Práctica 5 – Matemática Aplicada. Grupo 1^oB

Métodos numéricos de resolución de problemas de valores iniciales.

Índice

1. Introducción.

En esta práctica veremos dos métodos que permiten calcular una solución aproximada a la solución exacta de una ecuación diferencial, lo cual es útil en casos en que no se puede tener una solución exacta o si la solución exacta es muy complicada y no resulta práctica.

En ambos casos nos planteamos un problema de valor inicial como el siguiente, encontrar una función $y(x)$ definida en un intevalo I que verifique

$$\begin{cases} y'(x) = F(x,y), & x \in I, \\ y(x_0) = y_0. \end{cases}$$

Sabemos que el problema tiene solución única si la función F es continua en un recinto del plano que contiene a (x_0, y_0) y existe, y es continua, la parcial $\dfrac{\partial F}{\partial y}$ en dicho recinto.

Los métodos que planteamos consideran ciertos valores x_i en el intervalo I y hallan valores aproximados a valores $y(x_i)$ de la solución.

Ejercicio 1. *Considere el problema*

$$e^x \cdot \frac{dy}{dx} + x \cdot \operatorname{sen}(y) = xy^2, \quad y(1) = 2.$$

Plantee en papel el problema como se ha descrito en la introducción, determinando la función F, x_0 e y_0.

2. El método de Euler. Un ejemplo.

El método consiste en construir una poligonal que aproxime la solución exacta del problema, llamada también curva integral. Para entender la idea del método comenzaremos viendo un ejemplo concreto. Nos planteamos el problema de valor inicial

$$\begin{cases} y'(x) = F(x, y) = x + y, & x \in (1, 2), \\ y(1) = 0. \end{cases}$$

La solución del problema es una función $y(x)$ que debe cumplir que $y(1) = 0$. Además, usando F, sabemos cuanto debe valer la derivada de $y(x)$ en $x = 1$, ya que $y'(1) = F(1, 0) = 1 + 0 = 1$.

Con esta información podemos hallar la recta que pasa por el punto $(x_0, y_0) = (1, 0)$ y tiene pendiente $F(x_0, y_0) = 1$,

$$r_0(x) = y_0 + F(x_0, y_0) \cdot (x - x_0) = 0 + 1 \cdot (x - 1) = x - 1.$$

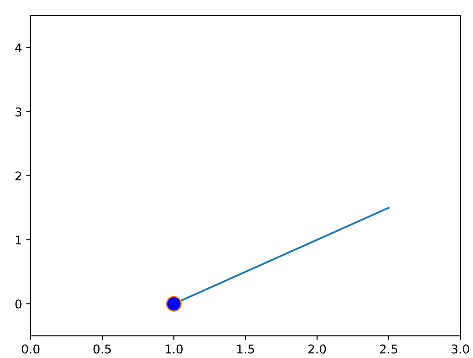

Figura 1: Recta tangente en el punto $(x_0, y_0) = (1, 0)$

La recta que hemos calculado es la recta tangente a la función $y(x)$, solución exacta de la ecuación diferencial.

Si pudiéramos representar la solución exacta $y(x)$ y la recta tangente, tendríamos una gráfica como la de la figura 2.

La recta tangente se usa como una aproximación a la función $y(x)$ en el entorno del punto $x = 1$. Si usáramos la recta tangente para hallar $y(2)$ podríamos obtener una mala aproximación, pero si consideramos un punto cercano a $x = 1$ como $x_1 = 1.1$ la aproximación sería ‘mejor’.

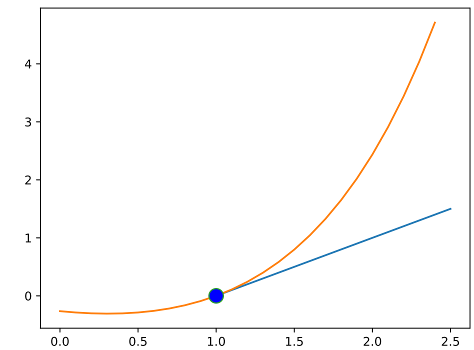

Figura 2: Solución exacta y recta tangente en el punto $(x_0, y_0) = (1, 0)$

Así, usamos la recta tangente que hemos calculado para hallar una aproximación de $y(1.1)$,

