"El saber de mis hijos hará mi grandeza"

Universidad De Sonora

División de Ciencias Exactas y Naturales

Licenciatura en Física

Física Computacional I

Actividad 11

"Solución Numérica de la Ecuación de Onda."

Ismael Espinoza Arias

Profesor Carlos Lizárraga Celaya

Hermosillo, Sonora a abril 15 de 2021

1 Ecuación de onda

Consideremos una perturbación, generada en una cuerda, que viaja hacia la derecha con una rapidez constante u. Si convenimos en denotar por x a las diferentes posiciones a lo largo de la dirección de propagación de la perturbación y por y a las posiciones perpendiculares a ésta, la forma de la perturbación quedará descrita por una función y = f(x,t). Si durante la propagación de la perturbación no existe disipación de energía, la forma y el tamaño de la perturbación no cambiarán a medida que ésta se desplaza. (En particular, la propagación de la luz en el espacio vacío cumple con este requisito.) En tal caso se debe cumplir necesariamente que:

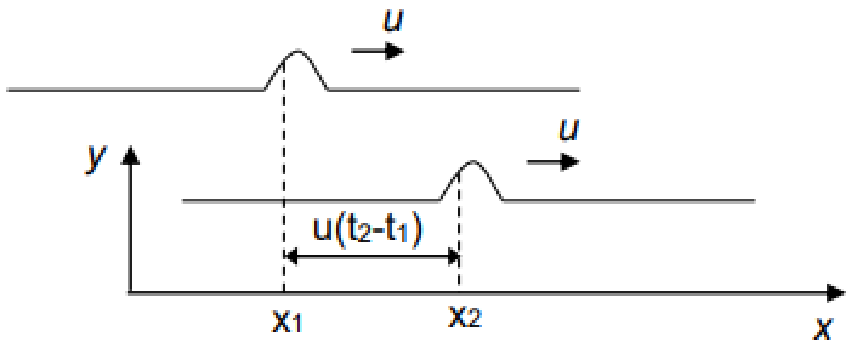


Figura 6.1. Una perturbación ondulatoria en una cuerda.

en donde $x2 = x1 + u(t2 - t1), yaqueu(t2 - t1)$ es la distancia que ha avanzado la perturbación en el intervalo t2 - t1. Este requisito se cumple si la función que describe a la perturbación tiene la forma:

$$f(x,t) = y(x - ut)$$

ya que entonces:

$$f(x2,t2) = y(x2 - ut2) = y[x1 + u(t2 - t1) - ut2] = y(x1 - ut1) = f(x1,t1)$$

La ecuación de onda es una ecuación diferencial parcial de segundo orden en el tiempo y las coordenadas espaciales y tiene la forma

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + f(x,y,z,t)$$

donde c^2 es la velocidad de propagación de la información. La función $u(x,y,z,t)$ representa la presión en una onda acústica, la intensidad de un campo electromagnético, el desplazamiento respecto a un nivel de referencia como lo puede ser la amplitud de una onda superficial en la superficie del agua o el desplazamiento respecto a la horizontal de una cuerda vibrante.

En una dimensión, por ejemplo el caso de una cuerda vibrante, la Ecuación de Onda se simplifica a

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} \right) + f(x,t) \qquad x \in (0,L], t \in (0,T]$$

Requerimos definir 4 condiciones: 2 iniciales (derivada de segundo orden en t) y 2 a la frontera (segundo orden en el espacio), para encontrar la solución.

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

Se requiere también especificar el valor de la constante c y la función $I(x)$.

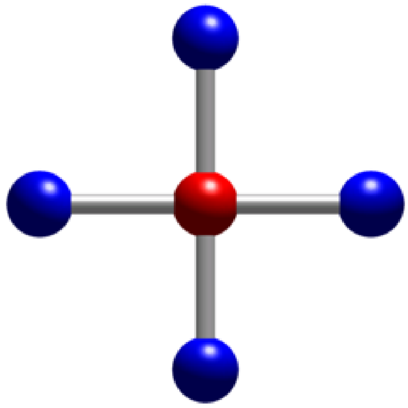
2 Solución de la Ecuación de Onda en una dimensión por el Método de Diferencias Finitas.

Podemos comenzar aproximando las segundas derivadas por diferencias finitas centradas de segundo orden.

Si h es el incremento en la dirección $x = \Delta x$ y $k = \Delta t$ es el incremento en el tiempo. Entonces en un punto de la malla discreta (x,t) tendremos

$$\frac{u(x,t+k) - 2u(x,t) + u(x,t-k)}{k^2} = c^2 \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2}$$

La ecuación anterior define un esténcil computacional de 5 puntos y se respresenta como:



Importación de bibliotecas

Empezamos importando nuestras bibliotecas, el primer paso para empezar con cualquier actividad.

