivada espacial usando la ecuación de onda (1)

$$u_i^1 = u_i^0 + h_t\, v_i^0 + \frac{1}{2}\, h_t^2\, c^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,0} + O(h_t^3)$$

donde $v_i^0 = v^0(x_i)$.

Consideraciones para el método

Para la segunda deriva espacial, usamos el esquema de segundo orden

$$u_i^1 = u_i^0 + h_t\, v_i^0 + \frac{1}{2}\, h_t^2\, c^2\, \frac{u_{i+1}^0 - 2\, u_i^0 + u_{i-1}^0}{h_x^2} + O(h_t^3 + h_x^2)$$

Parámero de propagación

Identificando el parámetro λ, tendremos la expresión final para los componentes de la primera solución propagada: $\mathbf{u}^1$, en términos de la solución inicial $\mathbf{u^0}$, y la primera derivada temporal inicial $\mathbf{v}^0$:

$$u_i^1 = \frac{1}{2}\,\lambda\,u_{i-1}^0 + (1-\lambda)\,u_i^0 + \frac{1}{2}\,\lambda\,u_{i+1}0 + h_t\,v_i^0 \qquad i = 2, 3, \ldots, N_x - 1 \tag{15}$$

Estabilidad del método

De acuerdo al método general para el análisis de estabilidad de Von Neumann, sustituimos el modo propio genérico $\varepsilon_i^n = \xi(k)^n \, \exp(\ell \, k \, i \, h_x)$ (recuerda que $\ell = \sqrt{-1}$, para diferenciar con el índice i) como una solución en la ecuación de diferencias (11), para encontrar:

$$\begin{aligned}\xi = [\lambda \, \exp(-\ell \, k \, h_x) + \\ + 2\,(1 - \lambda) + \lambda \, \exp(\ell \, k \, h_x)] \, \xi - \xi^{-1}\end{aligned}$$

El parámetro de propagación

Combinando las dos exponenciales en términos de $2\,\lambda\ \cos(k\,h_x)$ y utilizando la identidad $1 - \cos x = 2\ \sin^2(x/2)$, el factor de amplificación satisface

$$\frac{1}{2}(\xi + \xi^{-1}) = 1 - 2\,\lambda\,sin^2(k\,h_x/2) \qquad (16)$$

Factor de amplificación

Considerando que el lado derecho de la expresión es un real, entonces $\xi + \xi^{-1}$ debe ser real también.

Entonces debe ocurrir que $|\xi| = |\xi|^{-1}$, que nos lleva

$$\frac{1}{2}\left(\xi + \xi^{-1}\right) = |\xi| \, \cos\varphi$$

donde φ es una fase arbitraria.

Consideración para la estabilidad

Tomando en cuenta que la condición de estabilidad $|\xi| < 1$, la relación anterior impone que el término del lado derecho de la ecuación (16)

$$-1 \leq 1 - 2\,\lambda\, \sin^2(k\,h_x/2) \leq 1$$

Consideración para la estabilidad

Tomando en cuenta que la condición de estabilidad $|\xi| < 1$, la relación anterior impone que el término del lado derecho de la ecuación (16)

$$-1 \leq 1 - 2\,\lambda\,\sin^2(k\,h_x/2) \leq 1$$

que se puede satisfacer siempre que

$$0 \leq \lambda \leq 1 \tag{17}$$

Condición de estabilidad

Como se definió λ ec. (10), entonces la condición de estabilidad para la propagación de la función de onda $u(x,t)$, basado en un esquema explícito de diferencias es

$$h_t \leq \frac{h_x}{c} \tag{18}$$

La función `PropagOnda`

La siguiente función `PropagOnda` utiliza el algoritmo (10) - (12) para resolver un problema de Cauchy para la ecuación de onda (1), propagando las dos soluciones iniciales $u_0[\,]$ y $u_1[\,]$ en el intervalo h_t.

Variables que intervienen

La velocidad de fase constante c, el paso espacial h_x, el número de nodos espaciales N_x, se proporcionan a la función mediante los argumentos c, hx y nx.

Arreglo con la solución

La solución propagada se devuelve en el arreglo $u[\,]$, y se supone que el ciclo de propagación del programa principal desplazará hacia atrás el contenido de las matrices $u[\,]$, $u_1[\,]$ y $u_0[\,]$, liberando al arreglo $u[\,]$, para un nuevo paso de propagación.