$$y(1.1) \simeq r(1.1) = 1.1 - 1 = 0.1.$$

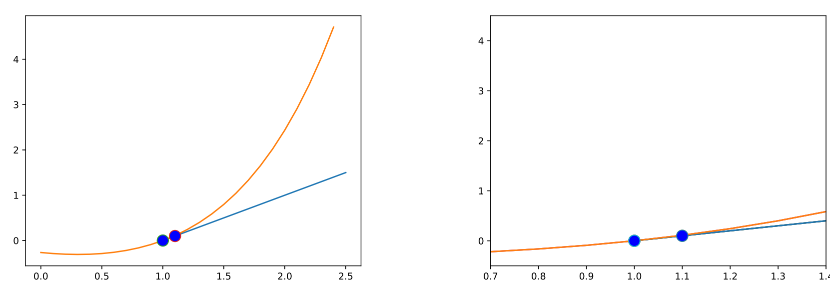

Figura 3: Solución exacta, $(x_0, y_0) = (1, 0)$, $(x_1, y_1) = (1.1, 0.1)$.

Ahora, con este valor aproximado de $y(1.1)$ podemos de nuevo definir la recta tangente en el punto 1.1, que pasa por $y(1.1) \simeq 0.1$ con pendiente $F(1.1, 0.1)$.

$$r_1(x) = y_1 + F(x_1, y_1) \cdot (x - x_1) = 1.1 + F(1.1, 0.1) \cdot (x - 1.1).$$

Evaluando en la expresión de F tenemos $F(1.1, 0.1) = 1.1 + 0.1 = 1.2$ y por tanto $r_1(x) = 1.1 + 1.2 \cdot (x - 1.1)$.

Esta recta tangente $r_1(x)$ sirve como aproximación cerca de $x_1 = 1.1$, y la usaremos para aproximar $y(x)$ en el punto $x_2 = 1.2$.

De esta forma se pueden ir construyendo rectas tangentes que nos van acercando al valor de $y(x)$ en $x_{10} = 2$.

3. Descripción y programación del método de Euler.

Nos planteamos de forma general el siguiente problema de valor inicial

$$\begin{cases} y'(x) = F(x, y), & x \in (x_0, b), \\ y(x_0) = y_0. \end{cases}$$

Dividiremos el código en dos partes.

En la primera parte del programa cargamos los módulos `numpy` y, si queremos hacer gráficas, `matplotlib`. Luego definimos el punto (x_0, y_0) con la condición inicial, y también la función $F(x, y)$ que define la ecuación diferencial y que nos servirá para hallar las pendientes de las rectas tangentes. En b definimos el último valor en el que queremos hallar la solución, con lo que obtendremos una aproximación en el intervalo (x_0, b). El intervalo se dividirá en n intevalos más pequeños, cada uno de ellos de longitud h.

Ejemplo 2 (Método de Euler 1/2).

```
from numpy import *
from matplotlib.pyplot import *

x0=1.0
y0=0.0

def F(x,y):
    return x+y

b = 2
n = 10
h = (b - x0)/n
```

Usaremos dos array `X` e `Y` para guardar los valores (x_i, y_i) que aproximan puntos de la gráfica de $y(x)$. El array `X` contiene los puntos $x_0, x_0 + h, x_0 + 2h, \ldots, x_0 + h \cdot n = b$, y lo definimos mediante `arange` que es similar a `range`, pero permite usar números decimales.

Para hallar `Y[1]`, partimos de los valores `X[0]` e `Y[0]` y usamos la primera recta tangente $r_0(x) = y_0 + F(x_0, y_0) \cdot (x - x_0)$. Al evaluar la recta en x_1, tenemos $y_1 = r_0(x_1) = y_0 + F(x_0, y_0) \cdot (x_1 - x_0)$ que en Python se escribe `Y[1]=Y[0] + F(X[0], Y[0]) * (X[1] - X[0])` .

Para hallar la aproximación $y(x_i)$, usamos la tangente que pasa por el punto (x_{i-1}, y_{i-1}), y que podemos escribir como

$$r_{i-1}(x) = y_{i-1} + F(x_{i-1}, y_{i-1}) \cdot (x - x_{i-1}),$$

al sustituir x_i en esta recta tangente obtenemos la aproximación $y(x_i) \simeq r_{i-1}(x_i)$. En Python escribiremos

```
Y[i]=Y[i-1] + F(X[i-1], Y[i-1]) * (X[i] - X[i-1])
```

dentro de un bucle que itera en i, con el que obtenemos todas las rectas tangentes y las aproximaciones a la solución en los nodos definidos.