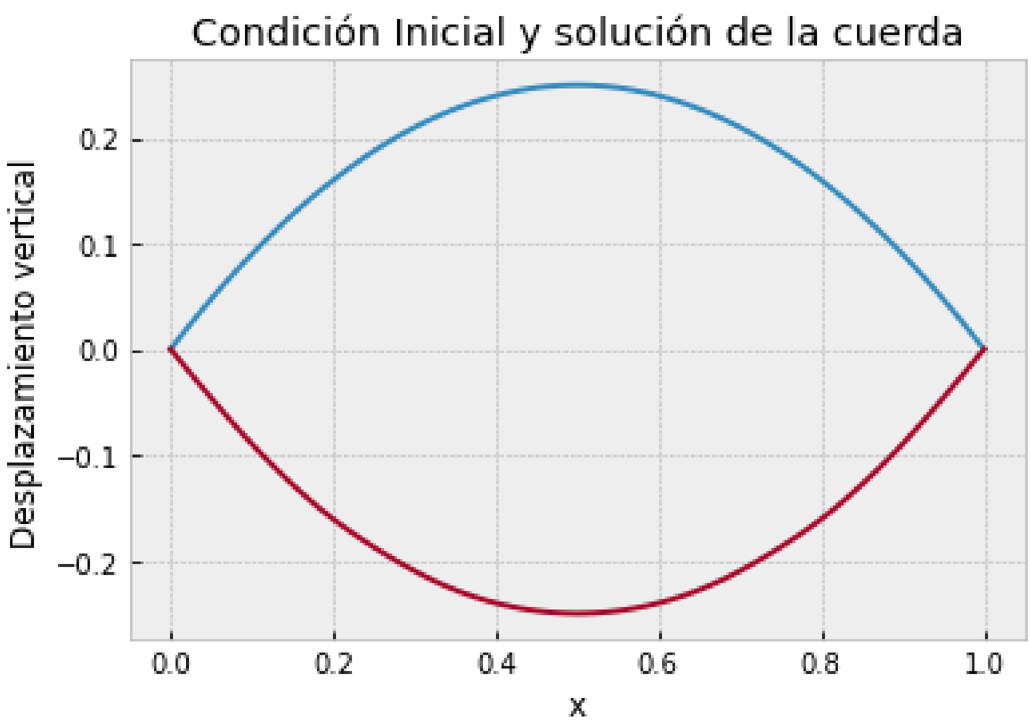
```
# Importamos algunas bibliotecas necesarias

import numpy as np
import matplotlib.pyplot as plt
from ipywidgets import interact
from numpy import linspace, zeros

%matplotlib inline
plt.style.use('bmh')
```

Donde podemos ver que las bibliotecas utilizadas, son numpy, scipy, matplotlib, toolkits, ect. Donde nos ayudaran a resolver nuestro siguiente problema.

Realización de la actividad

Ahora para la actividad, vamos a definir nuestros valores y resolver la ecuación de onda usando ecuaciones diferenciales parciales.

```python
def solver(I, V, f, c, L, Nx, C, T, user_action=None):
    """Resuelve u_tt=c^2*u_xx + f sobre (0,L)x(0,T]."""
    x = linspace(0, L, Nx+1) # Malla en x
    dx = x[1] - x[0]
    dt = C*dx/c
    Nt = int(round(T/dt))
    t = linspace(0, Nt*dt, Nt+1) # Malla en t
    C2 = C**2 # Simplificar la ecuación

    if f is None or f == 0 :
        f = lambda x, t: 0

    if V is None or V == 0:
        V = lambda x: 0
    u = zeros(Nx+1) # Solución al nuevo tiempo
    u_1 = zeros(Nx+1) # Solución 1 paso atrás en el tiempo
    u_2 = zeros(Nx+1) # Solución 2 pasos atrás en el tiempo
```



Ejercicio 1

Modifique el algoritmo de diferencias finitas empleado anteriormente y resuelva la ecuación de onda amortiguada en una dimensión, dada por la ecuación

$$\frac{\partial^2 u}{\partial t^2} + b\frac{\partial u}{\partial t} = c^2\left(\frac{\partial^2 u}{\partial x^2}\right) \qquad x \in (0,L], t \in (0,T]$$

donde $b \geq 0$ y c son constantes.

Se proporcionan las condiciones iniciales y a la frontera para encontrar la solución.

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

Utilice diferencias finitas centradas de segundo orden para aproximar la primer derivada $\partial u/\partial t$.

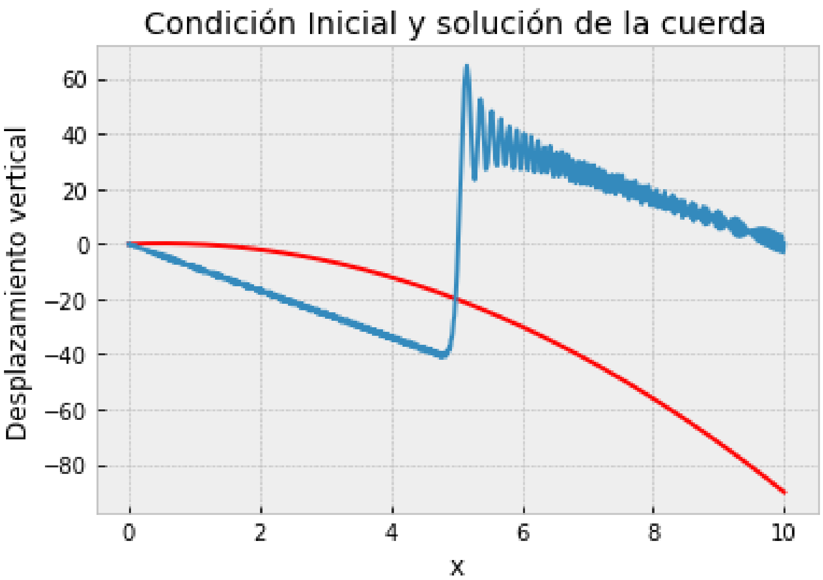
$$\frac{\partial}{\partial t}u(x,t) \approx \frac{u(x,t+k) - u(x,t-k)}{2k}$$

Suponga las mismas características del ejemplo presentado anteriormente $L = 10$, $c = 100\text{m/s}$, $t = (0,0.25)$, y coeficiente de amortiguamiento $b = 0.5$ con condiciones iniciales $u(x,0) = x(1-x)$ y $\partial u(x,0)/\partial t = 0$ y condiciones a la frontera $u(0,t) = u(L,t) = 0$.

