SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
TECNOLÓGICA

INSTITUTO TECNOLÓGICO
DE OAXACA

INSTITUTO TECNOLÓGICO DE OAXACA

“Tecnología Propia e Independencia Económica”

INGENIERIA EN SISTEMAS COMPUTACIONALES

Tópicos de ciencia de los datos

Documentación de códigos

Docente: Pedro Antonio Peralta Regalado

15:00 – 16:00 HRS Grupo: 4SC

Presenta:

Agustín Cruz Everardo Álvaro **17161051**

Oaxaca de Juárez

Contents

Documentacion de Código

1. Sistemas de ecuaciones lineales.

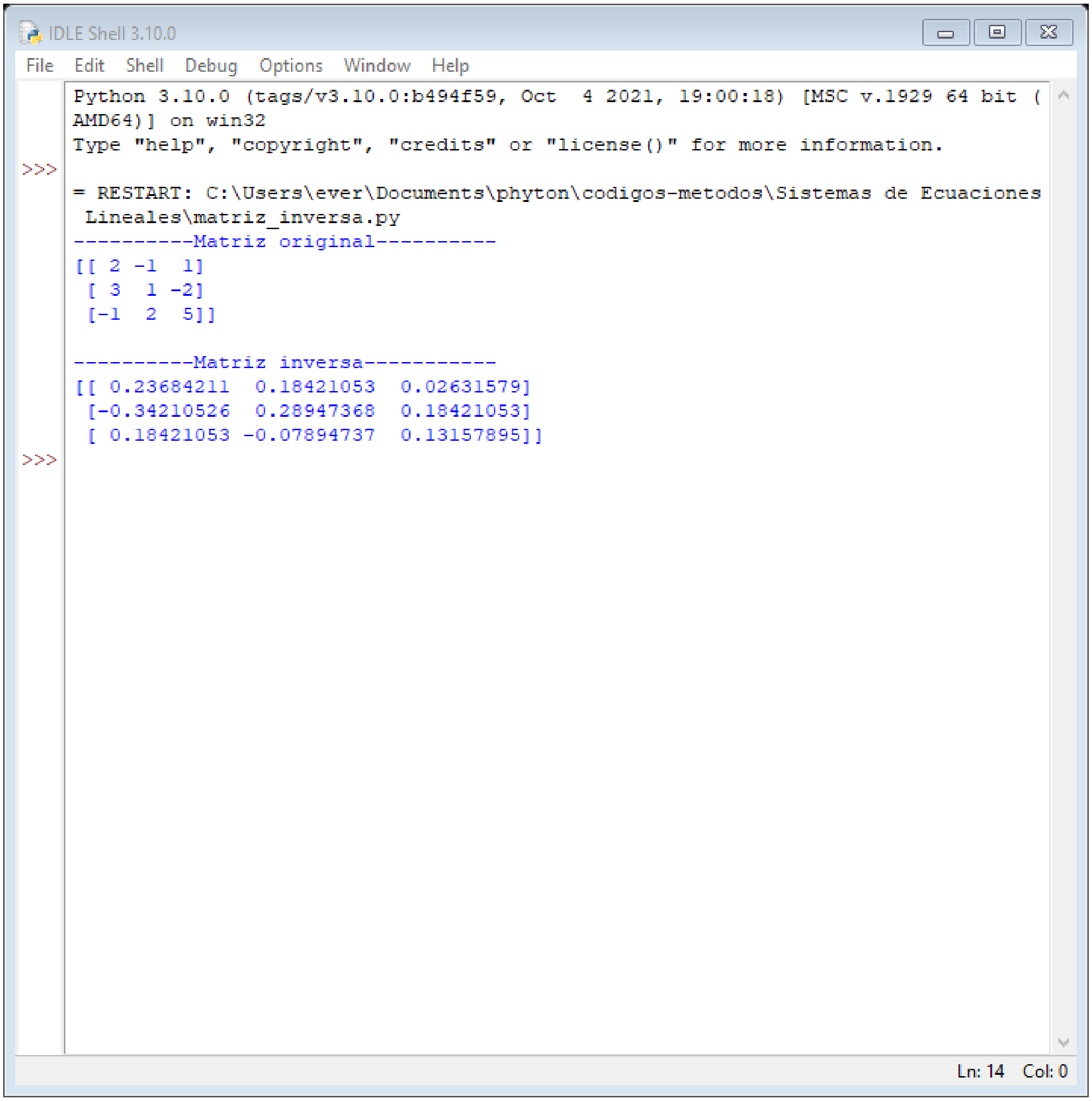
a) Método de Gauss.

```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones
 Lineales\gauss.py
Matriz aumentada:
[[ 2. -1.  1.  2.]
 [ 3.  1. -2.  9.]
 [-1.  2.  5. -5.]]
Pivoteo parcial por filas
[[ 3.  1. -2.  9.]
 [-1.  2.  5. -5.]
 [ 2. -1.  1.  2.]]
eliminacion hacia adelante
[[ 3.          1.         -2.          9.        ]
 [ 0.          2.33333333  4.33333333 -2.        ]
 [ 0.          0.          5.42857143 -5.42857143]]
eliminación hacia atrás
[[ 1.  0.  0.  2.]
 [ 0.  1.  0.  1.]
 [ 0.  0.  1. -1.]]
solución de X:
[[ 2.]
 [ 1.]
 [-1.]]
>>>
```

b) Factorización LU y PLU.

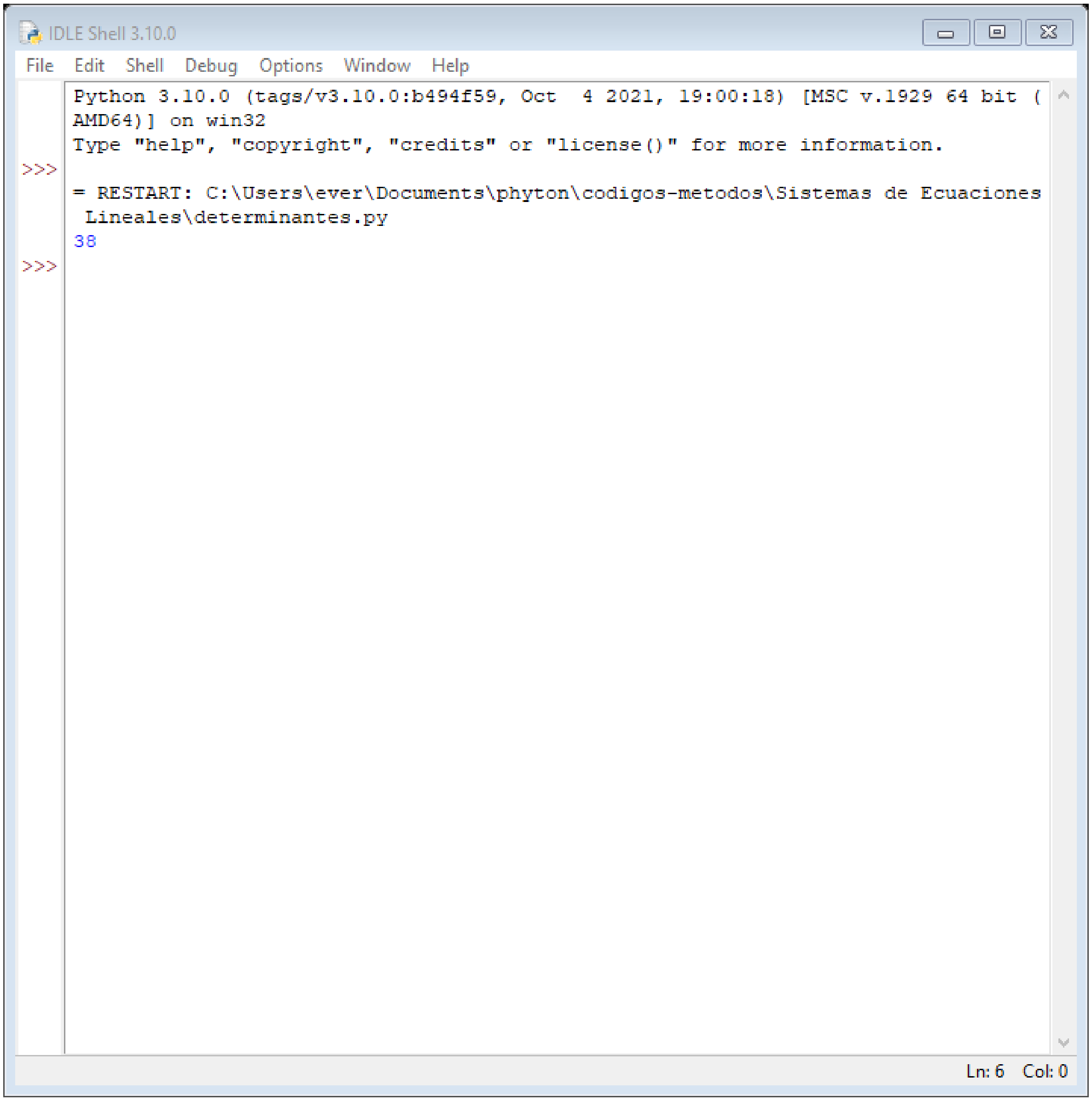
```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones
 Lineales\FactorizacionLU.py

Warning (from warnings module):
  File "C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones Li
neales\FactorizacionLU.py", line 4
    epsilon = np.finfo(np.float).eps
DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To
 silence this warning, use `float` by itself. Doing this will not modify any beh
avior and is safe. If you specifically wanted the numpy scalar type, use `np.flo
at64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdo
cs/release/1.20.0-notes.html#deprecations
L =
[[1. 0.]
 [3. 1.]]
U =
[[ 1.  1.]
 [ 0. -7.]]
>>>
Ln: 17   Col: 0
```

