Introducción al CFD - Parte 01

April 28, 2020

1 Introducción a la Dinámica de Fluidos Computacional con el Lenguaje de Programación Python (Clase N01)

1.1 Introducción

Comenzamos teniendo algunas consideraciones en cuanto a conocimientos y tecnologías que ya tengas instaladas en tu ordenador.

- Conocimientos en:
  - Ecuaciones diferenciales parciales.
  - Fundamentos de la mecánica de fluidos.
  - Programación básica en algún lenguaje
  - Métodos numéricos.
- Tecnologías instaladas:
  - Interprete de Python dentro de un entorno virtual (Mediante PIP o anaconda).
  - Qt Designer 5.
  - Git (Mac , GNU/Linux o Windows).

La motivación para este curso ya esta presente en cada uno de ustedes desde el instante que se animaron a llevarlo, constará de varias horas de estudio y posiblmente tengamos que repasar algún concepto, pero ello va bien porque buscamos que el estudiante pueda tener claro los conceptos.

2 CAPITULO N01 Programación Aplicada:

Python es un lenguaje de programación: * Orientado a objetos * Multipropósito * Interpretado * Fácil de aprender * Multiplataforma * Open-source

2.1 Mira a Python como un lenguaje ”pegamento” con otros

Con el tiempo notarás que has aprendido un lenguaje moderno y con un campo enorme de aplicación, aunque debo señalar desde un principio que como tal a este lenguaje lo sostiene su enorme comunidad en todo el mundo, siempre que tengas dudas búsca, un buen lugar al que puedes entrar a consultar es a StackOverFlow, pero ojo, preguntar es un privilegio, sugiero la verdad puedas primero leerte esto ”Cómo hacer preguntas de manera inteligente”; regresando al titulo de este parrafo lo digo porque Python es fabuloso, pero ojo, ”hay programas que se sobre salen en ciertas áreas y pues Python también podría hacer la tarea, pero tienes que evaluar bien”, conocer Python

para mi es como andar sobre un todo-terreno, puedes hacer de todo, incluso hay artificios que podrías llevar, pero al final la decisión vendrá del desarrollador y esto dependerá de su expertís; alguna vez conversando con alguien con mayor experiencia en desarrollo me hizo entender esto; para nuestro curso va bien, así que sigamos adelante.

2.2 Conocimientos previos

2.2.1 Python es un lenguaje interpretado

Significa que a medida que vayas escribiendo tu código este podrá interactuar directamente con el ordenador, las ventajas pues que vas desarrollando y puedes ir ejecutando; esto es algo que no hacen los lenguajes compilados ya que este último tendría que convertir tu código fuente a un archivo binario que solamente es entendido por la CPU; entonces si sabemos que Python es un lenguaje interpretado significa que es mejor que otros lenguajes ¿verdad?, sinceramente esa es una pregunta difícil de responder ya que depende de tus indicadores que pongas, si te refieres a que es mejor en cuanto a legibilidad de lectura, pues si; te resultará fácil de comprender el código, además que como lenguaje interpretado te tomará menos lineas de código para escribir un programa; los lenguajes compilados suelen tener muchas filas; - oye pero si me dices todo esto sigo pensando que Python es un mejor lenguaje - , no , no he terminado de contarte, Python es cierto que tiene un montón de ventajas, pero en cuanto a velocidad de calculo pues queda corto ante los lenguajes compilados y te guste o no siempre será así y es que es lógico, la tarea de compilar ya dije que trata de que tienes que crear un nuevo archivo en binario y por ser algo que el CPU lo entiende más "natural" hace todos sus calculos más rápidos, por ese motivo es que Python queda chico en velocidad, si vas a trabajar resolviendo grandes ecuaciones, meterte a computación de alto rendimiento (HPC) que tal vez puedas pensar que no va tan bien; pero sabes con Python se pueden hacer artificios, podrías por ejemplo escribir cierto código en Fortran y este podría ser ejecutado desde un script principal , llamaríamos a ese archivo Python con F2PY por ejemplo, o sino podríamos paralizar las tareas en el CPU y la GPU para el lenguaje Python con Numba.

2.2.2 Entornos virtuales

Un entorno virtual no es otra cosa que un ambiente de trabajo aislado , te permite contar con una versión del interprete de Python aislado teniendo la opción de poder interactuar con otras librerías que podrás instalar.

¿Por qué necesito un crear un entorno virtual?

Python nació en 1991, desde esa fecha ha venido evolucionando y teniendo muchas mejoras; pero como tal, es cierto que hay librerías que solamente pueden interactuar con ciertas versiones de otras dependencias, bueno se pueden ocasionar conflictos de versiones y para evitar ello creamos los famosos "entornos virtuales", así de simple.

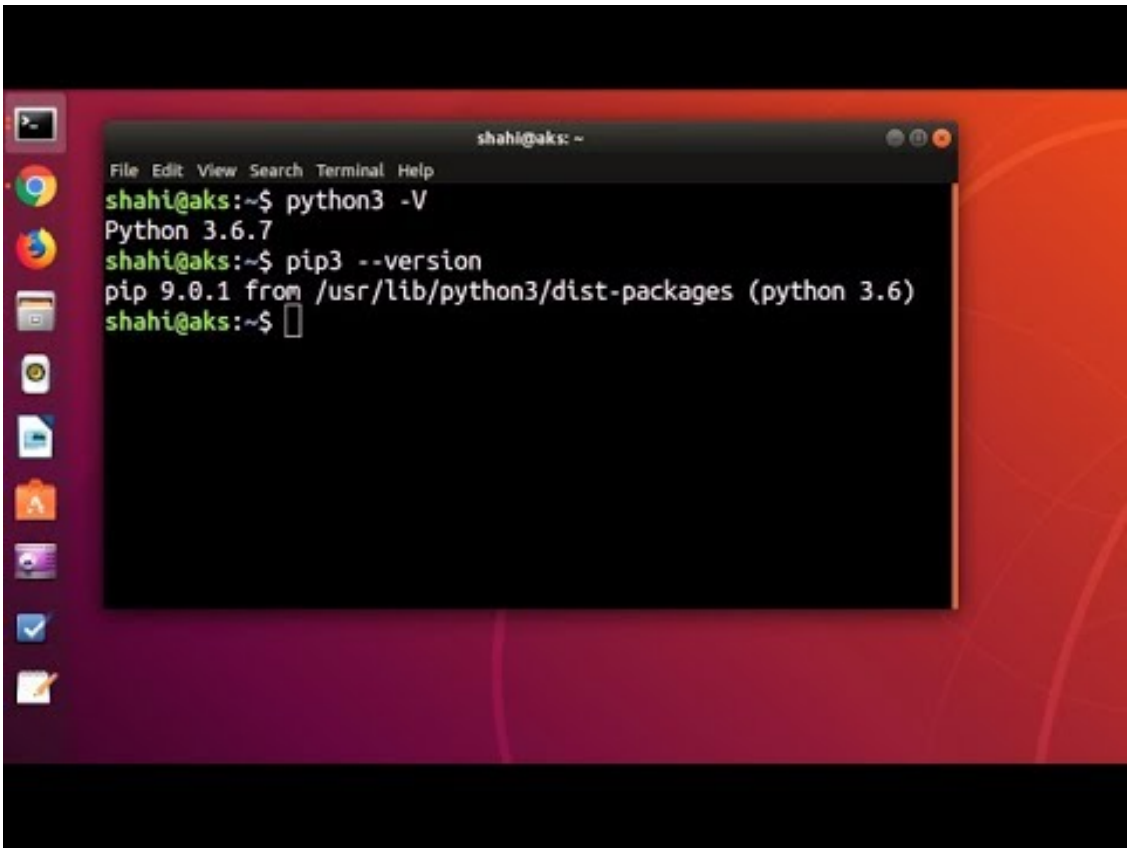
Facilitamos enlace para la gestión de los entornos: * Documentación para entornos virtuales con PIP * Documentación para entornos virtuales con Conda

[1]:
```
from IPython.display import YouTubeVideo
# Vídeo en inglés para instalar Anaconda en Ubuntu OS y derivados
YouTubeVideo('DY0DB_NwEu0')
```

[1]:

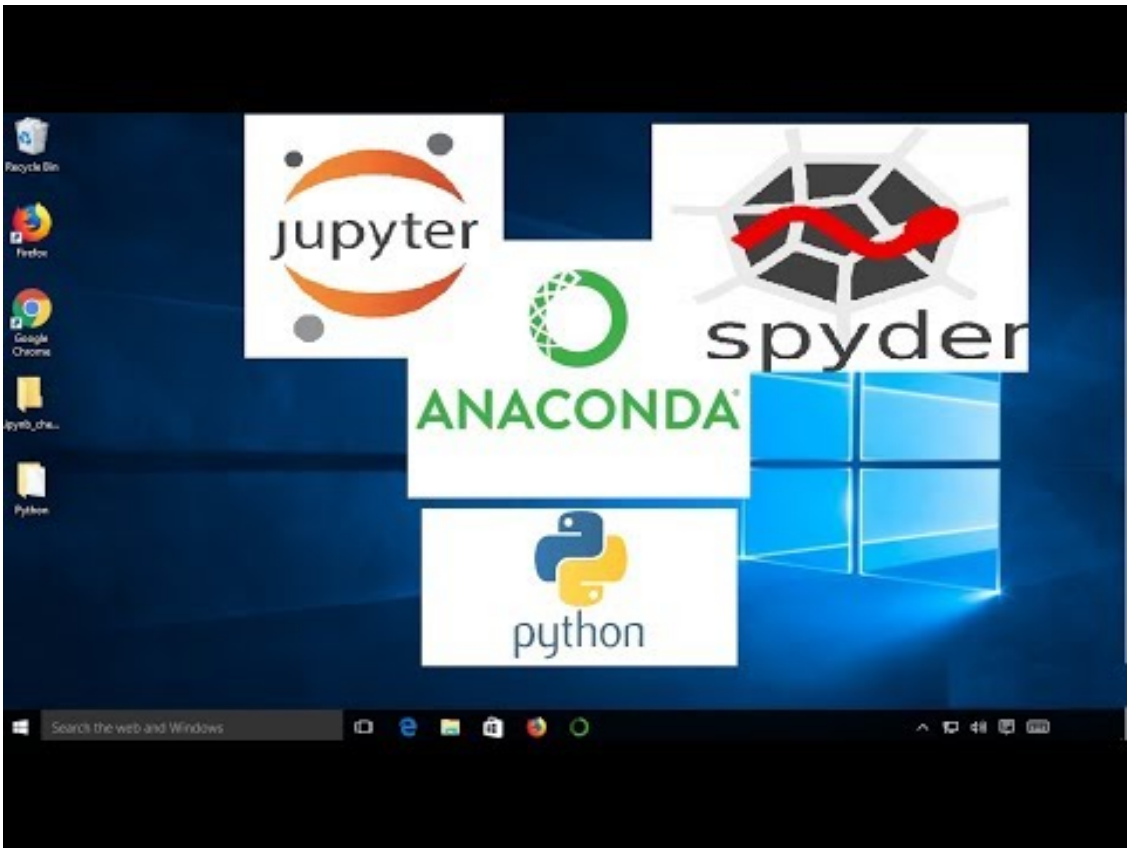
[2]:
```
# Vídeo en inglés para instalar Python3-pip en Ubuntu OS y derivados
YouTubeVideo('FfkPLekXuL4')
```

[2]:



[3]:
```
# Vídeo en inglés para instalar Python3-pip en Ubuntu OS y derivados
YouTubeVideo('5mDYijMfSzs')
```

[3]:



[4]:
```
# Vídeo en inglés para instalar Python3-pip en Ubuntu OS y derivados
YouTubeVideo('gFNApsyhpKk')
```

[4]:

Nota: Los gestores de paquetes para Python más populares son PIP y CONDA, si usas GNU/Linux tu sistema operativo ya tiene todo listo, si eres de Windows necesitarás instalar Python, posiblemente les interese comenzar con Anaconda ya que es relativamente fácil para los "novatos".

2.3 01.01. Variables:

[5]:
```
x1 = 96
x2 = 17
x3 = 35
x4 = 96.
x5 = 17.0
x6 = 17.00
x7 = 'avión'
x8 = 'casaca'
x9 = 'cuaderno'
```

2.4 01.02. Tipos de objetos

[6]:
```
type(x1)
```

[6]:
```
int
```

[7]:
```
print(type(x1))
print(type(x2))
print(type(x3))
print(type(x4))
print(type(x5))
print(type(x6))
print(type(x7))
print(type(x8))
print(type(x9))
```

```
<class 'int'>
<class 'int'>
<class 'int'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'str'>
<class 'str'>
<class 'str'>
```

2.5 01.03. Operadores matemáticos

[8]:
```
12 + 12.
```

[8]:
```
24.0
```

[9]:
```
12 + 12
```

[9]:
```
24
```

[10]:
```
14/7
```

[10]:
```
2.0
```

[11]:
```
14/7.
```

[11]:
```
2.0
```

[12]:
```
15/7
```

[12]: 2.142857142857143

Para la versión 2 del interprete de Python se tenía que agregar la siguiente línea de comando en caso era de interés ejecutar divisiones entre enteros y estos de manera automática se quisiera presente el resultado como un decimal.

[13]:
```
from __future__ import division
```

[14]:
```
14/7
```

[14]: 2.0

2.6 01.04. La identación

A diferencia de otros lenguajes de programación en Python no se utilizan signos de colección para ejecutar bucles o estructuras repetitivas, esto se logra gracias a que se respeta la identación, la idea es que el programador use o las barras espaciadoras o las tabulaciones pero no en simultaneo ambas ya que el hacer esto traería inconsistencia en el código.

2.7 01.05. Esquema For

[15]:
```
for i in range(10):
    print("Bienvenido a este curso, un saludo.")
```

```
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
```

[3]:
```
for i in range(4,10,2):
    print("Bienvenido a este curso, un saludo.")
```

```
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
Bienvenido a este curso, un saludo.
```

2.8 01.06. Esquema If

[4]:
```
a = int(input("Ingrese valor a: "))
b = int(input("Ingrese valor b: "))
if a > b:
  print("a es mayor que b")
if b > a:
  print("b es mayor que a")
```

```
Ingrese valor a: 19
Ingrese valor b: 10
a es mayor que b
```

[5]:
```
# Ejemplo
tproyecto = int(input("En cuantos meses harás ese proyecto?: "))
if tproyecto <= 6:
  print("Ese proyecto durará poco tiempo.")
if tproyecto > 6:
  print("Ese proyecto durará mucho tiempo.")
```

```
En cuantos meses harás ese proyecto?: 8
Ese proyecto durará mucho tiempo.
```

[6]:
```
# Cálculo del impuesto
salario = float(input("Ingrese la cantidad de su salario: "))
x = salario
impuesto = 0
if x > 1800:
  impuesto = impuesto + ((x - 1800) * 0.18)
  x = 1800
if x > 950:
  impuesto = impuesto + ((x - 950) * 0.18)
print("El salario será: s/.%6.2f , Impuesto a pagar: s/.%6.2f" %  (salario,␣
 ↪impuesto))
```

```
Ingrese la cantidad de su salario: 1800
El salario será: s/.1800.00 , Impuesto a pagar: s/.153.00
```

2.9 01.07. Esquema If ... else

[7]:
```
precio = float(input("Ingrese precio base de la laptop que desea consultar por␣
 ↪marca:"))
if precio <= 3600:
  print("Laptop con procesador core i5, 12 GB de Ram, tarjeta Nvidia")
else:
  print("Laptop con procesador core i7, 32 GB de Ram, tarjeta Nvidia")
```

```
Ingrese precio base de la laptop que desea consultar por marca:4000
Laptop con procesador core i7, 32 GB de Ram, tarjeta Nvidia
```

2.10 01.08. Estructuras anidadas

[8]:
```python
# Ejemplo para el recibo del agua:
volumen = float(input("¿Cuál es el volúmen de agua que se utilizó este mes: "))
if volumen < 100:
  precio = 0.20
else:
  if volumen < 200:
    precio = 0.15
  else:
    precio = 0.10
print("Este mes tocará pagar: s/.%6.2f" % (volumen * precio))
```

```
¿Cuál es el volúmen de agua que se utilizó este mes: 700
Este mes tocará pagar: s/. 70.00
```

2.11 01.09. Esquema if... elif ... else

[9]:
```python
zona = int(input("Ingrese la zona del evento: "))
if zona == 1:
  print("Su ingreso es a las 10:00 am por la puerta número 5")
elif zona == 2:
  print("Su ingreso es a las 10:30 am por la puerta número 8")
elif zona == 3:
  print("Su ingreso es a las 11:00 am por la puerta número 6")
else:
  print("Su ingreso es a las 11:30 am por la puerta número 4")
```

```
Ingrese la zona del evento: 3
Su ingreso es a las 11:00 am por la puerta número 6
```

2.12 01.10. Esquema While

[10]:
```python
x = 0
while x <= 10:
  print(x)
  x = x + 1
```

```
0
1
2
3
```

```
4
5
6
7
8
9
10
```

2.13 Guía de estilo al momento de programar en Python PEP - 8

El PEP-8 es una sugerencia que se hace a los nuevos, es decir si vas a comenzar algo es mejor que lo hagas de buena manera y aprendas los hábitos que se recomiendan, pasa que en el mundo del desarrollo puede que te toque tocar código ajeno y para ello deberas entender que es lo que hizo la otra persona, aunque si no se pone un guía de como programar gastarás más tiempo en tratar de entender que empezar a programar, son como 88 normas, sugiero que los leas en:

Guía de estilo PEP-8 en ingles

2.14 01.11. Matrices

Hemos de usar la librería Numpy

[13]:
```
import numpy as np
```

[19]:
```
var01 = np.array([14, 42, 37, 74])
```

[20]:
```
print(var01)
```

```
[14 42 37 74]
```

[21]:
```
var01[3]
```

[21]: 74

[22]:
```
var01[0:4]
```

[22]: array([14, 42, 37, 74])

[23]:
```
var01[0], var01[3]
```

[23]: (14, 74)

[24]:
```
var02 = np.array([1, -3, 0, 1, -5])
```

[29]:
```
var02[-5]
```

[29]: 1

[31]:
```
var02[0:-1]
```

[31]: array([1, -3, 0, 1])

[30]:
```
var02[4:5]
```

[30]: array([-5])

[33]:
```
var02[4]
```

[33]: -5

[31]:
```
var02[4:-1]
```

[31]: array([], dtype=int64)

[35]:
```
var02[-2:-1]
```

[35]: array([1])

2.15 01.12. Array de una dimensión

[44]:
```
var03 = np.linspace(-10, 10, 5)
var03
```

[44]: array([-10., -5.,   0.,   5., 10.])

[35]:
```
print(type(var03))
```

```
<class 'numpy.ndarray'>
```

[36]:
```
print(var03)
```

```
[-10.  -5.   0.   5.  10.]
```

2.15.1 Creamos una copia de var02

1° Con la sintaxis foo02 = foo01 lo que se está creando es una declaración que referencia los valores de foo01 a foo01, se está haciendo un puntero.

[37]:
```
var04 = var03
```

[38]:
```
var04
```

[38]:
```
array([-10.,  -5.,   0.,   5.,  10.])
```

[39]:
```
var04[0]
```

[39]:
```
-10.0
```

[40]:
```
var04[0] = 0
```

[41]:
```
var04
```

[41]:
```
array([ 0., -5.,  0.,  5., 10.])
```

[45]:
```
var03
```

[45]:
```
array([-10.,  -5.,   0.,   5.,  10.])
```

[46]:
```
var03[0]
```

[46]:
```
-10.0
```

[49]:
```
var03[0] = 80
```

[50]:
```
var03
```

[50]:
```
array([80., -5.,  0.,  5., 10.])
```

[51]:
```
var04
```

[51]:
```
array([ 0., -5.,  0.,  5., 10.])
```

Observamos que si cambiamos el valor del primer elemento a var03 en consecuencia simulteanea ocurré lo mismo para var04, por ello se identifica la referenciación que existe entre ambos.

[52]:
```
print(var03)
```

```
[80. -5.  0.  5. 10.]
```

[53]:
```
type(var03)
```

[53]:
```
numpy.ndarray
```

Creamos una copia de var03 como var05

var05[:] = var03[:] Este es el esquema correcto para crear una copia con todos sus elementos para una matriz, aunque antes deberíamos de contar con una matriz de las mismas dimensiones vacía.