Ejemplo 3 (Método de Euler 2/2).

```
X=arange(x0, b+h, h)
Y=zeros(n+1)
Y[0]=y0
Y[1]=Y[0] + F(X[0], Y[0]) * (X[1] - X[0])

for i in range(2,n+1):
    Y[i]=Y[i-1] + F(X[i-1], Y[i-1]) * (X[i] - X[i-1])
```

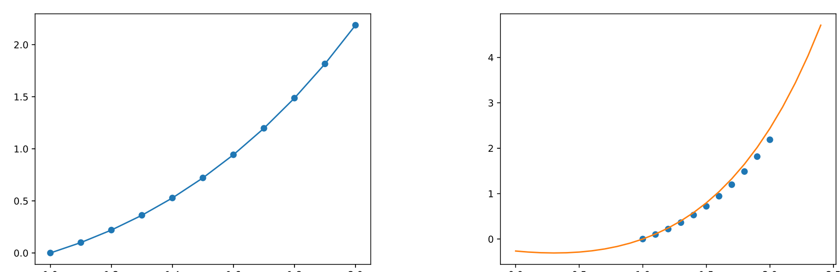

Figura 4: Poligonal de Euler. Comparación con la solución exacta.

En el archivo `Euler.py`, tras estas dos partes, se incluyen unas líneas que no es necesario estudiar, en las que se comparan los resultados con la solución exacta y se representan las tangentes, formando la poligonal de Euler.

Ejemplo 4 (`www.ugr.es/local/anpalom`).
`Euler.py`

Ejercicio 5.

Se quiere emplear el método de Euler con $h = 0.05$ para hallar una solución aproximada de $y(2)$ en la misma ecuación diferencial del ejemplo anterior ($y'(x) = F(x, y) = x + y$, $y(1) = 0$).

1. *Halle el número n de partes en las que se debe dividir el intervalo $[1, 2]$ y obtenga dicha solución aproximada.*
2. *Use el mismo paso $h = 0.05$ para hallar una aproximación a $y(3)$.*

Ejercicio 6.

Se considera el problema de valor inicial $y'(x) = 3y^2 - x^2$, $y'(0) = 0.15$.

1. *Halle una aproximación de $y(3)$ usando $h = 0.1$.*
2. *Halle una aproximación de $y(3)$ usando $h = 0.05$.*
3. *Suponiendo que $y(3)$ vale -1.673381304457, ¿cuál de los dos valores de h permite obtener la mejor aproximación de $y(3)$?*

Ejercicio 7. *Dentro del bucle del método de Euler, ¿se puede sustituir* `(X[i] - X[i-1])` *por* `h`*? Justifique la respuesta.*

Ejercicio 8. *Halle una aproximación de la solución de*

$$\begin{cases} y'(x) = 2y - \frac{3}{10}y^2, \\ y(0) = \frac{1}{10}. \end{cases}$$

en $x = 4$ usando el método de Euler en el intervalo $(0, 4)$ para $n = 25$.

Sabiendo que la solución exacta es $y(x) = \dfrac{20e^{2x}}{197 + 3e^{2x}}$ ¿Qué error absoluto se está cometiendo? (Valor exacto menos valor aproximado).

(Sol. $y(4) \simeq 6.5164$. Error$=0.00653$.)

4. Método de Runge-Kutta de cuarto orden.

En esta sección describimos el método de Runge-Kutta de cuarto orden. No veremos la deducción del método, ni el orden del error.

Consideramos un problema de valor inicial como el de antes

$$\begin{cases} y'(x) = F(x, y), & x \in (x_0, b), \\ y(x_0) = y_0. \end{cases}$$

Para este problema planteamos un método para aproximar el valor de la solución en un punto $x = b$, para lo cual dividimos el intervalo (x_0, b) en n trozos de longitud $h = (b - x_0)/n$. Así los nodos que dividen el intervalo serán $x_0, x_0 + h, x_0 + 2h, \ldots, x_0 + h \cdot n = b$.