c) Inversa de una matriz.



```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones
 Lineales\matriz_inversa.py
----------Matriz original----------
[[ 2 -1  1]
 [ 3  1 -2]
 [-1  2  5]]

----------Matriz inversa-----------
[[ 0.23684211  0.18421053  0.02631579]
 [-0.34210526  0.28947368  0.18421053]
 [ 0.18421053 -0.07894737  0.13157895]]
>>>
Ln: 14   Col: 0
```

d) Determinantes.



e) Gauss Seidel.

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones
 Lineales\gauss-seidel.py
respuesta X:
[[ 3. ]
 [-2.5]
 [ 7. ]]
verificar A.X=B:
[[  7.84999999]
 [-19.3       ]
 [ 71.4       ]]
>>>
Ln: 13   Col: 0
```

f) Método de las potencias directa/inversa.

IDLE Shell 3.10.0

File Edit Shell Debug Options Window Help

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Sistemas de Ecuaciones
 Lineales\MetodoDelasPotencias.py
[[2. 1.]
 [1. 1.]]
[1. 1.]
[3. 2.]
[0.85065081 0.52573111]
[8.35246383e-41 1.61560147e-15 1.00000000e+00]
[100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204
 100.20749204 100.20749204 100.20749204 100.20749204 100.20749204]
>>> 
```

Ln: 31 Col: 0

2. Ecuaciones no lineales.

a) Método de bisección.

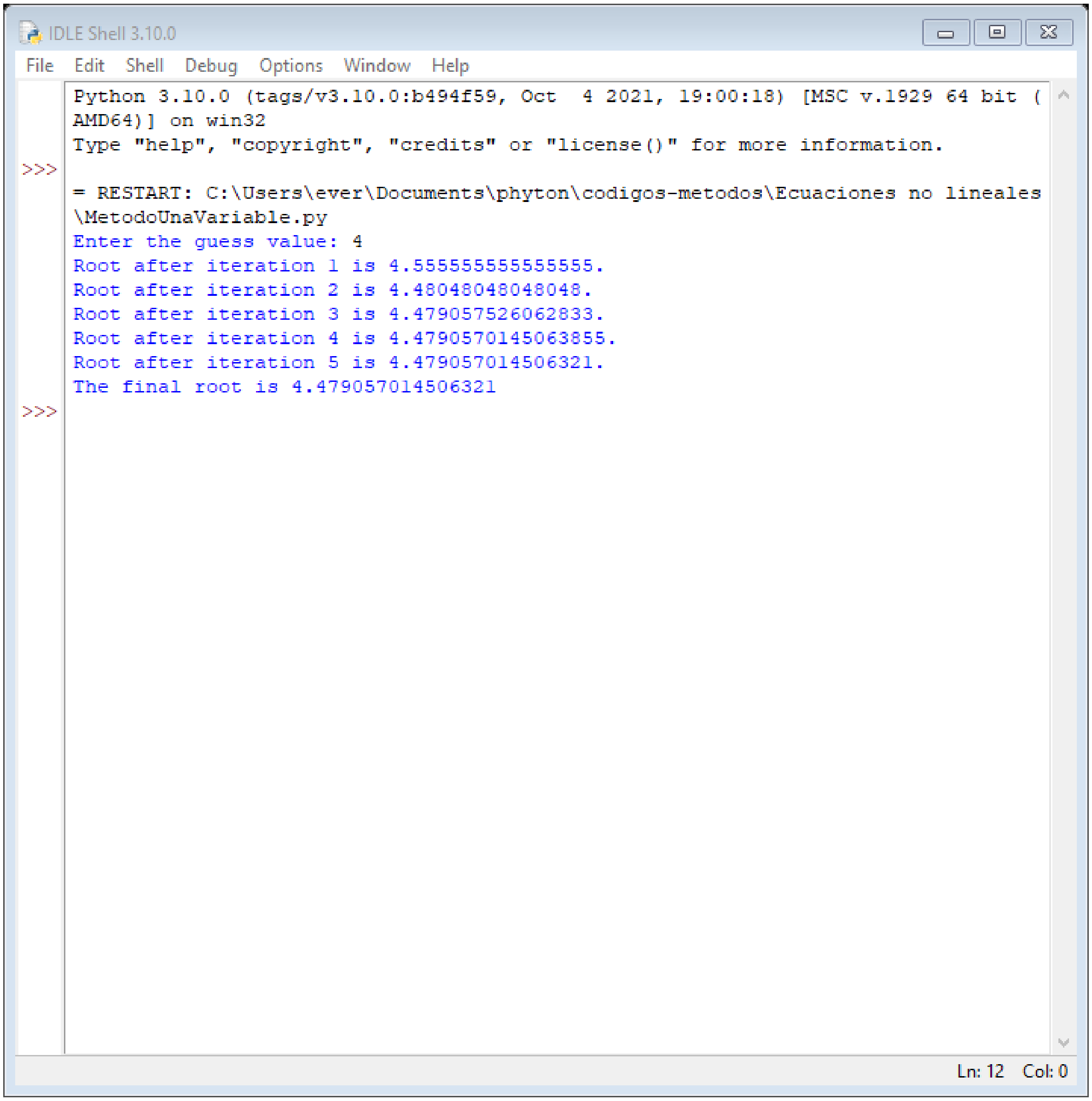
```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones no lineales
\A_MetodoBiseccion.py
Biseccion Funcion pol(x) = x^3 + 4x^2 -10:
i = 1 , p = 1.500000000000
i = 2 , p = 1.250000000000
i = 3 , p = 1.375000000000
i = 4 , p = 1.312500000000
i = 5 , p = 1.343750000000
i = 6 , p = 1.359375000000
i = 7 , p = 1.367187500000
i = 8 , p = 1.363281250000
i = 9 , p = 1.365234375000
i = 10, p = 1.364257812500
i = 11, p = 1.364746093750
i = 12, p = 1.364990234375
i = 13, p = 1.365112304688
i = 14, p = 1.365173339844
i = 15, p = 1.365203857422
i = 16, p = 1.365219116211
i = 17, p = 1.365226745605
i = 18, p = 1.365230560303
i = 19, p = 1.365228652954
i = 20, p = 1.365229606628
i = 21, p = 1.365230083466
i = 22, p = 1.365229845047
i = 23, p = 1.365229964256
i = 24, p = 1.365230023861
i = 25, p = 1.365229994059
i = 26, p = 1.365230008960
i = 27, p = 1.365230016410
Biseccino funcion trig(x) = xcos(x-1) - sen(x):
i = 1 , p = 5.000000000000
i = 2 , p = 5.500000000000
i = 3 , p = 5.750000000000
i = 4 , p = 5.625000000000
i = 5 , p = 5.562500000000
```

b) Método de falsa posición.

