

TD Calcul Numérique

BOUTON Nicolas

October 30, 2020

Information : Les codes se trouvent à la fin du pdf.

Exercice 1

1. Voir la fonction **pointgauche** dans le fichier "exo1.sci".

Calcul :

$$I \approx h \sum_{i=0}^{n-1} f(a + ih)$$

Explication du code :

- Le code est trivial et bien commenté.

Résultat numérique :

Pour l'appel de fonction suivant :

```
res = pointgauche(0, 10, mysquare, 100);
```

où :

- **0** correspond au début de l'intervale
- **10** correspond à la fin de l'intervale
- **mysquare** correspond à la fonction x^2 qui se trouve dans "myfunc.sci"
- **100** correspond au nombre de pas

On obtient : **res = 328.35**

2. Voir la fonction **trapeze** dans le fichier "exo1.sci".

Calcul :

$$I \approx h \left(\frac{1}{2}f(a) + \sum_{i=1}^{n-1} f(a + ih) + \frac{1}{2}f(b) \right)$$

Explication du code :

- Le code est trivial et bien commenté.

Résultat numérique :

Pour l'appel de fonction suivant :

```
res = trapeze(0, 10, mysquare, 100);
```

où :

- **0** correspond au début de l'intervale

- **10** correspond à la fin de l'intervale
- **mysquare** correspond à la fonction x^2 qui se trouve dans "myfunc.sci"
- **100** correspond au nombre de pas

On obtient : **res = 333.35**

3. Voir la fonction **int_simpson** dans le fichier "exo1.sci".

Calcul :

$$I \approx \frac{h}{6} \left[f(a) + 2 \left(\sum_{i=1}^{n-1} f(a + ih) \right) + f(b) + 4 \left(\sum_{i=0}^{n-1} f \left(a + \left(i + \frac{1}{2} \right) h \right) \right) \right]$$

Explication du code :

- La boucle contenant **somme_a_1h** calcul la somme :

$$\sum_{i=1}^{n-1} f(a + ih)$$

- La boucle contenant **somme_a_i2h** calcul la somme :

$$\sum_{i=0}^{n-1} f(a + (i + \frac{1}{2})h)$$

Résultat numérique :

Pour l'appel de fonction suivant :

res = int_simpson(0, 10, mysquare, 100);

où :

- **0** correspond au début de l'intervale
- **10** correspond à la fin de l'intervale
- **mysquare** correspond à la fonction x^2 qui se trouve dans "myfunc.sci"
- **100** correspond au nombre de pas

On obtient : **res = 333.33333**

4. Voir la fonction **sin_pi_x** dans le fichier "exo1.sci".

Sur l'intervale $[0, 1]$ et un nombre de pas de 10 on obtient :

- 0.6313752 pour **pointgauche**
- 0.6313752 pour **trapeze**
- 0.6366219 pour **int_simpson**

On voit que **pointgauche** et **trapeze** ont la même approximation, cela peut s'expliquer par leurs taux de convergences qui est le même.

Maintenant si on prend un nombre de pas de 20 on obtient :

- 0.6353102 pour **pointgauche**
- 0.6353102 pour **trapeze**
- 0.6366119 pour **int_simpson**

Si on compare l'écart des résultat on obtient :

- 0.0040550 pour **pointgauche**
- 0.0040550 pour **trapeze**
- 0.00001 pour **int_simpson**

On voit que l'erreur relative de **pointgauche** et **trapeze** est plus grand que **int_simpson**, ce qui confirme leurs taux de convergence plus bas que **int_simpson**.

Exercice 2

On a le système suivant :

$$\begin{cases} p(-3) = 3 \\ p(-1) = 7 \\ p(3) = 7 \\ p(5) = -3 \end{cases}$$

Utilisons la méthode des **différence divisé** :

Première étape :

$$\begin{array}{cc} x_i & y_i \\ -3 & 3 \\ -1 & 7 \end{array} \left\} \frac{7 - 3}{-1 - (-3)} = 2$$
$$\begin{array}{cc} x_i & y_i \\ -1 & 7 \\ 3 & 7 \end{array} \left\} \frac{7 - 7}{3 - (-1)} = 0$$

$$\left. \begin{array}{cc} x_i & y_i \\ 3 & 7 \\ 5 & -3 \end{array} \right\} \frac{-3-7}{5-3} = -5$$

Deuxième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -3 & 2 \\ 3 & 0 \end{array} \right\} \frac{0-2}{3-(-3)} = -\frac{2}{6}$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & 0 \\ 5 & -5 \end{array} \right\} \frac{-5-0}{5-(-1)} = -\frac{5}{6}$$

Troisième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -3 & -\frac{2}{6} \\ 5 & -\frac{5}{6} \end{array} \right\} \frac{-\frac{5}{6}-(-\frac{2}{6})}{5-(-3)} = \frac{-\frac{3}{6}}{8} = -\frac{3}{48}$$

Maintenant on peut calculer $p(x)$:

$$\begin{aligned}
p(x) &= 3 + 2(x - (-3)) \\
&\quad + \left(-\frac{2}{6}\right)(x - (-3))(x - (-1)) \\
&\quad - \frac{3}{48}(x - (-3))(x - (-1))(x - 3)
\end{aligned}$$

$$\begin{aligned}
p(x) &= 3 + 2x + 6 \\
&\quad - \frac{2(x+3)(x+1)}{6} \\
&\quad - \frac{3(x+3)(x+1)(x-3)}{48}
\end{aligned}$$

$$\begin{aligned}
p(x) &= \frac{144 + 96x + 288}{48} \\
&\quad - \frac{(2x+6)(x+1)}{6} \\
&\quad - \frac{(3x+9)(x+1)(x-3)}{48}
\end{aligned}$$

$$\begin{aligned}
p(x) &= \frac{144 + 96x + 288}{48} \\
&\quad - \frac{2x^2 + 2x + 6x + 6}{6} \\
&\quad - \frac{(3x^2 + 3x + 9x + 9)(x-3)}{48}
\end{aligned}$$

$$\begin{aligned}
p(x) &= \frac{144 + 96x + 288}{48} \\
&\quad - \frac{2x^2 + 8x + 6}{6} \\
&\quad - \frac{3x^3 - 9x^2 + 11x^2 - 33x + 9x - 27}{48}
\end{aligned}$$

$$\begin{aligned}
p(x) &= \frac{144 + 96x + 288}{48} \\
&\quad - \frac{16x^2 + 64x + 48}{48} \\
&\quad - \frac{3x^3 + 2x^2 - 24x - 27}{48}
\end{aligned}$$

$$p(x) = \frac{144 + 96x + 288 - (16x^2 + 64x + 48) - (3x^3 + 2x^2 - 24x - 27)}{48}$$

$$p(x) = \frac{-3x^3 - 18x^2 + 56x + 411}{48}$$

$$p(x) = -\frac{3}{48}x^3 - \frac{6}{16}x^2 + \frac{7}{6}x + \frac{411}{48}$$

Exercice 3

1. Calculons $f(-2)$, $f(-1)$, $f(0)$, $f(1)$, $f(2)$:

Calcul de $f(-2)$:

$$f(-2) = \ln(2 \cos \left(\frac{\pi(-2)}{4} \right)^2 + 1)$$

$$f(-2) = \ln(2(0)^2 + 1)$$

$$f(-2) = \ln(1)$$

$$f(-2) = 0$$

Calcul de $f(-1)$:

$$f(-1) = \ln(2 \cos \left(\frac{\pi(-1)}{4} \right)^2 + 1)$$

$$f(-1) = \ln(2 \left(\frac{\sqrt{2}}{2} \right)^2 + 1)$$

$$f(-1) = \ln(2 \left(\frac{2}{4} \right) + 1)$$

$$f(-1) = \ln(1 + 1)$$

$$f(-1) = \ln(2)$$

$$f(-1) = 0.6931471$$

Calcul de $f(0)$:

$$f(0) = \ln(2 \cos \left(\frac{\pi(0)}{4} \right)^2 + 1)$$

$$f(0) = \ln(1)$$

$$f(0) = 0$$

Calcul de $f(1)$:

$$f(1) = \ln(2 \cos \left(\frac{\pi(1)}{4} \right)^2 + 1)$$

$$f(1) = \ln(2 \left(\frac{\sqrt{2}}{2} \right)^2 + 1)$$

$$f(1) = \ln(2 \left(\frac{2}{4} \right) + 1)$$

$$f(1) = \ln(1 + 1)$$

$$f(1) = \ln(2)$$

$$f(1) = 0.6931471$$

Calcul de $f(2)$:

$$f(2) = \ln(2 \cos \left(\frac{\pi(2)}{4} \right)^2 + 1)$$

$$f(2) = \ln(2(0)^2 + 1)$$

$$f(2) = \ln(1)$$

$$f(2) = 0$$

Récapitulatif :

$$\begin{cases} f(-2) = 0 \\ f(-1) = \ln(2) \\ f(0) = 0 \\ f(1) = \ln(2) \\ f(2) = 0 \end{cases}$$

2. Essayons de trouver un polynôme p de degré inférieur ou égale à 3 tel que :

$$\begin{cases} p(-2) = f(-2) \\ p(-1) = f(-1) \\ p(1) = f(1) \\ p(2) = f(2) \end{cases}$$

Appliquons la méthode des différences divisées :

$$\begin{cases} p(-2) = 0 \\ p(-1) = \ln(2) \\ p(1) = \ln(2) \\ p(2) = 0 \end{cases}$$

Première étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & 0 \\ -1 & \ln(2) \end{array} \right\} \frac{\ln(2) - 0}{-1 - (-2)} = \ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & \ln(2) \\ 1 & \ln(2) \end{array} \right\} \frac{\ln(2) - \ln(2)}{1 - (-1)} = 0$$

$$\left. \begin{array}{cc} x_i & y_i \\ 1 & \ln(2) \\ 2 & 0 \end{array} \right\} \frac{0 - \ln(2)}{2 - 1} = -\ln(2)$$

Deuxième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & \ln(2) \\ 1 & 0 \end{array} \right\} \frac{0 - \ln(2)}{1 - (-2)} = -\frac{\ln(2)}{3}$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & 0 \\ 2 & -\ln(2) \end{array} \right\} \frac{-\ln(2) - 0}{2 - (-1)} = -\frac{\ln(2)}{3}$$

Troisième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & -\frac{\ln(2)}{3} \\ 2 & -\frac{\ln(2)}{3} \end{array} \right\} \frac{-\frac{\ln(2)}{3} - (-\frac{\ln(2)}{3})}{2 - (-2)} = 0$$

Essayons de calculer p :

$$\begin{aligned}
p(x) &= 0 \\
&+ \ln(2)(x - (-2)) \\
&- \frac{\ln(2)}{3}(x - (-2))(x - (-1)) \\
&+ 0(x - (-2))(x - (-1))(x - 1) \\
p(x) &= \ln(2)(x + 2) \\
&- \frac{\ln(2)}{3}(x + 2)(x + 1) \\
p(x) &= \ln(2)x + 2\ln(2) \\
&- \frac{\ln(2)}{3}(x^2 + x + 2x + 2) \\
p(x) &= \ln(2)x + 2\ln(2) \\
&- \frac{\ln(2)x^2 + 3\ln(2)x + 2\ln(2)}{3} \\
p(x) &= \frac{3\ln(2)x + 6\ln(2)}{3} \\
&- \frac{\ln(2)x^2 + 3\ln(2)x + 2\ln(2)}{3} \\
p(x) &= \frac{3\ln(2)x + 6\ln(2) - \ln(2)x^2 - 3\ln(2)x - 2\ln(2)}{3} \\
p(x) &= \frac{-\ln(2)x^2 + 4\ln(2)}{3} \\
p(x) &= -\frac{\ln(2)}{3}x^2 + \frac{4\ln(2)}{3}
\end{aligned}$$

3. Essayons de trouver un polynôme q de degré inférieur ou égale à 4 tel que :

$$\begin{cases} q(-2) = f(-2) \\ q(-1) = f(-1) \\ q(0) = f(0) \\ q(1) = f(1) \\ q(2) = f(2) \end{cases}$$

Appliquons la méthode des différences divisées :

$$\begin{cases} q(-2) = 0 \\ q(-1) = \ln(2) \\ q(0) = 0 \\ q(1) = \ln(2) \\ q(2) = 0 \end{cases}$$

Première étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & 0 \\ -1 & \ln(2) \end{array} \right\} \frac{\ln(2) - 0}{-1 - (-2)} = \ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & \ln(2) \\ 0 & 0 \end{array} \right\} \frac{0 - \ln(2)}{0 - (-1)} = -\ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ 0 & 0 \\ 1 & \ln(2) \end{array} \right\} \frac{\ln(2) - 0}{1 - 0} = \ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ 1 & \ln(2) \\ 2 & 0 \end{array} \right\} \frac{0 - \ln(2)}{2 - 1} = -\ln(2)$$

Deuxième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & \ln(2) \\ 0 & -\ln(2) \end{array} \right\} \frac{-\ln(2) - \ln(2)}{0 - (-2)} = -\ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & -\ln(2) \\ 1 & \ln(2) \end{array} \right\} \frac{\ln(2) - (-\ln(2))}{1 - (-1)} = \ln(2)$$

$$\left. \begin{array}{cc} x_i & y_i \\ 0 & \ln(2) \\ 2 & -\ln(2) \end{array} \right\} \frac{-\ln(2) - \ln(2)}{2 - 0} = -\ln(2)$$

Troisième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & -\ln(2) \\ 1 & \ln(2) \end{array} \right\} \frac{\ln(2) - (-\ln(2))}{1 - (-2)} = \frac{2\ln(2)}{3}$$

$$\left. \begin{array}{cc} x_i & y_i \\ -1 & \ln(2) \\ 2 & -\ln(2) \end{array} \right\} \frac{-\ln(2) - \ln(2)}{2 - (-1)} = -\frac{2\ln(2)}{3}$$

Quatrième étape :

$$\left. \begin{array}{cc} x_i & y_i \\ -2 & \frac{2\ln(2)}{3} \\ 2 & -\frac{2\ln(2)}{3} \end{array} \right\} \frac{-\frac{2\ln(2)}{3} - \frac{2\ln(2)}{3}}{2 - (-2)} = -\frac{\frac{4\ln(2)}{3}}{4} = -\frac{\ln(2)}{3}$$

Essayons de calculer q:

$$\begin{aligned}
q(x) &= 0 \\
&+ \ln(2)(x - (-2)) \\
&- \ln(2)(x - (-2))(x - (-1)) \\
&+ \frac{2\ln(2)}{3}(x - (-2))(x - (-1))(x - 0) \\
&- \frac{\ln(2)}{3}(x - (-2))(x - (-1))(x - 0)(x - 1)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)(x + 2) \\
&- \ln(2)(x + 2)(x + 1) \\
&+ \frac{2\ln(2)}{3}(x + 2)(x + 1)x \\
&- \frac{\ln(2)}{3}(x + 2)(x + 1)x(x - 1)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)(x + 2) \\
&- \ln(2)(x^2 + x + 2x + 2) \\
&+ \frac{2\ln(2)}{3}(x^2 + x + 2x + 2)x \\
&- \frac{\ln(2)}{3}(x^2 + x + 2x + 2)x(x - 1)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)(x + 2) \\
&- \ln(2)(x^2 + 3x + 2) \\
&+ \frac{2\ln(2)}{3}(x^3 + 3x^2 + 2x) \\
&- \frac{\ln(2)}{3}(x^3 + 3x^2 + 2x)(x - 1)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)(x + 2) \\
&- \ln(2)(x^2 + 3x + 2) \\
&+ \frac{2\ln(2)}{3}(x^3 + 3x^2 + 2x) \\
&- \frac{\ln(2)}{3}(x^4 + 3x^3 + 2x^2 - x^3 - 3x^2 - 2x)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)(x + 2) \\
&- \ln(2)(x^2 + 3x + 2) \\
&+ \frac{2\ln(2)}{3}(x^3 + 3x^2 + 2x) \\
&- \frac{\ln(2)}{3}(x^4 + 2x^3 - x^2 - 2x)
\end{aligned}$$

$$\begin{aligned}
q(x) &= \ln(2)x + 2\ln(2) \\
&- \ln(2)x^2 - 3\ln(2)x - 2\ln(2) \\
&+ \frac{2\ln(2)}{3}x^3 + \frac{6\ln(2)}{3}x^2 + \frac{4\ln(2)}{3}x \\
&- \frac{\ln(2)}{3}x^4 - \frac{2\ln(2)}{3}x^3 + \frac{\ln(2)}{3}x^2 + \frac{2\ln(2)}{3}x
\end{aligned}$$

$$\begin{aligned}
q(x) &= -\frac{\ln(2)}{3}x^4 \\
&\quad + \frac{2\ln(2)}{3}x^3 - \frac{2\ln(2)}{3}x^3 \\
&\quad - \ln(2)x^2 + \frac{6\ln(2)}{3}x^2 + \frac{\ln(2)}{3}x^2 \\
&\quad + \ln(2)x - 3\ln(2)x + \frac{4\ln(2)}{3}x + \frac{2\ln(2)}{3}x \\
&\quad + 2\ln(2) - 2\ln(2) \\
q(x) &= -\frac{\ln(2)}{3}x^4 \\
&\quad - \frac{3\ln(2) - 6\ln(2) - \ln(2)}{3}x^2 \\
&\quad + \frac{3\ln(2) - 9\ln(2) + 4\ln(2) + 2\ln(2)}{3}x \\
q(x) &= -\frac{\ln(2)}{3}x^4 + \frac{4\ln(2)}{3}x^2
\end{aligned}$$

4. Différence entre q et p :

$$\begin{aligned}
q - p &= -\frac{\ln(2)}{3}x^4 + \frac{4\ln(2)}{3}x^2 - \left(-\frac{\ln(2)}{3}x^2 + \frac{4\ln(2)}{3}\right) \\
q - p &= -\frac{\ln(2)}{3}x^4 + \frac{4\ln(2)}{3}x^2 + \frac{\ln(2)}{3}x^2 - \frac{4\ln(2)}{3} \\
q - p &= -\frac{\ln(2)}{3}x^4 + \frac{5\ln(2)}{3}x^2 - \frac{4\ln(2)}{3}
\end{aligned}$$

Exercice 4

1. Voir la fonction **polyLag** dans le fichier "exo4.sci".

Explication du code :

- J'ai créé une fonction **differences** qui calcul :

$$\frac{x - x_j}{x_i - x_j}$$

qui est dans la séquence de produit du polynôme de Lagrange associé au point x :

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

- La fonction **polyLag** récupère la taille du tableau et vérifie si il y a au moins 1 argument.
Ensuite initialise le tableau de résultat et calcul les polynôme de Lagrange associé au point x .

Résultat numérique :

Pour le point $x = 5$ et le vecteur $v = [0, 10, 5, 20]$

On obtient : $res = [0, 0, 1, 0]$

Appel de fonction : **res = polyLag(5, v);**

2. Voir la fonction **myinterpol** dans le fichier "exo4.sci".

Explication du code :

- Cette fonction reprends ce que fait **polyLag** mais ajoute à la fin de la boucle la multiplication avec la fonction **func** et fait donc la somme de toute les itération de **i**.

Résultat numérique :

Pour la fonction x^2

Pour $x = 5$

Pour le vecteur $v = [0, 10, 5, 20]$

On obtient : $res = 25$

Appel de fonction : **res = myinterpol(mysquare, 5, v);**

où **mysquare** est la fonction x^2 qui se trouve dans "myfunc.sci".

Exercice 5

1. Euler Explicite

- a. Déterminons y_{i+1} en fonction de y_i :

$$y_{i+1} = y_i + hf(y_i)$$

$$y_{i+1} = y_i + h \frac{1}{2y_i + 1}$$

$$y_{i+1} = y_i + \frac{h}{2y_i + 1}$$

- b. Voir la fonction **EulerExplicite** dans le fichier "exo5.sci".

Explication du code :

- Le code est triviale et bien commenté, il vérifie les conditions sur les entrées et calcul ensuite les y_i avec la méthode d'Euler explicite.

Résultat numérique :

Pour l'appel suivant :

res = EulerExplicite(0.88, 16) où :

- **0.88** correspond à la fin de l'intervale
- **16** le nombre de pas

On obtient :

i	y_i
0	0
1	0.055
2	0.1045495
3	0.150038
4	0.1923432
5	0.2320634
6	0.2696284
7	0.30536
8	0.3395062
9	0.3722635
10	0.4037907
11	0.4342181
12	0.4636545
13	0.4921917
14	0.5199081
15	0.5468713
16	0.5731401

2. Heun

a. Déterminons y_{i+1} en fonction de y_i :

$$y_{i+1} = y_i + \frac{h}{2}(f(y_i + hf(y_i)) + f(y_i))$$

$$y_{i+1} = y_i + \frac{h}{2} \left(\frac{1}{2 \left(y_i + h \frac{1}{2y_i+1} \right) + 1} + \frac{1}{2y_i + 1} \right)$$

$$y_{i+1} = y_i + \frac{h}{2} \left(\frac{1}{2y_i + \frac{2h}{2y_i+1} + 1} + \frac{1}{2y_i + 1} \right)$$

b. Voir la fonction **Heun** dans le fichier "exo5.sci".

Explication du code :

- Le code est triviale et bien commenté, il vérifie les conditions sur les entrées et calcul ensuite les y_i avec la méthode d'Heun.

Résultat numérique :

Pour l'appel suivant :

res = Heun(0.88, 16) où :

- **0.88** correspond à la fin de l'intervalle
- **16** le nombre de pas

On obtient :

i	y_i
0	0
1	0.0522748
2	0.1000097
3	0.144216
4	0.1855772
5	0.2245808
6	0.2615893
7	0.2968807
8	0.330674
9	0.3631453
10	0.3944384
11	0.4246731
12	0.45395
13	0.4823547
14	0.5099608
15	0.5368322
16	0.5630245

3. Euler Implicite

a. Déterminons un polynôme :

$$y_{i+1} = y_i + hf(y_{i+1})$$

$$y_{i+1} = y_i + h \frac{1}{2y_{i+1} + 1}$$

$$y_{i+1} = \frac{y_i(2y_{i+1} + 1) + h}{2y_{i+1} + 1}$$

$$y_{i+1}(2y_{i+1} + 1) = y_i(2y_{i+1} + 1) + h$$

$$2y_{i+1}^2 + y_{i+1} = y_i + 2y_{i+1}y_i + h$$

$$2y_{i+1}^2 + y_{i+1} - 2y_{i+1}y_i = y_i + h$$

$$2y_{i+1}^2 + y_{i+1} - 2y_{i+1}y_i - y_i - h = 0$$

$$2y_{i+1}^2 + (1 - 2y_i)y_{i+1} - y_i - h = 0$$

b. Calculons le discriminant :

$$\begin{aligned}\Delta &= b^2 - 4ac \\ \Delta &= (1 - 2y_i)^2 - [4 * 2 * (-y_i - h)] \\ \Delta &= (1 - 2y_i)^2 + 8y_i + 8h \\ \Delta &= 1 - 4y_i + (2y_i)^2 + 8y_i + 8h \\ \Delta &= 1 + 4y_i + (2y_i)^2 + 8h \\ \Delta &= (2y_i + 1)^2 + 8h\end{aligned}$$

c. Déterminons y_{i+1} en fonction de y_i et h :

$$\begin{aligned}y_{i+1} &= \frac{-b + \sqrt{\Delta}}{2a} \\ y_{i+1} &= \frac{2y_i - 1 + \sqrt{(2y_i + 1)^2 + 8h}}{4}\end{aligned}$$

d. Voir la fonction **EulerImplicite** dans le fichier "exo5.sci".

Explication du code :

- Le code est triviale et bien commenté, il vérifie les conditions sur les entrées et calcul ensuite les y_i avec la méthode d'EulerImplicite avec la relation ci-dessus.

Résultat numérique :

Pour l'appel suivant :

res = EulerImplicite(0.88, 16) où :

- **0.88** correspond à la fin de l'intervale
- **16** le nombre de pas

On obtient :

i	y_i
0	0
1	0.05
2	0.0961308
3	0.1391563
4	0.1796201
5	0.2179249
6	0.2543787
7	0.2892231
8	0.3226516
9	0.3548221
10	0.3858652
11	0.4158906
12	0.4449914
13	0.4732473
14	0.5007273
15	0.5274915
16	0.5535927

4. Affichage

a. Correspondances des colonnes :

- C2 : Euler Explicite, car les résultats obtenus sont inférieurs aux résultats exactes.
- C3 : Heun, car c'est la méthode qui à la convergence la plus forte
- C4 : Euler Implicite, car les résultats obtenus sont supérieurs aux résultats exactes.

b. Voir la fonction **AfficheRes** dans le fichier "exo5.sci".

c. Calculons le taux d'erreurs des 3 méthodes :

Taux d'erreur avec la méthode d'Euler Explicite :

$$\begin{aligned}|e - e'| &= |0,5630146 - 0,5731401| \\ &= 0.0101255 \\ &= 1.01255\%\end{aligned}$$

Taux d'erreur avec la méthode d'Heun :

$$\begin{aligned}|e - e'| &= |0,5630146 - 0,5630245| \\ &= 0.0000099 \\ &= 0.00099\%\end{aligned}$$

Taux d'erreur avec la méthode d'Euler Implicite :

$$\begin{aligned}|e - e'| &= |0,5630146 - 0,5535927| \\ &= 0.0094219 \\ &= 0.94219\%\end{aligned}$$

Nous voyons clairement que les deux méthodes d'Euler sont presque équivalentes en termes de convergences excepté qu'elles convergent dans des sens différents, les valeurs d'Euler Explicite sont de plus en plus grandes que celles attendues et les valeurs d'Euler Implicite sont de plus en plus petites des valeurs attendues. On voit également que le taux d'erreur est quasiment nulle pour la méthode d'Heun.

Exercice 6

Partie I

Dans cette partie on a le système suivant :

$$\begin{cases} -u''(t) + c(t)u(t) = f(t), t \in [0, 1] \\ u(0) = \alpha, u'(0) = \beta \end{cases}$$

Et pour tout $t \geq 0$:

$$U(t) = \begin{bmatrix} u(t) \\ u'(t) \end{bmatrix}$$

1. On a :

$$U'(t) = \begin{bmatrix} u'(t) \\ u''(t) \end{bmatrix}$$

Calculons $A(t)U(t) + B(t)$:

où

$$A(t) = \begin{bmatrix} 0 & 1 \\ c(t) & 0 \end{bmatrix}, B(t) = \begin{bmatrix} 0 \\ -f(t) \end{bmatrix}, t \in I.$$

$$A(t)U(t) = \begin{bmatrix} 0 & 1 \\ c(t) & 0 \end{bmatrix} \begin{bmatrix} u(t) \\ u'(t) \end{bmatrix} = \begin{bmatrix} u'(t) \\ c(t)u(t) \end{bmatrix}$$

$$A(t)U(t) + B(t) = \begin{bmatrix} u'(t) \\ c(t)u(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -f(t) \end{bmatrix} = \begin{bmatrix} u'(t) \\ c(t)u(t) - f(t) \end{bmatrix}$$

or

$$U'(t) = \begin{bmatrix} u'(t) \\ u''(t) \end{bmatrix} = \begin{bmatrix} u'(t) \\ c(t)u(t) - f(t) \end{bmatrix}$$

Donc si u est solution de (E) et de (C) alors :

$$U'(t) = A(t)U(t) + B(t)$$

2. (a) On a l'équation suivante :

$$U'(t) = A(t)U(t) + B(t)$$

Et on a :

$$\begin{cases} U_{i+1} = U_i + hF(t_i, U_i) \\ U_0 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{cases}$$

$$\begin{cases} U_{i+1} = U_i + hA(t_i)U_i + hB(t_i) \\ U_0 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{cases}$$

$$\begin{cases} U_{i+1} = U_i + h \begin{bmatrix} 0 & 1 \\ c(t_i) & 0 \end{bmatrix} U_i + h \begin{bmatrix} 0 \\ -f(t_i) \end{bmatrix} \\ U_0 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{cases}$$

(b) On a :

$$\begin{cases} U_{i+1} = U_i + hA(t_i)U_i + hB(t_i) \\ U_0 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{cases}$$

Posons

$$U_i = \begin{bmatrix} u_i \\ w_i \end{bmatrix}$$

$$\begin{aligned} \begin{bmatrix} u_{i+1} \\ w_{i+1} \end{bmatrix} &= \begin{bmatrix} u_i \\ w_i \end{bmatrix} + h \begin{bmatrix} 0 & 1 \\ c(t_i) & 0 \end{bmatrix} \begin{bmatrix} u_i \\ w_i \end{bmatrix} + h \begin{bmatrix} 0 \\ -f(t_i) \end{bmatrix} \\ &= \begin{bmatrix} u_i \\ w_i \end{bmatrix} + h \begin{bmatrix} w_i \\ c(t_i)u_i \end{bmatrix} + h \begin{bmatrix} 0 \\ -f(t_i) \end{bmatrix} \end{aligned}$$

On a :

$$\begin{cases} u_{i+1} = u_i + hw_i \\ w_{i+1} = w_i + hc(t_i)u_i - hf(t_i) \end{cases}$$

$$\begin{cases} w_i = \frac{-u_{i+1}+u_i}{h} \\ -\frac{u_{i+1}-2u_i+u_{i-1}}{h^2} + c(t_i)u_i = f(t_i) \end{cases}$$

Donc on a :

$$-\frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} + c(t_{i-1})u_{i-1} = f(t_{i-1})$$

(c) Voir la fonction **resouScd** dans "exo6.sci".

Partie II

1. La formule de Taylor à l'ordre 4 donne :

$$\begin{cases} u(x_{i+1}) = u(x_i) + h_i u'(x_i) + \frac{h_i^2}{2} u''(x_i) + \frac{h_i^3}{3!} u^{(3)}(x_i) + \frac{h_i^4}{4!} u^{(4)}(x_i) + O(h_i^4) \\ u(x_{i-1}) = u(x_i) - h_i u'(x_i) + \frac{h_i^2}{2} u''(x_i) - \frac{h_i^3}{3!} u^{(3)}(x_i) + \frac{h_i^4}{4!} u^{(4)}(x_i) + O(h_i^4) \end{cases}$$

$$h_i = x_{i+1} - x_i$$

Comme on suppose que la subdivision est uniforme :

$$u(x_{i+1}) - u(x_{i-1}) = 2u(x_i) + h^2 u''(x_i) + \frac{h^4}{12} u^{(4)}(x_i) + O(h^4)$$

D'où

$$\begin{aligned} u''(x_i) &= \frac{u(x_{i+1}) - u(x_{i-1}) - 2u(x_i)}{h^2} + O(h^2) \\ &\approx \frac{u(x_{i+1}) - u(x_{i-1}) - 2u(x_i)}{h^2} \end{aligned}$$

Cela inspire le problème approché :

$$\begin{cases} -\frac{u(x_{i+1}) - u(x_{i-1}) - 2u(x_i)}{h^2} + c(x_i)u(x_i) = f(x_i), 1 \leq i \leq n-1 \\ u_0 = u_n = 0 \end{cases}$$

$$\begin{cases} -\frac{u_{i+1} - u_{i-1} - 2u_i}{h^2} + c(x_i)u_i = f(x_i), 1 \leq i \leq n-1 \\ u_0 = u_n = 0 \end{cases}$$

Ce problème approché est un système linéaire :

$$A = \begin{bmatrix} \frac{2}{h^2} + c(t_1) & -\frac{1}{h^2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + c(t_2) & -\frac{1}{h^2} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} + c(t_3) & -\frac{1}{h^2} & 0 & 0 & 0 \\ - & - & - & - & - & - & - \\ 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + c(t_{n-3}) & -\frac{1}{h^2} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + c(t_{n-2}) & -\frac{1}{h^2} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + c(t_{n-1}) \end{bmatrix}$$

et

$$X = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ - \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{bmatrix}$$

et

$$B = \begin{bmatrix} f(u_1) \\ f(u_2) \\ f(u_3) \\ - \\ f(u_{n-3}) \\ f(u_{n-2}) \\ f(u_{n-1}) \end{bmatrix}$$

2.

$$\begin{aligned}
h^2 W^T A W &= h^2 W^T \begin{bmatrix} \frac{2w_1}{h^2} + c(u_1)w_1 - \frac{w_2}{h^2} \\ -\frac{w_1}{h^2} + \frac{2w_2}{h^2} + c(u_1)w_2 - \frac{w_3}{h^2} \\ \dots \\ -\frac{w_{n-3}}{h^2} + \frac{2w_{n-2}}{h^2} + c(u_{n-2})w_{n-2} - \frac{w_{n-1}}{h^2} \\ -\frac{w_{n-2}}{h^2} + \frac{2w_{n-1}}{h^2} + c(u_{n-1})w_{n-1} \end{bmatrix} \\
&= h^2 \begin{pmatrix} w_1 \frac{2w_1}{h^2} + w_1 c(u_1)w_1 - w_1 \frac{w_2}{h^2} \\ -w_2 \frac{w_1}{h^2} + w_2 \frac{2w_2}{h^2} + w_2 c(u_1)w_2 - w_2 \frac{w_3}{h^2} \\ \dots \\ -w_{n-2} \frac{w_{n-3}}{h^2} + w_{n-2} \frac{2w_{n-2}}{h^2} + w_{n-2} c(u_{n-2})w_{n-2} - w_{n-2} \frac{w_{n-1}}{h^2} \\ -w_{n-1} \frac{w_{n-2}}{h^2} + w_{n-1} \frac{2w_{n-1}}{h^2} + w_{n-1} c(u_{n-1})w_{n-1} \end{pmatrix} \\
&= \dots \\
&\quad w_1^2 + h^2 c(t_1)w_1^2 - w_1 w_2 \\
&\quad - w_1 w_2 + w_2^2 + h^2 c(t_2)w_2^2 - w_2 w_3 \\
&= \dots \\
&\quad - w_{n-2} w_{n-3} + w_{n-2}^2 + h^2 c(t_{n-2})w_{n-2}^2 - w_{n-2} w_{n-1} \\
&\quad - w_{n-1} w_{n-2} + w_{n-1}^2 + h^2 c(t_{n-1})w_{n-1}^2 \\
&= \sum_{i=1}^{n-1} (2 + h^2 c(ih))w_i^2 - 2 \sum_{i=2}^{n-1} w_{i-1} w_i
\end{aligned}$$

NOTE : $[]$ = *matrice*, $()$ = *equation*

3. $A = A^T$ donc A est symétrique car elle est tridiagonale et les valeurs sur les diagonales $i = j - 1$ et $i = j + 1$ sont constant et égaux.

4. Voir la fonction **partie2** dans "exo6.sci".

5. Pour l'appel suivant :

> **partie2(10, zer, mycube)**

où

- **10** : est le nombre de pas
- **zer** : est la fonction 0 qui est dans "exo6.sci"
- **mycube** : est la fonction cube qui est dans "myfunc.sci"

On obtient :

i	$f(t_i)$
0	0.0102483
1	0.0204867
2	0.030645
3	0.0405333
4	0.0497817
5	0.05778
6	0.0636183
7	0.0660267
8	0.063315
9	0.0533133
10	0.0333117

Normalement $u_0 = 0$ et $u_N = u_{10} = 0$.

Ici on a $u_0 = 0.0102483$ et $u_{10} = 0.0333117$.

Partie III

1. On a :

$$\begin{cases} -u''(t) + c(t)u(t) = f(t) \\ u(0) = 0, u(1) = 0 \end{cases}$$

Trouver $u \in V$ tq :

$$\begin{aligned} \forall v \in V, \quad & \int_a^b u'(x)v'(x)dx + \int_a^b c(x)u(x)v(x)dx \\ & = \int_a^b f(x)v(x)dx \end{aligned}$$

Le problème s'écrit alors :

$$\begin{cases} \text{Trouver } u \in V \text{ tq} \\ \forall v \in V, a(u, v) = l(v) \end{cases}$$

où :

$$\begin{aligned} l : V &\rightarrow R \\ v &\rightarrow \int_a^b f(x)v(x)dx \end{aligned}$$

l est linéaire.

$$a : V \times V \rightarrow R$$

$$(w, v) \rightarrow \int_a^b w'(x)v'(x) + c(x)w(x)v(x)dx$$

2. Approchons le problème :

Considérons un espace $V_h \subset V$ de dimension finie.

Posons $n = \dim(V_h)$

Soit w_1, \dots, w_n une base de V_h

Considérons le problème approché :

$$\Longleftrightarrow \left\{ \begin{array}{l} \text{Trouver } u_h \in V_h \text{ tq :} \\ \forall v_h \in V_h, a(u_h, v_h) = l(v_h) \end{array} \right.$$

$$\Longleftrightarrow \left\{ \begin{array}{l} \text{Trouver } u_h \in V_h \text{ tq :} \\ \forall 1 \leq i \leq n, a(u_h, w_i) = l(w_i) \end{array} \right.$$

Cherchons $u_h = \sum_{k=1}^n \alpha_k w_k$ tq :

$$\Longleftrightarrow \forall 1 \leq i \leq n, a(\sum_{k=1}^n \alpha_k w_k, w_i) = l(w_i)$$

$$\Longleftrightarrow \forall 1 \leq i \leq n, \sum_{k=1}^n a(w_k, w_i) \alpha_k = l(w_i)$$

Ce qui nous donne $AU = F$:

$$\begin{bmatrix} a(w_1, w_1) & \dots & a(w_1, w_n) \\ \dots & \dots & \dots \\ a(w_n, w_1) & \dots & a(w_n, w_n) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \dots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} l(w_1) \\ \dots \\ l(w_n) \end{bmatrix}$$

3. Soit $w_k = I_h^{-1}(e_k)$ où $\{e_1, \dots, e_n\}$ est la base canonique de R^{n-1} .

$$w_k = I_h^{-1}(e_k) \Longleftrightarrow I_h(w_k) = e_k$$

$$\Longleftrightarrow (w_k(t_1), \dots, w_k(t_k), \dots, w_k(t_{n-1}))$$

$$= (0, \dots, 1, \dots, 0)$$

$$\Longleftrightarrow w_k(t_j) = \delta_{k,j} \quad 1 \leq j \leq n-1$$

Mettre la figure.

$$w_k(t) = \begin{cases} \frac{t-t_{k-1}}{t_k-t_{k-1}} & \text{si } t \in [t_{k-1}, t_k] \\ \frac{t-t_{k+1}}{t_k-t_{k+1}} & \text{si } t \in [t_k, t_{k+1}] \\ 0 & \text{si } t \notin [t_{k-1}, t_{k+1}] \end{cases}$$

Si la subdivision est uniforme on a :

$$w_k(t) = \begin{cases} \frac{t-t_{k-1}}{h} & \text{si } t \in [t_{k-1}, t_k] \\ -\frac{t-t_{k+1}}{h} & \text{si } t \in [t_k, t_{k+1}] \\ 0 & \text{sinon} \end{cases}$$

4.

5. Lorsque $|i - j| > 1$, $a(i, j) = 0$. A est une matrice tridiagonale.

6.

7.

Exercice 7

Exercice 8

Annexe

BOUTON NICOLAS

```
1 // Exercice 1
2
3 // Calcul sin(PI * x)
4 function [y] = sin_pi_x(x)
5     y = sin(%pi * x);
6 endfunction
7
8 //*****
9
10 // Calcul l'intégrale de func avec la méthode du calcul des aires
11 // des rectangles à gauche
12 function [res] = pointgauche(a, b, func, N)
13
14     // Initialise à 0 la somme
15     res = 0;
16
17     // Calcul de h
18     h = (b-a)/N;
19
20     // Calcul de la somme
21     for i = 0:(N - 1)
22         res = res + func(a + i * h);
23     end
24
25     // Multiplication final
26     res = res * h;
27 endfunction
28
29 //*****
30
31 // Calcul l'intégrale de func avec la méthode des trapèzes
32 function [res] = trapeze(a, b, func, N)
33
34     // Initialise à 0 la somme
35     res = 0;
36
37     // Calcul de h
38     h = (b - a) / N;
39
40     // Calcul de petite multiplication
41     f_a = (1 / 2) * func(a);
42     f_b = (1 / 2) * func(b);
43
44     // Calcul de la somme
45     for i = 1:(N - 1)
46         res = res + func(a + i * h);
47     end
```

```

48
49 // Ajoute f_a et f_b à res
50 res = res + f_a + f_b;
51
52 // Multiplication final
53 res = res * h;
54 endfunction
55
56 // *****
57
58 // Calcul l'intégrale de func avec la méthode de simpson
59 function [res] = int_simpson(a, b, func, N)
60
61 // Initialise à 0 la somme
62 res = 0;
63
64 // Calcul de h
65 h = (b - a) / N;
66 h_6 = h / 6;
67
68 // Ajoute f(a) au résultat
69 res = res + func(a);
70
71 // Calcul de la somme f(x_k)
72 somme_a_ih = 0;
73
74 for i = 1:N-1
75     somme_a_ih = somme_a_ih + func(a + i * h);
76 end
77
78 res = res + 2 * somme_a_ih;
79
80 // Ajoute f(b) au résultat
81 res = res + func(b);
82
83 // Calcul de la somme f(a + (i + 1/2) * h)
84 somme_a_i2h = 0;
85
86 for i = 0:(N - 1)
87     somme_a_i2h = somme_a_i2h + func(a + (i + 1 / 2) * h);
88 end
89
90 res = res + 4 * somme_a_i2h;
91
92 // Multiplication final
93 res = res * h_6;
94 endfunction

```

```
1 // Exercice 2
2
3 function [y] = polynome(x)
4     y = - (3 / 48) * (x * x * x) - (6 / 16) * (x * x) + (7 / 6) * x + (411 /
5     48);
6 endfunction
```

src/exo2.sci

```
1 // Exercice 3
2
3 // Calcul x au carré
4 function [y] = square(x)
5     y = x * x;
6 endfunction
7
8 //*****
9
10 // Calcul la fonction f
11 function [y] = func(x)
12     y = log(2 * cos(square((%pi * x) / 4)) + 1);
13 endfunction
```

src/exo3.sci

```

1 // Exercice 4
2
3 // Calcul une séquence de produit spécifique
4 function [res] = differences(x, xj, xi)
5     res = (x - xj) / (xi - xj);
6 endfunction
7
8 //*****
9
10 // Trouve les pôlinomes d'interpolation de Lagrange associés aux point xi[i]
11 function [Lag] = polyLag(x, xi)
12
13     // Récupère la taille de xi
14     n = size(xi, 's');
15
16     // Vérifie la taille de xi
17     if n < 1
18         Lag = 0;
19         return
20     end
21
22     // Initialise à zéro le vecteur de résultat
23     Lag = zeros(n);
24
25     for i = 1:n
26         // Initialise Lag[i] à 1 pour ne pas faire une multiplication par zé
27         ro
28             Lag(i) = 1;
29
30             // Calcul de Li(x)
31             for j = 1:n
32                 if i ~= j
33                     Lag(i) = Lag(i) * differences(x, xi(j), xi(i));
34                 end
35             end
36         end
37     endfunction
38
39 //*****
40
41 // Calcul le polynôme d'interpolation de la fonction func aux point du tableau
42 // xi
43 function [p] = myinterpol(func, x, xi)
44
45     // Récupère la taille de xi
46     n = size(xi, 's');
47
48     // Vérifie la taille de xi
49     if n < 1

```



```

49         p = 0;
50         return
51     end
52
53     // Initialisation du résultat
54     p = 0;
55
56
57     // Calcul le polynôme d'interpolation
58     for i = 1:n
59         // Initialisation de la variable temporaire Lag
60         // qui est le polynôme de Lagrange associé au point
61         // correspondant à i
62         Lag = 1;
63
64         for j = 1:n
65             // Si i != j
66             if i ~= j
67                 Lag = Lag * differences(x, xi(j), xi(i));
68             end
69         end
70         p = p + func(xi(i)) * Lag;
71     end
72
73 endfunction

```

```

1 // Exercice 5
2
3 // Calcul la fonction  $f(x) = (1 / (2x + 1))$ 
4 function [res] = func(x)
5     res = 1 / ( 2 * x + 1);
6 endfunction
7
8 // *****
9
10 // Calcul le valeurs yi avec la méthode d'Euler Explicite
11 function [y] = EulerExplicite(T, N)
12     // Vérifie les conditions de N
13     if N < 1
14         y = 0;
15         return
16     end
17
18     // Vérifie les conditions de T
19     if T <= 0
20         y = 0;
21         return
22     end
23
24     // Calcul de h
25     h = T / N;
26
27     // Initialise le tableau
28     y = zeros(N + 1);
29
30     // Calcul des yi de 0 à N-1
31     for i = 1:N
32         y(i + 1) = y(i) + h * func(y(i));
33     end
34 endfunction
35
36 // *****
37
38 // Calcul le valeurs yi avec la méthode de Heun
39 function [y] = Heun(T, N)
40     // Vérifie les conditions de N
41     if N < 1
42         y = 0;
43         return
44     end
45
46     // Vérifie les conditions de T
47     if T <= 0
48         y = 0;
49         return
50     end

```

```

51
52 // Calcul de h
53 h = T / N;
54 h_2 = h / 2;
55
56 // Initialise le tableau à 0
57 y = zeros(N + 1);
58
59 // Calcul des yi de 0 à N-1
60 for i = 1:N
61     y(i + 1) = y(i) + h_2 * (func(y(i) + h * func(y(i))) + func(y(i)));
62 end
63 endfunction
64
65 // *****
66
67 // Calcul le valeurs yi avec la méthode d'Euler Implicite
68 function [y] = EulerImplicite(T, N)
69     // Vérifie les conditions de N
70     if N < 1
71         y = 0;
72         return
73     end
74
75     // Vérifie les conditions de T
76     if T <= 0
77         y = 0;
78         return
79     end
80
81     // Calcul de h
82     h = T / N;
83
84     // Initialise le tableau à 0
85     y = zeros(N + 1);
86
87     // Calcul des yi de 0 à N-1
88     for i = 1:N
89         // Calcul (2yi + 1)^2
90         yi_x2_1 = (2 * y(i) + 1) * (2 * y(i) + 1);
91
92         // Calcul la racine carré qui est dans la fonction
93         sqrt_res = sqrt(yi_x2_1 + 8 * h);
94
95         // Calcul yi+1
96         y(i + 1) = (2 * y(i) - 1 + sqrt_res) / 4;
97     end
98 endfunction
99
100 // *****
101

```

```
102 // Affiche les différents résultat
103 function [] = AfficheRes(T, N)
104
105     C4 = EulerExplicite(T, N);
106     C3 = Heun(T, N);
107     C2 = EulerImplicite(T, N);
108
109     print(%io(2), "Euler Implicite = ");
110     print(%io(2), C2);
111     print(%io(2), "Heun = ");
112     print(%io(2), C3);
113     print(%io(2), "Euler Explicite = ");
114     print(%io(2), C4);
115 endfunction
```

```

1  // Exercice 6
2
3  // Calcul tous les Ui de 0 à N
4  function [U] = resouScd(N, alpha, beta, cofc, foncf)
5
6      // Initialise les tableaux
7      U = zeros(N + 1);
8      A = zeros(N + 1, N + 1);
9      B = zeros(N + 1);
10     C = zeros(N + 1);
11
12     // Calcul h
13     h = 1 / N;
14     h2 = h * h;
15
16     // Initialise u0 et u1
17     U(1) = alpha;
18     U(2) = U(1) + h * beta;
19
20     // Calcul toutes les fonction f
21     for i = 1:N+1
22         ti = i * h;
23         B(i) = foncf(ti);
24     end
25
26     // Calcul toutes les coefficients c
27     for i = 1:N+1
28         ti = i * h;
29         C(i) = cofc(ti);
30     end
31
32     // Initilaise la matrice A
33     // Si i = j, alors a(i, j) = (2 / h2) + C(i);
34     // Sinon si i = j+1 ou i = j-1, alors a(i, j) = -(1 / h2);
35     for i = 1:N+1
36         for j = 1:N+1
37             if (i == j)
38                 A(i, j) = (2 / h2) + C(i);
39             elseif (i == j-1 || i == j+1)
40                 A(i, j) = -(1 / h2);
41             end
42         end
43     end
44
45     // Calcul le résultat final
46     U = inv(A) * B;
47
48 endfunction
49
50 // *****

```

```

51
52 // Fonction 0
53 function [y] = zer(a)
54     y = 0;
55 endfunction
56
57 //*****
58
59 // Calcul tous les Ui de 0 à N
60 function [U] = partie2(N, cofc, foncf)
61
62     // Initialise les tableaux
63     U = zeros(N + 1);
64     A = zeros(N + 1, N + 1);
65     B = zeros(N + 1);
66     C = zeros(N + 1);
67
68     // Calcul h
69     h = 1 / N;
70     h2 = h * h;
71
72     // Calcul toutes les fonction f
73     for i = 1:N+1
74         ti = i * h;
75         B(i) = foncf(ti);
76     end
77
78     // Calcul toutes les coefficients c
79     for i = 1:N+1
80         ti = i * h;
81         C(i) = cofc(ti);
82     end
83
84     // Initilaise la matrice A
85     // Si i = j, alors a(i, j) = (2 / h2) + C(i);
86     // Sinon si i = j+1 ou i = j-1, alors a(i, j) = -(1 / h2);
87     for i = 1:N+1
88         for j = 1:N+1
89             if (i == j)
90                 A(i, j) = (2 / h2) + C(i);
91             elseif (i == j-1 || i == j+1)
92                 A(i, j) = -(1 / h2);
93             end
94         end
95     end
96
97     // Calcul le résultat final
98     U = inv(A) * B;
99
100 endfunction

```

src/exo6.sci