ESCUELA POLITÉCNICA NACIONAL	Tarea No.
Metodos Numericos – Computación	**12**
NOMBRE: Ivonne Carolina Ayala	

[Tarea 12] Ejercicios Unidad 05-A | ODE Método de Euler

Solo resolver ejercicios 3, 4 y 5.

3. Utilice el método de Euler para aproximar las soluciones para cada uno de los siguientes problemas de valor inicial.

4. Aquí se dan las soluciones reales para los problemas de valor inicial en el ejercicio 3. Calcule el error real en las aproximaciones del ejercicio 3.

```python
import numpy as np

def euler_method(f, t0, y0, t_fin, h):
    """
    f: función que define la EDO, es decir f(t,y) = dy/dt
    t0: valor inicial de t
    y0: valor inicial de y
    t_fin: valor final de t hasta donde iterar
    h: paso de integración

    Retorna:
    t_vals: array con los valores de t
    y_vals: array con la aproximación de y en cada t
    """
    # Número de pasos
    N = int(np.round((t_fin - t0)/h))

    # Arreglos para almacenar soluciones
    t_vals = np.zeros(N+1)
    y_vals = np.zeros(N+1)

    # Condiciones iniciales
    t_vals[0] = t0
```

```
    y_vals[0] = y0

    # Iteración de Euler
    for n in range(N):
        t_n = t_vals[n]
        y_n = y_vals[n]

        # y_{n+1} = y_n + h * f(t_n, y_n)
        y_vals[n+1] = y_n + h*f(t_n, y_n)
        t_vals[n+1] = t_n + h

    return t_vals, y_vals
```

Ejercicio 3

a. $y' = \frac{y}{t} - \left(\frac{y}{t}\right)^2$, $1 \leq t \leq 2$, $y(1) = 1$, con $h = 0.1$