Realizamos la resolución de nuestro ejercicio.

```
# Se define una función que resuelve la Ecuación de Onda

def solver(b,I, V, f, c, L, Nx, C, T, user_action=None):
    """Resuelve u_tt=c^2*u_xx + f sobre (0,L)x(0,T]."""
    x = linspace(0, L, Nx+1) # Malla en x
    dx = x[1] - x[0]
    dt = C*dx/c
    Nt = int(round(T/dt))
    t = linspace(0, Nt*dt, Nt+1) # Malla en t
    C2 = C**2 # Simplificar la ecuación
    k=dt
    C3=b*k/2
    C1=(1/(1+(b*k/2)))
    C4=(1/(1+C1*C3+C1))
    #C3=1/(1/)
    if f is None or f == 0 :
        f = lambda x, t: 0
```

Donde llegamos al siguiente resultado:



Ejercicio 2

Haga el desarrollo del algoritmo de diferencias finitas centradas para resolver la ecuación de onda en 1 dimensión si se tiene un término de forzamiento $f(x,t)$

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2}\right) + f(x,t) \qquad x \in (0,L], t \in (0,T]$$

Con las condiciones iniciales y a la frontera para encontrar la solución.

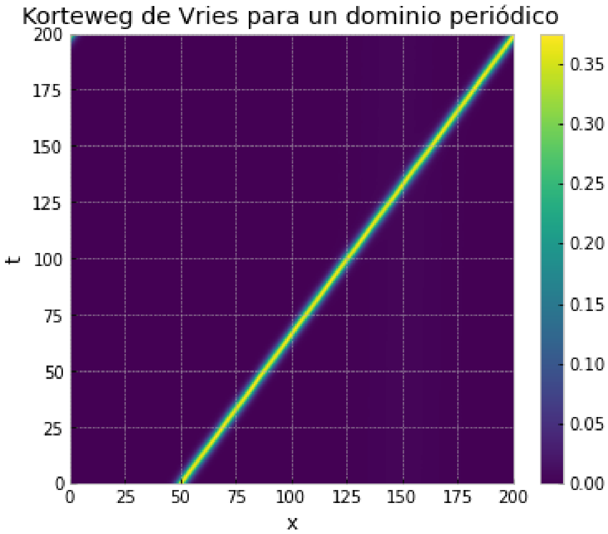
$$
\begin{aligned}
u(x,0) &= I(x) \\
\frac{\partial}{\partial t} u(x,0) &= 0 \\
u(0,t) &= 0 \\
u(L,t) &= 0
\end{aligned}
$$

Realizamos la resolución de nuestro ejercicio.

```
# Se define una función que resuelve la Ecuación de Onda

def solver(I, V, f, c, L, Nx, C, T, user_action=None):
    """Resuelve u_tt=c^2*u_xx + f sobre (0,L)x(0,T]."""
    x = linspace(0, L, Nx+1) # Malla en x
    dx = x[1] - x[0]
    dt = C*dx/c
    Nt = int(round(T/dt))
    t = linspace(0, Nt*dt, Nt+1) # Malla en t
    C2 = C**2 # Simplificar la ecuación

    if f is None or f == 0 :
        f = lambda x, t: 0

    if V is None or V == 0:
        V = lambda x: 0
    u = zeros(Nx+1) # Solución al nuevo tiempo
    u_1 = zeros(Nx+1) # Solución 1 paso atrás en el tiempo
    u_2 = zeros(Nx+1) # Solución 2 pasos atrás en el tiempo
```

Donde llegamos al siguiente resultado:

```
Calculando la solución.
Graficando una representación de la interacción de 2 ondas solitarias.
```



Ejercicio 3

Resuelva la Ecuación KdV, para el caso de 2 solitones comenzando en $x01 = 0.25 * L$ y $x02 = 0.75 * L$, con velocidades $c1 = 0.75$ y $c2 = 0.01$ e integre hasta que una de las ondas llegue a la frontera.

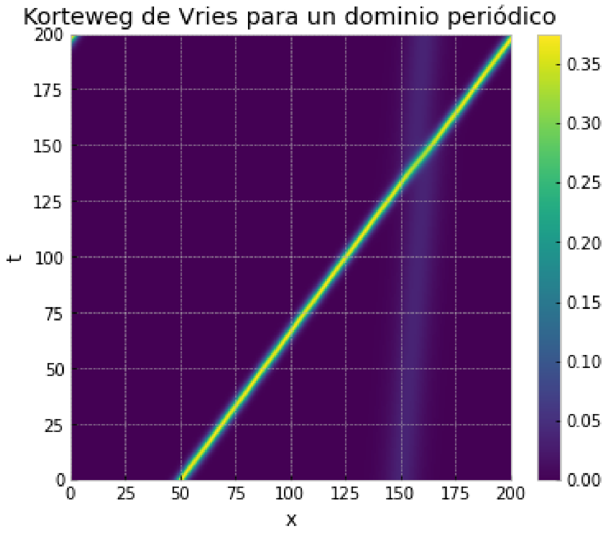
Grafique las soluciones como en el ejemplo que se proporcionó.