Notamos que nos imprime error, ello ocurre porque hacer una copia de esta manera exige que primero se genere una matriz vacía con la misma cantidad de elmeentos de var03, para ello hacemos:

[56]:
```
print(var05)
```

```
␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-56-08a2d84fe924> in <module>
   ----> 1 print(var05)


       NameError: name 'var05' is not defined
```

[57]:
```
print(np.empty_like(var03))
```

```
[80. -5.  0.  5. 10.]
```

[58]:
```
var05 = np.empty_like(var03)
print(var05)
```

```
[80. -5.  0.  5. 10.]
```

[59]:
```
len(var05)
```

[59]: 5

2.15.2 De la librería Numpy: np.empty_like(varX)

Debemos contar con la variable 'a' , esta guarda en memoria a una matriz de dimensiones nxm , lo que se desea es crear una matriz equivalente en las dimensiones, por lo tanto hacemos:

[60]:
```
a = ([418, 7, 3], [-4, -5, 6])
print(a)
```

```
([418, 7, 3], [-4, -5, 6])
```

[61]:
```
np.empty_like(a)
```

[61]:
```
array([[39398272,        0,        0],
       [       0,        0,        0]])
```

La nueva matriz generada a partir de 'a' tiene la misma cantidad de elementos y en las mismas posiciones, aunque los elementos son random

[62]:
```
np.empty_like(var03)
```

[62]:
```
array([80., -5.,  0.,  5., 10.])
```

[63]:
```
var003 = [30, 5, 0, 5, 10]
```

[64]:
```
np.empty_like(var003)
```

[64]:
```
array([          43784848,     140093243260928,                    0,
                       0, 2322214077744902952])
```

Acabamos de descubrir que la función 'np.empty_like(var)' crea una nueva matriz equivalente siempre y cuando sus terminos hayan sido guardados de la forma de números enternos, en el caso sean decimales la nueva matriz será identica a la anterior.

[65]:
```
a = np.array([[1., 2., 3.],[4., 5., 6.]])
```

[66]:
```
np.empty_like(a)
```

[66]:
```
array([[2.06494855e-316, 0.00000000e+000, 0.00000000e+000],
       [0.00000000e+000, 0.00000000e+000, 0.00000000e+000]])
```

[67]:
```
type(a)
```

[67]:
```
numpy.ndarray
```

[68]:
```
np.empty_like(var03)
```

[68]:
```
array([1.5e-322, 2.5e-323, 0.0e+000, 2.5e-323, 4.9e-323])
```

[69]:
```
var05 = np.empty_like(var03)
```

[70]:
```
print(var05)
```

```
[1.5e-322 2.5e-323 0.0e+000 2.5e-323 4.9e-323]
```

Para mejor entendimiento de np.empty_like(a) revisar el siguiente enlace: Documentación desde Scipy

[71]:
```
var05[:] = var03[:]
```

[72]:
```
print(var05)
```

```
[80. -5.  0.  5. 10.]
```

[73]:
```
type(var05)
```

[73]:
```
numpy.ndarray
```

2.16 01.13. Matemática básica con una matriz de 1D en Numpy

[79]:
```
a = [1, 2, 3,]
print(a)
print(a+a)
```

```
[1, 2, 3]
[1, 2, 3, 1, 2, 3]
```

[80]:
```
b = a
type(b)
```

[80]:
```
list
```

[81]:
```
import numpy as np
```

[83]:
```
b = np.array(a)
print(type(b))
print(b)
```

```
<class 'numpy.ndarray'>
[1 2 3]
```

[84]:
```
print(b+b)
print(b + 1)
print(b**2)
print(np.sin(b))
```

```
[2 4 6]
[2 3 4]
[1 4 9]
[0.84147098 0.90929743 0.14112001]
```

[87]:
```
A = 5
print("Imprimir los " + str(A) + " primeros números usando la función 'range'")
for contador in range(A):
    #print( i, end=', ')
    print(contador, end=', ')
```

```
Imprimir los 5 primeros números usando la función 'range'
0, 1, 2, 3, 4,
```

2.17 Framework Qt, el complemento para el desarrollo de software GUI

Qt es un framework multiplataforma para el desarrollo de software, también es posible hacer software embebido para vehículos y equipos industriales.

La metodología que usaremos para crear aplicaciones será que usaremos el Qt Designer donde se usará para ir creando la interfaz de usuario, es posible luego convertir este archivo a uno con extensión .py, eso se logra bajo la sintaxis:

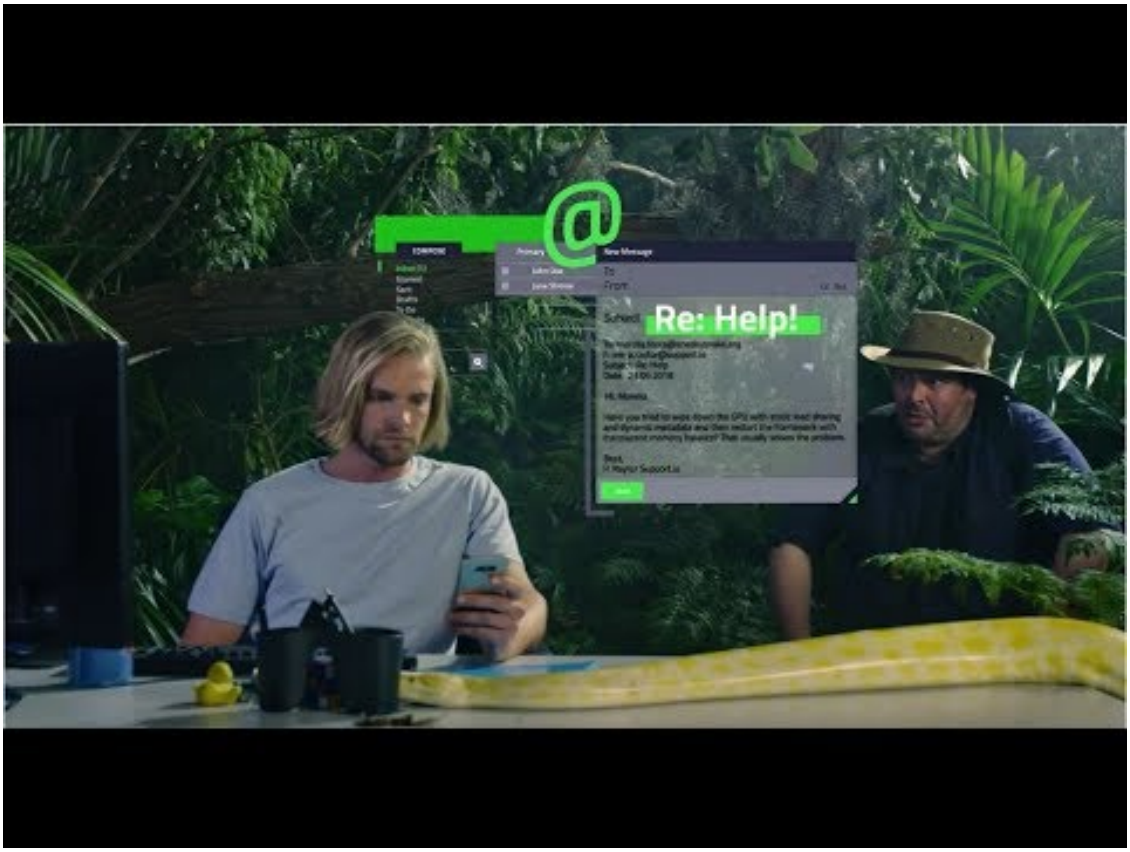
$ pyuic5 file_GUI.ui -o file_GUI.py

En este curso además tendremos la posibilidad de desarrollar nuestras aplicaciones propias aplicaciones, si hablamos del lenguaje Python existen las librerías PyQt, Tkinter, PySide, WxPython, WxPython; en nuestro caso usaremos PyQt en su versión 5 con el cual daremos la funcionalidad a los archivos en formato .ui que nos brinde el Qt Designer.

Para la instalación sugerimos hacerlo desde la página web oficial, prueben la versión open-source.

[74]:
```
from IPython.display import YouTubeVideo

# En este vídeo se muestra un spot que hace Qt a su librería PySide para␣
 ↪trabajar con Python
YouTubeVideo('ic0PnF04N-Q')
```

[74]:



3 CAPITULO N02 Convección

3.1 02.01. Ecuación de convección lineal 1D

Esta es la ecuación más sencilla con la que podremos aprender sobre dinámica de fluidos computacional, resulta interesante ya que es una ecuación pequeña y nos muestra mucho:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

La ecuación anterior manifiestaa la onda inicial con una velocidad c para su propagación. Contando con la condición inicial $u(x,0) = u_0(x)$. El resultado exacto de la ecuación es $u(x,t) = u_0(x - ct)$

La ecuación para resolverse en el ordenador debe discretizarse, usaremos el método de diferencias finitas donde para el tiempo será diferencias finitas adelantadas y para el caso del espacio serán diferencias finitas retrazadas. Para el espacio en el eje x para los puntos de $i = 0$ a N, el tiempo tiene un tamaño de paso Δt .

La ecuación para la coordenada x resulta:

$$\frac{\partial u}{\partial x} \approx \frac{u(x + \Delta x) - u(x)}{\Delta x}$$

Si la discretizamos es:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

Siendo n y $n + 1$ puntos consecutivos en el tiempo; $i - 1$ y i son aquellos puntos vecinos de la coordenada x discretProgramaciónizada. Si se brindan las condiciones iniciales, entonces la única incógnita en esta discretización queda ser u_i^{n+1}. Se logra resolver la incógnita y tener una ecuación que nos permita continuar en el tiempo, de la siguiente manera:

$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n)$$

Implementamos esto para el lenguaje Python y así mismo haremos nuestra GUI.

3.1.1 Esquema que seguiremos

La ecuación lineal de convección 1-D en su forma discretizada habíamos hecho la descomposición para la derivada parcial de u_i^{n+1} con respecto al tiempo ya que de esta manera hacemos diferencias finitas adelantadas, quedando:

$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n)$$

Por ello necesitamos como variables de entrada a:

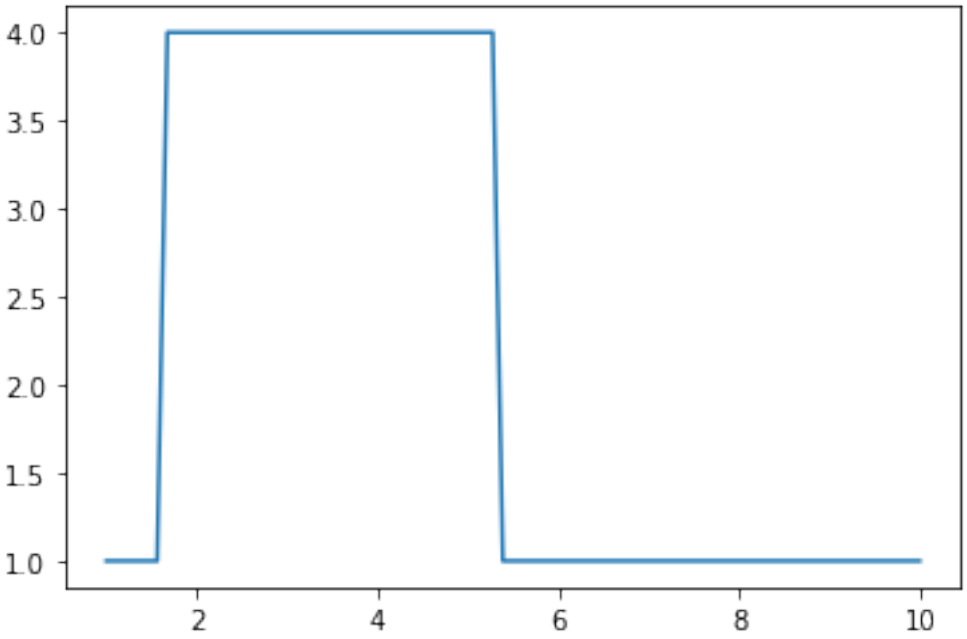
```
# Librerías
import numpy as np # Numpy contraída como np
import matplotlib.pyplot as plt # Matplotlib contraída como plt
```

Creamos una red de puntos uniforme con lo cual se llega a definir el dominio para 4 unidades de longitud de ancho x = [0,4], nx es el número de puntos para la malla, dx es el espacio entre cada punto de la malla.Ç

[111]:
```
import numpy as np # Numpy contraída como np
import matplotlib.pyplot as plt # Matplotlib contraída como plt


x = 10. # Longitud en el eje X
nx = 81 # Número de divisiones que se le hace a x
dx = x/(nx-1)
nt = 200 #Tiempo t total
dt = 0.010 # Paso del tiempo t
c = 1. # Velocidad de la onda
ue = 4
x1 = 0.8
x2 = 4.8

# Condiciones iniciales
u = np.ones(nx)      # numpy función ones() creando un array de 1 con nx␣
 ↪elmentos
u[int(x1/dx) : int(x2/dx+1)]=ue
print("Esto es u:")
print(u)
print("Esto es nuestra función sombrero:")
plt.plot(np.linspace(1,x,nx), u) # Es esquema es para hacer una gráfica simple␣
 ↪plot(X,Y)
plt.show()
```

```
Esto es u:
[1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
```

```
 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```
Esto es nuestra función sombrero:

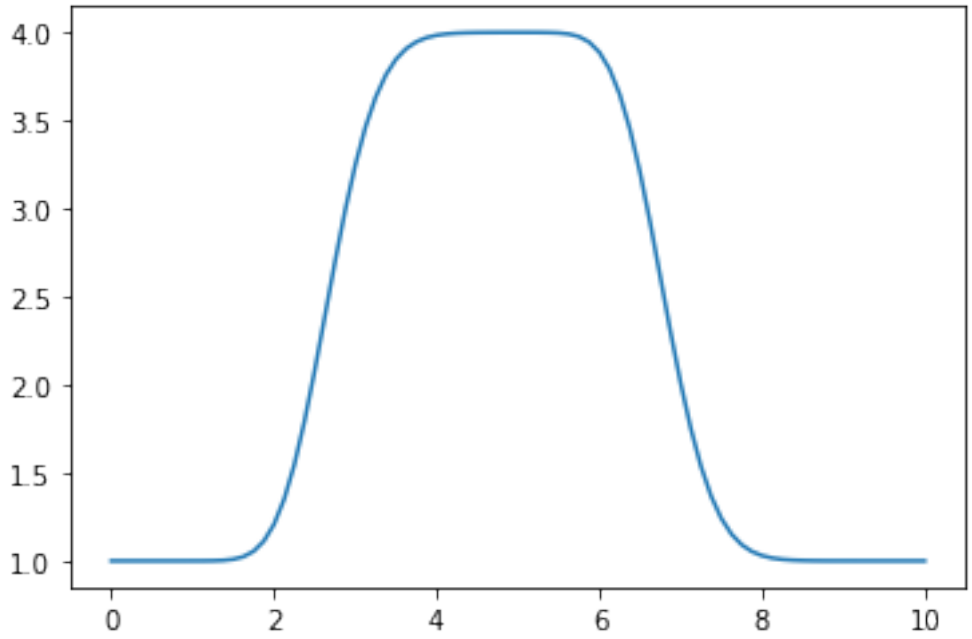


La forma de la función sombrero es definida por lineas rectas ya que esta asumiendo valores exactos que hemos asignado al vector u, en un principio habíamos dicho que u sería un arreglo conformado por ***unos*** con la misma cantidad de de elementos que ***nx***, luego según las condiciones iniciales definimos a ue al cual le dimos un valor mayor que sería reemplezado en ***u*** para las cotas x1 y x2, derecha e izquierda respectivamente, lo que estamos visualizando ocurre para el instante $t = 0$ s.

Ahora vamos a implementar la ecuación de convección ya discretizada, esto lo logramos haciendo un *for* que irá desde el *1* hasta el valor de ***nx***.

[112]:
```
un = np.ones(nx)

for n in range(nt):
    un[:] = u[:]
    for i in range(1,nx):
        u[i] = un[i]-c*(dt/dx)*(un[i]-un[i-1])
plt.plot(np.linspace(0,x,nx),u)
```

[112]:
```
[<matplotlib.lines.Line2D at 0x7f69fac9a6d8>]
```



[1]:
```
# Importando librerías
import sys # Función de Python
import numpy as np # Numpy contraída como np
import matplotlib.pyplot as plt # Matplotlib contraída como plt
from IPython.core.display import clear_output
from PyQt5.QtWidgets import QMainWindow, QApplication, QAction, QFileDialog #␣
 ↪PyQt5 el binding que hace posible la GUI
from UI_N01_V01 import * # GUI convertida de Qt Designer
```

[2]:
```
# Clase principal
class MyForm(QMainWindow): # Clase de la ventana principal
    def __init__(self): # Inicializamos el programa
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.pushButton_Paso01_LC_solver.clicked.connect(self.
 ↪dispsolver_Paso01_LC_01) # El botón es conectado a nuestro método que␣
 ↪resolverá la ecuación lineal de convección 1-D
        self.show() # Muestra resultados de la tarea ejecutada en la línea␣
 ↪anterior

    def dispsolver_Paso01_LC_01(self): # Método que usamos para resolver la␣
 ↪ecuación de convección lineal 1-D
        Paso01_LC_1D_nx = int(self.ui.lineEdit_Paso01_LC_01_nx.text()) # Input␣
 ↪para el número de puntos en el que se divide la distancia en el eje x
```

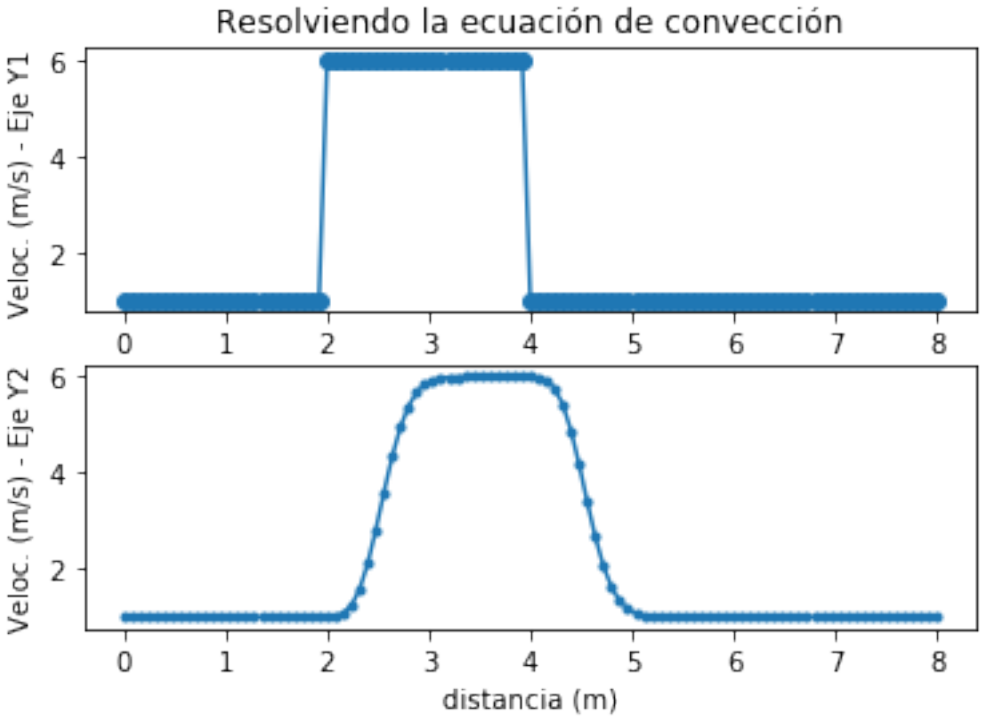
```
        Paso01_LC_1D_x = float(self.ui.lineEdit_Paso01_LC_02_x.text()) # Input␣
↪de la distancia en el eje x
        Paso01_LC_1D_dx = Paso01_LC_1D_x/(Paso01_LC_1D_nx-1)
        Paso01_LC_1D_nt = int(self.ui.lineEdit_Paso01_LC_03_nt.text()) # Número␣
↪de veces que el dt se repite
        Paso01_LC_1D_dt = float(self.ui.lineEdit_Paso01_LC_04_dt.text()) # dt␣
↪es el tamaño del paso del tiempo
        Paso01_LC_1D_c = float(self.ui.lineEdit_Paso01_LC_05_c.text()) #␣
↪velocidad de la onda , c = 1
        Paso01_LC_1D_ue = float(self.ui.lineEdit_Paso01_LC_06_u.text()) # Es la␣
↪velocidad inicial u_0 , u=2
        Paso01_LC_1D_x1 = float(self.ui.lineEdit_Paso01_LC_07_x1.text()) # es␣
↪la u inicial u_0 para el tramo izquierdo
        Paso01_LC_1D_x2 = float(self.ui.lineEdit_Paso01_LC_08_x2.text()) # Es␣
↪la u inicial u_0 para el tramo derecho