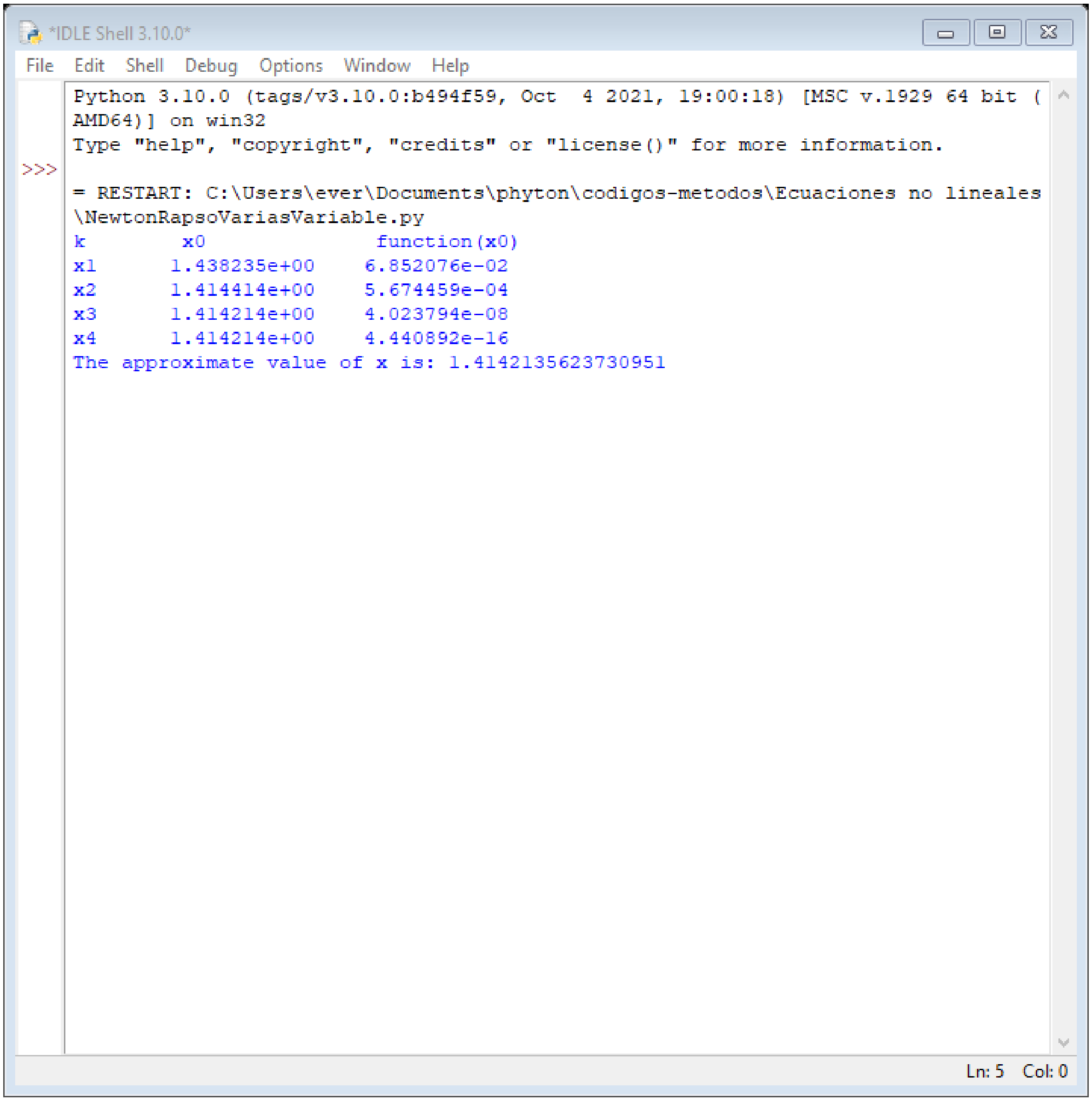
```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones no lineales
\B_falsa_posicion.py
Falsa posicion: pol(x) = x^3 + 4x^2 - 10
Iteracion => 0 , p = 1.263157894737
Iteracion => 1 , p = 1.338827838828
Iteracion => 2 , p = 1.358546341825
Iteracion => 3 , p = 1.363547440042
Iteracion => 4 , p = 1.364807031827
Iteracion => 5 , p = 1.365123717884
Iteracion => 6 , p = 1.365203303663
Iteracion => 7 , p = 1.365223301986
Iteracion => 8 , p = 1.365228327026
Iteracion => 9 , p = 1.365229589674
Iteracion => 10, p = 1.365229906941
Iteracion => 11, p = 1.365229986660
Iteracion => 12, p = 1.365230006692
Iteracion => 13, p = 1.365230011725
Falsa posicin: trig(x) =  xcos(x-1) - sin(x)
Iteracion => 0 , p = 5.235657374722
Iteracion => 1 , p = 5.569477410510
Iteracion => 2 , p = 5.597623035312
Iteracion => 3 , p = 5.599220749873
Iteracion => 4 , p = 5.599307996036
Iteracion => 5 , p = 5.599312749633
Iteracion => 6 , p = 5.599313008600
Iteracion => 7 , p = 5.599313022708
Iteracion => 8 , p = 5.599313023477
Falsa posicion: pote(x) = 7^x - 13
Iteracion => 0 , p = 0.500000000000
Iteracion => 1 , p = 0.835058241104
Iteracion => 2 , p = 1.045169783424
Iteracion => 3 , p = 1.168847080360
Iteracion => 4 , p = 1.238197008244
Iteracion => 5 , p = 1.275862101477
Iteracion => 6 , p = 1.295933755010
Iteracion => 7 , p = 1.306516642232
                                                              Ln: 119   Col: 0
```

c) Método de Newton/Raphson.

Una variable.

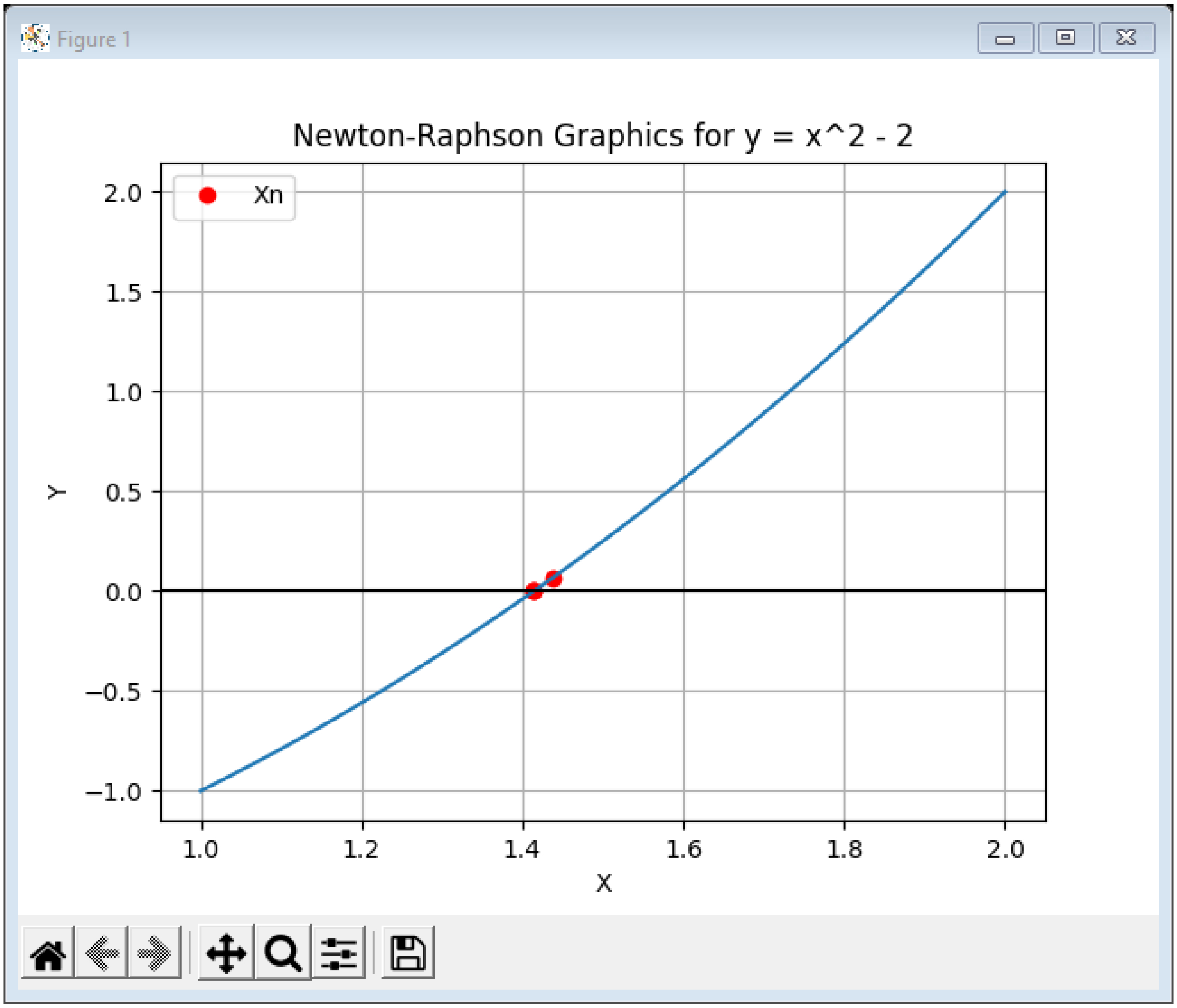


Varias variables.