Función PropagOnda I

Código (1): Función `PropagOnda` para el ejercicio

```python
def PropagOnda(u0, u1, u, nx, c, hx, ht):
    lam = c*ht/hx
    lam = lam*lam
    lam2 = 2e0*(1e0 - lam)

    u[1] = u0[1]; u[nx] = u0[nx]
    for i in range(2,nx):
        u[i] = lam*u1[i-1] + lam2*u1[i] + lam*u1[i+1] - u0[i]
```

Ejercicio

Utilizaremos el algoritmo propuesto para resolver la clásica ecuación de onda:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad t > 0 \tag{19}$$

Paquete de onda inicial

Queremos ver cómo evoluciona y se propaga la solución con un paquete de onda inicial centrado en el origen:

$$u(x, 0) = \frac{\sin x \, \Delta k}{x \, \Delta k}, \qquad -x_{\text{max}} \leq x \leq x_{\text{max}} \tag{20}$$

Condiciones iniciales y de frontera

La primera derivada parcial de u con respecto al tiempo es $(\partial u/\partial t)(x,0) = 0$, y la ecuación de onda está sujeta a las CDF de Dirichlet:

$$u(\pm x_{\text{max}}, t) = \frac{\sin x_{\text{max}}\,\Delta t}{x_{\text{max}}\,\Delta k}, \qquad t > 0 \tag{21}$$

donde Δk k denota el ancho del intervalo de números de onda que contribuyen al paquete de onda, y cuanto mayor es su valor, menor es el ancho medio del paquete.

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$
3. $x_{\mathsf{max}} = 100$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$
3. $x_{\text{max}} = 100$
4. $h_x = 5 \times 10^{-2}$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$
3. $x_{\text{max}} = 100$
4. $h_x = 5 \times 10^{-2}$
5. $h_t = 5 \times 10^{-3}$ a $t_{\text{max}} = 40$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$
3. $x_{\text{max}} = 100$
4. $h_x = 5 \times 10^{-2}$
5. $h_t = 5 \times 10^{-3}$ a $t_{\text{max}} = 40$

Valores necesarios

Usaremos los siguientes valores para el ejercicio:

1. $c = 10$
2. $\Delta\, k = 1$
3. $x_{\text{max}} = 100$
4. $h_x = 5 \times 10^{-2}$
5. $h_t = 5 \times 10^{-3}$ a $t_{\text{max}} = 40$

De tal manera que $\lambda = 1$, lo que nos garantiza la estabilidad en la propagación de la solución.

El moduloEcuacionOnda

Dentro del módulo **`moduloEcuacionOnda`** se tienen las siguientes tres funciones:

1. **`PropagOnda`**.

El moduloEcuacionOnda

Dentro del módulo **`moduloEcuacionOnda`** se tienen las siguientes tres funciones:

1. **`PropagOnda`**.
2. **`Init`**.

El moduloEcuacionOnda

Dentro del módulo **`moduloEcuacionOnda`** se tienen las siguientes tres funciones:

1. **`PropagOnda`**.
2. **`Init`**.
3. **`generaArchivo`**.

La función `PropagOnda`

Ya se presentó la operación de la función que resuelve la propagación de la solución de la ecuación de onda.

La función `Init`

Inicializa en la malla espacial la solución inicial

$$u(x, 0) = \frac{\sin x \; \Delta k}{x \; \Delta k}, \qquad -x_{\text{max}} \leq x \leq x_{\text{max}} \quad (22)$$

y establece las CDF del problema.

Código de la función Init I

Código (2): Nombre Codigo

```
def Init(u0,u1,x,nx,c,dk,hx,ht):
    for i in range(1, nx+1):
        u0[i] = np.sin(dk*x[i])/(dk*x[i]) if dk*x[i] else 1e0

    lam = c*ht/hx
    lam = lam*lam
    lam2 = 2e0 * (1e0 - lam)
    u1[1] = u0[1]
    u1[nx] = u0[nx]
    for i in range(2,nx):
        v0 = 0e0
```

Código de la función `Init` II

```
        u1[i] = 0.5e0*(lam*u0[i-1]+lam
    2*u0[i]+lam*u0[i+1])-ht*v0
```

La función `generaArchivo`

La información que nos devuelve la propagación de la solución $u(x,t)$ está dentro de las listas x y u, considerando h_t, el tamaño del conjunto solución es de 4000 datos.