        # Barra de progreso
        x = 0
        while x < 100:
            x += 0.0001
            self.ui.progressBar_Paso01_LC_01.setValue(x)

        # Ecuación de Convección - Función sombrero
        print("Ecuación de convección - Función sombrero")
        u = np.ones(Paso01_LC_1D_nx) # Creamos una ecuación de puros unos con␣
↪la cantidad de elementos de nx
        u[int(Paso01_LC_1D_x1/Paso01_LC_1D_dx) : int(Paso01_LC_1D_x2/
↪Paso01_LC_1D_dx)] = Paso01_LC_1D_ue # Remplezamos el valor de u_0 en el␣
↪tramo de las cotas en el eje x para x1 y x2

        # Dominio01:
        Paso01_LC_1D_x1 = np.linspace(0,Paso01_LC_1D_x, Paso01_LC_1D_nx)
        print("Dominio01 x1:")
        print(Paso01_LC_1D_x1)

        # Rango01:
        print("Rango01 u:")
        print(u)

        y1 = u
        print("Rango01 y1:")
        print(y1)

        # Gráfico
        plt.subplot(2,1,1)
        plt.plot(Paso01_LC_1D_x1, y1, 'o-')
```

```
        plt.title('Resolviendo la ecuación de convección')
        plt.ylabel('Veloc. (m/s) - Eje Y1')

        self.ui.label_Paso01_LC_response_01.setText("uini: " + str(y1))

        #Ecuación de Convección - Discretizada
        print("Ecuación de convección - Discretizada")
        un = np.ones(Paso01_LC_1D_nx)

        for n in range(Paso01_LC_1D_nt):
          un[:] = u[:]
          for i in range(1,Paso01_LC_1D_nx):
              u[i] = un[i]-Paso01_LC_1D_c*Paso01_LC_1D_dt/
↪Paso01_LC_1D_dx*(un[i]-un[i-1])

        # Dominio02
        print("Dominio02 x2:")
        Paso01_LC_1D_x2 = np.linspace(0,Paso01_LC_1D_x, Paso01_LC_1D_nx)
        print(Paso01_LC_1D_x2)

        # Rango02
        print("Rango02 u:")
        print(u)
        y2 = u
        print("Rango02 y2:")
        print(y2)

        plt.subplot(2,1,2)
        plt.plot(Paso01_LC_1D_x2, y2, '.-')
        plt.xlabel('distancia (m)')
        plt.ylabel('Veloc. (m/s) - Eje Y2')

        plt.show()

        self.ui.label_Paso01_LC_response_02.setText("Ufin : " + str(y2))

if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

```
Ecuación de convección - Función sombrero
Dominio01 x1:
[0.   0.08 0.16 0.24 0.32 0.4  0.48 0.56 0.64 0.72 0.8  0.88 0.96 1.04
 1.12 1.2  1.28 1.36 1.44 1.52 1.6  1.68 1.76 1.84 1.92 2.   2.08 2.16
 2.24 2.32 2.4  2.48 2.56 2.64 2.72 2.8  2.88 2.96 3.04 3.12 3.2  3.28
```

```
 3.36 3.44 3.52 3.6  3.68 3.76 3.84 3.92 4.   4.08 4.16 4.24 4.32 4.4
 4.48 4.56 4.64 4.72 4.8  4.88 4.96 5.04 5.12 5.2  5.28 5.36 5.44 5.52
 5.6  5.68 5.76 5.84 5.92 6.   6.08 6.16 6.24 6.32 6.4  6.48 6.56 6.64
 6.72 6.8  6.88 6.96 7.04 7.12 7.2  7.28 7.36 7.44 7.52 7.6  7.68 7.76
 7.84 7.92 8.  ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6.
 6. 6. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6.
 6. 6. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[0.   0.08 0.16 0.24 0.32 0.4  0.48 0.56 0.64 0.72 0.8  0.88 0.96 1.04
 1.12 1.2  1.28 1.36 1.44 1.52 1.6  1.68 1.76 1.84 1.92 2.   2.08 2.16
 2.24 2.32 2.4  2.48 2.56 2.64 2.72 2.8  2.88 2.96 3.04 3.12 3.2  3.28
 3.36 3.44 3.52 3.6  3.68 3.76 3.84 3.92 4.   4.08 4.16 4.24 4.32 4.4
 4.48 4.56 4.64 4.72 4.8  4.88 4.96 5.04 5.12 5.2  5.28 5.36 5.44 5.52
 5.6  5.68 5.76 5.84 5.92 6.   6.08 6.16 6.24 6.32 6.4  6.48 6.56 6.64
 6.72 6.8  6.88 6.96 7.04 7.12 7.2  7.28 7.36 7.44 7.52 7.6  7.68 7.76
 7.84 7.92 8.  ]
Rango02 u:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.0016575  1.01586463 1.07573754 1.24110082 1.5777332
 2.11634502 2.82167002 3.59896696 4.33462299 4.94183114 5.38422565
 5.67149481 5.83906849 5.927459   5.96985037 5.98842183 5.99588357
 5.99864254 5.99958409 5.99988142 5.9999685  5.99999219 5.99999819
 5.99999961 5.99999992 5.99834249 5.98413537 5.92426246 5.75889918
 5.4222668  4.88365498 4.17832998 3.40103304 2.66537701 2.05816886
 1.61577435 1.32850519 1.16093151 1.072541   1.03014963 1.01157817
 1.00411643 1.00135746 1.00041591 1.00011858 1.0000315  1.00000781
 1.00000181 1.00000039 1.00000008 1.00000002 1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
Rango02 y2:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
```

```
1.         1.         1.         1.         1.         1.
1.         1.         1.         1.         1.         1.
1.         1.0016575  1.01586463 1.07573754 1.24110082 1.5777332
2.11634502 2.82167002 3.59896696 4.33462299 4.94183114 5.38422565
5.67149481 5.83906849 5.927459   5.96985037 5.98842183 5.99588357
5.99864254 5.99958409 5.99988142 5.9999685  5.99999219 5.99999819
5.99999961 5.99999992 5.99834249 5.98413537 5.92426246 5.75889918
5.4222668  4.88365498 4.17832998 3.40103304 2.66537701 2.05816886
1.61577435 1.32850519 1.16093151 1.072541   1.03014963 1.01157817
1.00411643 1.00135746 1.00041591 1.00011858 1.0000315  1.00000781
1.00000181 1.00000039 1.00000008 1.00000002 1.         1.
1.         1.         1.         1.         1.         1.
1.         1.         1.         1.         1.         1.
1.         1.         1.         1.         1.         1.
1.         1.         1.         1.         1.        ]
```




```
An exception has occurred, use %tb to see the full traceback.

SystemExit: 0
```

```
/home/jhongesell/Documentos/Informatica/Entornos_virtuales/Entornos_pip_venv/ent
```

```
orno01N01/lib/python3.6/site-packages/IPython/core/interactiveshell.py:3334:
UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

3.2 02.02. Ecuación de convección no-lineal 1D

Es la siguiente ecuación de convección representada en su forma no-lineal:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

El valor de la velocidad de propagación de la onda c ya no se encuentra disponible, en su lugar se presenta a u quien reemplaza a la constante anterior hablada, ya hemos visto antes como se hace la discretización por lo tanto para este caso nos queda:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

Recordemos que nuestro interés es obtener el valor de u_i^{n+1} , por ello si despejamos queda:

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n)$$

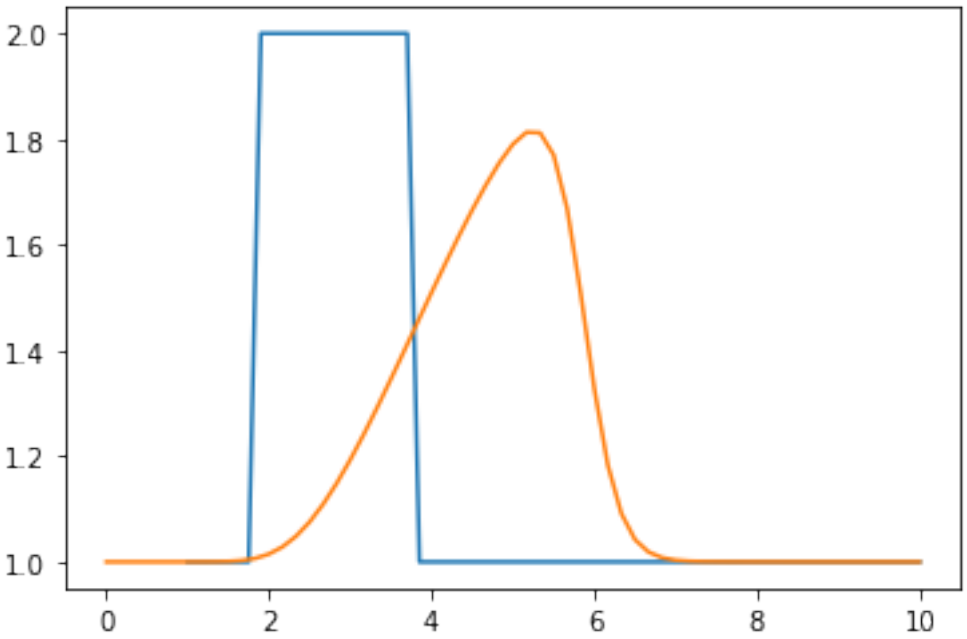
Para nuestro programa GUI sería lo siguiente:

[6]:
```
# Librerías
import numpy as np # Numpy contraída como np
import matplotlib.pyplot as plt # Matplotlib contraída como plt

x = 10. # Longitud en el eje X
nx = 61 # Número de divisiones que se le hace a x
dx = x/(nx-1)
nt = 200 # Tiempo t total
dt = 0.010 # Paso del tiempo t
ue = 2
x1 = 1
x2 = 3

# Condiciones iniciales
u = np.ones(nx)      # numpy función ones() creando un array de 1 con nx␣
 ↪elmentos
u[int(x1/dx) : int(x2/dx+1)]=ue
print("Esto es u:")
print(u)
print("Esto es nuestra función sombrero:")
```

```
plt.plot(np.linspace(1,x,nx), u) # Es esquema es para hacer una gráfica simple
 ↪plot(X,Y)

un = np.ones(nx)

for n in range(nt):
    un[:] = u[:]
    for i in range(1,nx):
        u[i] = un[i]-un[i]*(dt/dx)*(un[i]-un[i-1])
plt.plot(np.linspace(0,x,nx),u)
```

```
Esto es u:
[1. 1. 1. 1. 1. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Esto es nuestra función sombrero:
```

[6]:
```
[<matplotlib.lines.Line2D at 0x7f0507757b38>]
```



[1]:
```
# Importando librerías
import sys # Función de Python
import numpy as np # Numpy contraída como np
import matplotlib.pyplot as plt # Matplotlib contraída como plt
from IPython.core.display import clear_output
from PyQt5.QtWidgets import QMainWindow, QApplication, QAction, QFileDialog #
 ↪PyQt5 el binding que hace posible la GUI
```

```
from UI_N01_V02 import * # GUI convertida de Qt Designer
```

[2]:
```
# Clase principal
class MyForm(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.pushButton_Paso01_LC_solver.clicked.connect(self.
↪dispsolver_Paso01_LC)
        self.ui.pushButton_Paso01_NLC_solver.clicked.connect(self.
↪dispsolver_Paso01_NCL)
        self.show()

    def dispsolver_Paso01_LC(self):
        Paso01_LC_1D_nx = int(self.ui.lineEdit_Paso01_LC_01_nx.text())
        Paso01_LC_1D_x = float(self.ui.lineEdit_Paso01_LC_02_x.text())
        Paso01_LC_1D_dx = Paso01_LC_1D_x/(Paso01_LC_1D_nx-1)
        Paso01_LC_1D_nt = int(self.ui.lineEdit_Paso01_LC_03_nt.text())
        Paso01_LC_1D_dt = float(self.ui.lineEdit_Paso01_LC_04_dt.text())
        Paso01_LC_1D_c = float(self.ui.lineEdit_Paso01_LC_05_c.text())
        Paso01_LC_1D_ue = float(self.ui.lineEdit_Paso01_LC_06_u.text())
        Paso01_LC_1D_x1 = float(self.ui.lineEdit_Paso01_LC_07_x1.text())
        Paso01_LC_1D_x2 = float(self.ui.lineEdit_Paso01_LC_08_x2.text())

        # Barra de progreso
        x = 0
        while x < 100:
            x += 0.0001
            self.ui.progressBar_Paso01_LC_01.setValue(x)

        # Ecuación de Convección - Función sombrero
        print("Ecuación de convección - Función sombrero")
        u = np.ones(Paso01_LC_1D_nx)
        u[int(Paso01_LC_1D_x1/Paso01_LC_1D_dx) : int(Paso01_LC_1D_x2/
↪Paso01_LC_1D_dx+1)] = Paso01_LC_1D_ue

        # Dominio01:
        Paso01_LC_1D_x1 = np.linspace(0,Paso01_LC_1D_x, Paso01_LC_1D_nx)
        print("Dominio01 x1:")
        print(Paso01_LC_1D_x1)

        # Rango01:
        print("Rango01 u:")
        print(u)

        y1 = u
```

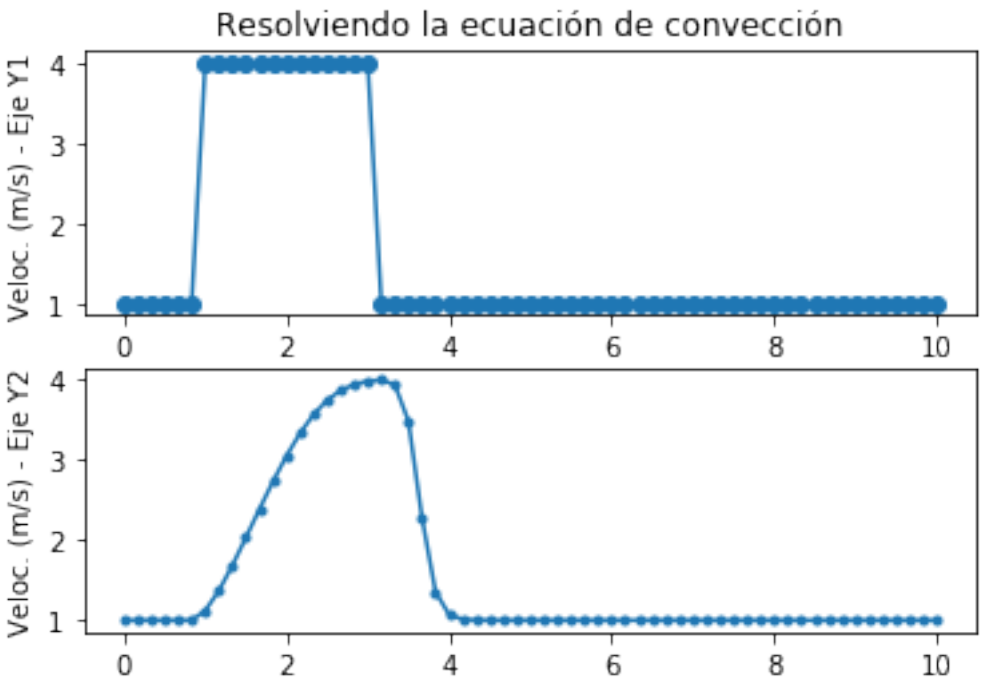
```
        print("Rango01 y1:")
        print(y1)

        # Gráfico
        plt.subplot(2,1,1)
        plt.plot(Paso01_LC_1D_x1, y1, 'o-')
        plt.title('Resolviendo la ecuación de convección')
        plt.ylabel('Veloc. (m/s) - Eje Y1')

        self.ui.label_Paso01_LC_response_01.setText("uini: " + str(y1))

        #Ecuación de Convección - Discretizada
        print("Ecuación de convección - Discretizada")
        un = np.ones(Paso01_LC_1D_nx)

        for n in range(Paso01_LC_1D_nt):
          un[:] = u[:]
          for i in range(1,Paso01_LC_1D_nx):
              u[i] = un[i]-Paso01_LC_1D_c*Paso01_LC_1D_dt/
↪Paso01_LC_1D_dx*(un[i]-un[i-1])

        # Dominio02
        print("Dominio02 x2:")
        Paso01_LC_1D_x2 = np.linspace(0,Paso01_LC_1D_x, Paso01_LC_1D_nx)
        print(Paso01_LC_1D_x2)

        # Rango02
        print("Rango02 u:")
        print(u)
        y2 = u
        print("Rango02 y2:")
        print(y2)

        plt.subplot(2,1,2)
        plt.plot(Paso01_LC_1D_x2, y2, '.-')
        plt.xlabel('distancia (m)')
        plt.ylabel('Veloc. (m/s) - Eje Y2')

        plt.show()

        self.ui.label_Paso01_LC_response_02.setText("Ufin : " + str(y2))
    def dispsolver_Paso01_NCL(self):
        print("Hello")
        Paso01_NLC_1D_nx = int(self.ui.lineEdit_Paso01_NLC_01_nx.text())
        Paso01_NLC_1D_x = float(self.ui.lineEdit_Paso01_NLC_02_x.text())
        Paso01_NLC_1D_dx = Paso01_NLC_1D_x/(Paso01_NLC_1D_nx-1)
        Paso01_NLC_1D_nt = int(self.ui.lineEdit_Paso01_NLC_03_nt.text())
```

```
        Paso01_NLC_1D_dt = float(self.ui.lineEdit_Paso01_NLC_04_dt.text())
        Paso01_NLC_1D_ue = float(self.ui.lineEdit_Paso01_NLC_06_ue.text())
        Paso01_NLC_1D_x1 = float(self.ui.lineEdit_Paso01_NLC_07_x1.text())
        Paso01_NLC_1D_x2 = float(self.ui.lineEdit_Paso01_NLC_08_x2.text())

        # Barra de progreso
        x = 0
        while x < 100:
            x += 0.0001
            self.ui.progressBar_Paso01_NLC_01.setValue(x)

        # Ecuación de COnvección - Función sombrero
        print("Ecuación de convección - Función sombrero")
        u = np.ones(Paso01_NLC_1D_nx)
        u[int(Paso01_NLC_1D_x1/Paso01_NLC_1D_dx) : int(Paso01_NLC_1D_x2/
↪Paso01_NLC_1D_dx+1)] = Paso01_NLC_1D_ue

        # DominioO1:
        Paso01_NLC_1D_x1 = np.linspace(0,Paso01_NLC_1D_x, Paso01_NLC_1D_nx)
        print("Dominio x1:")
        print(Paso01_NLC_1D_x1)

        # RangoO1:
        print("RangoO1 u:")
        print(u)

        y1 = u
        print("RangoO1 y1:")
        print(y1)

        # Gráfico
        plt.subplot(2,1,1)
        plt.plot(Paso01_NLC_1D_x1, y1, 'o-')
        plt.title('Resolviendo la ecuación de convección')
        plt.ylabel('Veloc. (m/s) - Eje Y1')

        self.ui.label_Paso01_NLC_response_01.setText("uini: " + str(y1))

        # Ecuación de convección - Discretizada
        print("Ecuación de convección - Discretizada")
        un = np.ones(Paso01_NLC_1D_nx)

        for n in range(Paso01_NLC_1D_nt):
            un[:] = u[:]
            for i in range(1,Paso01_NLC_1D_nx):
                u[i] = un[i]-un[i]*Paso01_NLC_1D_dt/
↪Paso01_NLC_1D_dx*(un[i]-un[i-1])
```

```
        # Dominio02
        print("Dominio02 x2:")
        Paso01_NLC_1D_x2 = np.linspace(0,Paso01_NLC_1D_x, Paso01_NLC_1D_nx)
        print(Paso01_NLC_1D_x2)