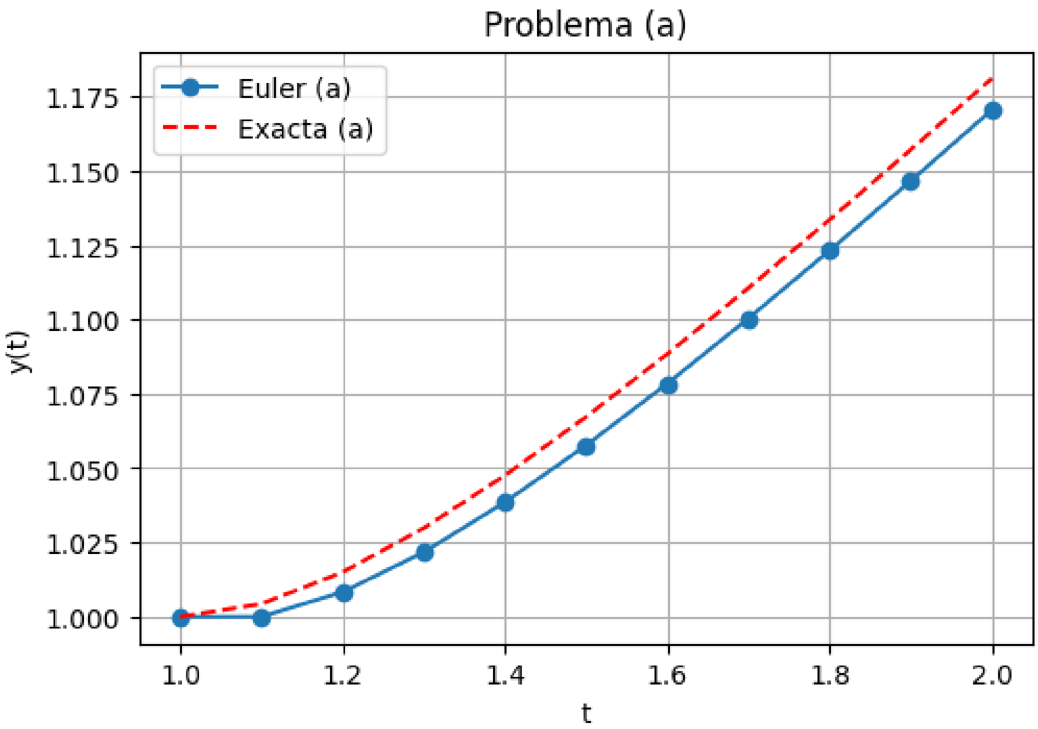
Ejercicio 4

a. $y(t) = \frac{t}{1+\ln t}$

```
def f_a(t, y):
    return (y/t) - (y/t)**2

def y_exacta_a(t):
    return t / (1 + np.log(t))

t0_a = 1.0
y0_a = 1.0
t_fin_a = 2.0
h_a = 0.1

t_a, y_a = euler_method(f_a, t0_a, y0_a, t_fin_a, h_a)

y_a_exact = y_exacta_a(t_a)
error_a = np.abs(y_a - y_a_exact)

print(" t      Euler       Exacta       Error")
for i in range(len(t_a)):
    print(f"{t_a[i]:.2f}   {y_a[i]:.6f}   {y_a_exact[i]:.6f}   {error_a[i]:.6e}")
```

```
 t      Euler      Exacta      Error
1.00   1.000000   1.000000   0.000000e+00
1.10   1.000000   1.004282   4.281728e-03
1.20   1.008264   1.014952   6.687851e-03
1.30   1.021689   1.029814   8.124217e-03
1.40   1.038515   1.047534   9.019185e-03
1.50   1.057668   1.067262   9.594162e-03
1.60   1.078461   1.088433   9.971593e-03
1.70   1.100432   1.110655   1.022289e-02
1.80   1.123262   1.133654   1.039151e-02
1.90   1.146724   1.157228   1.050484e-02
2.00   1.170652   1.181232   1.058065e-02
```

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.plot(t_a, y_a, 'o-', label='Euler (a)')
plt.plot(t_a, y_a_exact, 'r--', label='Exacta (a)')
plt.title("Problema (a)")
plt.xlabel("t")
plt.ylabel("y(t)")
plt.grid(True)
plt.legend()
plt.show()
```



Ejercicio 3

b. $y' = 1 + \frac{y}{t} + \left(\frac{y}{t}\right)^2$, $1 \leq t \leq 3$, $y(1) = 0$, con $h = 0.2$

Ejercicio 4

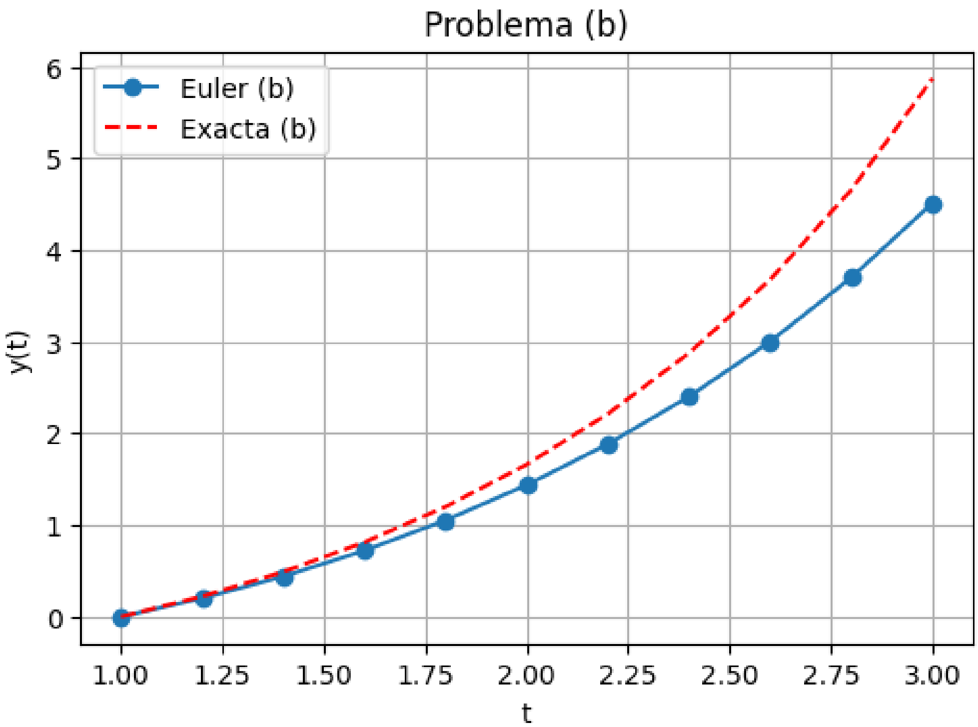
b. $y(t) = t \tan(\ln t)$