La función `generaArchivo`

Para manipular esta información en una gráfica, nos conviene almacenar el contenido de los arreglos en un archivo.

Esta función crea un archivo de texto plano (`*.txt`) y le asigna el nombre para el tiempo t.

La función `generaArchivo` I

Código (3): Código para generar archivos de texto plano con los arreglos solución

```
def generaArchivo(t, x, u, nx):
    archivo = "solucion_onda_{0:4.2f}.txt".format(t)
    out = open(archivo, "w")
    out.write("t = {0:4.2f}\n".format(t))
    out.write("     x          u\n")
    for i in range(1, nx+1):
        out.write("{0:10.5f}{1:10.5f}\n".format(x[i], u[i]))

```

La función `generaArchivo` II

```
    print('Se guardaron los datos en el archivo ' + archivo)
    out.close
```

Código principal I

Código (4): Código principal para el problema de la ecuación de onda

```
from moduloEcuacionOnda import Init,
    PropagOnda, generaArchivo

c    = 10e0
dk   = 1e0
xmax = 100e0
hx   = 5e-2
tmax = 40e0
ht   = 5e-3
nout = 500

nx = 2*(int)(xmax/hx+0.5)+1
```

Código principal II

```
nt = (int)(tmax/ht+0.5)
nx2 = int(nx/2)

u0 = [0]*(nx+1)
u1 = [0]*(nx+1)
u  = [0]*(nx+1)
x  = [0]*(nx+1)

for i in range(1,nx+1): x[i] = (i-nx2-1)*hx

Init(u0,u1,x,nx,c,dk,hx,ht)
generaArchivo(0,x,u1,nx)
```

Código principal III

```
for it in range(1,nt+1):
    t = it*ht
    PropagOnda(u0,u1,u,nx,c,hx,ht)

    for i in range(1,nx):
        u0[i] = u1[i]
        u1[i] = u[i]

    if (it % nout == 0 or it == nt):
       generaArchivo(t,x,u,nx)
```

Aclaración del uso de la función `int`

En el código anterior, vimos el uso de la función **`int`** para obtener un valor entero:

Código (5): Uso de la función `int`

```
nx = 2*(int)(xmax/hx+0.5)+1

nt = (int)(tmax/ht+0.5)

nx2 = int(nx/2)
```

Modo (`int`)(valor)

De esta manera estamos usando la función **`int`** entre paréntesis, que es válido, la diferencia es la manera en la que se ocupan los argumentos.

De la forma **`(int) (valor)`**, tenemos **`(int) (self)`**, donde `self` se refiere al objeto sobre el cual se está invocando, en este caso, la función **`int`**.

Modo (`int`)(valor)

Esta parte se comprende mejor cuando consideramos que el objeto que se invoca es una clase, pero hasta el momento, hemos manejado a la función **`int`**, como una función integrada de `python`.

Modo (`int`)(valor)

Es válido manejarlo de esta manera, ya que ambas expresiones son equivalentes, es decir

```
(int)(valor) = int(valor)
```

El lado derecho de la expresión, es el modo normal que hemos manejado en el curso para usar una función.

Datos obtenidos

Luego de haber ejecutado el programa, tendremos un conjunto de 17 archivos de texto plano: `solucion_onda_0.00.txt`, `solucion_onda_2.50.txt`, etc.

Gráficas de la solución

Con cada archivo de datos, podemos elaborar una gráfica y ver el comportamiento de la propagación de la onda.

Para darle un poco más de emoción, mostraremos las gráficas de cada archivo de texto, para simular una animación.

