Corrigé DM3 - Résolution numérique d'une équation différentielle avec conditions initiales

1. **a.** On pose $v(t) = u'(t)$. Comme u est deux fois dérivable sur $]0,+\infty[$, v est dérivable sur $]0,+\infty[$ et $v'(t) = u''(t)$. On a alors $\forall t > 0$, $\begin{cases} t^2u''(t) + tu'(t) - u(t) = 2t \\ u(1) = 1 \quad \text{et} \quad u'(1) = 1 \end{cases} \Leftrightarrow \begin{cases} t^2v'(t) + tv(t) - u(t) = 2t \\ u(1) = 1 \quad \text{et} \quad v(1) = 1 \end{cases}$

On a bien $\boxed{(E) \Leftrightarrow (E') : \begin{cases} v(t) = u'(t) \\ v'(t) = -\frac{1}{t}v(t) + \frac{1}{t^2}u(t) + \frac{2}{t} \\ u(1) = 1 \quad \text{et} \quad v(1) = 1 \end{cases}}$.

b. À l'aide des relations données,

```python
def tuv(a,b,n):
    h = (b-a)/n
    Lt = [a+k*h for k in range(n+1)]
    u, v = 1, 1
    Lu, Lv = [u], [v]
    for t in Lt[:-1]:
        u, v = u+h*v, v+h*(-v/t+u/t**2+2/t)
        Lu.append(u)
        Lv.append(v)
    return Lt, Lu, Lv
```

c. On utilise les listes t_i et u_i obtenues avec la fonction précédente et la fonction `plot`.

```python
import matplotlib.pyplot as plt
a, b = 1, 21
n = 100
Lt, Lu, Lv = tuv(a,b,n)
plt.plot(Lt,Lu)
plt.show()
```

2. **a.** Soit u une solution de (E). On pose, pour tout $t > 0$, $u(t) = ty(t)$.

Comme $t > 0$, on peut écrire $y(t) = \dfrac{u(t)}{t}$ et y est deux fois dérivable sur $\mathbb{R}_+^*$.

On a alors $\forall t > 0$, $u(t) = ty(t)$, $u'(t) = y(t) + ty'(t)$ et $u''(t) = 2y'(t) + ty''(t)$.

On en déduit que : u est solution de (E)

$$\Leftrightarrow \begin{cases} t^2u''(t) + tu'(t) - u(t) = 2t \\ u(1) = 1 \quad \text{et} \quad u'(1) = 1 \end{cases} \Leftrightarrow \begin{cases} t^2(2y'(t) + ty''(t)) + t(y(t) + ty'(t)) - ty(t) = 2t \\ y(1) = 1 \quad \text{et} \quad y(1) + y'(1) = 1 \end{cases}.$$

Donc $\boxed{u \text{ solution de } (E) \Leftrightarrow y \text{ solution de } (E'') : \begin{cases} t^3y''(t) + 3t^2y'(t) = 2t \\ y(1) = 1 \quad \text{et} \quad y(1) + y'(1) = 1 \end{cases}}$.

b. Comme $t^3y''(t) + 3t^2y'(t)$ est la dérivée de $t^3y'(t)$, y solution de

$$(E'') \Leftrightarrow \begin{cases} t^3y'(t) = t^2 + A \\ y(1) = 1 \quad \text{et} \quad y'(1) = 0 \end{cases}.$$

on obtient alors $\begin{cases} y'(t) = \frac{1}{t} + \frac{A}{t^3} \\ y(1) = 1 \quad \text{et} \quad y'(1) = 0 \end{cases}$ soit $\begin{cases} y(t) = \ln t - \frac{A}{2t^2} + B \\ y(1) = 1 \quad \text{et} \quad y'(1) = 0 \end{cases}$ car $t > 0$.

Comme $y(1) = 1 \quad \text{et} \quad y'(1) = 0 \Leftrightarrow -\dfrac{A}{2} + B = 1$ et $1 + A = 0$, on obtient $A = -1$ et $B = \dfrac{1}{2}$.

Finalement, $\boxed{\text{la solution de } (E'') \text{ est } y(t) = \ln t + \frac{1}{2t^2} + \frac{1}{2}}$

et $\boxed{\text{la solution de } (E) \text{ est } u(t) = ty(t) = t\ln t + \frac{1}{2t} + \frac{t}{2}}$.

c. Il reste à créer la fonction u et la liste des $u(t)$:

```
import numpy as np
def u(t):
    return t*np.log(t)+1/(2*t)+t/2
a, b = 1, 21
n = 100
Lt, Lu, Lv = tuv(a,b,n)
plt.plot(Lt,Lu,'r',label = "solution approchée")
Ly = [u(t) for t in Lt]
plt.plot(Lt,Ly,'g',label = "solution exacte")
plt.legend(loc = "best")
plt.show()
```