```
def f_b(t, y):
    return 1 + (y/t) + (y/t)**2

def y_exacta_b(t):
    return t * np.tan(np.log(t))

t0_b = 1.0
y0_b = 0.0
t_fin_b = 3.0
h_b = 0.2

t_b, y_b = euler_method(f_b, t0_b, y0_b, t_fin_b, h_b)

y_b_exact = y_exacta_b(t_b)
```

```python
error_b = np.abs(y_b - y_b_exact)

print(" t      Euler       Exacta       Error")
for i in range(len(t_b)):
    print(f"{t_b[i]:.2f}   {y_b[i]:.6f}   {y_b_exact[i]:.6f}   {error_b[i]:.6e}")
```

```
 t      Euler       Exacta       Error
1.00   0.000000   0.000000   0.000000e+00
1.20   0.200000   0.221243   2.124277e-02
1.40   0.438889   0.489682   5.079277e-02
1.60   0.721243   0.812753   9.150998e-02
1.80   1.052038   1.199439   1.474006e-01
2.00   1.437251   1.661282   2.240306e-01
2.20   1.884261   2.213502   3.292410e-01
2.40   2.402270   2.876551   4.742818e-01
2.60   3.002837   3.678475   6.756382e-01
2.80   3.700601   4.658665   9.580644e-01
3.00   4.514277   5.874100   1.359823e+00
```

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.plot(t_b, y_b, 'o-', label='Euler (b)')
plt.plot(t_b, y_b_exact, 'r--', label='Exacta (b)')
plt.title("Problema (b)")
plt.xlabel("t")
plt.ylabel("y(t)")
plt.grid(True)
plt.legend()
plt.show()
```



Ejercicio 3

c. $y' = -(y+1)(y+3)$, $0 \leq t \leq 2$, $y(0) = -2$, con $h = 0.2$

Ejercicio 4

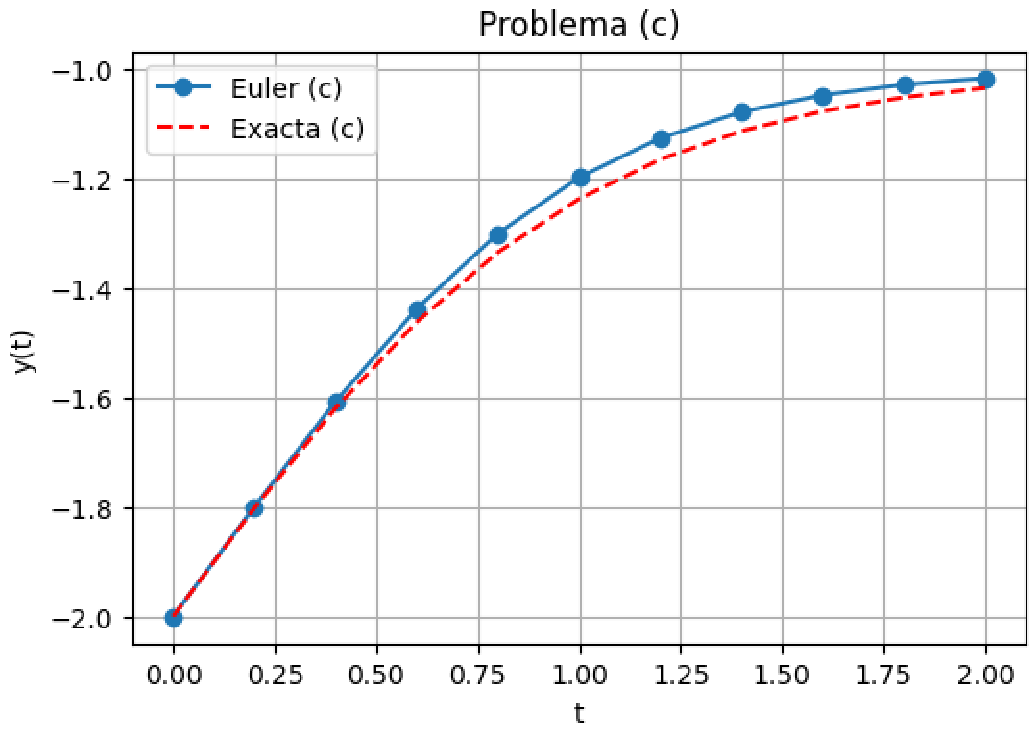
c. $y(t) = -3 + \frac{2}{1+e^{-2t}}$

```
def f_c(t, y):
    return -(y+1)*(y+3)

def y_exacta_c(t):
    return -3 + 2/(1 + np.exp(-2*t))

t0_c = 0.0
y0_c = -2.0
t_fin_c = 2.0
h_c = 0.2

t_c, y_c = euler_method(f_c, t0_c, y0_c, t_fin_c, h_c)

y_c_exact = y_exacta_c(t_c)
```

```
error_c = np.abs(y_c - y_c_exact)

print(" t      Euler       Exacta       Error")
for i in range(len(t_c)):
    print(f"{t_c[i]:.2f}   {y_c[i]:.6f}   {y_c_exact[i]:.6f}   {error_c[i]:.6e}")
```

```
 t      Euler       Exacta       Error
0.00   -2.000000   -2.000000   0.000000e+00
0.20   -1.800000   -1.802625   2.624680e-03
0.40   -1.608000   -1.620051   1.205104e-02
0.60   -1.438733   -1.462950   2.421763e-02
0.80   -1.301737   -1.335963   3.422626e-02
1.00   -1.199251   -1.238406   3.915462e-02
1.20   -1.127491   -1.166345   3.885445e-02
1.40   -1.079745   -1.114648   3.490300e-02
1.60   -1.049119   -1.078331   2.921237e-02
1.80   -1.029954   -1.053194   2.324000e-02
2.00   -1.018152   -1.035972   1.782058e-02
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.plot(t_c, y_c, 'o-', label='Euler (c)')
plt.plot(t_c, y_c_exact, 'r--', label='Exacta (c)')
plt.title("Problema (c)")
plt.xlabel("t")
plt.ylabel("y(t)")
plt.grid(True)
plt.legend()
plt.show()
```