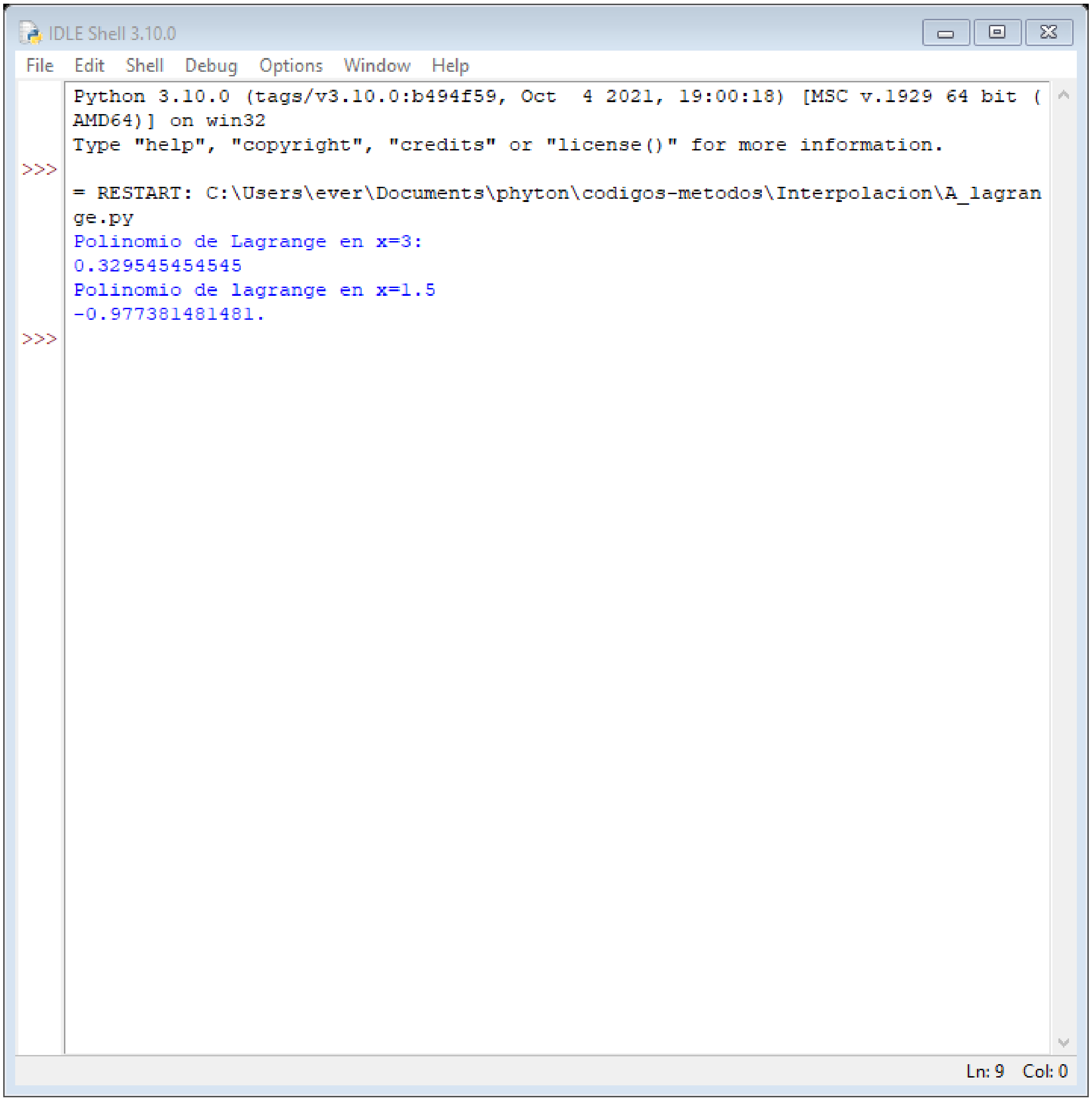
```
*IDLE Shell 3.10.0*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones no lineales
\NewtonRapsoVariasVariable.py
k         x0               function(x0)
x1      1.438235e+00     6.852076e-02
x2      1.414414e+00     5.674459e-04
x3      1.414214e+00     4.023794e-08
x4      1.414214e+00     4.440892e-16
The approximate value of x is: 1.4142135623730951

Ln: 5   Col: 0
```