Código para ver las gráficas I

Código (6): Código para mostrar las gráficas

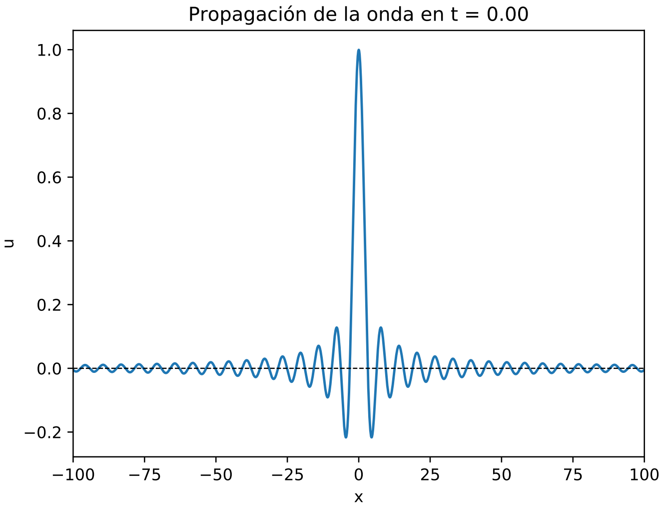
```
mport numpy as np
import matplotlib.pyplot as plt

for i in np.arange(0.0, 42.0, 2.5):
    tiempo = '{0:4.2f}'.format(i)
    archivo = 'solucion_onda_' + tiempo + '.txt'
    arreglos = np.genfromtxt(archivo, skip_header=2)

    plt.clf()
```

Código para ver las gráficas II

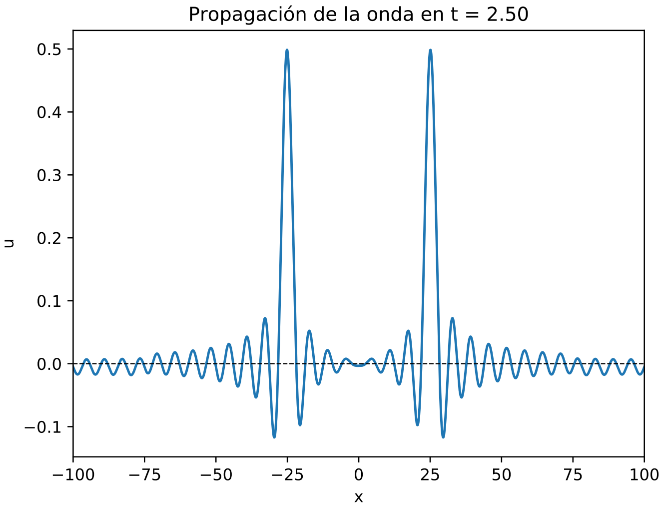
```
    plt.plot(arreglos[:,0],arreglos[:,1])
    plt.axhline(y=0,ls='dashed',lw=0.75,c='k')
    plt.title('Propagacion de la onda en t = ' + tiempo)
    plt.xlabel('x')
    plt.ylabel('u')
    plt.xlim([-100,100])
    plt.pause(1)
plt.show()
```

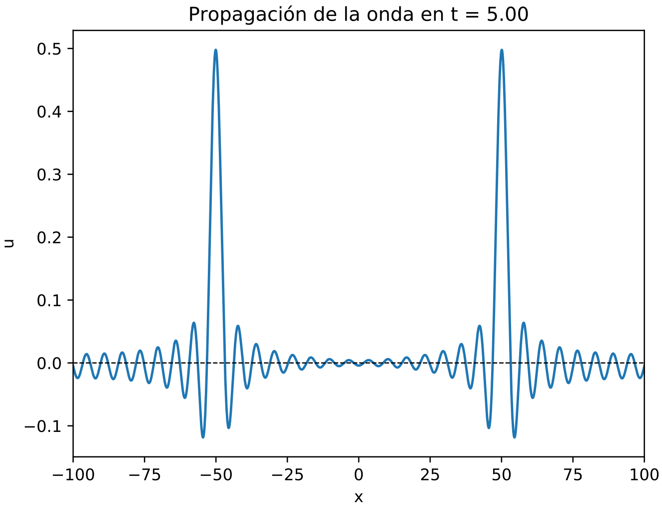
Gráficas obtenidas

A continuación se muestran las gráficas obtenidas con la solución de la propagación de la onda durante el intervalo de tiempo establecido.