        # Rango02
        print("Rango02 u:")
        print(u)
        y2 = u
        print("Rango02 y2:")
        print(y2)

        plt.subplot(2,1,2)
        plt.plot(Paso01_NLC_1D_x2, y2, '.-')
        plt.ylabel('Veloc. (m/s) - Eje Y2')

        plt.show()

        self.ui.label_Paso01_NLC_response_02.setText("Ufin : " + str(y2))


if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

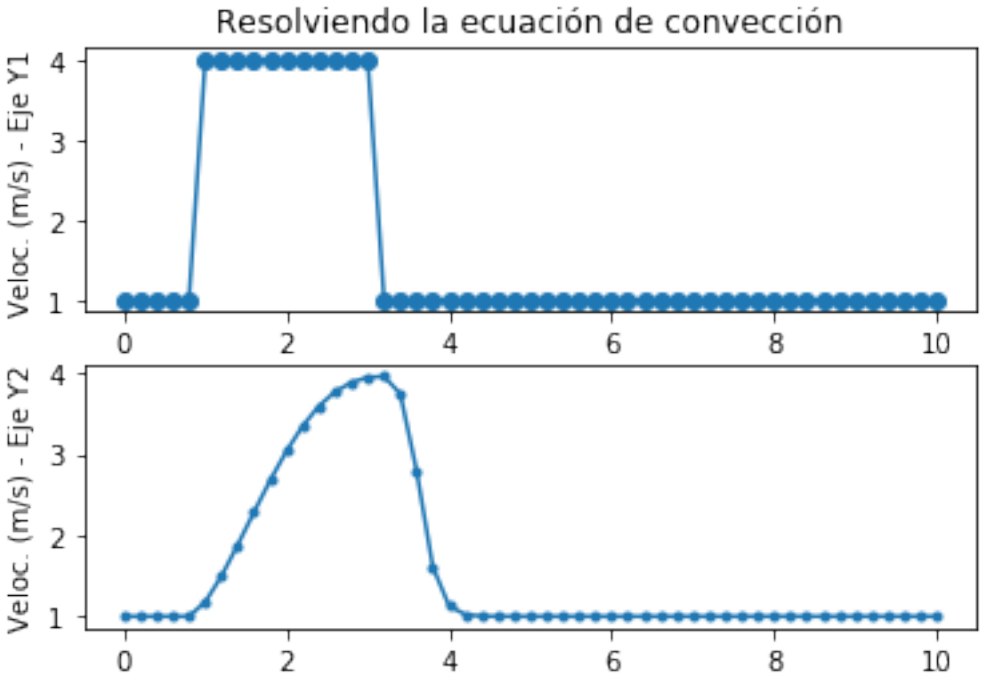
```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.          0.16666667  0.33333333  0.5         0.66666667  0.83333333
  1.          1.16666667  1.33333333  1.5         1.66666667  1.83333333
  2.          2.16666667  2.33333333  2.5         2.66666667  2.83333333
  3.          3.16666667  3.33333333  3.5         3.66666667  3.83333333
  4.          4.16666667  4.33333333  4.5         4.66666667  4.83333333
  5.          5.16666667  5.33333333  5.5         5.66666667  5.83333333
  6.          6.16666667  6.33333333  6.5         6.66666667  6.83333333
  7.          7.16666667  7.33333333  7.5         7.66666667  7.83333333
  8.          8.16666667  8.33333333  8.5         8.66666667  8.83333333
  9.          9.16666667  9.33333333  9.5         9.66666667  9.83333333
 10.        ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Rango01 y1:
```

```
[1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.          0.16666667  0.33333333  0.5         0.66666667  0.83333333
  1.          1.16666667  1.33333333  1.5         1.66666667  1.83333333
  2.          2.16666667  2.33333333  2.5         2.66666667  2.83333333
  3.          3.16666667  3.33333333  3.5         3.66666667  3.83333333
  4.          4.16666667  4.33333333  4.5         4.66666667  4.83333333
  5.          5.16666667  5.33333333  5.5         5.66666667  5.83333333
  6.          6.16666667  6.33333333  6.5         6.66666667  6.83333333
  7.          7.16666667  7.33333333  7.5         7.66666667  7.83333333
  8.          8.16666667  8.33333333  8.5         8.66666667  8.83333333
  9.          9.16666667  9.33333333  9.5         9.66666667  9.83333333
 10.        ]
Rango02 u:
[1.         1.         1.         1.         1.         1.
 1.12058959 1.36410401 1.67891836 2.02551686 2.37910902 2.72239388
 3.04105221 3.32212317 3.55463638 3.73191231 3.85414921 3.92908498
 3.96942225 3.9835785  3.92384426 3.47436295 2.27920953 1.35547818
 1.07084704 1.01205988 1.00182097 1.00024453 1.00002915 1.00000308
 1.00000029 1.00000002 1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.        ]
Rango02 y2:
[1.         1.         1.         1.         1.         1.
 1.12058959 1.36410401 1.67891836 2.02551686 2.37910902 2.72239388
 3.04105221 3.32212317 3.55463638 3.73191231 3.85414921 3.92908498
 3.96942225 3.9835785  3.92384426 3.47436295 2.27920953 1.35547818
 1.07084704 1.01205988 1.00182097 1.00024453 1.00002915 1.00000308
 1.00000029 1.00000002 1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.        ]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.2  0.4  0.6  0.8  1.   1.2  1.4  1.6  1.8  2.   2.2  2.4  2.6
  2.8  3.   3.2  3.4  3.6  3.8  4.   4.2  4.4  4.6  4.8  5.   5.2  5.4
  5.6  5.8  6.   6.2  6.4  6.6  6.8  7.   7.2  7.4  7.6  7.8  8.   8.2
  8.4  8.6  8.8  9.   9.2  9.4  9.6  9.8 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.2  0.4  0.6  0.8  1.   1.2  1.4  1.6  1.8  2.   2.2  2.4  2.6
  2.8  3.   3.2  3.4  3.6  3.8  4.   4.2  4.4  4.6  4.8  5.   5.2  5.4
  5.6  5.8  6.   6.2  6.4  6.6  6.8  7.   7.2  7.4  7.6  7.8  8.   8.2
  8.4  8.6  8.8  9.   9.2  9.4  9.6  9.8 10. ]
Rango02 u:
[1.         1.         1.         1.         1.         1.1772267
 1.49634249 1.88187353 2.28609021 2.67932438 3.03968691 3.3493219
 3.59519201 3.77220726 3.88555656 3.94913659 3.95967874 3.74695999
```

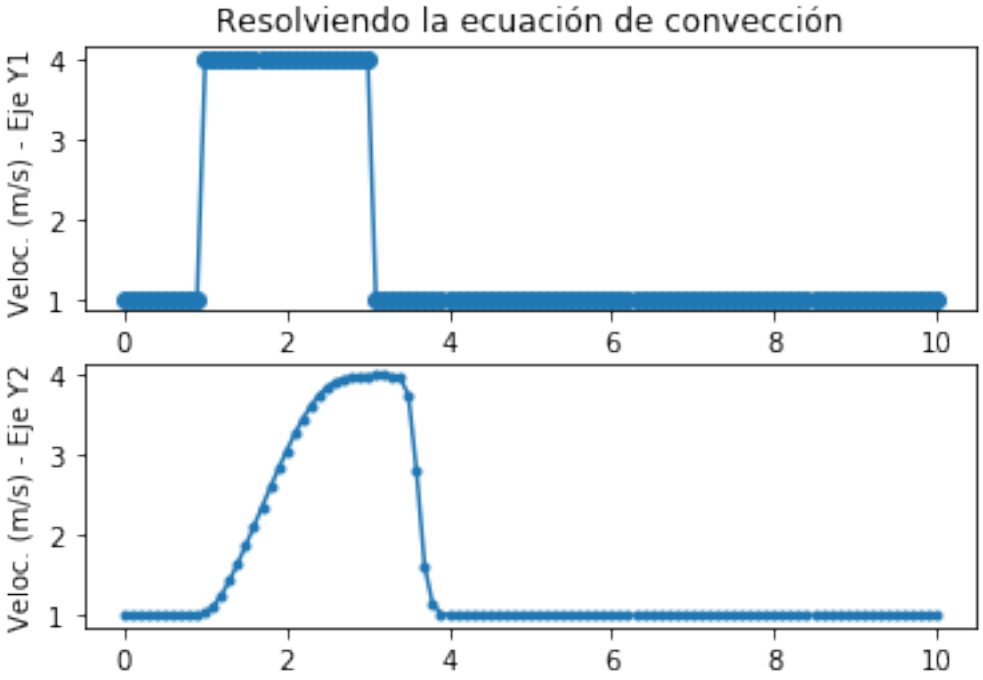
```
 2.80056313 1.61239644 1.13045423 1.02238849 1.00333587 1.00043652
 1.00005022 1.00000508 1.00000045 1.00000004 1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.        ]
Rango02 y2:
[1.         1.         1.         1.         1.         1.1772267
 1.49634249 1.88187353 2.28609021 2.67932438 3.03968691 3.3493219
 3.59519201 3.77220726 3.88555656 3.94913659 3.95967874 3.74695999
 2.80056313 1.61239644 1.13045423 1.02238849 1.00333587 1.00043652
 1.00005022 1.00000508 1.00000045 1.00000004 1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.        ]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
```

```
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.02766816 1.11217175
 1.25425744 1.43889705 1.65090135 1.87923455 2.11637333 2.3569136
 2.59646952 2.83090969 3.05583148 3.2662401  3.45651018 3.62081157
 3.75417318 3.85406715 3.92180487 3.96266395 3.98428353 3.994219
 3.99815695 3.99948922 3.99971116 3.99734524 3.97084484 3.75292701
 2.80585472 1.61311878 1.13548651 1.02547394 1.00439176 1.00069799
 1.00010151 1.00001341 1.0000016  1.00000017 1.00000002 1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
Rango02 y2:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.02766816 1.11217175
 1.25425744 1.43889705 1.65090135 1.87923455 2.11637333 2.3569136
 2.59646952 2.83090969 3.05583148 3.2662401  3.45651018 3.62081157
```

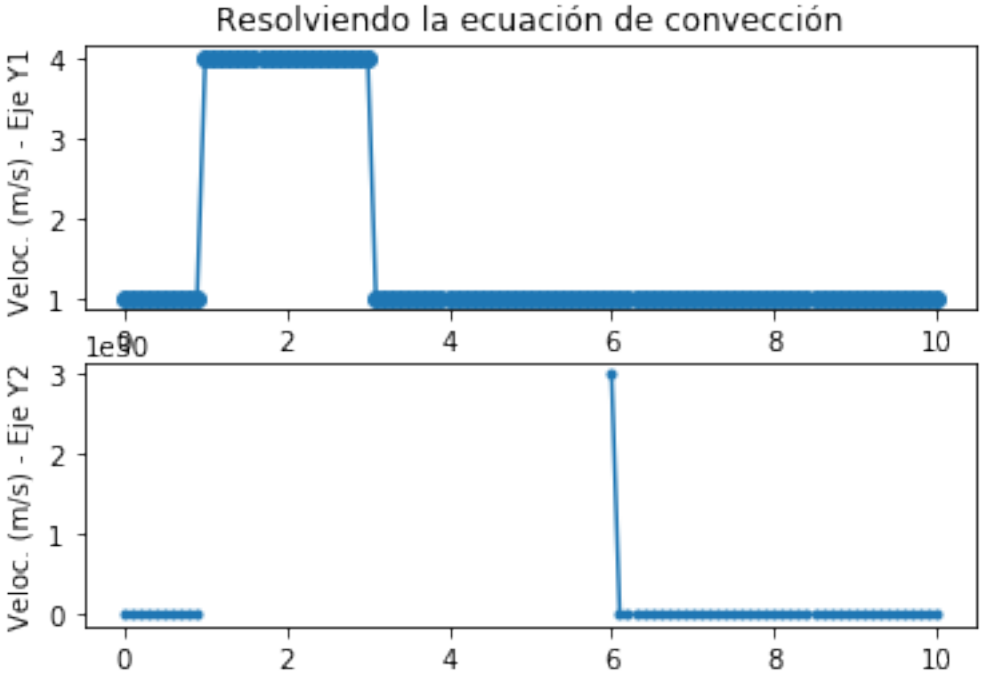
```
 3.75417318 3.85406715 3.92180487 3.96266395 3.98428353 3.994219
 3.99815695 3.99948922 3.99971116 3.99734524 3.97084484 3.75292701
 2.80585472 1.61311878 1.13548651 1.02547394 1.00439176 1.00069799
 1.00010151 1.00001341 1.0000016  1.00000017 1.00000002 1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
```

```
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
   -inf    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan   -inf
 3.e+30 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00]
Rango02 y2:
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
   -inf    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan   -inf
 3.e+30 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00]

/home/jhongesell/Documentos/Informatica/Entornos_virtuales/Entornos_pip_venv/ent
```

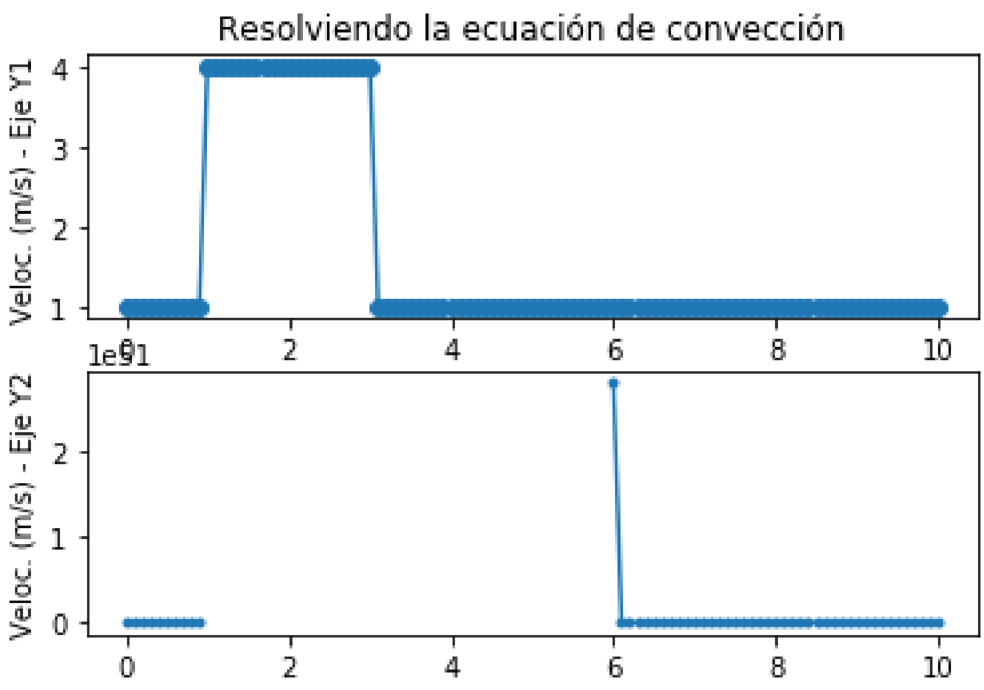
```
orno01N01/lib/python3.6/site-packages/ipykernel_launcher.py:133: RuntimeWarning:
overflow encountered in double_scalars
/home/jhongesell/Documentos/Informatica/Entornos_virtuales/Entornos_pip_venv/ent
orno01N01/lib/python3.6/site-packages/ipykernel_launcher.py:133: RuntimeWarning:
invalid value encountered in double_scalars
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
```

```
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00           -inf            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan           -inf
 2.79396772e+51 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00]
Rango02 y2:
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00           -inf            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
```

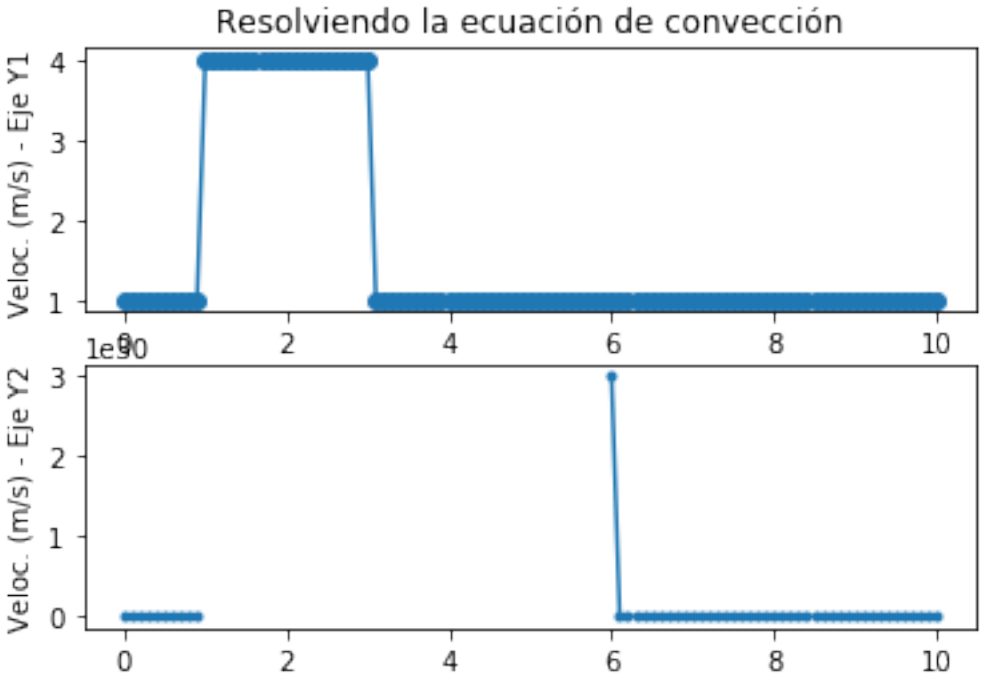
```
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan           -inf
 2.79396772e+51 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
```

```
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
   -inf    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan   -inf
 3.e+30 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00]
Rango02 y2:
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
   -inf    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan    nan
    nan    nan    nan    nan    nan    nan    nan    nan    nan   -inf
```

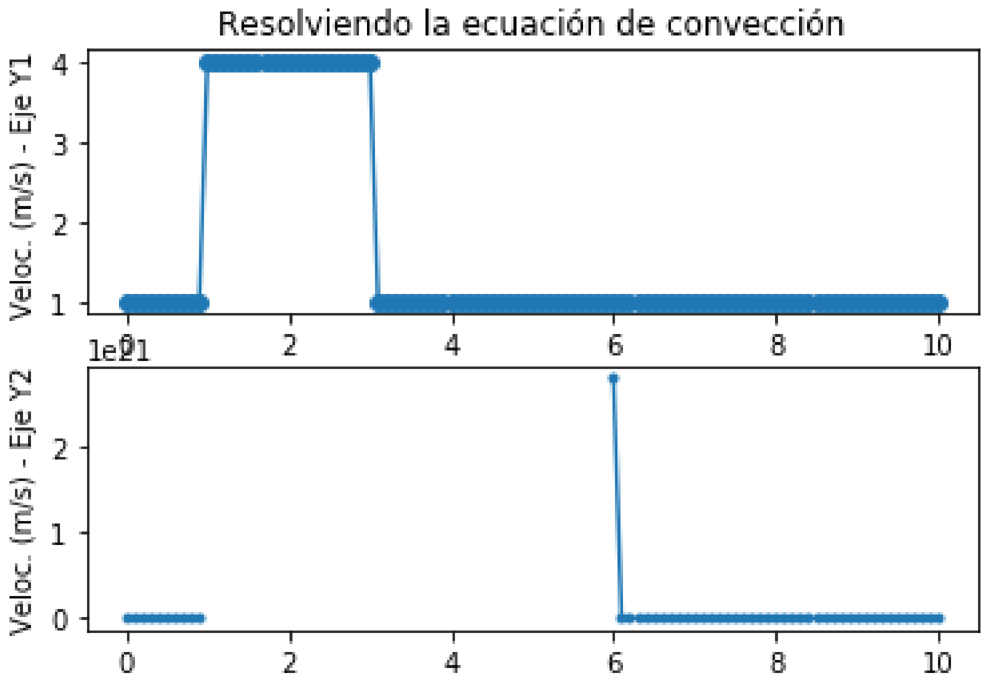
```
 3.e+30 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00
 1.e+00]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
```

```
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00           -inf            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan           -inf
 2.79396772e+21 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00]
Rango02 y2:
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00           -inf            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
            nan            nan            nan            nan
```