Realizamos la resolución de nuestro ejercicio.

```
#Solución de la Ecuación de KdV en un dominio periódico

import numpy as np
from scipy.integrate import odeint
from scipy.fftpack import diff as psdiff
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.axes_grid1 import make_axes_locatable

def kdv_exact(x, c):
    """Perfil de la solución exacta de la Ecuación KdV: 1 solitón sobre el eje real"""
    u = 0.5*c*np.cosh(0.5*np.sqrt(c)*x)**(-2)
    return u

def kdv(u, t, L):
    """Las ecuaciones diferenciales de la ecuación KdV, discretizada en x"""
    # Calcula las derivadas en x usando un método pseudoespectral (Transformada de Fourier)
    # Supone tener condiciones periódicas en la dirección x
    ux = psdiff(u, period=L)
    uxxx = psdiff(u, period=L, order=3)
```

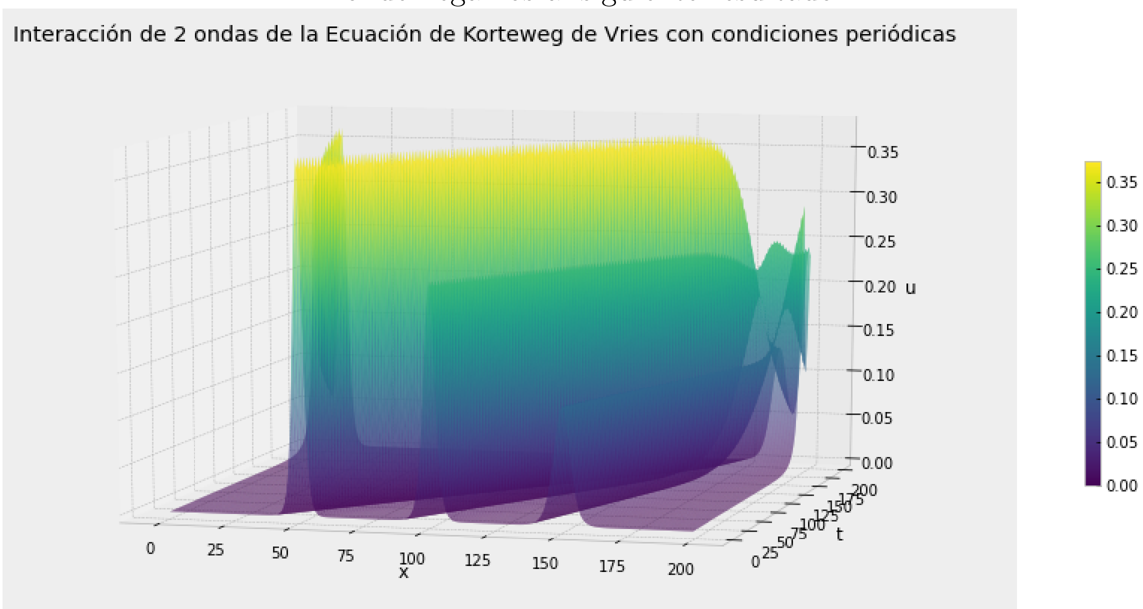
Donde llegamos al siguiente resultado:

```
Calculando la solución.
Graficando una representación de la interacción de 2 ondas solitarias.
```




```python
# Gráfica tipo superfices en 3D en evolución

fig = plt.figure(figsize=(16,8))
ax = fig.add_subplot(111, projection='3d')

SX, ST = np.meshgrid(x, t)
# Ver mapa de colores de Matplotlib
# https://matplotlib.org/stable/tutorials/colors/colormaps.html
ax.plot_surface(SX, ST, sol, cmap='viridis', rstride=1, cstride=1, alpha=0.75)

# Grafica una tabla de valores
m = cm.ScalarMappable(cmap='viridis')
m.set_array(sol)
plt.colorbar(m,shrink=0.5)
```

Donde llegamos al siguiente resultado:



Ejercicio 4

Resuelva la Ecuación KdV, para el caso de 3 solitones comenzando en $x01 = 0.25 * L$, $x02 = 0.5 * L$, y $x03 = 0.75 * L$, con velocidades $c1 = 0.75$, $c2 = 0.5$ y $c3 = 0.25$ e integre hasta que una de las ondas llegue a la frontera.

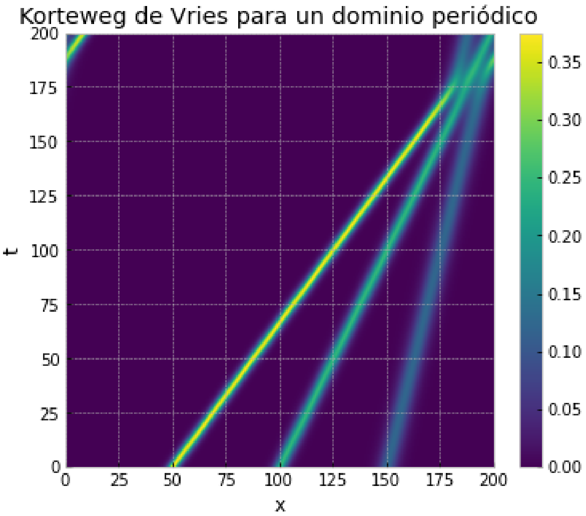
Grafique las soluciones.

Realizamos la resolución de nuestro ejercicio.

```
#Solución de la Ecuación de KdV en un dominio periódico

import numpy as np
from scipy.integrate import odeint
from scipy.fftpack import diff as psdiff
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.axes_grid1 import make_axes_locatable

def kdv_exact(x, c):
    """Perfil de la solución exacta de la Ecuación KdV: 1 solitón sobre el eje real"""
    u = 0.5*c*np.cosh(0.5*np.sqrt(c)*x)**(-2)
    return u
```