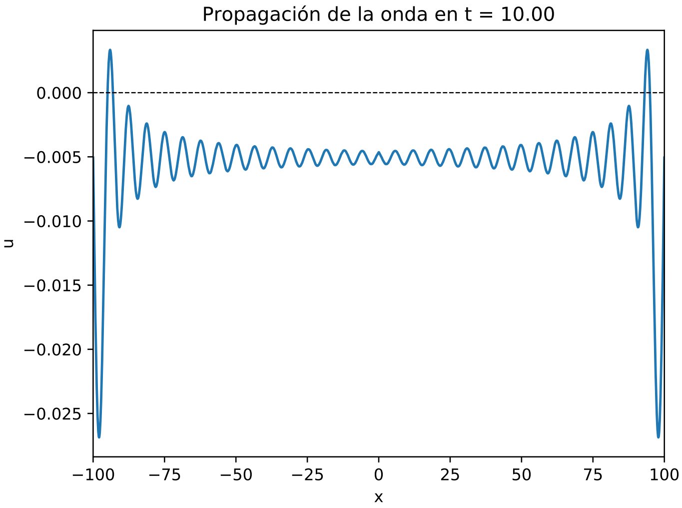


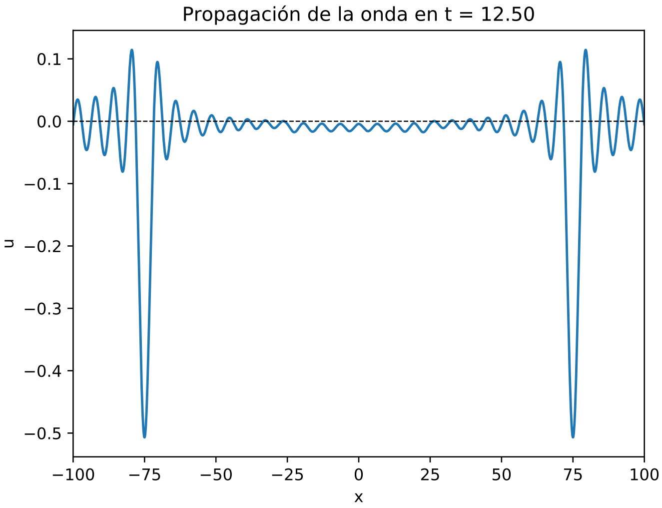
Propagación de la onda en t = 0.00
u
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−100
−75
−50
−25
0
25
50
75
100
x

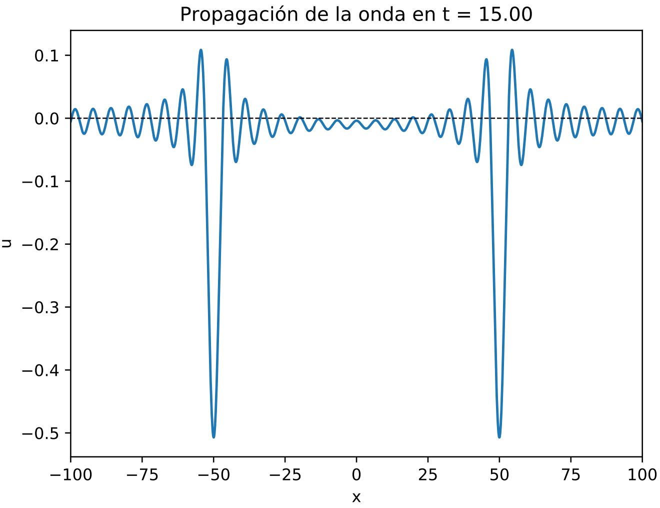

Propagación de la onda en t = 2.50
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
u
−100
−75
−50
−25
0
25
50
75
100
x

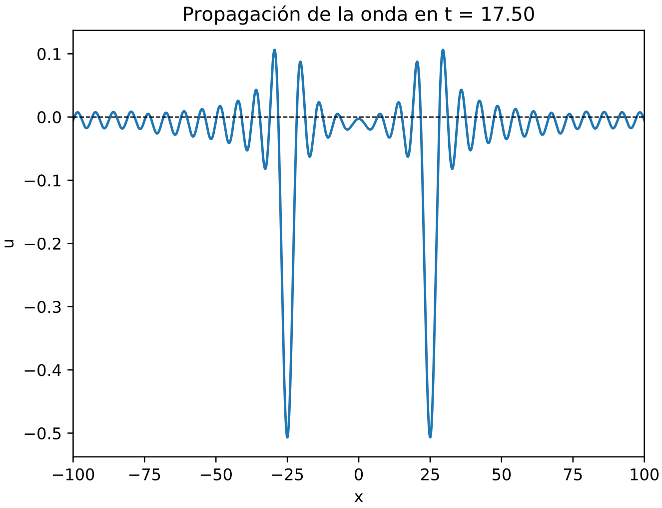

Propagación de la onda en t = 5.00
u
x
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−100
−75
−50
−25
0
25
50
75
100