Ejercicio 3

d. $y' = -5y + 5t^2 + 2t$, $0 \leq t \leq 1$, $y(0) = \frac{1}{3}$, con $h = 0.1$

Ejercicio 4

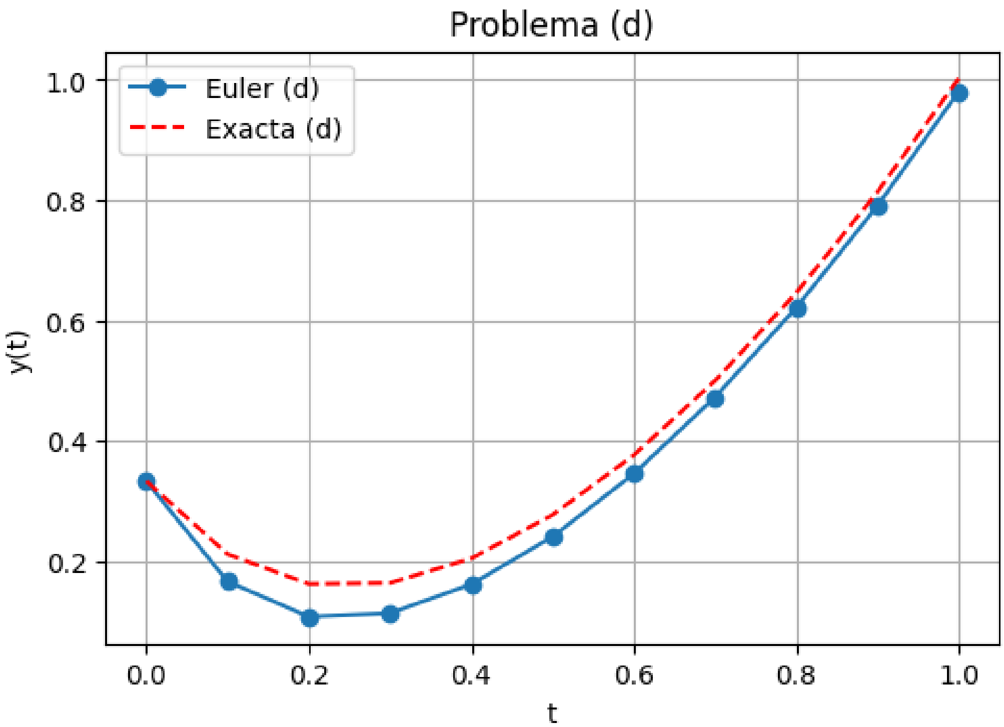
d. $y(t) = t^2 + \frac{1}{3}e^{-5t}$

```
def f_d(t, y):
    return -5*y + 5*(t**2) + 2*t

def y_exacta_d(t):
    return t**2 + (1.0/3.0)*np.exp(-5*t)

t0_d = 0.0
y0_d = 1.0/3.0
t_fin_d = 1.0
h_d = 0.1

t_d, y_d = euler_method(f_d, t0_d, y0_d, t_fin_d, h_d)

y_d_exact = y_exacta_d(t_d)
```

```python
error_d = np.abs(y_d - y_d_exact)

print(" t      Euler       Exacta       Error")
for i in range(len(t_d)):
    print(f"{t_d[i]:.2f}   {y_d[i]:.6f}   {y_d_exact[i]:.6f}   {error_d[i]:.6e}")
```

```
 t      Euler       Exacta       Error
0.00   0.333333   0.333333   0.000000e+00
0.10   0.166667   0.212177   4.551022e-02
0.20   0.108333   0.162626   5.429315e-02
0.30   0.114167   0.164377   5.021005e-02
0.40   0.162083   0.205112   4.302843e-02
0.50   0.241042   0.277362   3.632000e-02
0.60   0.345521   0.376596   3.107486e-02
0.70   0.472760   0.500066   2.730538e-02
0.80   0.621380   0.646105   2.472500e-02
0.90   0.790690   0.813703   2.301289e-02
1.00   0.980345   1.002246   2.190093e-02
```