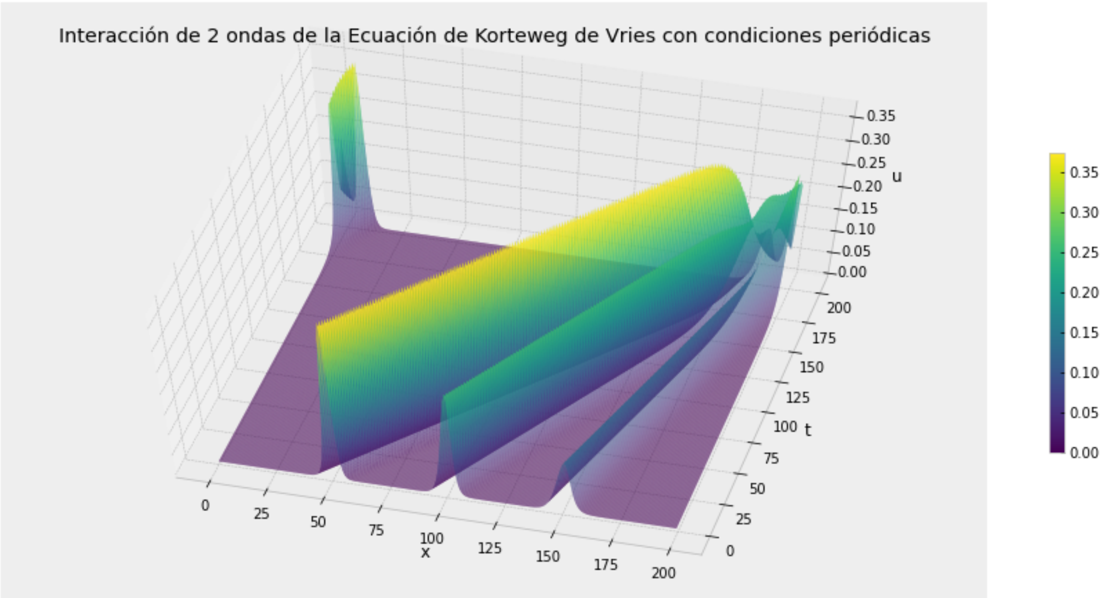
Donde llegamos al siguiente resultado:

```
Calculando la solución.
Graficando una representación de la interacción de 2 ondas solitarias.
```



```
# Gráfica tipo superfices en 3D en evolución

fig = plt.figure(figsize=(16,8))
ax = fig.add_subplot(111, projection='3d')

SX, ST = np.meshgrid(x, t)
# Ver mapa de colores de Matplotlib
# https://matplotlib.org/stable/tutorials/colors/colormaps.html
ax.plot_surface(SX, ST, sol, cmap='viridis', rstride=1, cstride=1, alpha=0.75)

# Grafica una tabla de valores
m = cm.ScalarMappable(cmap='viridis')
m.set_array(sol)
plt.colorbar(m,shrink=0.5)

ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u')
ax.view_init(elev=60, azim=-75)
ax.set_title('Interacción de 2 ondas de la Ecuación de Korteweg de Vries con condiciones periódicas')
plt.show()
```

Donde llegamos al siguiente resultado:



Ejercicio 5

En el ejemplo resuleto anterior, se mostró la evolución de la condición inicial

$$u_0^{(2,1)}(x,y,0) = sin(\pi x)\sin(\frac{\pi y}{2})$$

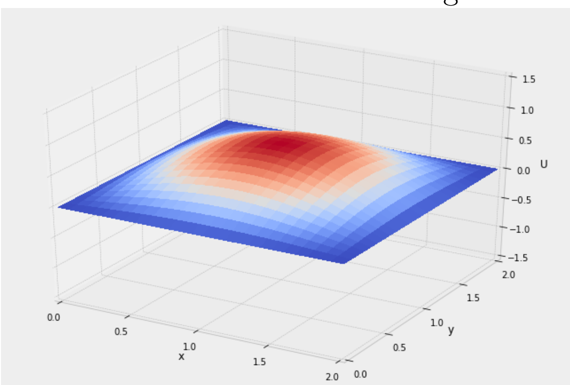
mostrando el *modo (1,2)* de oscilación natural de la membrana ([Ver estas animaciones](https://www.acs.psu.edu/drussell/Demos/MembraneSquare/Square.html)).

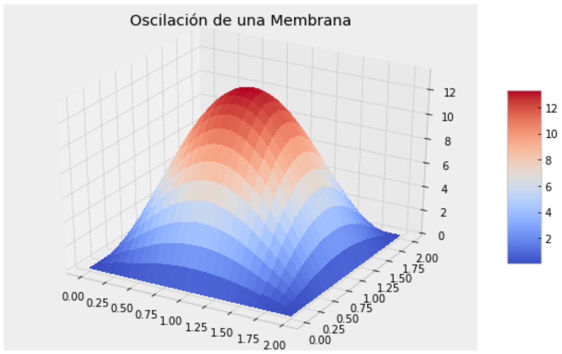
En este Ejercicio se pide mostrar la evolución del *modo (1,1)*, con la condición inicial

$$u_0^{(1,1)}(x,y,0) = \sin(\frac{\pi x}{2})\sin(\frac{\pi y}{2})$$

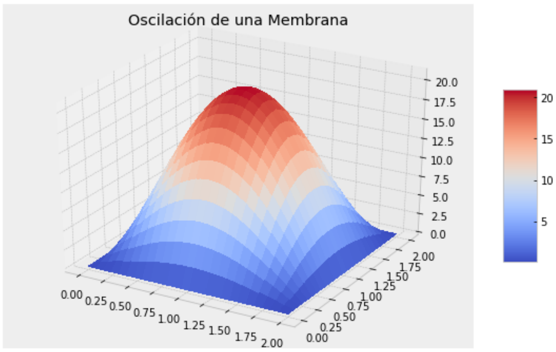
```
#Solución de la Ecuación de Onda en 2D: Utt = (Uxx+Uyy)
# Método de Diferencias Finitas

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.axes_grid1 import make_axes_locatable
%matplotlib inline

# Se define la Clase general que incluye todas las funciones
class WaveEquationFD:
```

Donde llegamos a la siguientes gráficas:

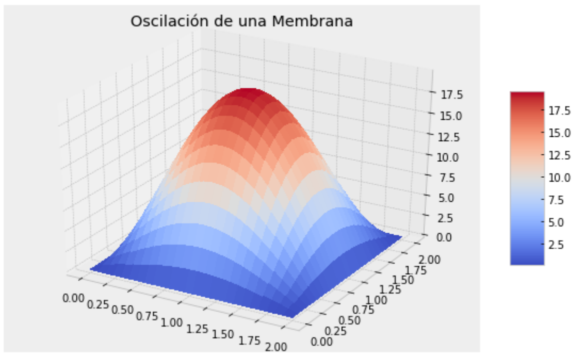




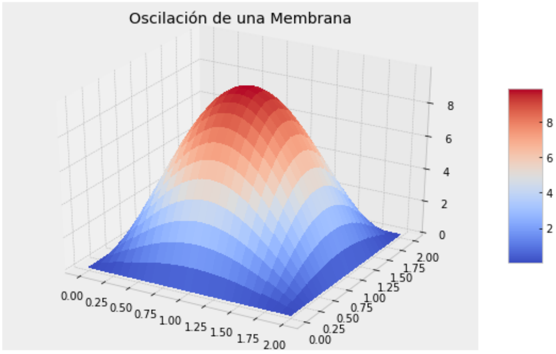
paso: 20.0 t= 0.585



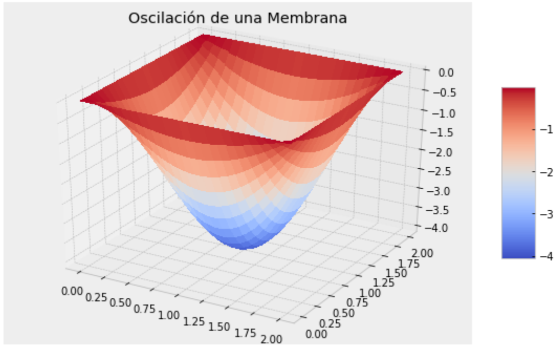
paso: 30.0 t= 0.885



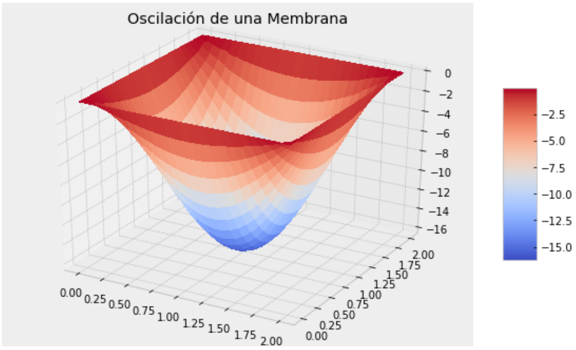
paso: 40.0 t= 1.185



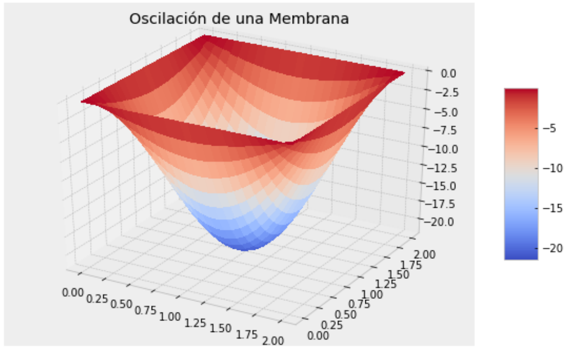
paso: 50.0 t= 1.4849999999999999



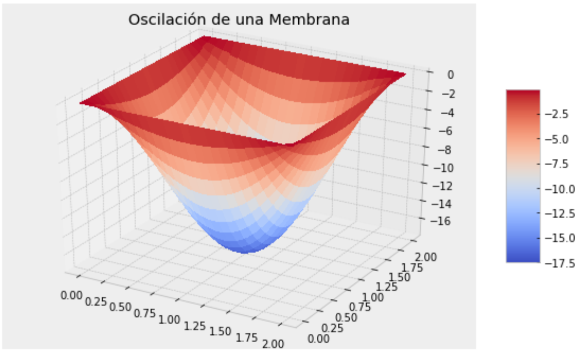
paso: 60.0 t= 1.785



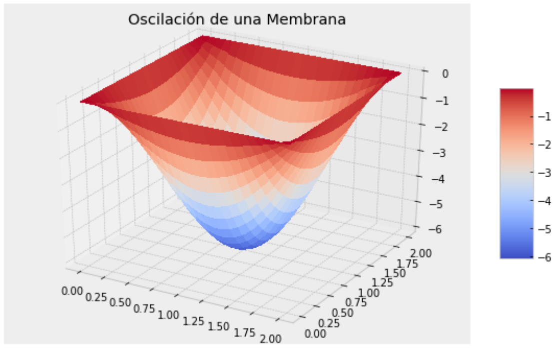
paso: 70.0 t= 2.085



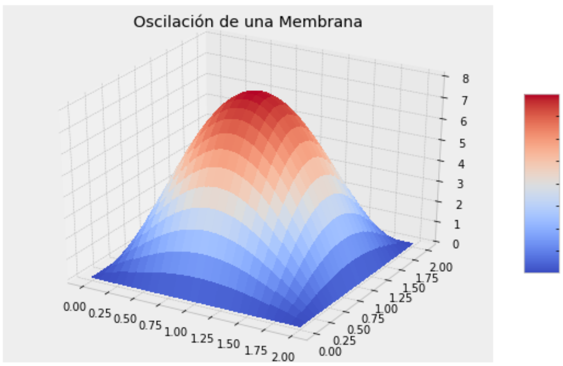
paso: 80.0 t= 2.385



paso: 90.0 t= 2.685



paso: 100.0 t= 2.985



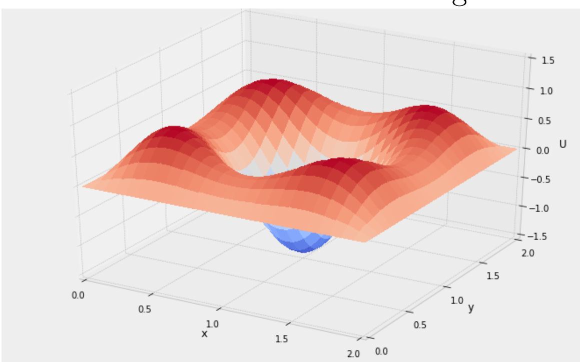
Ejercicio 6

En el mismo contexto que el problema anterior, muestra la evolución de la superposición *modos (3,1)+ (1,3)* dada la condición inicial