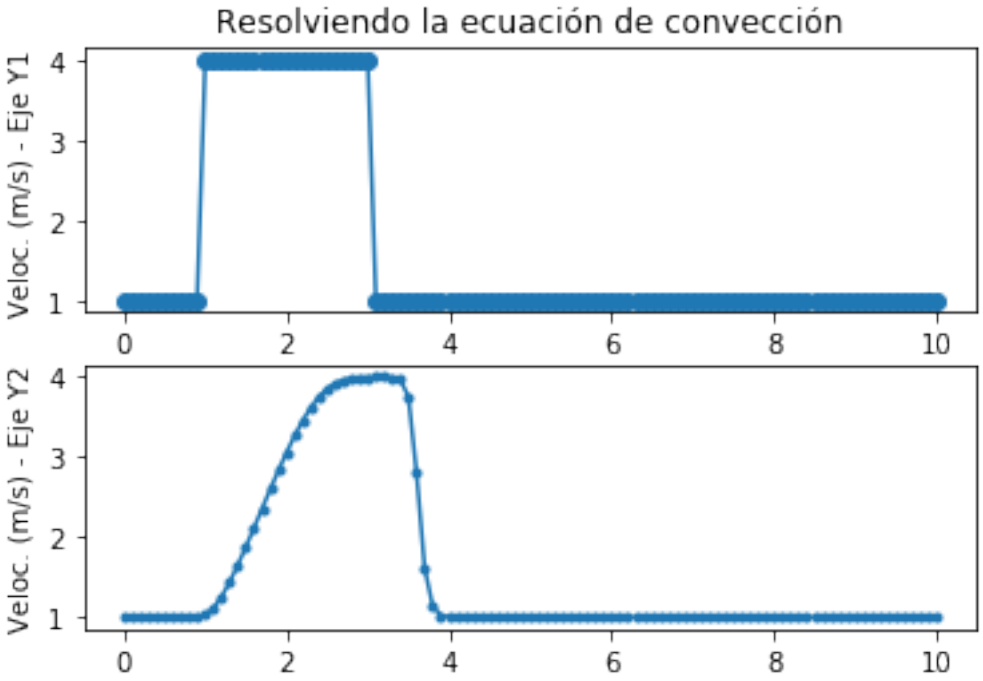
```
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan            nan
           nan            nan            nan           -inf
 2.79396772e+21 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
```

```
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.02766816 1.11217175
 1.25425744 1.43889705 1.65090135 1.87923455 2.11637333 2.3569136
 2.59646952 2.83090969 3.05583148 3.2662401  3.45651018 3.62081157
 3.75417318 3.85406715 3.92180487 3.96266395 3.98428353 3.994219
 3.99815695 3.99948922 3.99971116 3.99734524 3.97084484 3.75292701
 2.80585472 1.61311878 1.13548651 1.02547394 1.00439176 1.00069799
 1.00010151 1.00001341 1.0000016  1.00000017 1.00000002 1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
Rango02 y2:
```

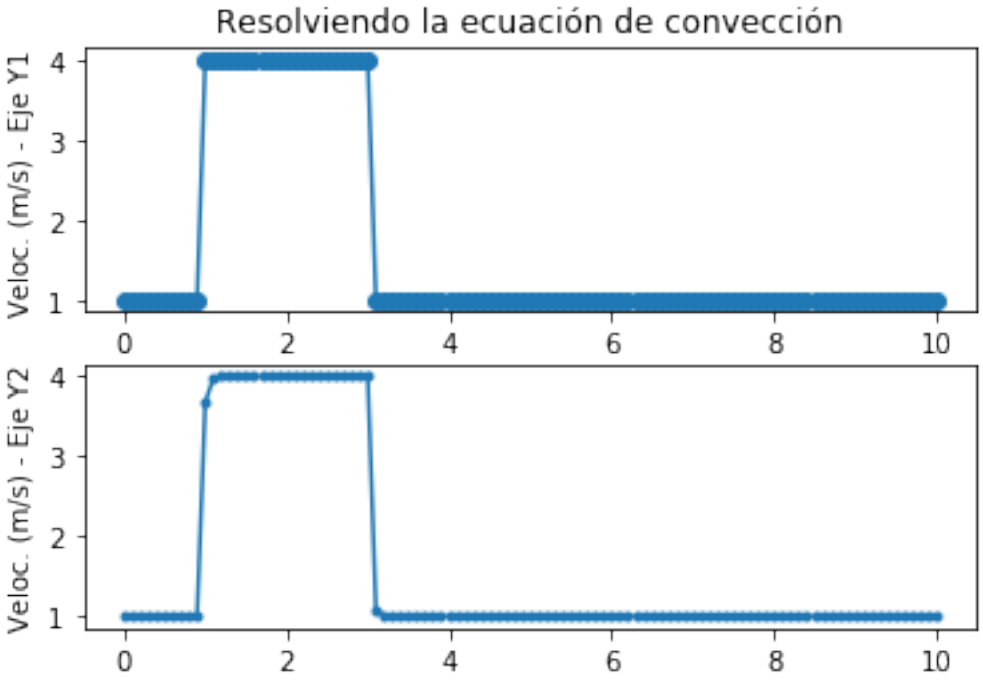
```
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.02766816 1.11217175
 1.25425744 1.43889705 1.65090135 1.87923455 2.11637333 2.3569136
 2.59646952 2.83090969 3.05583148 3.2662401  3.45651018 3.62081157
 3.75417318 3.85406715 3.92180487 3.96266395 3.98428353 3.994219
 3.99815695 3.99948922 3.99971116 3.99734524 3.97084484 3.75292701
 2.80585472 1.61311878 1.13548651 1.02547394 1.00439176 1.00069799
 1.00010151 1.00001341 1.0000016  1.00000017 1.00000002 1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
```




```
Hello
Ecuación de convección - Función sombrero
Dominio x1:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
```

```
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango01 u:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Rango01 y1:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
Ecuación de convección - Discretizada
Dominio02 x2:
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.   1.1  1.2  1.3
  1.4  1.5  1.6  1.7  1.8  1.9  2.   2.1  2.2  2.3  2.4  2.5  2.6  2.7
  2.8  2.9  3.   3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.   4.1
  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.   5.1  5.2  5.3  5.4  5.5
  5.6  5.7  5.8  5.9  6.   6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
  7.   7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.   8.1  8.2  8.3
  8.4  8.5  8.6  8.7  8.8  8.9  9.   9.1  9.2  9.3  9.4  9.5  9.6  9.7
  9.8  9.9 10. ]
Rango02 u:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         3.67324371 3.9811591
 3.99929572 3.99998094 3.9999996  3.99999999 4.         4.
 4.         4.         4.         4.         4.         4.
 4.         4.         4.         4.         4.         4.
 4.         1.09257803 1.0013177  1.00001218 1.00000008 1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.        ]
Rango02 y2:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         3.67324371 3.9811591
 3.99929572 3.99998094 3.9999996  3.99999999 4.         4.
```

```
4.          4.          4.          4.          4.          4.
4.          4.          4.          4.          4.          4.
4.          1.09257803 1.0013177  1.00001218 1.00000008 1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.          1.
1.          1.          1.          1.          1.        ]
```




```
An exception has occurred, use %tb to see the full traceback.


SystemExit: 0
```

```
/home/jhongesell/Documentos/Informatica/Entornos_virtuales/Entornos_pip_venv/ent
orno01N01/lib/python3.6/site-packages/IPython/core/interactiveshell.py:3334:
UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
```

```
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

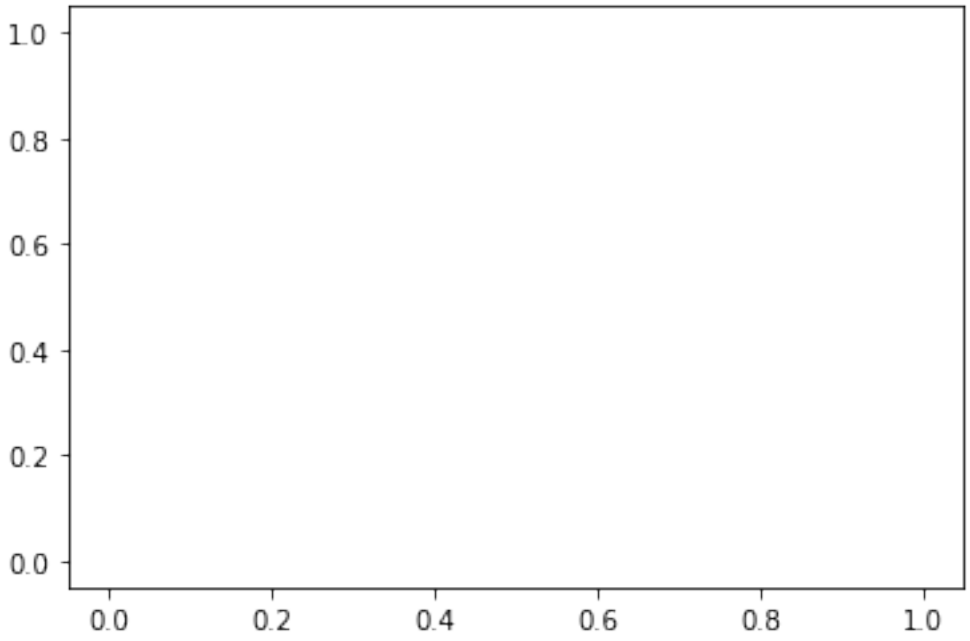
3.3 02.03. Convergencia y condición CFL

Si bien en este curso tenemos la posibilidad de cambiar los parámetros de nuestras ecuaciones, en algunos casos se pueden llegar a ver como estos hacen que la gráfica sea inestable, vamos a crear una función o método para una sola variable, recordemos que además de definir el espacio de trabajo en el eje x también señalamos el número de nodos o puntos nx en el cual este se divide, vamos a analizar ello creando métodos:

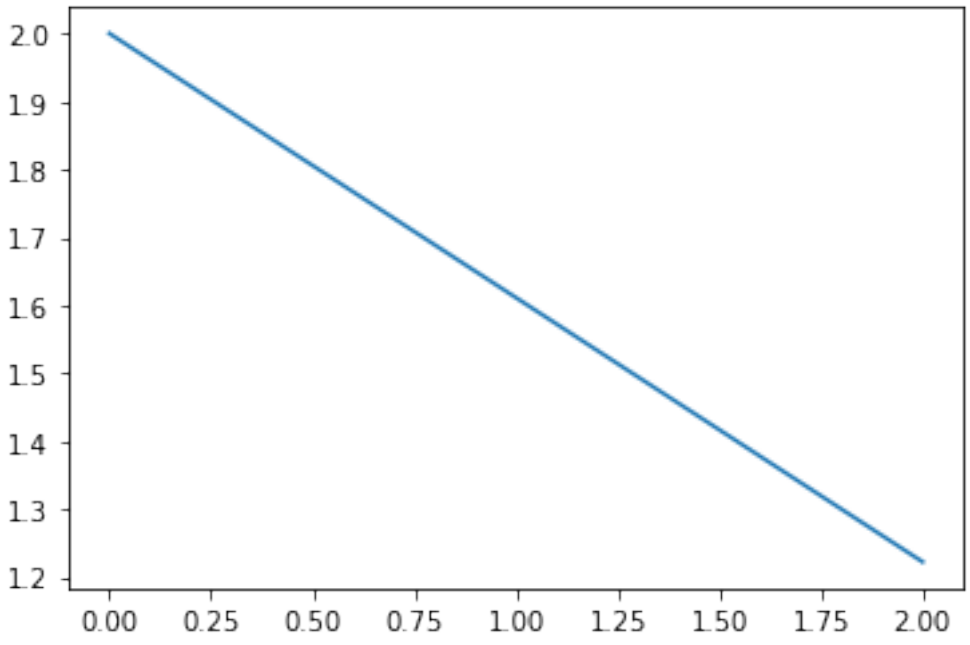
[3]:
```
%pylab inline
#función mágica que hace que los gráficos se muestren en la misma celda de␣
 ↪resultados

import numpy as np
import matplotlib.pyplot as plt

def conveccionlineal1D(nx):
    x = 2.
    dx = x/(nx-1)
    nt = 20
    dt = 0.025
    c = 1
    x1 = 0.5
    x2 = 1.
    ue = 2

    u = np.ones(nx)
    u[int(x1/dx) : int(x2/dx+1)]=ue

    un = np.ones(nx)

    for n in range(nt):
        un[:] = u[:]
        for i in range(1,nx):
            u[i] = un[i]-c*(dt/dx)*(un[i]-un[i-1])
    plt.plot(np.linspace(0,x,nx),u)
```