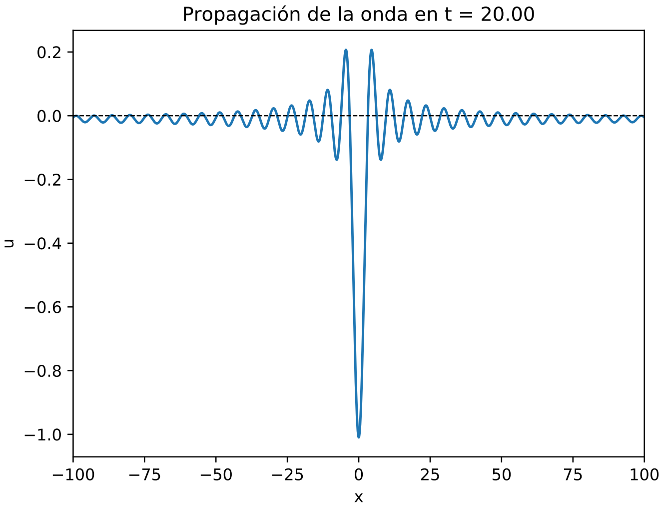

Propagación de la onda en t = 7.50
u
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−100
−75
−50
−25
0
25
50
75
100
x

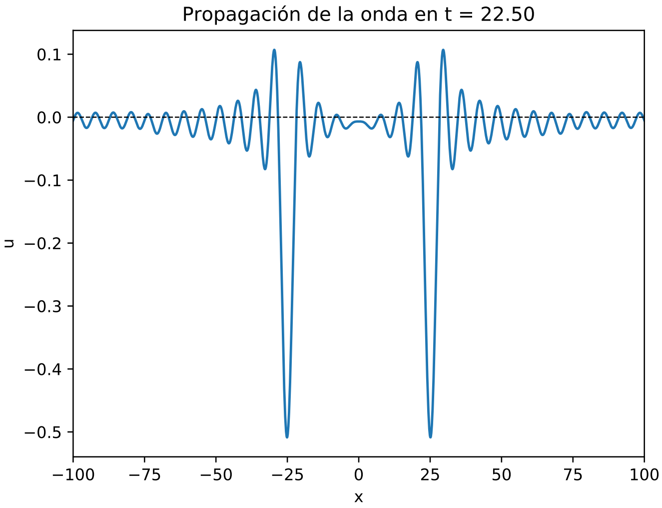

Propagación de la onda en t = 10.00
0.000
−0.005
−0.010
−0.015
−0.020
−0.025
u
−100
−75
−50
−25
0
25
50
75
100
x

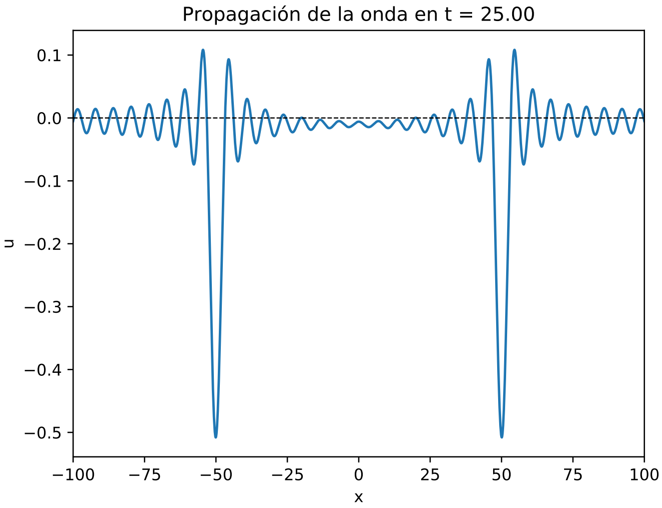

Propagación de la onda en t = 12.50
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
u
−100
−75
−50
−25
0
25
50
75
100
x