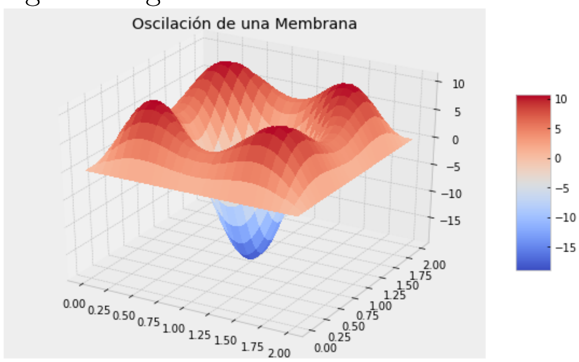
$$u_0^{(3,1)+(1,3)}(x,y,0) = \sin(\frac{3\pi x}{2})\sin(\frac{\pi y}{2}) + \sin(\frac{\pi x}{2})\sin(\frac{3\pi y}{2})$$

```python
#Solución de la Ecuación de Onda en 2D: Utt = (Uxx+Uyy)
# Método de Diferencias Finitas

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.axes_grid1 import make_axes_locatable
%matplotlib inline

# Se define la Clase general que incluye todas las funciones
class WaveEquationFD:

    def __init__(self, N, D, Mx, My):
        self.N = N
        print('N:',N)
        self.D = D
        print('D:',D)
        self.Mx = Mx
        print('Mx:',Mx)
        self.My = My
        print('My',My)
        self.tend = 6
```

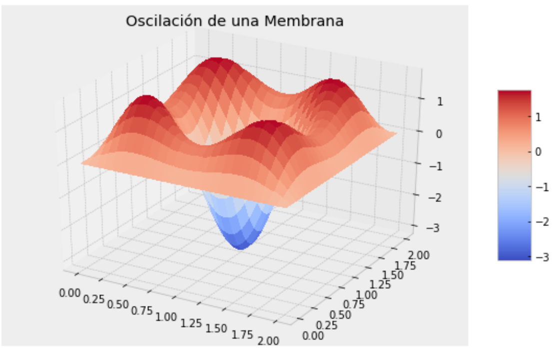
Donde llegamos a la siguientes gráficas:



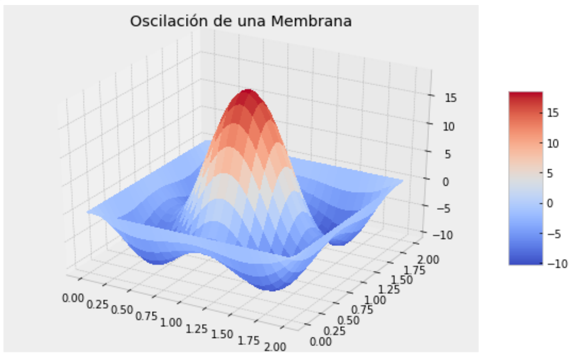
paso: 10.0 t= 0.285



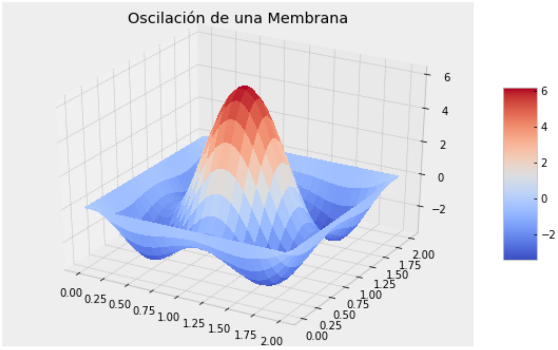
paso: 20.0 t= 0.585



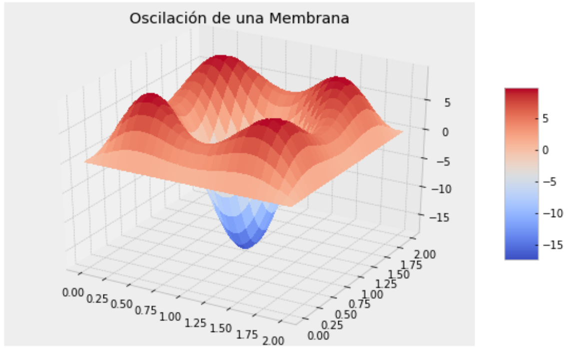
paso: 30.0 t= 0.885



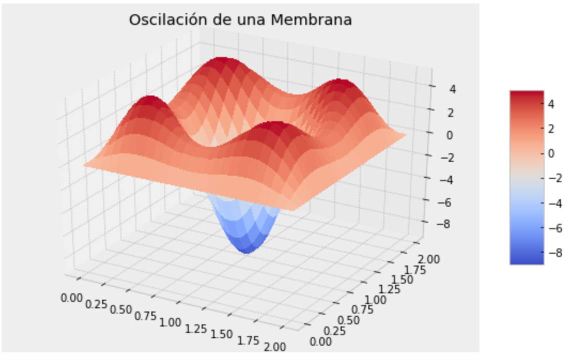
paso: 40.0 t= 1.185



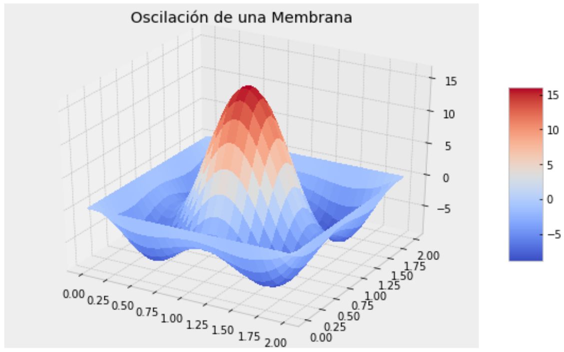
`paso: 50.0 t= 1.4849999999999999`



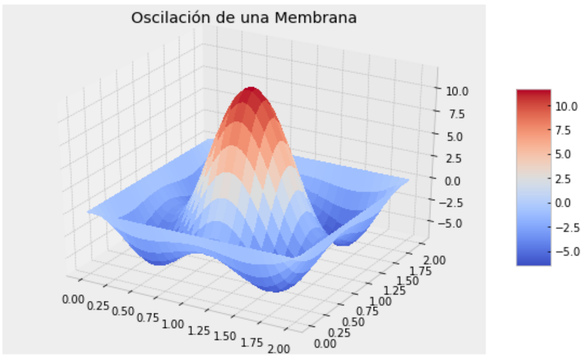
`paso: 60.0 t= 1.785`



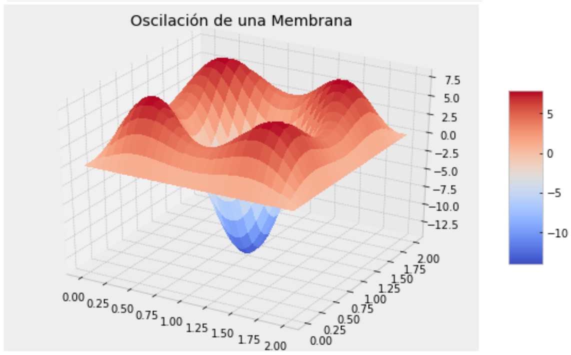
`paso: 70.0 t= 2.085`



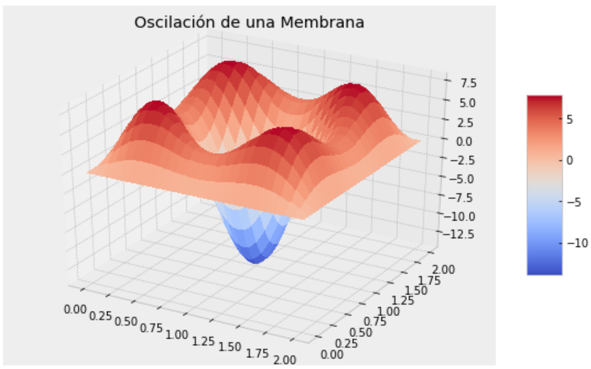
`paso: 80.0 t= 2.385`



`paso: 90.0 t= 2.685`



`paso: 100.0 t= 2.985`



Conclusiones, resultados y discusión

En esta activiadad se reforzaron los conocimientos adquridos en las clases de teoría, impartidas por el docente en el transcruso de la semana. En esta ocasión resolvimos y aplicamos algunos ejemplos de la ecuación de calor, donde graficamos las soluciones viendo como es que estas se comportan. También vimos 2 métodos aplicamos para realizar este ejercicio y de una forma utilizando las diferencias finitas.

Bibliografía

- *Wave Equation. (s. f.). Hyperphysics. Recuperado 21 de abril de 2021, de http://hyperphysics.phy-astr.gsu.edu/hbasees/Waves/waveq.html*

- *Tomé, C. (2018, 20 noviembre). Fase y ecuación de onda. Cuaderno de Cultura CientÃfica. https://culturacientifica.com/2018/11/20/fase-y-ecuacion-de-onda/.*

- *Ecuación de Onda. (s. f.). Ecuación de Onda. Recuperado 21 de abril de 2021, de https://www.compadre.org/osp/EJSS/4446/240.htm.*

Apéndice

1. ¿Qué aprendiste nuevo?
   Aprendí como se resulve la ecuación de calor con el uso de difrencias finitas y aplicaciones y gráficas que llaman la atención visualmente.

2. ¿Qué fue lo que más te llamó la atención de la ecuación del calor?
   Su resolución, ya ecuaciones diferenciales parciales aplicadas en un campo específico me gustan y me llaman mucho la atención, es por eso que como la ecuación de onda se resuelve de esta forma, me gusta.

3. ¿Qué mejorarías en esta actividad?
   Tratar de referenciar con información mas digerible rapidamente, podria decir.

4. ¿Algún comentario adicional antes de dejar de trabajar en Jupyter con Python?
   Es un lenguaje de los más usados y muy extenso, pero presiento que conocí una parte valiosa de él.

5. Cerramos la parte de trabajo con Python ¿Que te ha parecido?
   Fue una buena experiencia trabajar con este lenguaje, pues, es muy fácil de manejar a diferencia de otros, y logra cosas que otros no pueden o bien sea una tarea dificil,

tal es el caso de métodos numéricos, graficación, solución de sistemas de ecuaciones diferenciales, etc. Todo con la ayuda de sus bibliotecas.