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.plot(t_d, y_d, 'o-', label='Euler (d)')
plt.plot(t_d, y_d_exact, 'r--', label='Exacta (d)')
plt.title("Problema (d)")
plt.xlabel("t")
plt.ylabel("y(t)")
plt.grid(True)
plt.legend()
plt.show()
```



5. Utilice los resultados del ejercicio 3 y la interpolación lineal para aproximar los siguientes valores de $y(t)$. Compare las aproximaciones asignadas para los valores reales obtenidos mediante las funciones determinadas en el ejercicio 4.

```
def interp_lineal(t_vals, y_vals, t_star):
    for i in range(len(t_vals) - 1):
        if t_vals[i] <= t_star <= t_vals[i+1]:

            t_i, t_ip1 = t_vals[i], t_vals[i+1]
            y_i, y_ip1 = y_vals[i], y_vals[i+1]

            return y_i + (y_ip1 - y_i)*(t_star - t_i)/(t_ip1 - t_i)

    return None
```

a. $y(0.25)$ y $y(0.93)$

```python
t_a_req = [0.25, 0.93]
for t_star in t_a_req:
    y_star_interp = interp_lineal(t_c, y_c, t_star)
    y_star_exact = y_exacta_c(t_star)
    print(f"t = {t_star:.2f}, Interp = {y_star_interp:.6f},
          Exacta = {y_star_exact:.6f},
          Error = {abs(y_star_interp - y_star_exact):.6e}")
```

```
t = 0.25, Interp = -1.752000, Exacta = -1.755081, Error = 3.081338e-03
t = 0.93, Interp = -1.235121, Exacta = -1.269406, Error = 3.428487e-02
```

b. $y(1.25)$ y $y(1.93)$

```python
t_b_req = [1.25, 1.93]
for t_star in t_b_req:
    y_star_interp = interp_lineal(t_a, y_a, t_star)
    y_star_exact = y_exacta_a(t_star)
    print(f"t = {t_star:.2f}, Interp = {y_star_interp:.6f},
          Exacta = {y_star_exact:.6f},
          Error = {abs(y_star_interp - y_star_exact):.6e}")
```

```
t = 1.25, Interp = 1.014977, Exacta = 1.021957, Error = 6.979939e-03
t = 1.93, Interp = 1.153902, Exacta = 1.164390, Error = 1.048818e-02
```

c. $y(2.10)$ y $y(2.75)$

```python
t_c_req = [2.10, 2.75]
for t_star in t_c_req:
    y_star_interp = interp_lineal(t_b, y_b, t_star)
    y_star_exact = y_exacta_b(t_star)
    print(f"t = {t_star:.2f}, Interp = {y_star_interp:.6f},
          Exacta = {y_star_exact:.6f},
          Error = {abs(y_star_interp - y_star_exact):.6e}")
```

```
t = 2.10, Interp = 1.660756, Exacta = 1.924962, Error = 2.642057e-01
t = 2.75, Interp = 3.526160, Exacta = 4.394170, Error = 8.680099e-01
```

d. $y(0.54)$ y $y(0.94)$

```
t_d_req = [0.54, 0.94]
for t_star in t_d_req:
    y_star_interp = interp_lineal(t_d, y_d, t_star)
    y_star_exact = y_exacta_d(t_star)
    print(f"t = {t_star:.2f}, Interp = {y_star_interp:.6f},
          Exacta = {y_star_exact:.6f},
          Error = {abs(y_star_interp - y_star_exact):.6e}")
```

```
t = 0.54, Interp = 0.282833, Exacta = 0.314002, Error = 3.116850e-02
t = 0.94, Interp = 0.866552, Exacta = 0.886632, Error = 2.007968e-02
```