```
Populating the interactive namespace from numpy and matplotlib
```

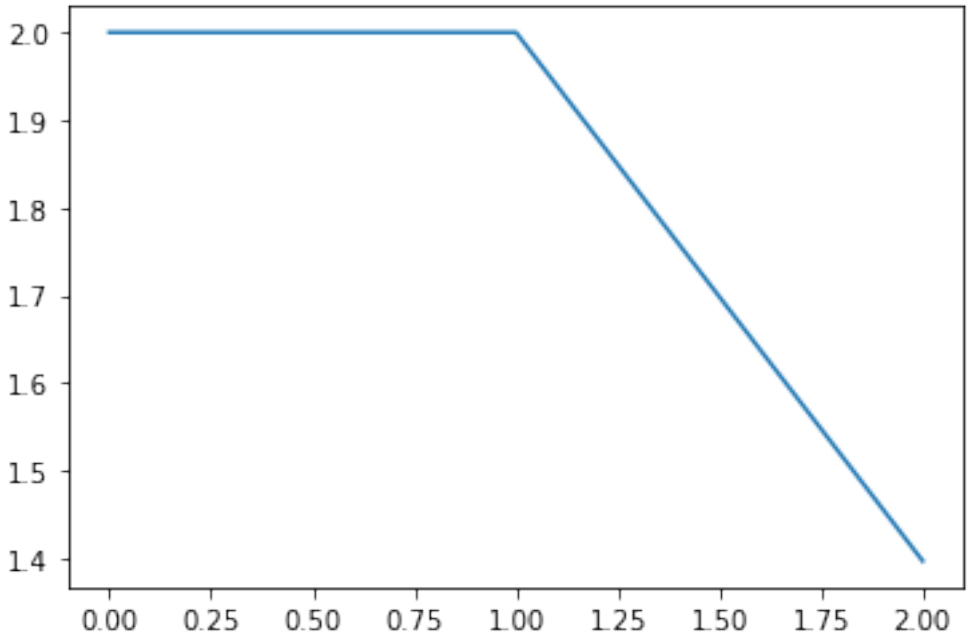
[4]:
```
conveccionlineal1D(0)
```



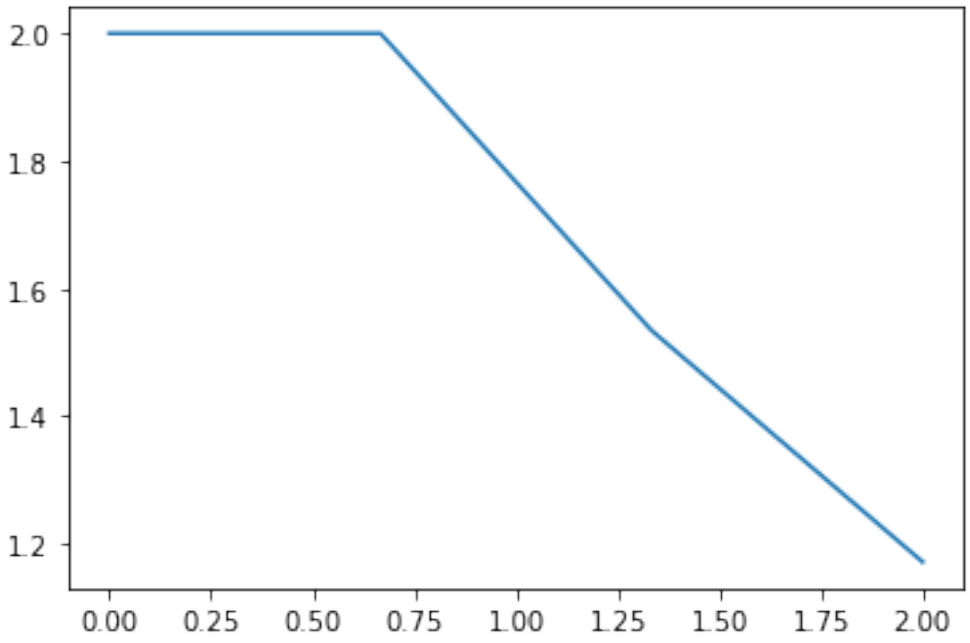
[5]:
```
conveccionlineal1D(2)
```



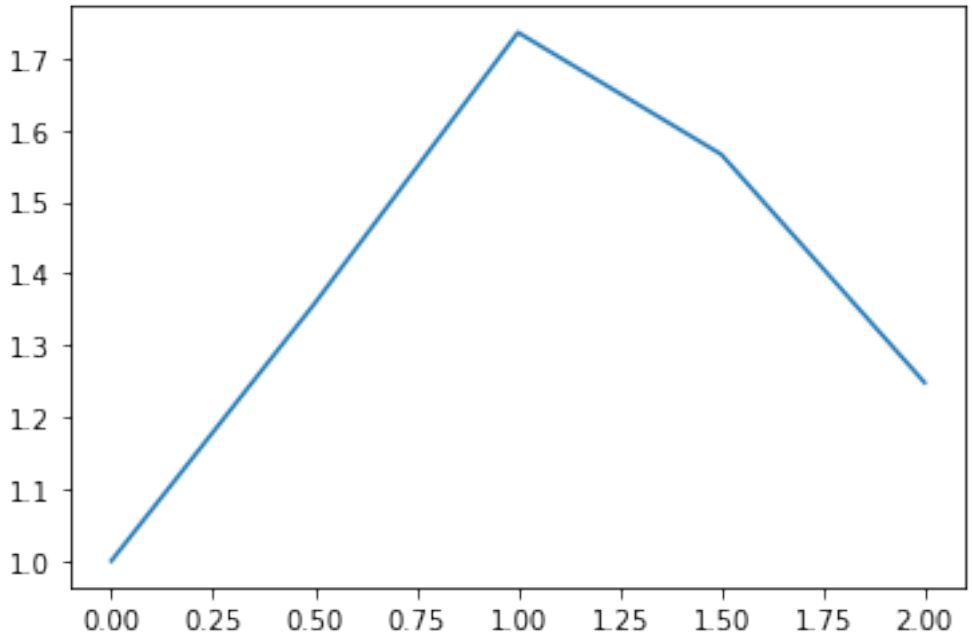
[7]:
```
conveccionlineal1D(3)
```



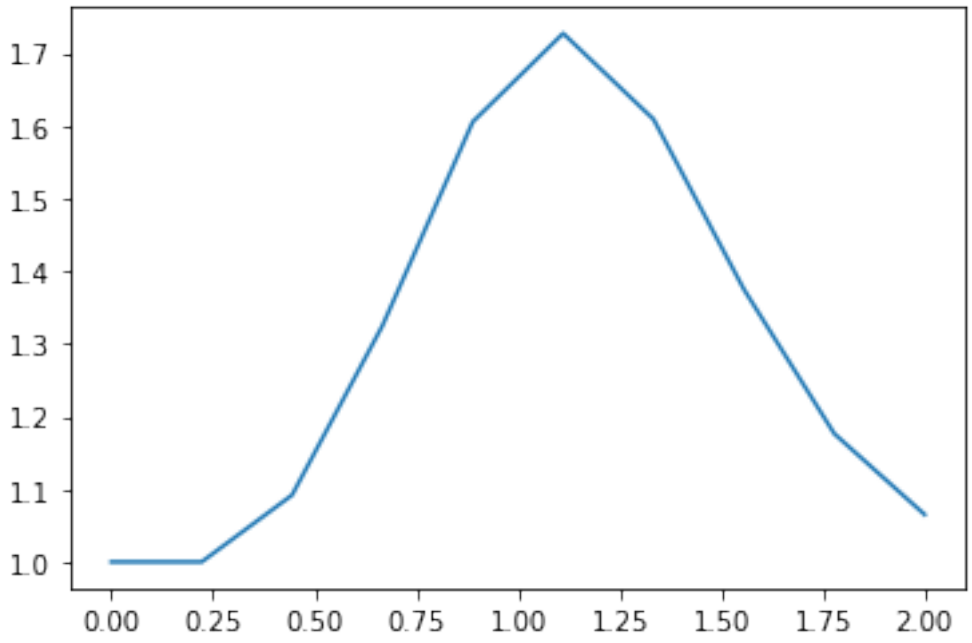
[8]:
```
conveccionlineal1D(4)
```



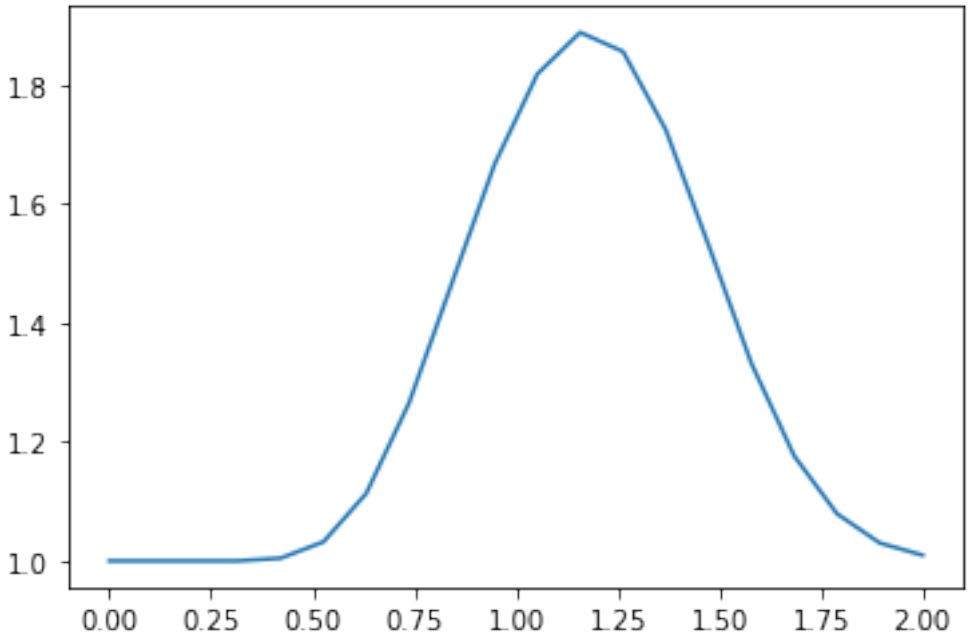
[9]:
```
conveccionlineal1D(5)
```



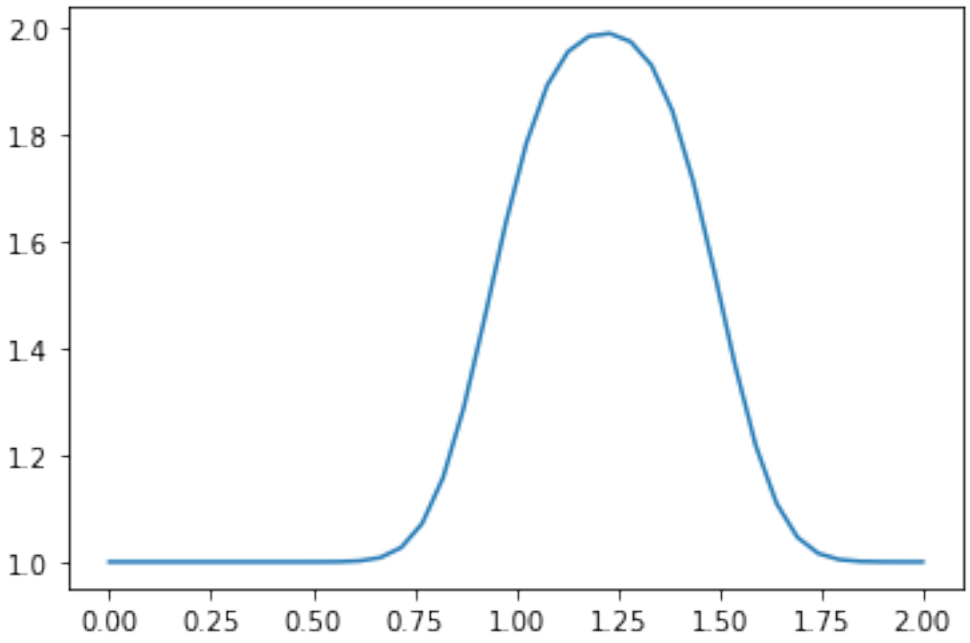
[10]:
```
conveccionlineal1D(10)
```



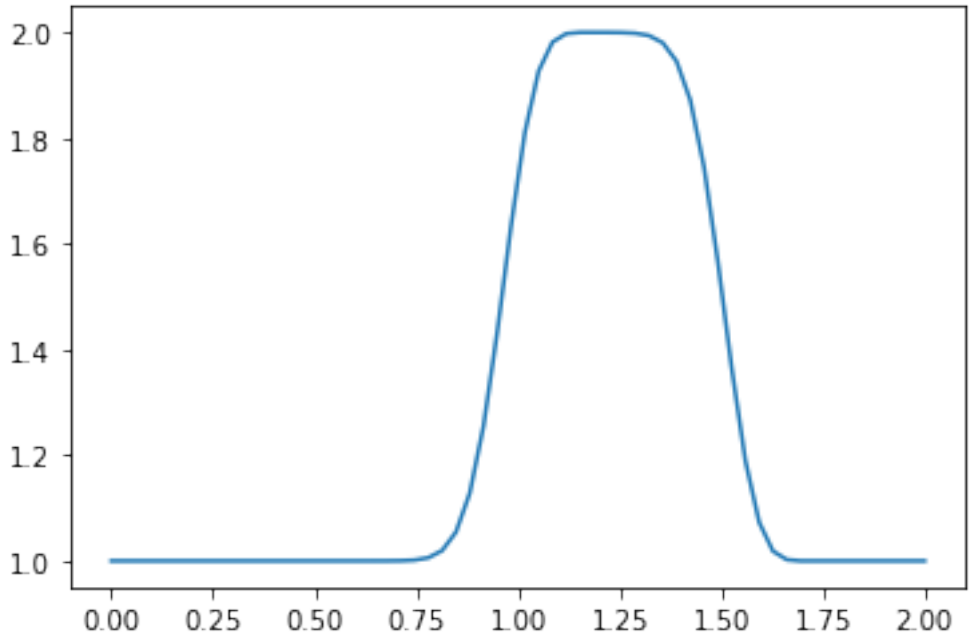
[11]:
```
conveccionlineal1D(20)
```



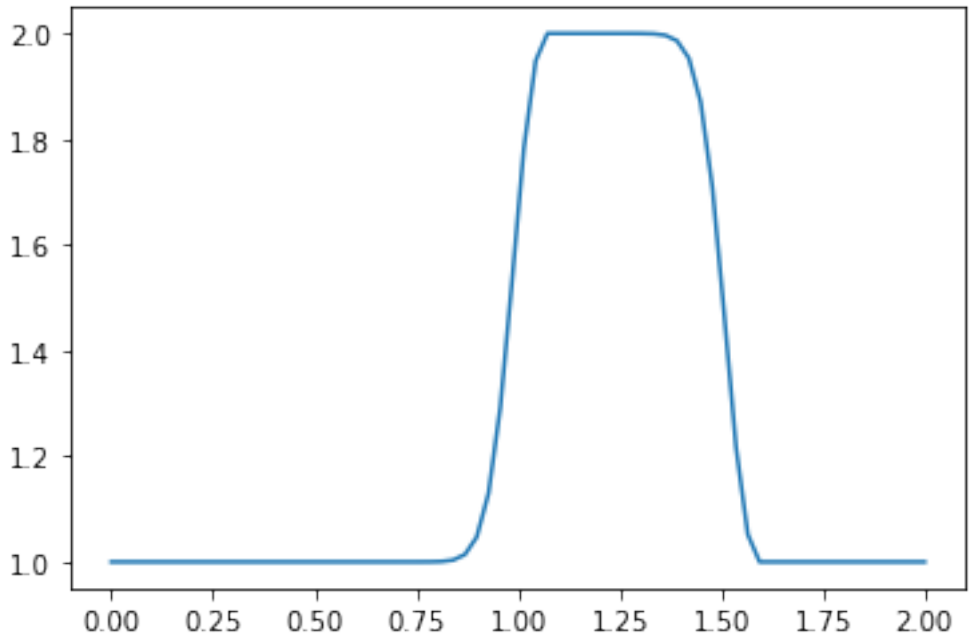
[12]:
```
conveccionlineal1D(40)
```



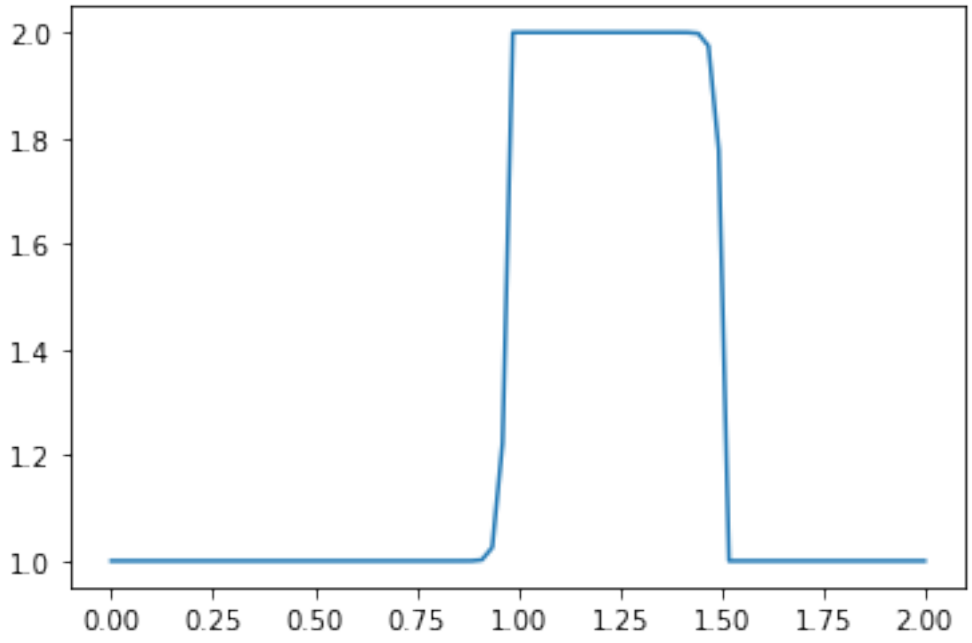
[14]:
```
conveccionlineal1D(60)
```



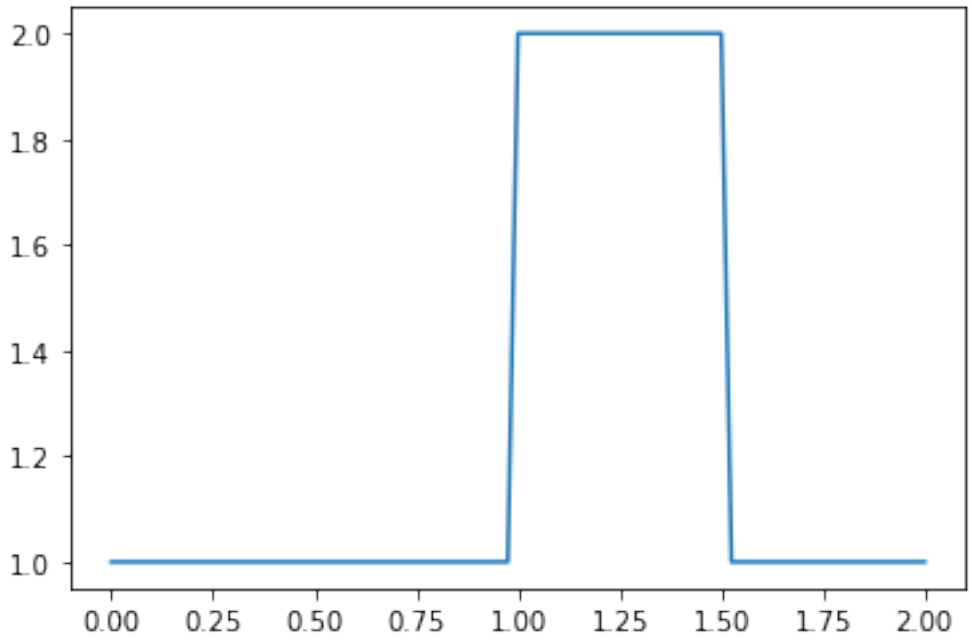
[23]:
```
conveccionlineal1D(70)
```



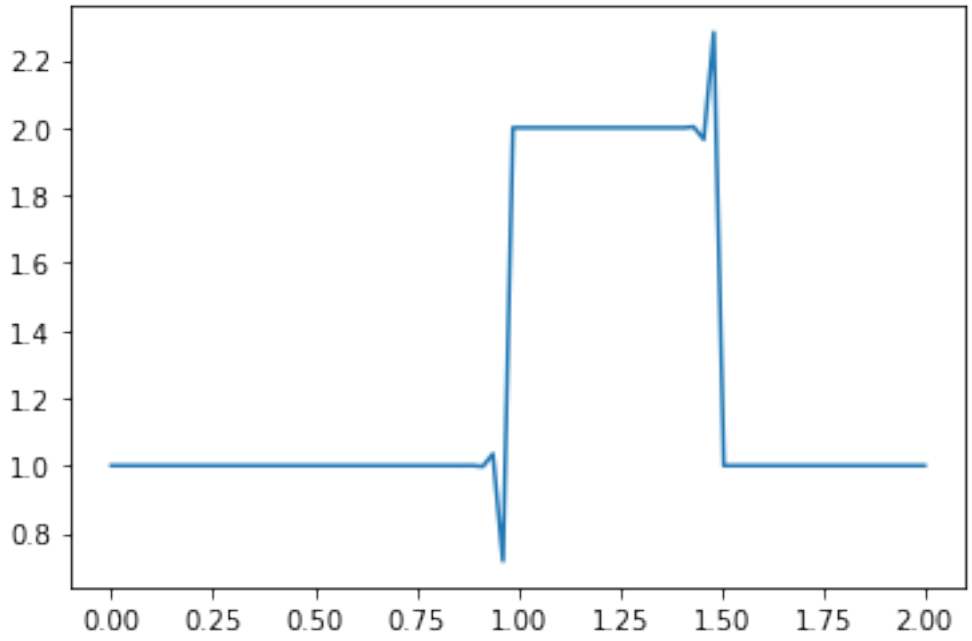
[24]:
```
conveccionlineal1D(80)
```



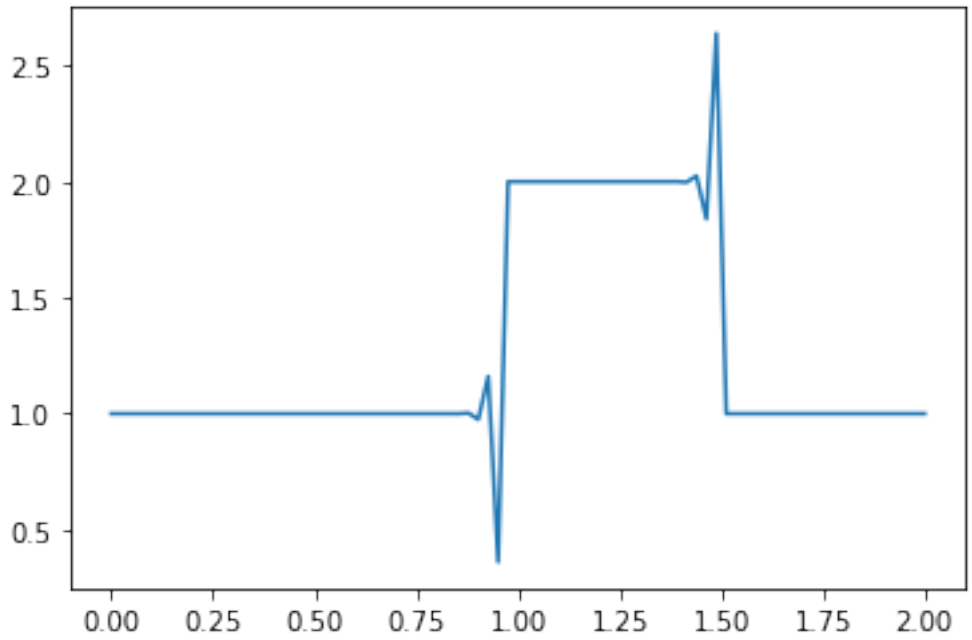
[27]:
```
conveccionlineal1D(81)
```



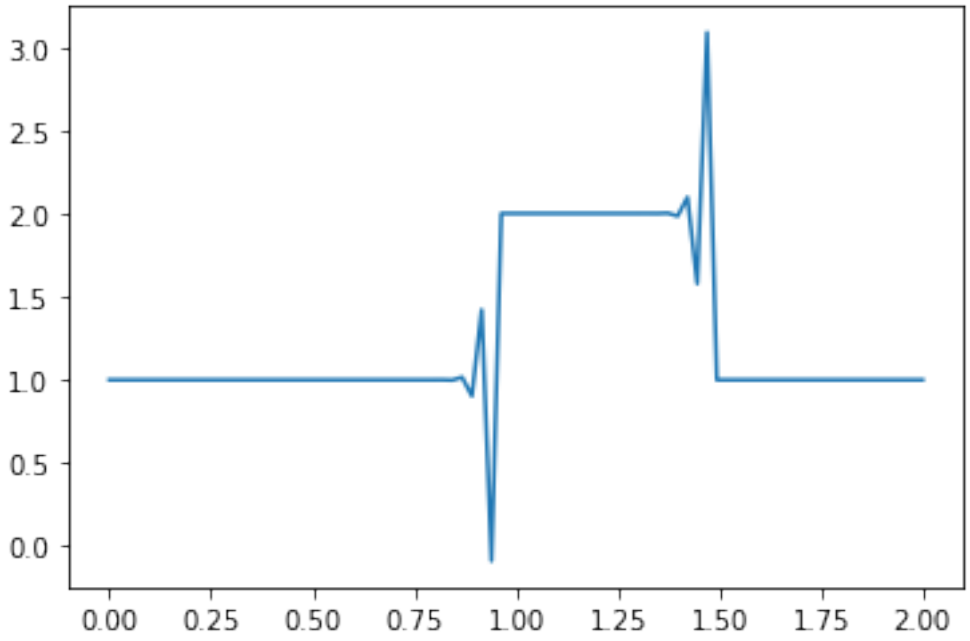
[28]:
```
conveccionlineal1D(82)
```



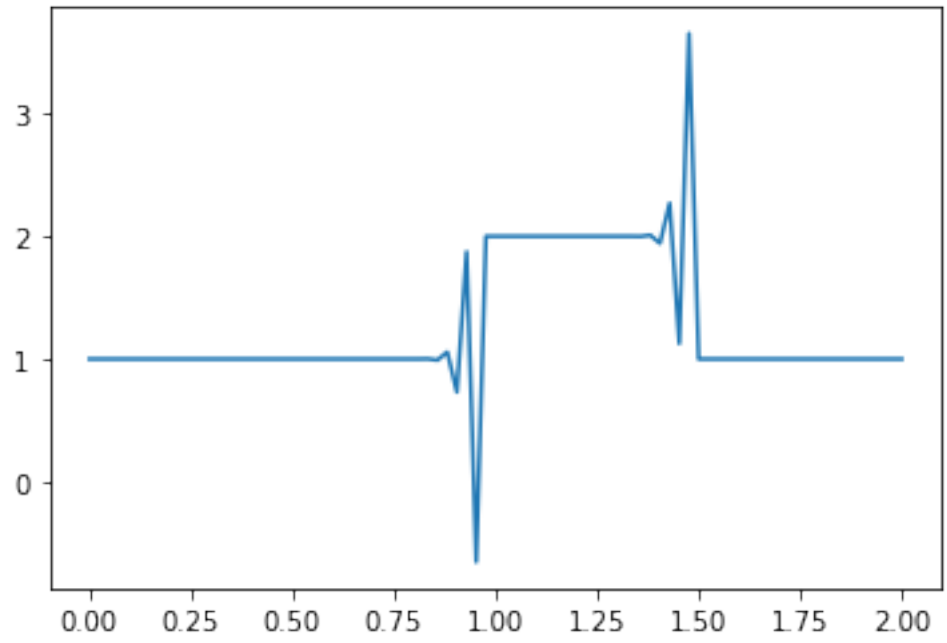
[29]:
```
conveccionlineal1D(83)
```



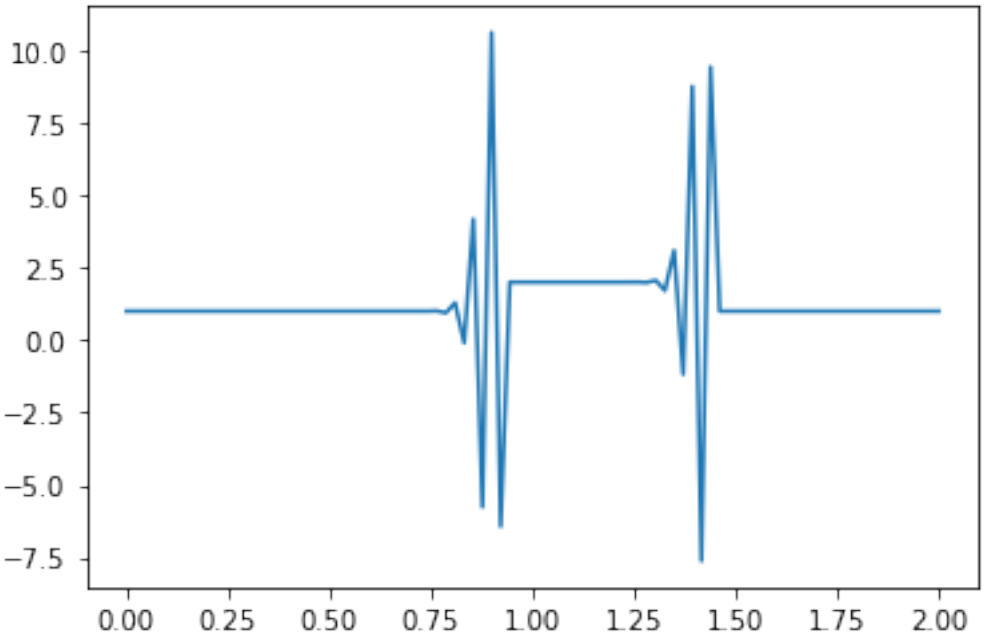
[30]:
```
conveccionlineal1D(84)
```



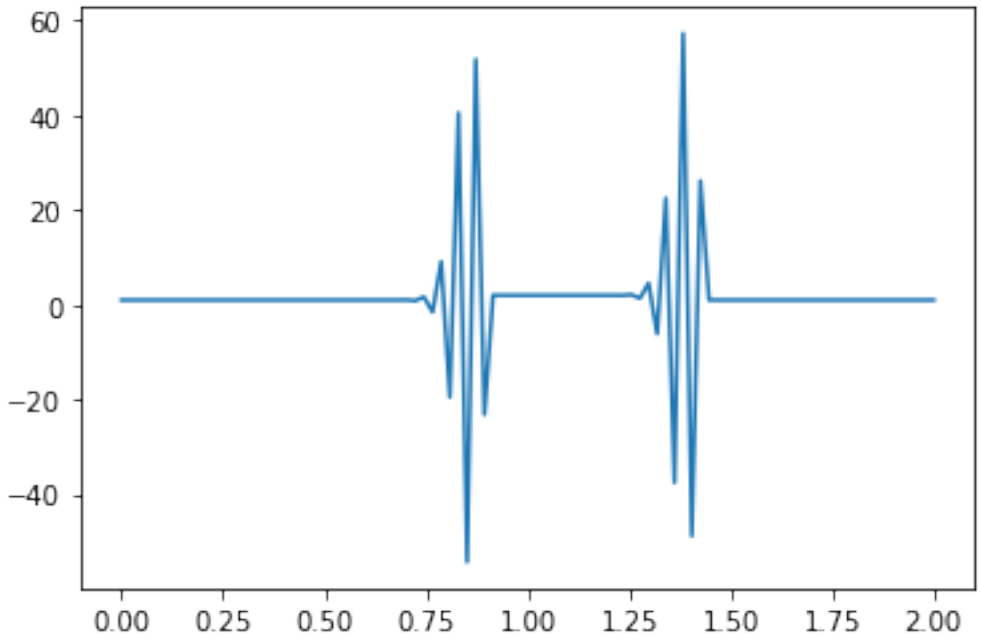
[31]:
```
conveccionlineal1D(85)
```



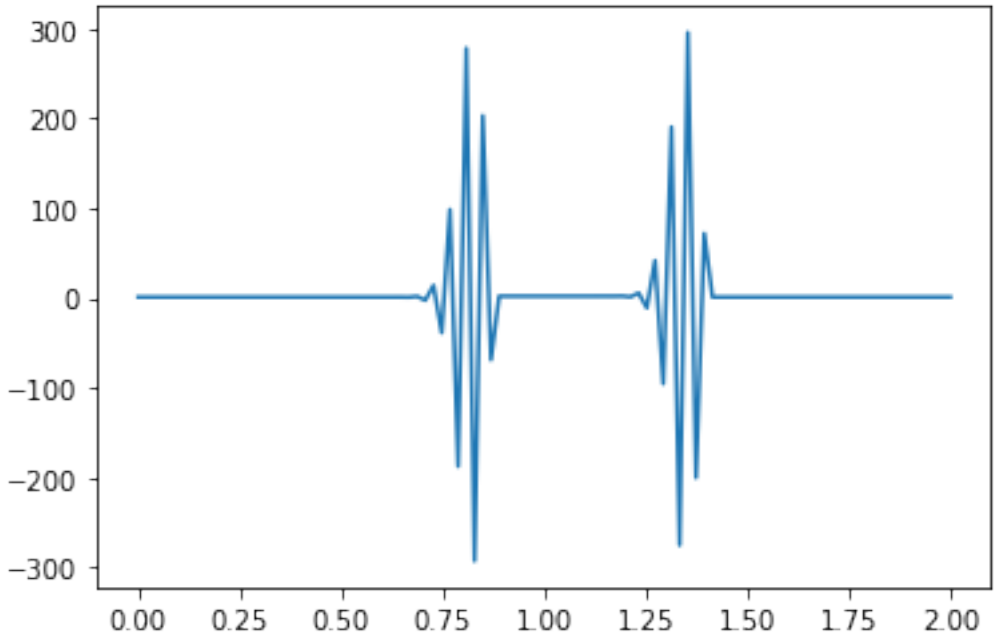
[33]:
```
conveccionlineal1D(90)
```



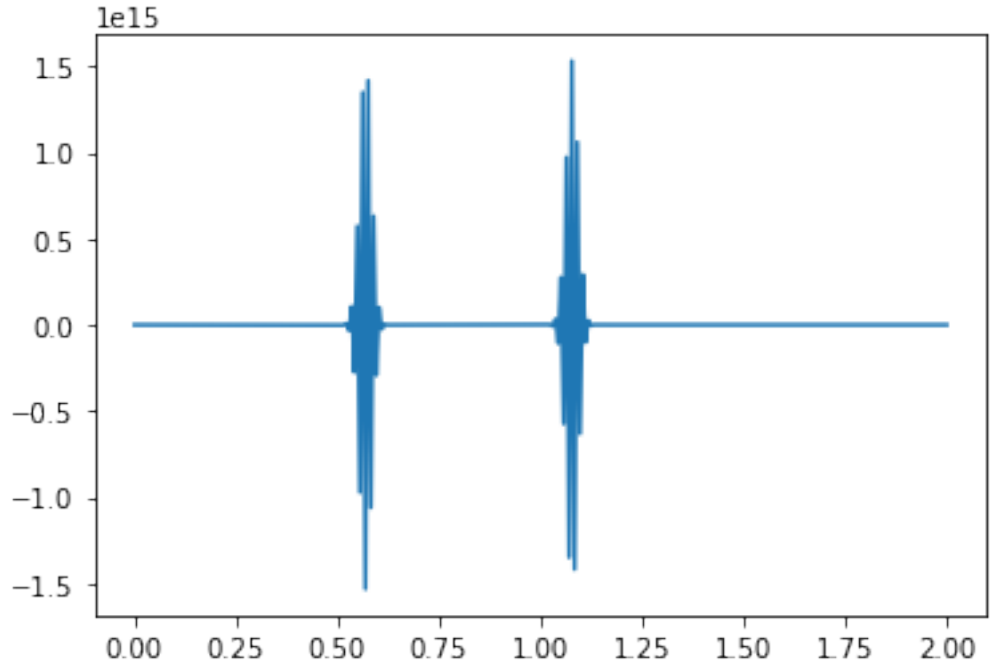
[34]:
```
conveccionlineal1D(95)
```



[35]:
```
conveccionlineal1D(100)
```



[37]:
```
conveccionlineal1D(300)