Grafica

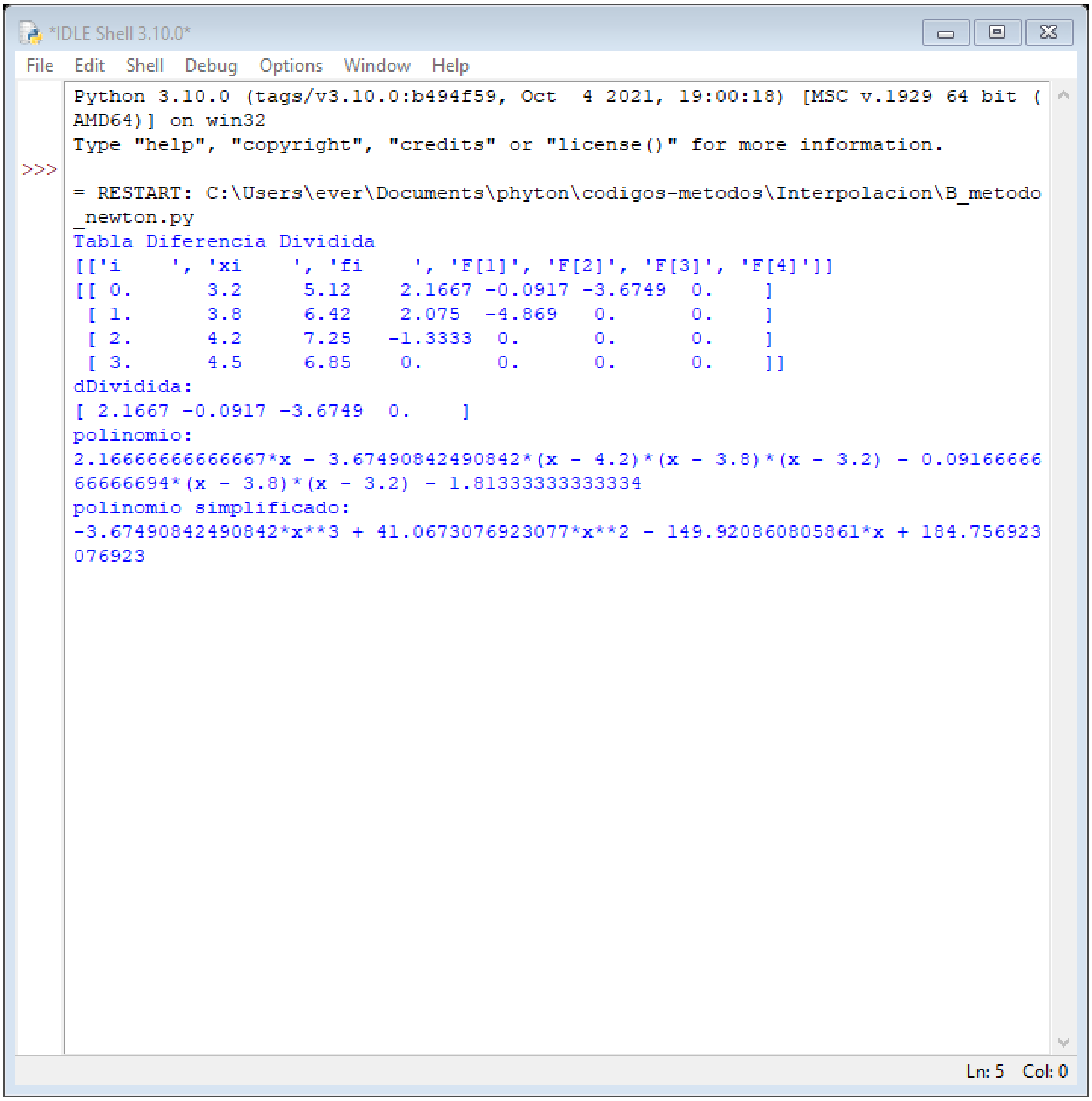


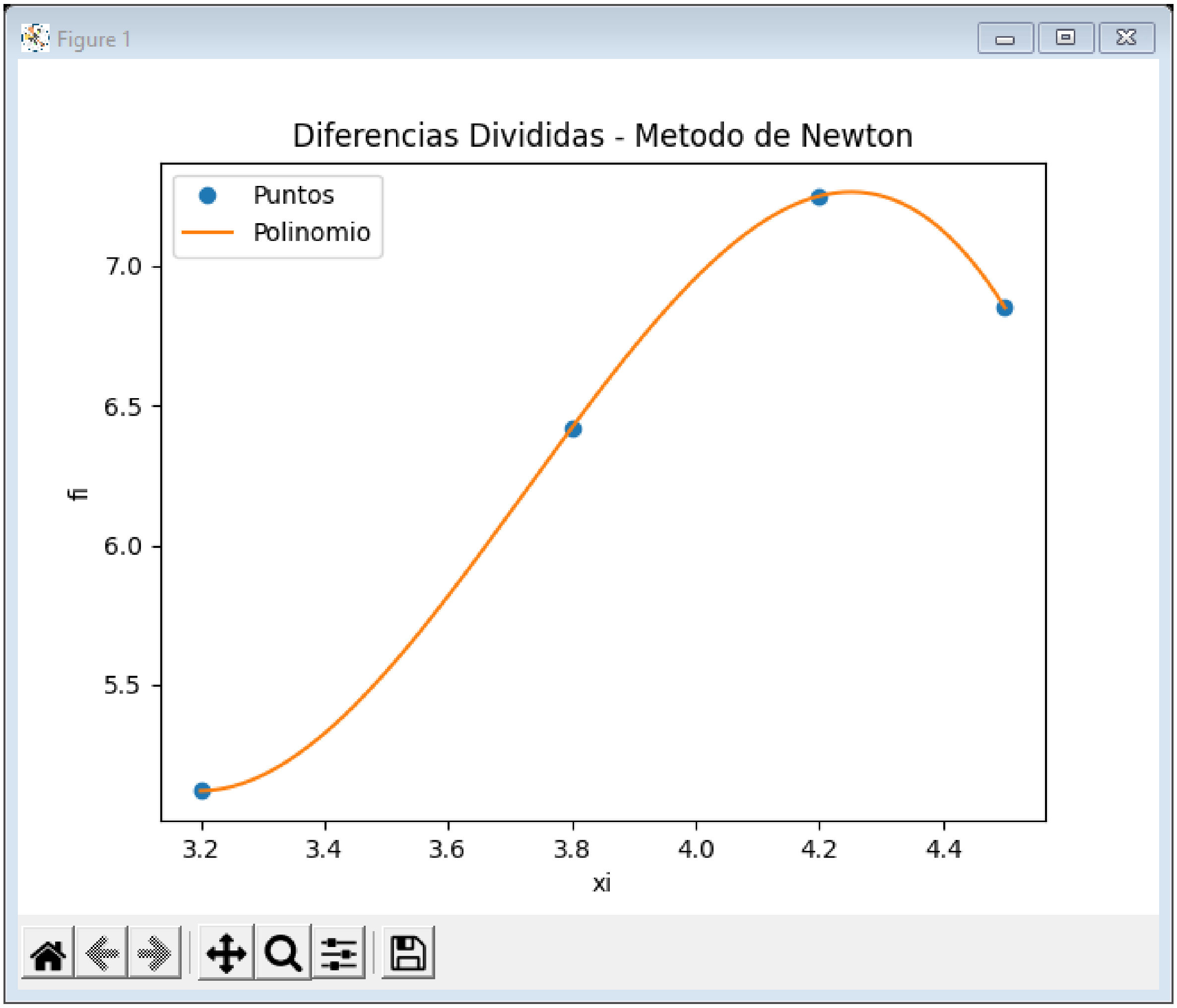
3. Interpolación.

Método de Lagrange.



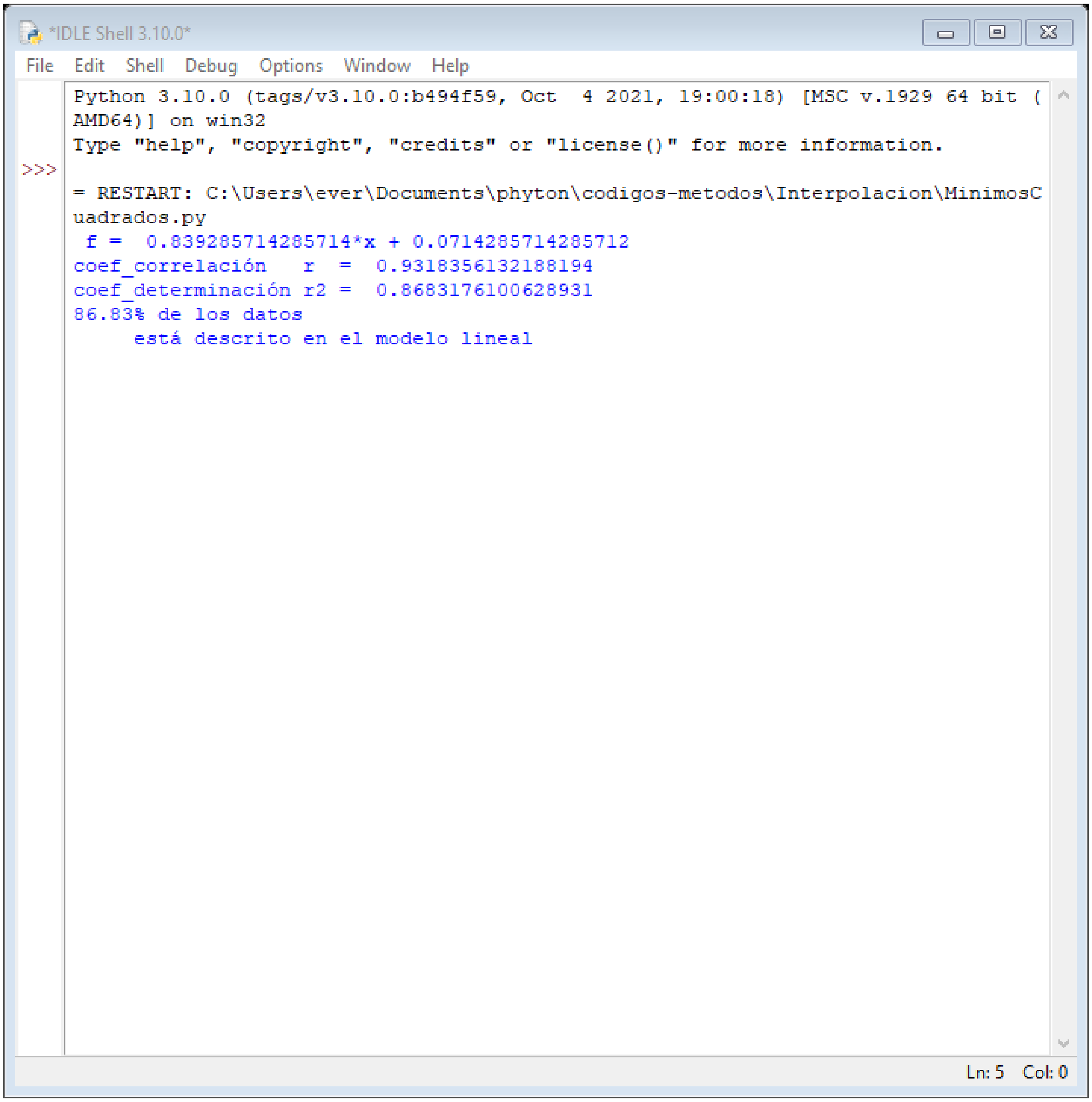
Método de Newton.



