# TP 9 de physique

## Questions préliminaires

2. — Script :

```
def subdivision1(tmax:float, n:int) -> list:
  Dt = tmax/n  # Pas de temps
  return [k*Dt for k in range(n+1)]
print(subdivision1(1., 5))
```

Renvoie : `[0.0, 0.2, 0.4, 0.6, 0.8, 1.0]`

— Script :

```
def subdivision2(tmax:float, Dt:float) ->  list:
  n = int(tmax/Dt)
  return [k*Dt for k in range(n+1)]
print(subdivision2(0.3, 0.07))
```

Renvoie : `[0.0, 0.07, 0.14, 0.21000000000000002, 0.28]`
(Ignorez l'erreur de python)

3.
```
def resolution(e, U:float, tau:float, tmax:float, Dt:float) -> (list, list):
  t = subdivision2(tmax, Dt)
  u = [U]
  uk = U
  r = Dt/tau
  for k in range(len(t)-1):
    uk = uk + r * (e(t[k])-uk)
    u.append(uk)
  return (t, u)

t, u = resolution(lambda t:0., 3., 1., 5., 0.1)
print(t, u)
```

Renvoie : `([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8,`

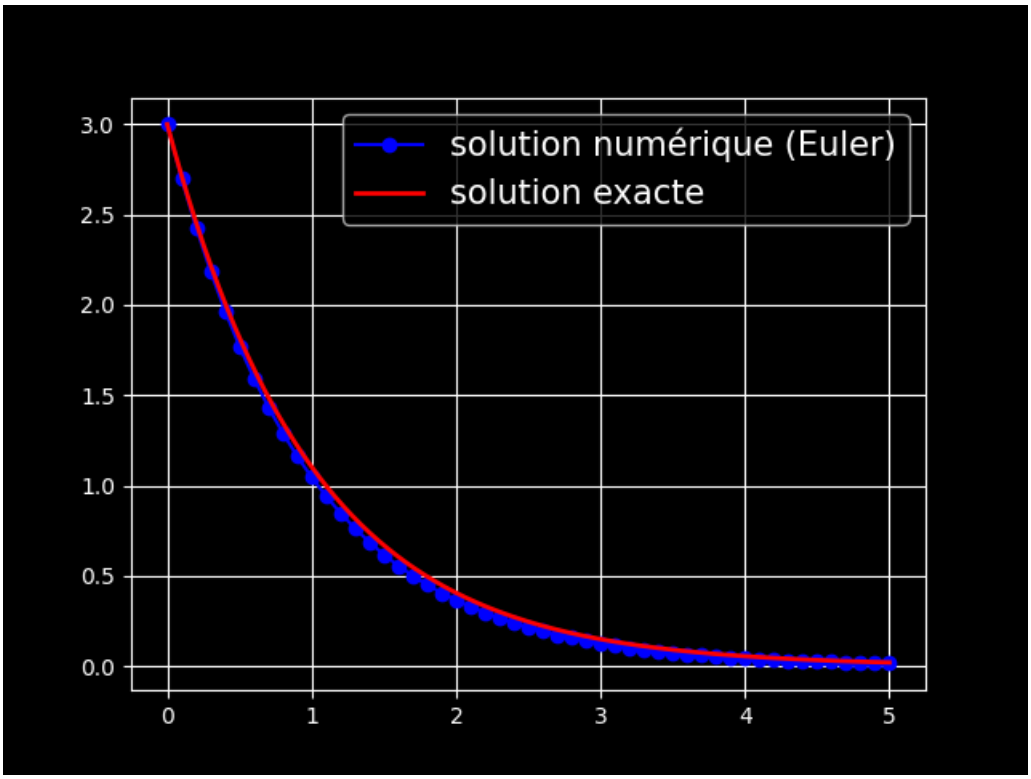
```
4.9, 5.0], [3.0, 2.7, 2.43, 2.187, 1.9683, 1.77147, 1.594323,
1.4348907, 1.29140163, 1.162261467, 1.046035 3203, 0.94143178827,
0.847288609443, 0.7625597484987, 0.68630377364883, 0.617673396283947,
0.5559060566555524, 0.5003154509899972, 0.4502839058909975, 0.40525551530189774,
0.36472996377170797, 0.3282569673945372, 0.29543127065508346,
0.26588814358957513, 0.2392993292306176, 0.21536939630755586,
0.19383245667680027, 0.17444921100912025, 0.15700428990820822,
0.1413038609173874, 0.12717347482564864, 0.11445612734308377,
0.10301051460877539, 0.09270946314789785, 0.08343851683310807,
0.07509466514979726, 0.06758519863481753, 0.060826678771335775,
0.054744010894202194, 0.049269609804781976, 0.044342648824303776,
0.0399083839418734, 0.03591754554768606, 0.03232579099291745,
0.029093211893625705, 0.026183890704263135, 0.023565501633836822,
0.02120895147045314, 0.019088056323407824, 0.01717925069106704,
0.015461325621960335])
```

(Les erreurs de python ont été corrigées)

4.
```python
def e_nulle(t):
  return 0

def u_exacte(t):
  return 3.*np.exp(-t/1.)

t_dense = np.linspace(0., 5., 10000)
plt.style.use('dark_background')
plt.clf()
plt.grid()
plt.plot(t, u, 'bo-', label="solution numérique (Euler)")
plt.plot(t_dense, u_exacte(t_dense), 'r', lw=2, label="solution exacte")
plt.legend(fontsize=15)
plt.savefig("p_tp_9")
```

Renvoie : 


Voir le PDF

---