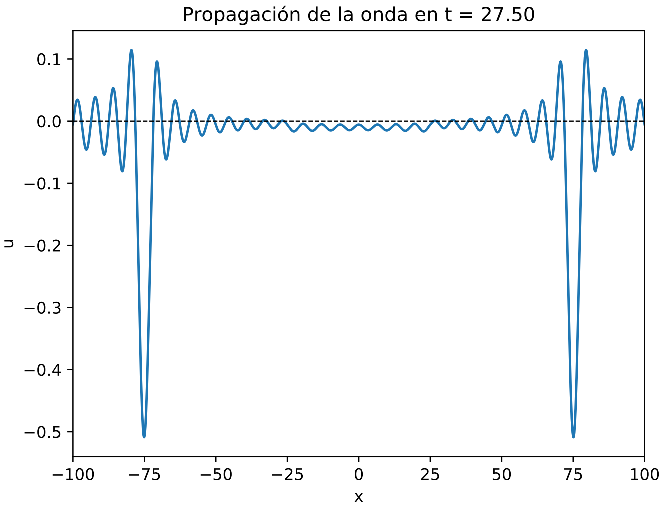

Propagación de la onda en t = 15.00
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
u
−100
−75
−50
−25
0
25
50
75
100
x

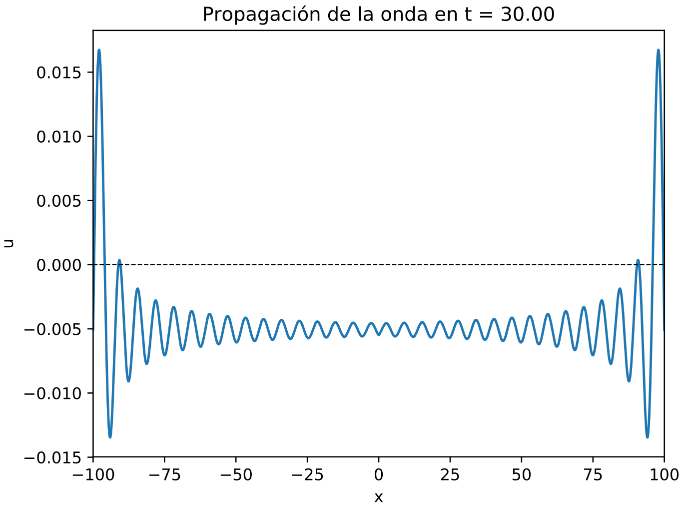

Propagación de la onda en t = 17.50
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
u
−100
−75
−50
−25
0
25
50
75
100
x

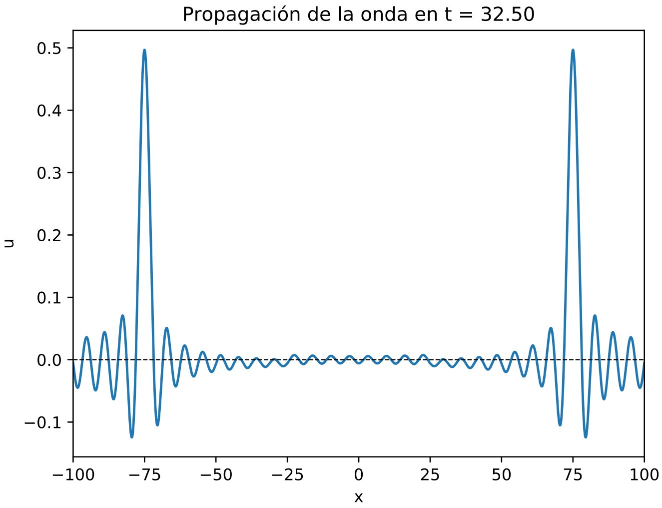

Propagación de la onda en t = 20.00
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
u
−100
−75
−50
−25
0
25
50
75
100
x

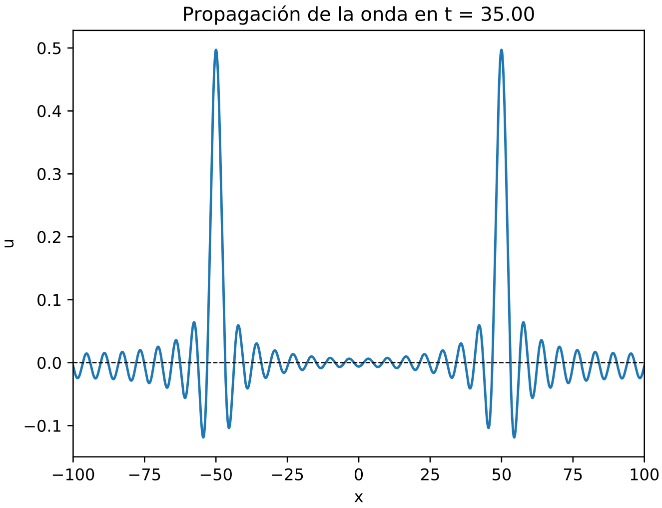

Propagación de la onda en t = 22.50
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
u
−100
−75
−50
−25
0
25
50
75
100
x