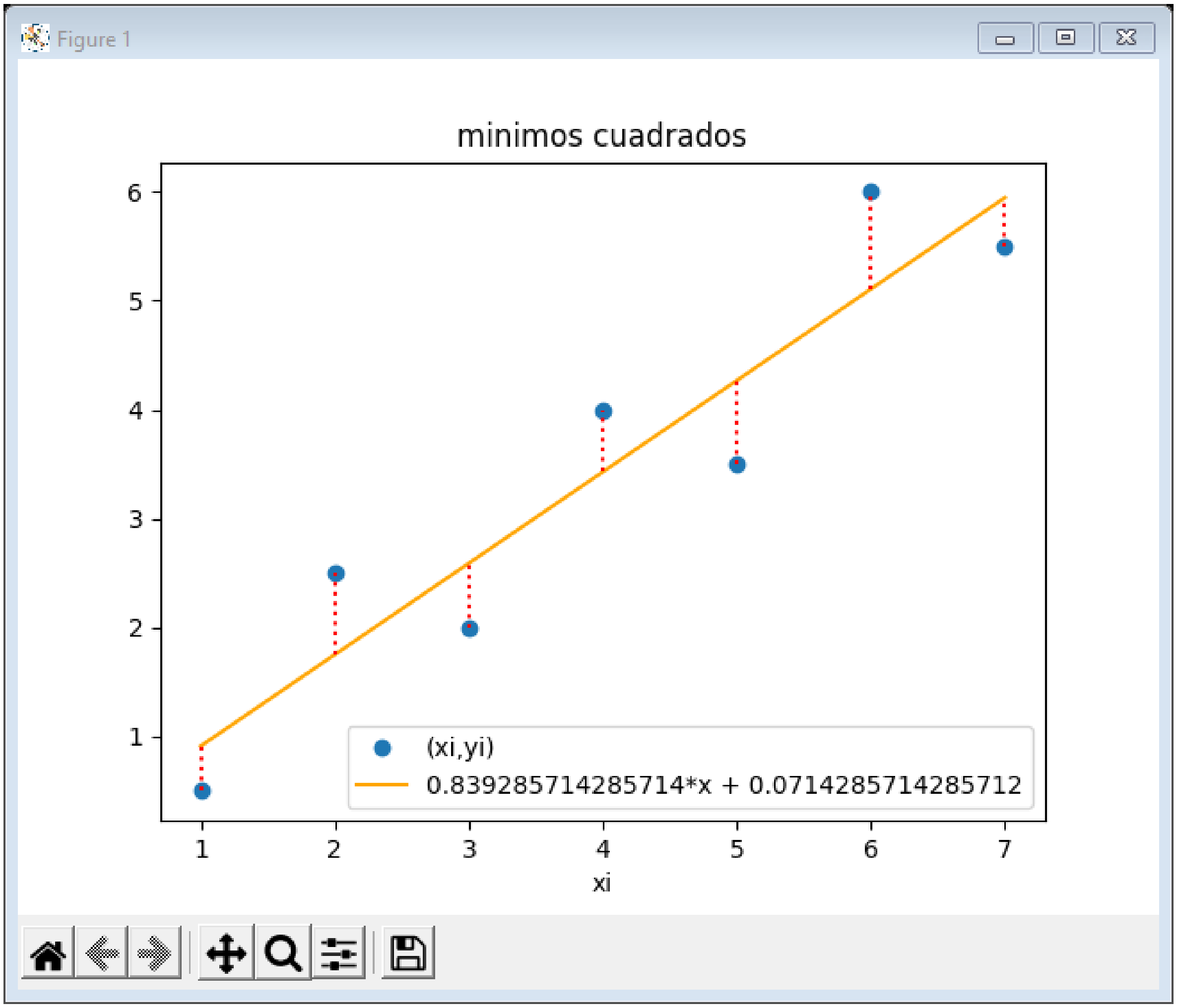


4 CAPÍTULO

a) Ajuste de un polinomio por mínimos cuadrados.



```
*IDLE Shell 3.10.0*
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Interpolacion\MinimosC
uadrados.py
 f =  0.839285714285714*x + 0.0714285714285712
coef_correlación   r  =  0.9318356132188194
coef_determinación r2 =  0.8683176100628931
86.83% de los datos
     está descrito en el modelo lineal
Ln: 5   Col: 0
```



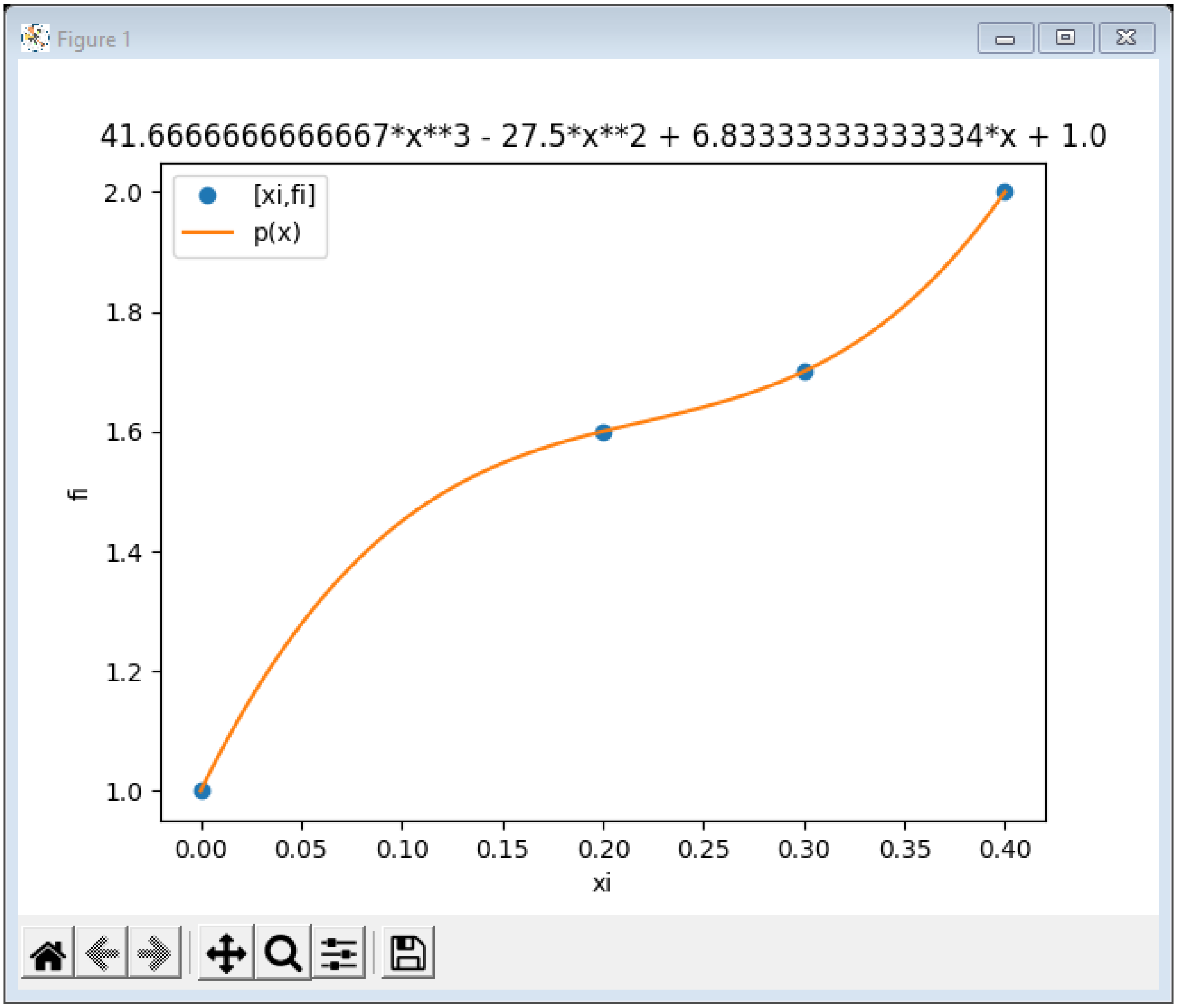
Interpoladores cúbicos.