```



La inestabilidad que vemos que se produce es más notorio a la medida que aumentamos el número de puntos, en un principio con los primeros valores

que asignabamos a *nx* eramos concientes que se parecía a función sombrero, pero conforme fue creciendo *nx* para un valor de 80 este iba alejandose a lo que habíamos visto hasta ese instante; lo que vemos que ocurre es que comienza a haber mayor difusión numérica y por ello la onda toma esas formas, consideraciones:

- La Convergencia y estabilidad de un esquema explícito dependerá del tamaño de los intervalos de tiempo y espacio.
- Para un método explícito luego de cierto tiempo si el error disminuye o ya no aumenta al pasar otra etapa de tiempo se dice que el esquema es estable.

Nota: Para el método implícito de Crack-Nicolson lo que se llegará a ver que es incondicionalmente estable.

3.4 02.04. Condición de Courant-Friedrichs-Lewys (CFL)

El CFL o también conocido como número de Courant se utiliza como una restricción para que exista una convergencia de las ecuaciones diferenciales parciales, es diferente a la estabilidad numérica. El CFL hace que el paso del tiempo deba ser menos a un valor ya que si no, la solución falla.

$$\sigma = \frac{u\Delta t}{\Delta x} \leq \sigma_{max}$$

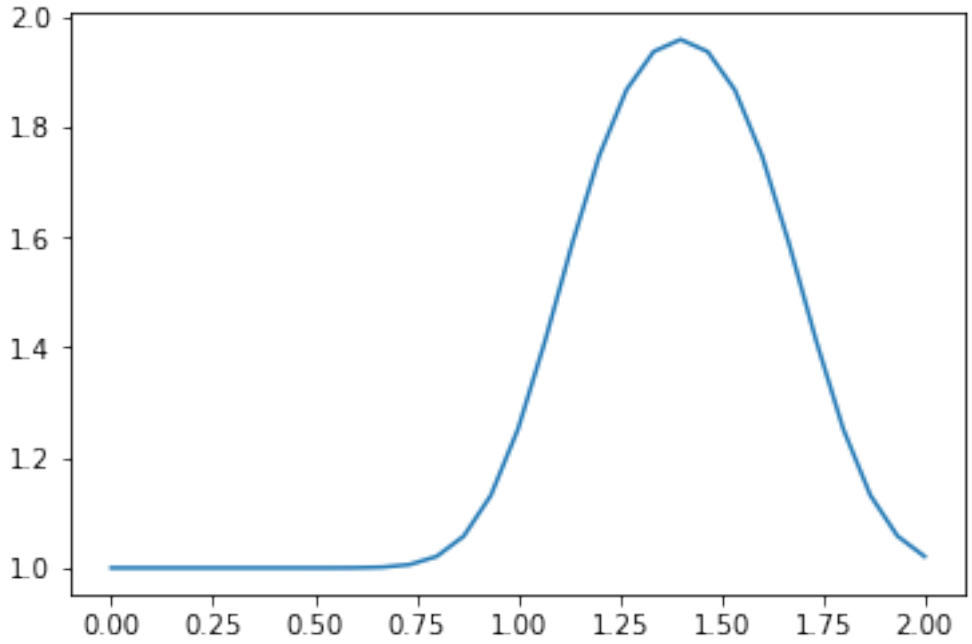
σ : Número de Courant u : Velocidad de la onda

Ahora vamos a implementar un código el cual incluya al número de Courant con la finalidad de que converga a una solución estable.

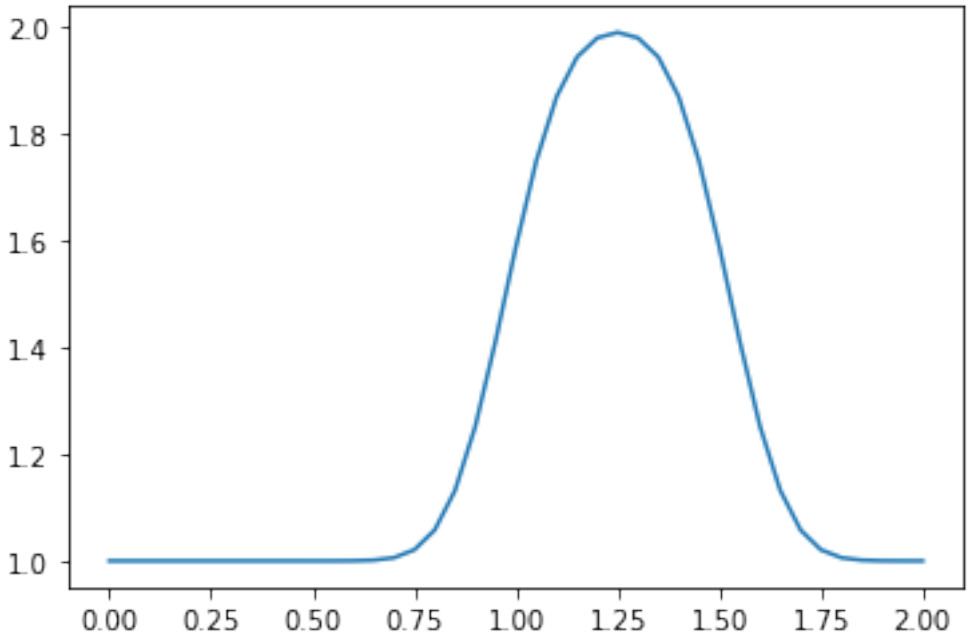
[15]:
```
import numpy as np
import matplotlib.pyplot as plt