Hasta ahora estamos usando la misma notación que en el método de Euler, pero calculamos los valores y_i, aproximaciones de $y(x_i)$, de una manera diferente.

Partiendo de un punto (x_{i-1}, y_{i-1}) conocido, definimos cuatro coeficientes

$$\begin{aligned} k_1 &= h \cdot F(x_{i-1}, y_{i-1}), \\ k_2 &= h \cdot F(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{1}{2}k_1), \\ k_3 &= h \cdot F(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{1}{2}k_2), \\ k_4 &= h \cdot F(x_{i-1} + h, y_{i-1} + k_3). \end{aligned}$$

Con estos coeficientes, hallamos

$$y_i = y_{i-1} + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4).$$

Dividimos el código en dos partes, en la primera inicializamos las variables y definimos la función F.

Ejemplo 9 (Runge-Kutta 1/2).

```
from numpy import *
from matplotlib.pyplot import *

x0=1.0
y0=0.0
```

```
def F(x,y):
    return x+y

b = 2
n = 10
h = (b - x0)/n
```

En la segunda parte calculamos las aproximaciones `Y[i]` usando un bucle.

Ejemplo 10 (Runge-Kutta 2/2).

```
X=arange(x0, b+h, h)
Y=zeros(n+1)
Y[0]=y0

for i in range(1,n+1):
    k1=h *F( X[i-1] , Y[i-1] )
    k2=h *F( X[i-1]+h/2 , Y[i-1]+k1/2 )
    k3=h *F( X[i-1]+h/2 , Y[i-1]+k2/2 )
    k4=h *F( X[i-1]+h , Y[i-1]+k3 )
    Y[i]=Y[i-1]+(k1+2*k2+2*k3+k4)/6
```

En el archivo `RungeKutta.py` se incluyen unas líneas, que no es necesario estudiar, en las que se comparan las aproximaciones calculadas con la solución exacta.

Ejemplo 11 (`www.ugr.es/local/anpalom`).
`RungeKutta.py`

Ejercicio 12. *Halle una aproximación de la solución de*

$$\begin{cases} y'(x) = 2y - \frac{3}{10}y^2, \\ y(0) = \frac{1}{10}. \end{cases}$$

en $x = 4$ *usando el método de Runge-Kutta de cuarto orden en el intervalo* $(0, 4)$ *para* $n = 25$.

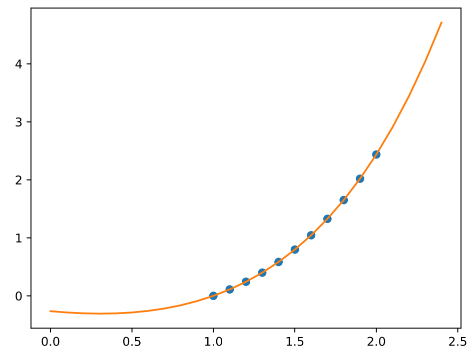

Figura 5: Solución exacta y aproximaciones usando el método de Runge-Kutta

Sabiendo que la solución exacta es $y(x) = \dfrac{20e^{2x}}{197 + 3e^{2x}}$ *¿Qué error absoluto se está cometiendo? (Valor exacto menos valor aproximado).*

(Sol. $y(4) \simeq 6.5229$ *. Error*$=4.95e-05.$*)*

Ejercicio 13. *Emplee el método de Runge-Kutta de cuarto orden con pasos* $h = 0.1$ *y* $h = 0.05$ *para hallar valores aproximados de*

$$\left\{ \begin{array}{l} xy'(x) + 2y(x) = 12x^2, \\ y(1) = -4. \end{array} \right.$$

en $x = 5$*. Compare los resultados obtenidos con el valor* $\dfrac{1868}{25}$.

Ejercicio 14. *Modifique el programa* `RungeKutta.py` *para implementar el método de Runge-Kutta de orden 2 que, partiendo de un punto* (x_{i-1}, y_{i-1}) *conocido, calcula dos coeficientes*

$$\begin{aligned} k_1 &= h \cdot F(x_{i-1}, y_{i-1}), \\ k_2 &= h \cdot F(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{1}{2}k_1), \end{aligned}$$

y con estos coeficientes calcula $y_i = y_{i-1} + k_2$.