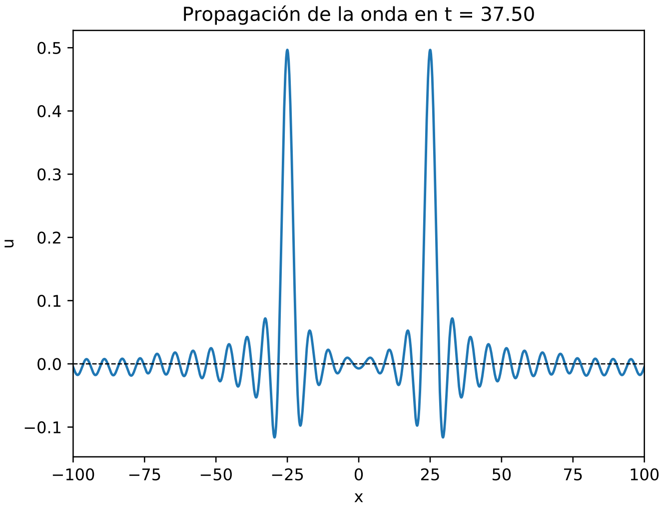

Propagación de la onda en t = 25.00
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
u
−100
−75
−50
−25
0
25
50
75
100
x

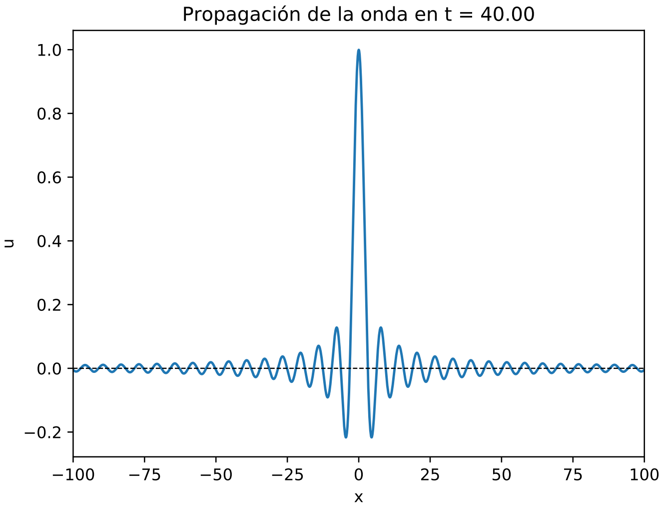

Propagación de la onda en t = 27.50
u
x
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−100
−75
−50
−25
0
25
50
75
100


Propagación de la onda en t = 30.00
u
x
0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
−100
−75
−50
−25
0
25
50
75
100


Propagación de la onda en t = 32.50
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
u
−100
−75
−50
−25
0
25
50
75
100
x


Propagación de la onda en t = 35.00
u
x
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−100
−75
−50
−25
0
25
50
75
100


Propagación de la onda en t = 37.50
u
x
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−100
−75
−50
−25
0
25
50
75
100


Propagación de la onda en t = 40.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
u
−100
−75
−50
−25
0
25
50
75
100
x

Solución completa

Con lo que hemos revisado, se le ha dado solución a la ecuación de onda, una EDP hiperbólica, mediante el esquema de diferencias finitas.

Ejercicio a cuenta

Utilizando la misma configuración del problema anterior de la ecuación de onda, ahora configura el paquete de onda inicial, utilizando comparativamente los anchos de intervalo de número de onda $\triangle\, k = 0.5$, 1 y 5.

Ejercicio a cuenta

Discute el efecto de cambiar $\Delta\, k$ y utiliza en cada caso la amplitud del paquete de onda final, que se reconstruye en el origen, para obtener un estimado del error.

Correlaciona la estimación del error con la media del ancho de los paquetes de onda iniciales.