IDLE Shell 3.10.0

File Edit Shell Debug Options Window Help

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Interpolacion\interpol
adoresCubicos.py
Matriz Vandermonde:
[[0.    0.    0.    1.   ]
 [0.008 0.04  0.2   1.   ]
 [0.027 0.09  0.3   1.   ]
 [0.064 0.16  0.4   1.   ]]
los coeficientes del polinomio:
[ 41.66666667 -27.5          6.83333333   1.        ]
Polinomio de interpolación:
41.6666666666667*x**3 - 27.5*x**2 + 6.83333333333334*x + 1.0

 formato pprint
                  3          2
41.6666666666667·x  - 27.5·x  + 6.83333333333334·x + 1.0
```

Ln: 5   Col: 0



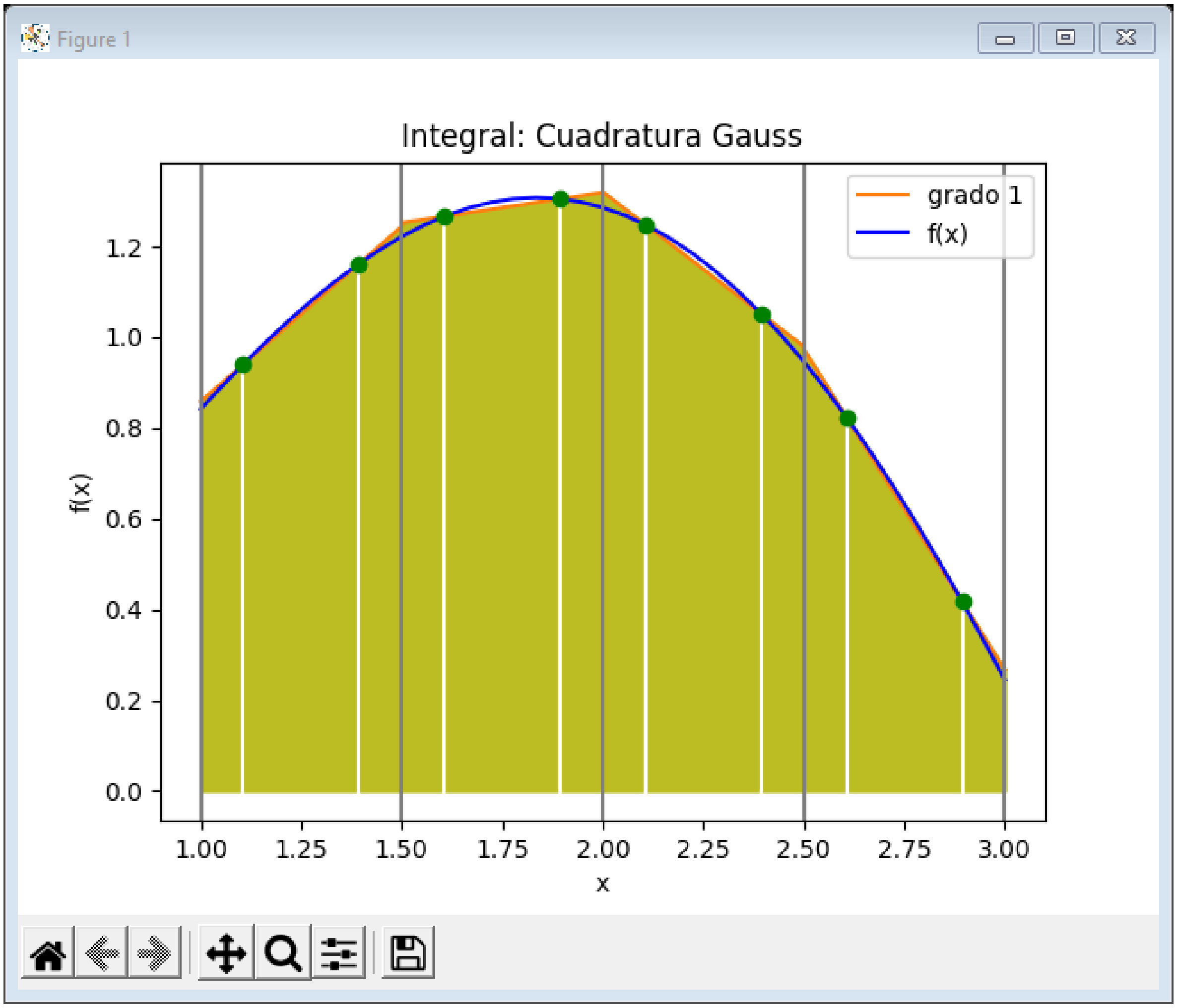
4. Cálculo numérico.

a) Derivación e integración de datos tabulados.

b) Derivación e integración de funciones.

c) Integrador en cuadraturas Gaussianas.

```
*IDLE Shell 3.10.0*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Calculo Numerico\Integ
radorencuadraturasGaussianas.py
Integral:  2.0535771900286597
Ln: 5   Col: 0
```

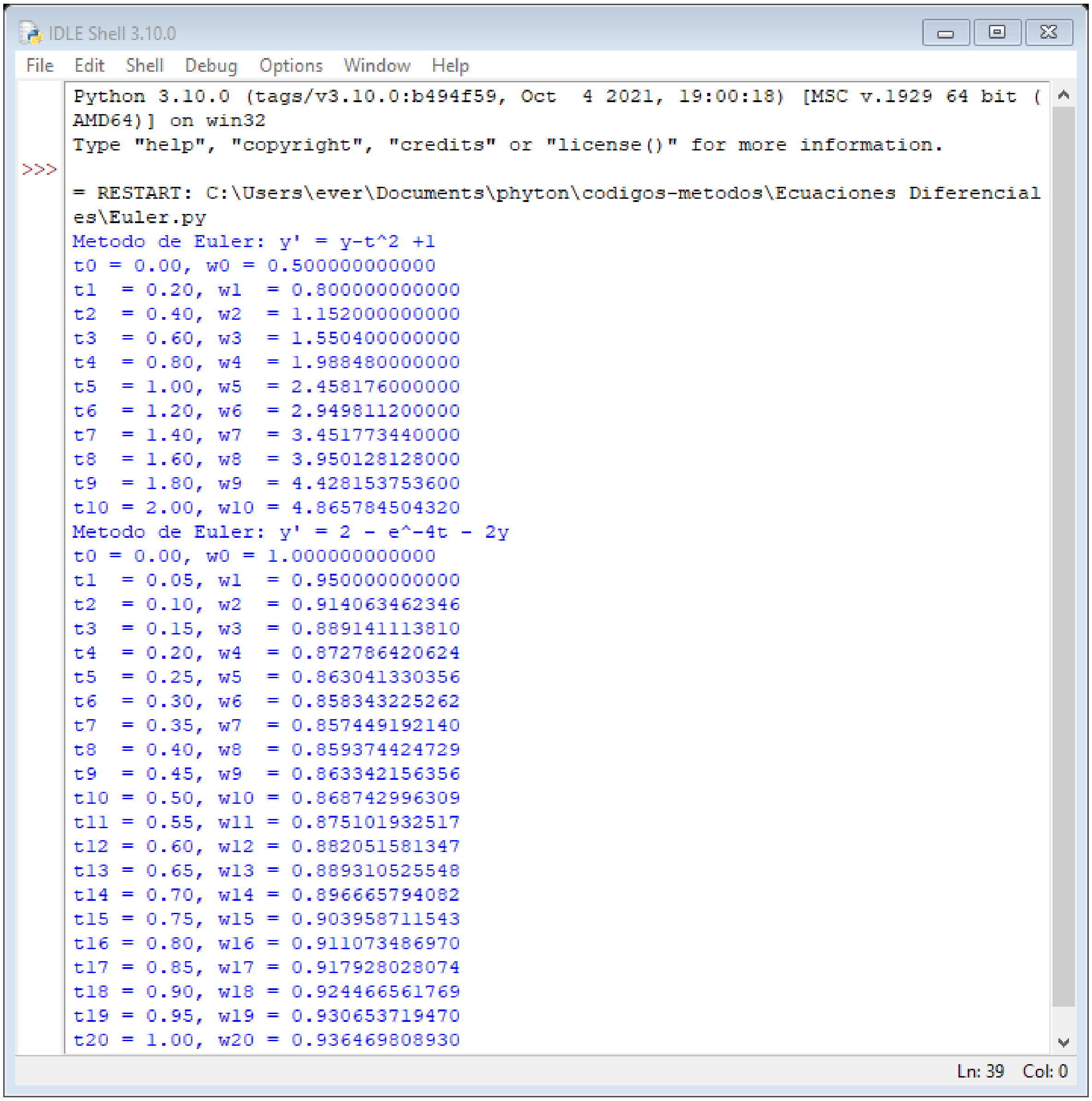


5. Ecuaciones diferenciales.

a) Métodos para resolver una ecuación diferencial, problema de condiciones iniciales.

Euler izquierdo.

Euler centrado.




```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones Diferencial
es\Euler.py
Metodo de Euler: y' = y-t^2 +1
t0 = 0.00, w0 = 0.500000000000
t1  = 0.20, w1  = 0.800000000000
t2  = 0.40, w2  = 1.152000000000
t3  = 0.60, w3  = 1.550400000000
t4  = 0.80, w4  = 1.988480000000
t5  = 1.00, w5  = 2.458176000000
t6  = 1.20, w6  = 2.949811200000
t7  = 1.40, w7  = 3.451773440000
t8  = 1.60, w8  = 3.950128128000
t9  = 1.80, w9  = 4.428153753600
t10 = 2.00, w10 = 4.865784504320
Metodo de Euler: y' = 2 - e^-4t - 2y
t0 = 0.00, w0 = 1.000000000000
t1  = 0.05, w1  = 0.950000000000
t2  = 0.10, w2  = 0.914063462346
t3  = 0.15, w3  = 0.889141113810
t4  = 0.20, w4  = 0.872786420624
t5  = 0.25, w5  = 0.863041330356
t6  = 0.30, w6  = 0.858343225262
t7  = 0.35, w7  = 0.857449192140
t8  = 0.40, w8  = 0.859374424729
t9  = 0.45, w9  = 0.863342156356
t10 = 0.50, w10 = 0.868742996309
t11 = 0.55, w11 = 0.875101932517
t12 = 0.60, w12 = 0.882051581347
t13 = 0.65, w13 = 0.889310525548
t14 = 0.70, w14 = 0.896665794082
t15 = 0.75, w15 = 0.903958711543
t16 = 0.80, w16 = 0.911073486970
t17 = 0.85, w17 = 0.917928028074
t18 = 0.90, w18 = 0.924466561769
t19 = 0.95, w19 = 0.930653719470
t20 = 1.00, w20 = 0.936469808930
```

Euler derecho.

■ Ejemplo 8.2 — Método de Euler modificado. Calcular la concentración final de A si la constante de velocidad de reacción $k = 2$ y la concentración inicial $C_{A_0} = 1.5$ en el tiempo 0, calcular en el tiempo $t_n = 0.6$

$$\mathrm{A} \xrightarrow{k=2} \mathrm{B}$$

$$\frac{dC_A}{dt} = -2C_A$$

Solución

```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones Diferencial
es\EulerHaciaDelante.py
>>> main()
x =  [0.   0.12 0.   0.   0.   0.  ]
y =  [1.5    1.1832 0.      0.      0.      0.    ]
Traceback (most recent call last):
```

Métodos de Runge/Kutta 3o orden.

Métodos de Runge/Kutta 4o orden.

Un tanque esférico de radio R está inicialmente lleno de agua. En el fondo del tanque hay un agujero de radio r, por el cual escapa el agua bajo la influencia de la gravedad. La ecuación diferencial que expresa la profundidad del agua como función del tiempo es

$$\frac{dy}{dt} + \frac{r^2\sqrt{2g}}{2R\sqrt{y} - \sqrt{y^3}} = 0$$

donde g = 32.2 ft/s

, R = 12 ft, r = 1/8 ft. La condición inicial es que en t = 0, y = 22. Encontrar

la altura del agua al minuto 1000

```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ever\Documents\phyton\codigos-metodos\Ecuaciones Diferencial
es\RangeK4.py
>>> main()
x =  [   0.  100.  200.  300.  400.  500.  600.  700.  800.  900. 1000.]
y =  [22.          20.9302613  20.13213069 19.46148227 18.86896063 18.33050353
 17.83222571 17.36524926 16.92348771 16.50255067 16.09914568]
```

Frontera

Ejemplo 1

— Problema de valor en la frontera por el método del disparo. Usando el

ejemplo de la barra metálica mostrado en la figura 8.11 y la ecuación 8.8, y los siguientes datos:

L = 10m Longitud de la barra

α = 0.01m

−2 Factor de dispersión de calor

Ta = 25◦C Temperatura ambiente

T(0) = 50◦C Temperatura de la barra en el extremo izquierdo

T(L) = 200◦C Temperatura de la barra en el extremo derecho

Solución

Convertir la ecuación de segundo orden en un sistema de ecuaciones de primer orden para aplicar

los métodos vistos.

dT

dx

= υ

d

2T

dx2

=

dυ

dx

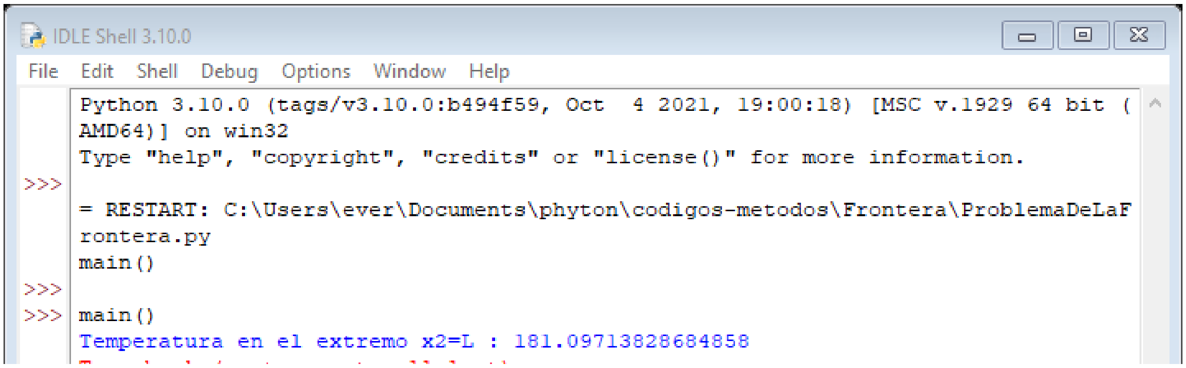
= α(T −Ta)

Esto es un sistema de ecuaciones ordinarias, con los valores iniciales T(x1 = 0) = 50 para la segunda

ecuación y υ(x1 = 0) =? un valor que debemos suponer para obtener un primer resultado para

T(x2 = L), si no es el valor esperado, entonces se hace otra suposición.

El primer valor supuesto será υ(x1 = 0) = 10



Bibliography

Enciclopedia virtual. (n.d.). Métodos numéricos. Obtenido de [*http://cca.org.mx*](http://cca.org.mx)

Método de la secante | La Guía de Matemática. (n.d.). Obtenido de [*https://matematica.laguia2000.com/general/metodo-de-la-secante-2*](https://matematica.laguia2000.com/general/metodo-de-la-secante-2)

Jorge Román. (2002). Integración Numérica: Método de Simpson. 2012, de GeoGebra Sitio web: [*https://www.geogebra.org/m/tWqP2wQs*](https://www.geogebra.org/m/tWqP2wQs)

Steve C. Chapra, Raymond P. Canales.. (2006). Métodos numéricos para ingenieros. México: McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V..

Gómez Fuentes, M. (2015). Métodos Numéricos Jacobi. Obtenido de Métodos Numéricos Jacobi: [*http://test.cua.uam.mx/MN/Methods/EcLineales/Jacobi/Jacobi.php*](http://test.cua.uam.mx/MN/Methods/EcLineales/Jacobi/Jacobi.php)

Moya Ramírez , O. Y. (Diciembre. de 2017). Unitec Campus En Línea. Obtenido de [*https://uniteconline.blackboard.com/webapps/portal/execute/tabs/tabAction?tab_tab_group_id=_53_1*](https://uniteconline.blackboard.com/webapps/portal/execute/tabs/tabAction?tab_tab_group_id=_53_1)

Rodríguez Guzmán, M. D. (Diciembre. de 2017). Unitec Campus en línea. Obtenido de [*https://uniteconline.blackboard.com/webapps/portal/execute/tabs/tabAction?tab_tab_group_id=_53_1*](https://uniteconline.blackboard.com/webapps/portal/execute/tabs/tabAction?tab_tab_group_id=_53_1)

Ajuste de curvas, la enciclopedia libre. (n.d.). Obtenido de [*https://es.wikipedia.org/wiki/Ajuste_de_curvas*](https://es.wikipedia.org/wiki/Ajuste_de_curvas)