def conveccionlineal1D(nx):
    x = 2.
    dx = x/(nx-1)
    nt = 20
    c = 1
    sigma = .5

    dt = sigma*dx

    ue = 2
    x1 = .5
    x2 = 1

    u = np.ones(nx)
    u[int(x1/dx) : int(x2/dx+1)] = ue
```

```
    un = np.ones(nx)

    for n in range(nt):  # teración a través del tiempo
        un[:] = u[:] ## copia los valores existentes de 'u' en 'un'
        for i in range(1,nx):
            u[i] = un[i]-c*dt/dx*(un[i]-un[i-1])

    plt.plot(np.linspace(0,x,nx),u)
```

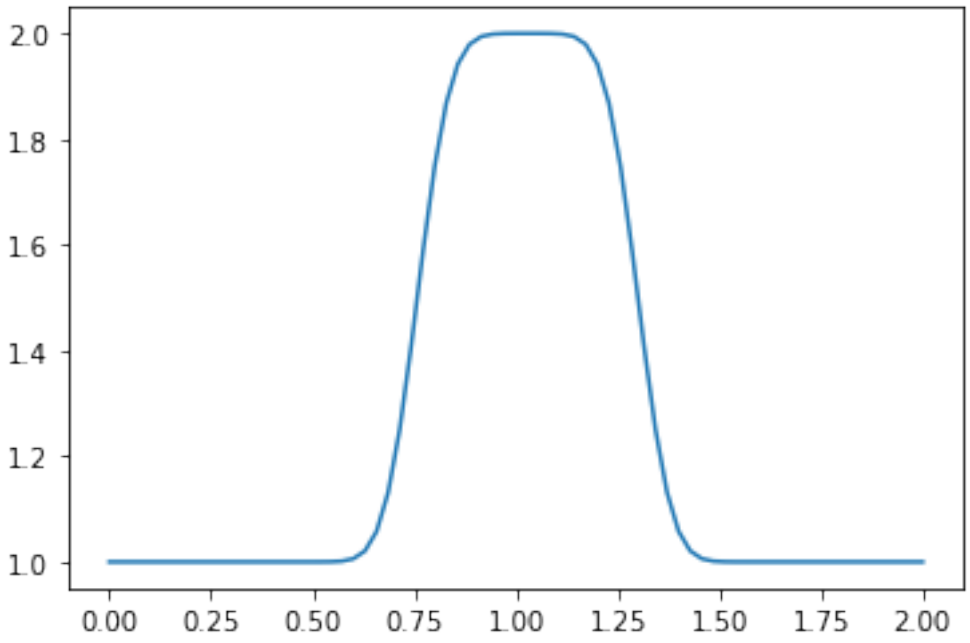
[16]:
```
conveccionlineal1D(31)
```



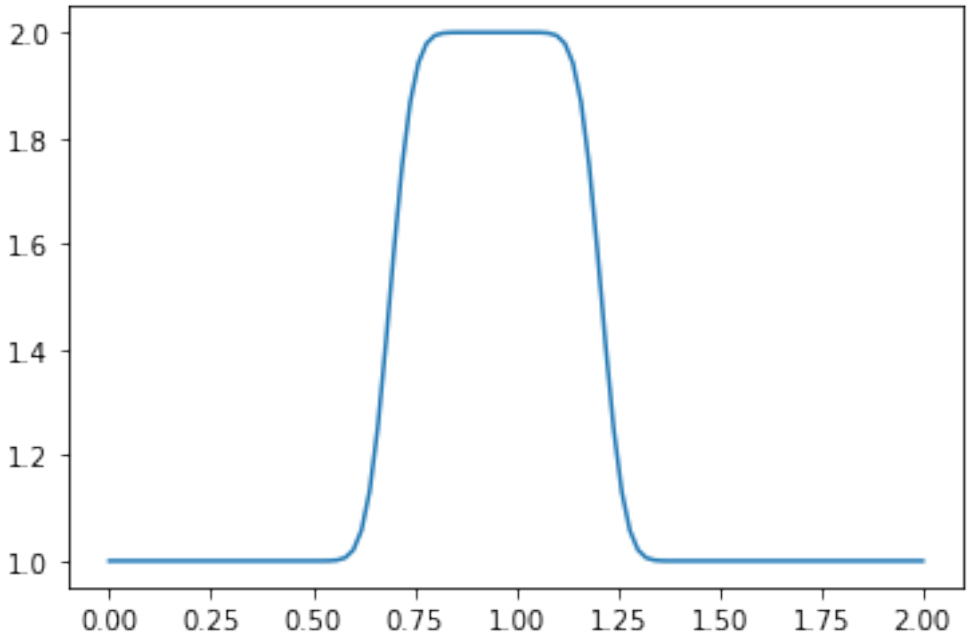
[17]:
```
conveccionlineal1D(41)
```



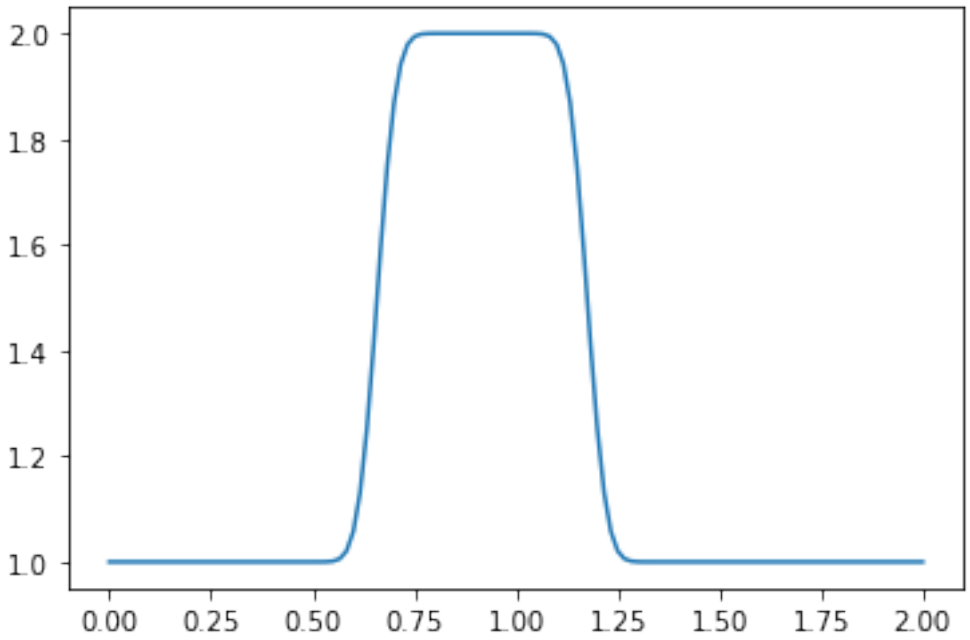
[18]:
```
conveccionlineal1D(71)
```



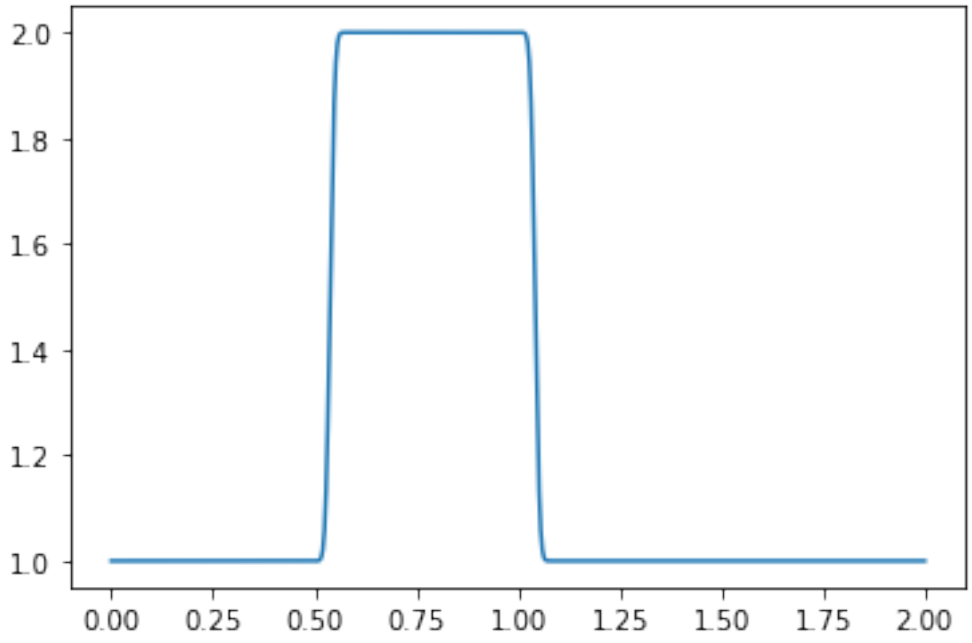
[19]:
```
conveccionlineal1D(101)
```



[20]:
```
conveccionlineal1D(121)
```



[22]:
```
conveccionlineal1D(501)
```



4 CAPITULO N03 Difusión

4.1 03.01. Ecuación de Difusión en 1-D

Se puede expresar de la siguiente forma la ecuación unidimensional:

$$\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} = 0$$

Para este caso al contar con una ecuación de segundo orden deberíamos saber como proceder para su discretización.

4.1.1 03.02. Discretizando entonces $\frac{\partial^2 u}{\partial x^2}$

Partiendo de la serie de expansión de Taylor de u_{i+1} y u_{i-1} en torno a u_i para la derivada de segundo orden tenemos:

$$u_{i+1} = u_i + \Delta x \frac{\partial u}{\partial x}\bigg|_i + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2}\bigg|_i + \frac{\Delta x^3}{3}\frac{\partial^3 u}{\partial x^3}\bigg|_i + O(\Delta x^4)$$

$$u_{i-1} = u_i - \Delta x \frac{\partial u}{\partial x}\bigg|_i + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2}\bigg|_i - \frac{\Delta x^3}{3}\frac{\partial^3 u}{\partial x^3}\bigg|_i + O(\Delta x^4)$$

Si hacemos la adición de estas dos ecuaciones notamos que las expresiones donde los terminos se encuentran en una posición par son anuladas por los signos que

llevan y al final para el caso de la $O(\Delta x^4)$ el valor que de su suma es muy pequeño por lo que no lo tomamos en cuenta, quedando:

$$u_{i+1} + u_{i-1} = 2u_i + \Delta x^2 \frac{\partial^2 u}{\partial x^2}\bigg|_i + O(\Delta x^4)$$

De lo anterior queríamos discretizar $\frac{\partial^2 u}{\partial x^2}\Big|_i$ así que:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + O(\Delta x^2)$$

4.1.2 03.03. Regresando a la ecuación de difusión lineal en 1-D

Escribimos la ecuación de difusión lineal en 1-D en su forma discretizada:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}$$

Observando esta ecuación nos queda por encontrar el valor de $O(\Delta x^4)$, para lo cual comenzamos despejando:

$$u_i^{n+1} = u_i^n + \frac{\nu \Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

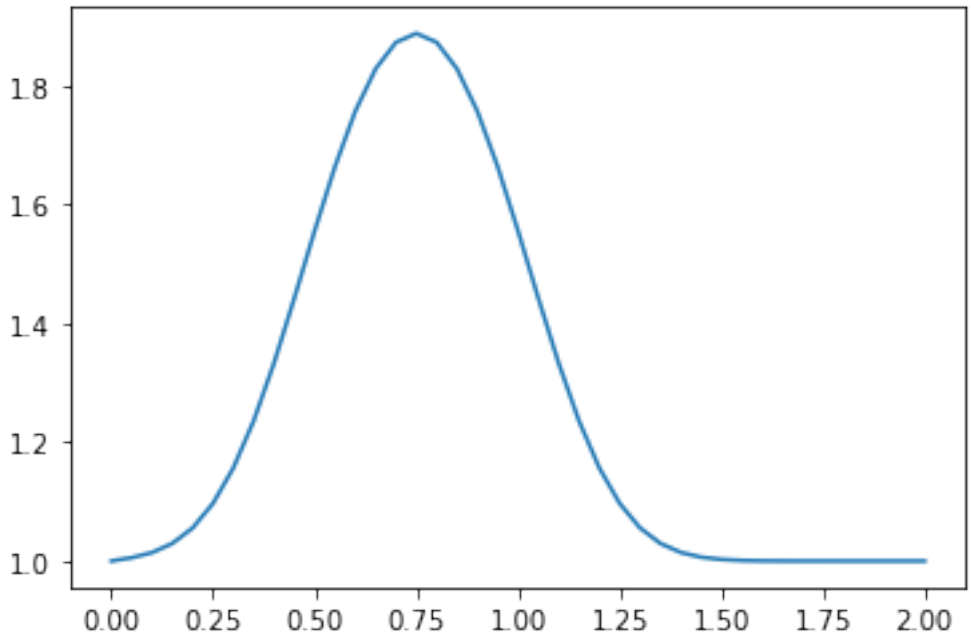
Vamos a implementar esta ecuación en nuestro código, las condiciones iniciales son $t = 0$, $u = 2$ para el intervalo $0.5 \leq x \leq 1$ y $u = 1$ para todo lo demás.

[27]:
```
%pylab inline

import numpy as np
import matplotlib.pyplot as plt

nx = 41
x = 2.
dx = x/(nx-1)
nt = 20
nu = 0.3
sigma = .3
dt = sigma*(dx**2)/nu

x1 = 0.5
x2 = 1
ue = 2

u = np.ones(nx)
u[int(x1/dx) : int(x2/dx+1)] = ue

un = np.ones(nx)
```

```
for n in range(nt):
    un[:] = u[:]
    for i in range(1, nx - 1):
        u[i] = un[i] + nu*(dt/dx**2)*(un[i+1]-2*un[i]+un[i-1])

plt.plot(np.linspace(0, x, nx), u)
```

```
Populating the interactive namespace from numpy and matplotlib
```

[27]: [<matplotlib.lines.Line2D at 0x7f9950274cf8>]



5 CAPITULO N04: Ecuación de Burgers

Para una dimensión espacial la ecuación de Burgers es:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$$

Como se vé esta entonces viene a ser resultado de la mezcla de las ecuaciones de convección y difusión, en este momento estamos viendo para el caso de 1 dimensión.

La discretizamos con diferencias delantadas en el tiempo y retrazadas para el espacio para las derivadas de orden 1º, para la derivada de orden 2º hacemos

diferencias centradas como vimos en las celdas anteriores, logrando:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}$$

Es nuestra tarea hallar el valor de u_i^{n+1} por lo que la despejamos:

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) + \nu \frac{\Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

5.1 04.02. Condiciones iniciales y de contorno

Vamos a explorar algunos atributos de la ecuación de Burgers con diferentes condiciones iniciales y de contorno respecto a las que ya hemos visto.

Para esta demostración usaremos:

$$u = -\frac{2\nu}{\phi}\frac{\partial \phi}{\partial x} + 4 \tag{1}$$

$$\phi = \exp\left(\frac{-x^2}{4\nu}\right) + \exp\left(\frac{-(x-2\pi)^2}{4\nu}\right) \tag{2}$$

La solución anaística es:

$$u = -\frac{2\nu}{\phi}\frac{\partial \phi}{\partial x} + 4 \tag{3}$$

$$\phi = \exp\left(\frac{-(x-4t)^2}{4\nu(t+1)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4\nu(t+1)}\right) \tag{4}$$

La condición de frontera:

$$u(0) = u(2\pi)$$

A esto se le denomina condición de frontera *periódica*.

5.2 04.03. Sympy para matemática simbólica

Vamos a utilizar la librería Sympy la cuál nos ayudará a evaluar la confición inicial dada por $\frac{\partial \phi}{\partial x}$, esto con fines de ágilidad de calculo.

[29]:
```
import numpy as np
import sympy
from sympy import init_printing
init_printing(use_latex=True)
```

```
x, nu, t = sympy.symbols('x nu t')
phi = sympy.exp(-(x-4*t)**2/(4*nu*(t+1))) + sympy.exp(-(x-4*t-2*np.pi)**2/
 ↪(4*nu*(t+1)))
phi
```

[29]:
$$e^{-\frac{(-4t+x-6.28318530717959)^2}{4\nu(t+1)}} + e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}$$

[30]:
```
phiprime = phi.diff(x)
phiprime
```

[30]:
$$-\frac{(-8t+2x)\,e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}}{4\nu\,(t+1)} - \frac{(-8t+2x-12.5663706143592)\,e^{-\frac{(-4t+x-6.28318530717959)^2}{4\nu(t+1)}}}{4\nu\,(t+1)}$$

[6]:
```
print(phiprime)
```

```
-(-8*t + 2*x)*exp(-(-4*t + x)**2/(4*nu*(t + 1)))/(4*nu*(t + 1)) - (-8*t + 2*x -
12.5663706143592)*exp(-(-4*t + x - 6.28318530717959)**2/(4*nu*(t + 1)))/(4*nu*(t
+ 1))
```

[31]:
```
from sympy.utilities.lambdify import lambdify

u = -2*nu*(phiprime/phi)+4
u
```

[31]:
$$-\frac{2\nu\left(-\frac{(-8t+2x)e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}}{4\nu(t+1)} - \frac{(-8t+2x-12.5663706143592)e^{-\frac{(-4t+x-6.28318530717959)^2}{4\nu(t+1)}}}{4\nu(t+1)}\right)}{e^{-\frac{(-4t+x-6.28318530717959)^2}{4\nu(t+1)}} + e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}} + 4$$

Lambdify

Lambdify transforma la experesión a una función que se pueda usar, solo bastará con decirle a lambdify qué variables son las de entrada y las de salida.

- Variables de entrada:
- (t,x,nu)
- Variables de salida:
- (u)

[32]:
```
ufunc = lambdify((t, x, nu), u)
ufunc(1,4,3)
```

[32]:
3.4917066420644494

Nuevamente viendo la ecuación de Burgers

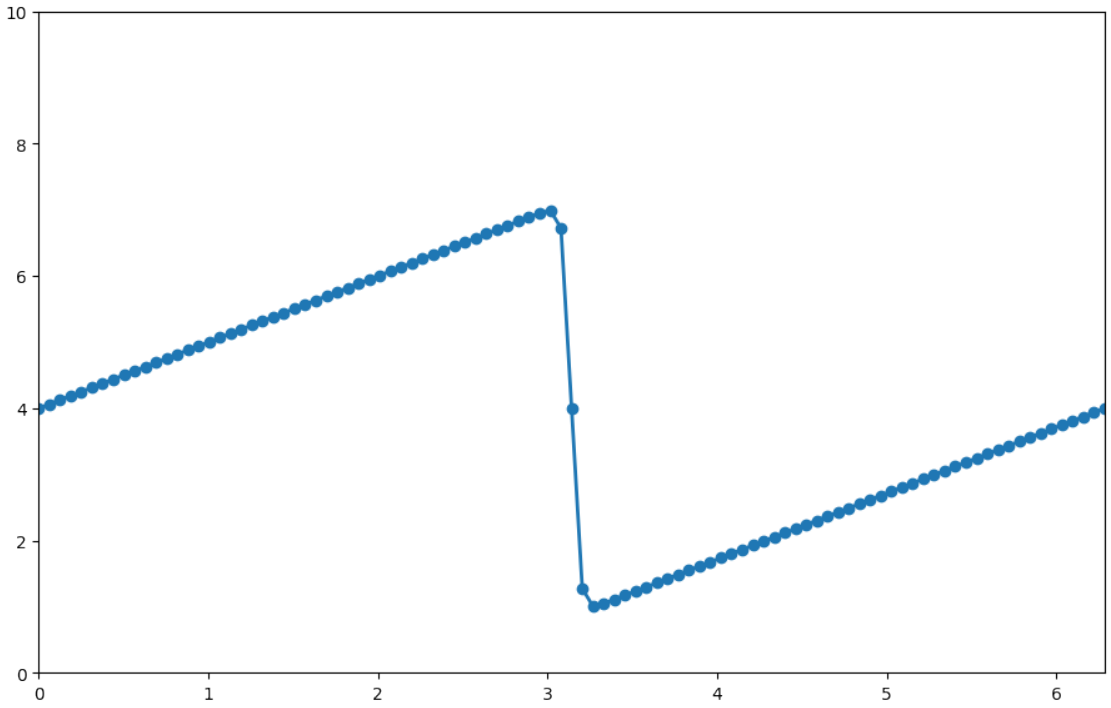
Ya contamos con las condiciones iniciales defindas y se puede continuar para finalizar el problema con su solución.

[33]:
```
import matplotlib.pyplot as plt

x = 2.
nx = 101
nt = 100
dx = x*np.pi/(nx-2)
nu = 0.07
dt = dx*nu

x = np.linspace(0, x*np.pi, nx)
un = np.empty(nx)
t = 0

u = np.asarray([ufunc(t, x0, nu) for x0 in x])
u
```

[33]:
```
array([4.        , 4.06283185, 4.12566371, 4.18849556, 4.25132741,
       4.31415927, 4.37699112, 4.43982297, 4.50265482, 4.56548668,
       4.62831853, 4.69115038, 4.75398224, 4.81681409, 4.87964594,
       4.9424778 , 5.00530965, 5.0681415 , 5.13097336, 5.19380521,
       5.25663706, 5.31946891, 5.38230077, 5.44513262, 5.50796447,
       5.57079633, 5.63362818, 5.69646003, 5.75929189, 5.82212374,
       5.88495559, 5.94778745, 6.0106193 , 6.07345115, 6.136283  ,
       6.19911486, 6.26194671, 6.32477856, 6.38761042, 6.45044227,
       6.51327412, 6.57610598, 6.63893783, 6.70176967, 6.76460125,
       6.82742866, 6.89018589, 6.95176632, 6.99367964, 6.72527549,
       4.        , 1.27472451, 1.00632036, 1.04823368, 1.10981411,
       1.17257134, 1.23539875, 1.29823033, 1.36106217, 1.42389402,
       1.48672588, 1.54955773, 1.61238958, 1.67522144, 1.73805329,
       1.80088514, 1.863717  , 1.92654885, 1.9893807 , 2.05221255,
       2.11504441, 2.17787626, 2.24070811, 2.30353997, 2.36637182,
       2.42920367, 2.49203553, 2.55486738, 2.61769923, 2.68053109,
       2.74336294, 2.80619479, 2.86902664, 2.9318585 , 2.99469035,
       3.0575222 , 3.12035406, 3.18318591, 3.24601776, 3.30884962,
       3.37168147, 3.43451332, 3.49734518, 3.56017703, 3.62300888,
       3.68584073, 3.74867259, 3.81150444, 3.87433629, 3.93716815,
       4.        ])
```

[34]:
```
plt.figure(figsize=(11,7), dpi=100)
plt.plot(x,u, marker='o', lw=2)
plt.xlim([0,2*np.pi])
plt.ylim([0,10])
```

[34]:
(0, 10)



Nos habíamos acostumbrado a ver a la función sombrero y esta es alejada a ella, le llamaremos "función de diente de sierra".

5.2.1 Condiciones de contorno periódicas

Las condiciones de contorno periódica hacen que nuestra función sombrero al llegar al extremo de la derecha pueda retornar hacia la izquierda, en los casos anteriores cuando aumentadabamos el nt simplemente nuestra función se desplazaba hacia la derecha hasta que esta dejaba de figurar en la pantalla, con las funciones de contorno esto ya no sucede sino que hay un retorno, un cambio de dirección.

Del capítulo 4 respecto a la ecuación de Burguer que combina a las ecuaciones de convección y difusión en su forma discretizada y despejada para u_i^{n+1} tenemos:

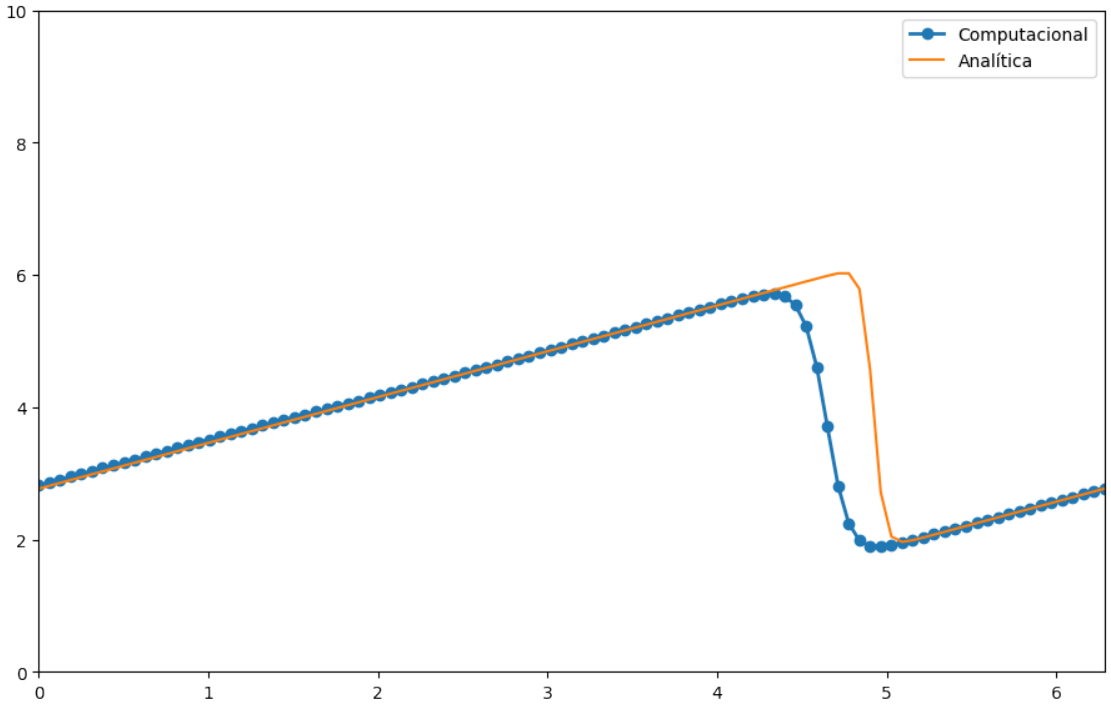
$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) + \nu \frac{\Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

[35]:
```
for n in range(nt):
    un[:]=u[:]
    for i in range(nx-1):
        u[i] = un[i] - un[i] * dt/dx *(un[i] - un[i-1]) + nu*dt/dx**2*\
                (un[i+1]-2*un[i]+un[i-1])
    u[-1] = un[-1] - un[-1] * dt/dx * (un[-1] - un[-2]) + nu*dt/dx**2*\
```

```
                (un[0]-2*un[-1]+un[-2])

u_analytical = np.asarray([ufunc(nt*dt, xi, nu) for xi in x])
```

[36]:
```
plt.figure(figsize=(11,7), dpi=100)
plt.plot(x,u, marker='o', lw=2, label='Computacional')
plt.plot(x, u_analytical, label='Analítica')
plt.xlim([0,2*np.pi])
plt.ylim([0,10])
plt.legend()
```

[36]: <matplotlib.legend.Legend at 0x7f9947ac2b00>



6 Bibliografía:

- Este es un contenido adaptado del curso de la Dra. Lorena Barba denominado "12 Steps to Navier-Stokes"

[ ]: