TD Calcul Numérique

BOUTON Nicolas

October 30, 2020

Information : Les codes se trouvent à la fin du pdf (page 28).

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_ih** 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 métode des **différence divisé** :
Premiére étape :

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

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

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

Deuxième étape :

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

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

Troisième étape :

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

Maintenant on peut calculer $p(x)$:

$$
\begin{aligned}
p(x) =& 3 + 2(x-(-3)) \\
&+ (-\frac{2}{6})(x-(-3))(x-(-1)) \\
&- \frac{3}{48}(x-(-3))(x-(-1))(x-3) \\
p(x) =& 3 + 2x + 6 \\
&- \frac{2(x+3)(x+1)}{6} \\
&- \frac{3(x+3)(x+1)(x-3)}{48} \\
p(x) =& \frac{144+96x+288}{48} \\
&- \frac{(2x+6)(x+1)}{6} \\
&- \frac{(3x+9)(x+1)(x-3)}{48} \\
p(x) =& \frac{144+96x+288}{48} \\
&- \frac{2x^2+2x+6x+6}{6} \\
&- \frac{(3x^2+3x+9x+9)(x-3)}{48} \\
p(x) =& \frac{144+96x+288}{48} \\
&- \frac{2x^2+8x+6}{6} \\
&- \frac{3x^3-9x^2+11x^2-33x+9x-27}{48} \\
p(x) =& \frac{144+96x+288}{48} \\
&- \frac{16x^2+64x+48}{48} \\
&- \frac{3x^3+2x^2-24x-27}{48} \\
p(x) =& \frac{144+96x+288-(16x^2+64x+48)-(3x^3+2x^2-24x-27)}{48} \\
p(x) =& \frac{-3x^3-18x^2+56x+411}{48} \\
p(x) =& -\frac{3}{48}x^3 - \frac{6}{16}x^2 + \frac{7}{6}x + \frac{411}{48}
\end{aligned}
$$

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 où 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 méthodes des différences divisé :

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

Premiére étape :

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

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

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

Deuxième étape :

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

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

Troisième étape :

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

<u>Essayons de calculer p :</u>

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

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

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

<u>Appliquons la méthodes des différences divisé :</u>

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

Premiére étape :

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

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

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

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

Deuxième étape :

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

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

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

Troisième étape :

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

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

Quatrième étape :

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

<u>Essayons de calculer q:</u>

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

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

4. Différence entre q et p :

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

Exercice 4

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

   Explication du code :

   - J'ai créer 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 :

$$
\begin{aligned}
y_{i+1} &= y_i + hf(y_i) \\
y_{i+1} &= y_i + h\frac{1}{2y_i + 1} \\
y_{i+1} &= y_i + \frac{h}{2y_i + 1}
\end{aligned}
$$

   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 entrés 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 entrés 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'intervale
- **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 :

$$\begin{aligned}
y_{i+1} &= y_i + hf(y_{i+1}) \\
y_{i+1} &= y_i + h\frac{1}{2y_{i+1}+1} \\
y_{i+1} &= \frac{y_i(2y_{i+1}+1)+h}{2y_{i+1}+1} \\
y_{i+1}(2y_{i+1}+1) &= y_i(2y_{i+1}+1)+h \\
2y_{i+1}^2 + y_{i+1} &= y_i + 2y_{i+1}y_i + h \\
2y_{i+1}^2 + y_{i+1} - 2y_{i+1}y_i &= y_i + h \\
2y_{i+1}^2 + y_{i+1} - 2y_{i+1}y_i - y_i - h &= 0 \\
2y_{i+1}^2 + (1-2y_i)y_{i+1} - y_i - h &= 0
\end{aligned}$$

b. Calculons le descriminant :

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

c. Déterminons y_{i+1} en fontion 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 entrés 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 exepté qu'elles convergent dans des sens différents, les valeurs d'Euler Explicite sont de plus en plus grandes que celles attendus et les valeurs d'Euler Implicite sont de plus en plus petites des valeurs ettendus. On voit également que le taux d'érreur 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 $t >= 0$:

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

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 :

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

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

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 :

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

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 :

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

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

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.

$$
\begin{aligned}
h^2W^TAW =& h^2W^T \begin{bmatrix} \frac{2w_1}{h^2} + c(u_1)w_1 - \frac{w_2}{h^2} \\ -\frac{w_1}{h^2} + \frac{2w_2}{h^2} + c(u_1)w_2 - \frac{w_3}{h^2} \\ ... \\ -\frac{w_{n-3}}{h^2} + \frac{2w_{n-2}}{h^2} + c(u_{n-2})w_{n-2} - \frac{w_{n-1}}{h^2} \\ -\frac{w_{n-2}}{h^2} + \frac{2w_{n-1}}{h^2} + c(u_{n-1})w_{n-1} \end{bmatrix} \\
=& h^2 \begin{pmatrix} w_1\frac{2w_1}{h^2} + w_1c(u_1)w_1 - w_1\frac{w_2}{h^2} \\ -w_2\frac{w_1}{h^2} + w_2\frac{2w_2}{h^2} + w_2c(u_1)w_2 - w_2\frac{w_3}{h^2} \\ ... \\ -w_{n-2}\frac{w_{n-3}}{h^2} + w_{n-2}\frac{2w_{n-2}}{h^2} + w_{n-2}c(u_{n-2})w_{n-2} - w_{n-2}\frac{w_{n-1}}{h^2} \\ -w_{n-1}\frac{w_{n-2}}{h^2} + w_{n-1}\frac{2w_{n-1}}{h^2} + w_{n-1}c(u_{n-1})w_{n-1} \end{pmatrix} \\
=& \begin{matrix} w_1^2 + h^2c(t_1)w_1^2 - w_1w_2 \\ -w_1w_2 + w_2^2 + h^2c(t_2)w_2^2 - w_2w_3 \\ ... \\ -w_{n-2}w_{n-3} + w_{n-2}^2 + h^2c(t_{n-2})w_{n-2}^2 - w_{n-2}w_{n-1} \\ -w_{n-1}w_{n-2} + w_{n-1}^2 + h^2c(t_{n-1})w_{n-1}^2 \end{matrix} \\
=& \sum_{i=1}^{n-1}(2 + h^2c(ih))w_i^2 - 2\sum_{i=2}^{n-1} w_{i-1}w_i
\end{aligned}
$$

NOTE : $[] = matrixe, () = equation$

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 :

$$\forall v \in V, \quad \int_a^b u'(x)v'(x)dx + \int_a^b c(x)u(x)v(x)dx = \int_a^b f(x)v(x)dx$$

Le problème s'écrit alors :

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

où :

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

l est linéaire.

$$\begin{array}{ll} a: & VxV \to R \\ & (w,v) \to \int_a^b w'(x)v'(x) + c(x)w(x)v(x)dx \end{array}$$

2. Approchons le problème :

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

   Posons $n = dim(V_h)$

   Soit $w_1, ..., w_n$ une base de V_h

   Considérons le problème approché :

$$\Longleftrightarrow \begin{cases} Trouver \quad u_h \in V_h \quad tq : \\ \forall v_h \in V_h, a(u_h, v_h) = l(v_h) \end{cases}$$

$$\Longleftrightarrow \begin{cases} Trouver \quad u_h \in V_h \quad tq : \\ \forall 1 \leq i \leq n, a(u_h, w_i) = l(w_i) \end{cases}$$

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

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

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

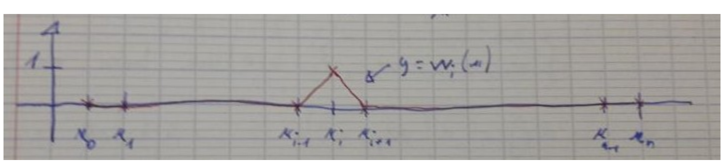
   Ce qui nous donne $AU = F$:

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

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

$$\begin{aligned} w_k = I_h^{-1}(e_k) &\Longleftrightarrow I_h(w_k) = e_k \\ &\Longleftrightarrow \begin{array}{l} \quad (w_k(t_1), ..., w_k(t_k), ..., w_k(t_{n-1})) \\ =(0, ..., 1, ..., 0) \end{array} \\ &\Longleftrightarrow w_k(t_j) = \delta_{k,j} \quad 1 \leq j \leq n-1 \end{aligned}$$

   <u>Voici la courbe de l'une des fonction :</u>



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

Si la subdivision est uniforme on a :

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

w_k est dérivable sur $[a, b]$ donc :

$$w_k'(t) = \begin{cases} \frac{1}{h} & si \quad t \in]t_{k-1}, t_k[\\ -\frac{1}{h} & si \quad t \in]t_k, t_{k+1}[\\ 0 & sinon \end{cases}$$

donc,

$$w_k'(t) = \begin{cases} \frac{1}{h} & si \quad t \in]t_{k-1}, t_k[\\ -\frac{1}{h} & si \quad t \in]t_k, t_{k+1}[\\ 0 & sinon \end{cases}$$

4.

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

6.

7.

Exercice 7

Exercice 8

Annexe

BOUTON NICOLAS

```
// Exercice 1

// Calcul sin(PI * x)
function [y] = sin_pi_x(x)
    y = sin(%pi * x);
endfunction

//*****************************************************

// Calcul l'intégrale de func avec la méthode du calcul des aires
// des rectangles à gauche
function [res] = pointgauche(a, b, func, N)

    // Initialise à 0 la somme
    res = 0;

    // Calcul de h
    h = (b-a)/N;

    // Calcul de la somme
    for i = 0:(N - 1)
        res = res + func(a + i * h);
    end

    // Multiplication final
    res = res * h;
endfunction

//*****************************************************

// Calcul l'intégrale de func avec la méthode des trapèzes
function [res] = trapeze(a, b, func, N)

    // Initialise à 0 la somme
    res = 0;

    // Calcul de h
    h = (b - a) / N;

    // Calcul de petite multiplication
    f_a = (1 / 2) * func(a);
    f_b = (1 / 2) * func(b);

    // Calcul de la somme
    for i = 1:(N - 1)
        res = res + func(a + i * h);
    end
```

```

    // Ajoute f_a et f_b à res
    res = res + f_a + f_b;

    // Multiplication final
    res = res * h;
endfunction

//*****************************************************

// Calcul l'intégrale de func avec la méthode de simpson
function [res] = int_simpson(a, b, func, N)

    // Initialise à 0 la somme
    res = 0;

    // Calcul de h
    h = (b - a) / N;
    h_6 = h / 6;

    // Ajoute f(a) au résultat
    res = res + func(a);

    // Calcul de la somme f(x_k)
    somme_a_ih = 0;

    for i = 1:N-1
        somme_a_ih = somme_a_ih + func(a + i * h);
    end

    res = res + 2 * somme_a_ih;

    // Ajoute f(b) au résultat
    res = res + func(b);

    // Calcul de la some f(a + (i + 1/2) * h)
    somme_a_i2h = 0;

    for i = 0:(N - 1)
        somme_a_i2h = somme_a_i2h + func(a + (i + 1 / 2) * h);
    end

    res = res + 4 * somme_a_i2h;

    // Multiplication final
    res = res * h_6;
endfunction
```

src/exo1.sci

```
// Exercice 2

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

src/exo2.sci

```
// Exercice 3

// Calcul x au carré
function [y] = square(x)
    y = x * x;
endfunction

//**********************************************************

// Calcul la fonction f
function [y] = func(x)
    y = log(2 * cos(square((%pi * x) / 4)) + 1);
endfunction
```

src/exo3.sci

```
// Exercice 4

// Calcul une séquence de produit scpécifique
function [res] = differences(x, xj, xi)
    res = (x - xj) / (xi - xj);
endfunction

//*****************************************************

// Trouve les pôlinomes d'interpolation de Lagrange associés aux point xi[i]
function [Lag] = polyLag(x, xi)

    // Récupère la taille de xi
    n = size(xi, '*');

    // Vérifie la taille de xi
    if n < 1
       Lag = 0;
       return
    end

    // Initialise à zéro le vecteur de résultat
    Lag = zeros(n);

    for i = 1:n
        // Initialise Lag[i] à 1 pour ne pas faire une multiplication par zéro
        Lag(i) = 1;

        // Calcul de Li(x)
        for j = 1:n
            if i ~= j
               Lag(i) = Lag(i) * differences(x, xi(j), xi(i));
        end
        end
    end

endfunction

//*****************************************************

// Calcul le polynôme d'interpolation de la function func aux point du tableau xi
function [p] = myinterpol(func, x, xi)

    // Récupère la taille de xi
    n = size(xi, '*');

    // Vérifie la taille de xi
    if n < 1
```

```
        p = 0;
        return
    end

    // Initialisation du résultat
    p = 0;


    // Calcul le polynôme d'interpolation
    for i = 1:n
        // Initialisation de la variable temporaire Lag
        // qui est le polynôme de Lagrange associé au point
        // correspondant à i
        Lag = 1;

        for j = 1:n
            // Si i != j
        if i ~= j
                Lag = Lag * differences(x, xi(j), xi(i));
        end
        end
            p = p + func(xi(i)) * Lag;
    end

endfunction
```

src/exo4.sci

```
// Exercice 5

// Calcul la fonction f(x) = (1 / (2x + 1))
function [res] = func(x)
      res = 1 / ( 2 * x + 1);
endfunction

//*****************************************************

// Calcul le valeurs yi avec la méthode d'Euler Explicite
function [y] = EulerExplicite(T, N)
      // Vérifie les conditions de N
      if N < 1
         y = 0;
         return
      end

      // Vérifie les conditions de T
      if T <= 0
         y = 0;
         return
      end

      // Calcul de h
      h = T / N;

      // Initialise le tableau
      y = zeros(N + 1);

      // Calcul des yi de 0 à N-1
      for i = 1:N
          y(i + 1) = y(i) + h * func(y(i));
      end
endfunction

//*****************************************************

// Calcul le valeurs yi avec la méthode de Heun
function [y] = Heun(T, N)
      // Vérifie les conditions de N
      if N < 1
         y = 0;
         return
      end

      // Vérifie les conditions de T
      if T <= 0
         y = 0;
         return
      end
```

```

    // Calcul de h
    h = T / N;
    h_2 = h / 2;

    // Initialise le tableau à 0
    y = zeros(N + 1);

    // Calcul des yi de 0 à N-1
    for i = 1:N
        y(i + 1) = y(i) + h_2 * (func(y(i) + h * func(y(i))) + func(y(i)));
    end
endfunction

//*******************************************************

// Calcul le valeurs yi avec la méthode d'Euler Implicite
function [y] = EulerImplicite(T, N)
    // Vérifie les conditions de N
    if N < 1
       y = 0;
       return
    end

    // Vérifie les conditions de T
    if T <= 0
       y = 0;
       return
    end

       // Calcul de h
    h = T / N;

    // Initialise le tableau à 0
    y = zeros(N + 1);

    // Calcul des yi de 0 à N-1
    for i = 1:N
        // Calcul (2yi + 1)^2
        yi_x2_1 = (2 * y(i) + 1) * (2 * y(i) + 1);

        // Calcul la racine carré qui est dans la fonction
        sqrt_res = sqrt(yi_x2_1 + 8 * h);

        // Calcul yi+1
        y(i + 1) = (2 * y(i) - 1 + sqrt_res) / 4;
    end
endfunction

//*******************************************************

```

```
// Affiche les différents résultat
function [] = AfficheRes(T, N)

    C4 = EulerExplicite(T, N);
    C3 = Heun(T, N);
    C2 = EulerImplicite(T, N);

    print(%io(2), "Euler Implicite = ");
    print(%io(2), C2);
    print(%io(2), "Heun = ");
    print(%io(2), C3);
    print(%io(2), "Euler Explicite = ");
    print(%io(2), C4);
endfunction
```

src/exo5.sci

```
// Exercice 6

// Calcul tous les Ui de 0 à N
function [U] = resouScd(N, alpha, beta, cofc, foncf)

    // Initialise les tableaux
    U = zeros(N + 1);
    A = zeros(N + 1, N + 1);
    B = zeros(N + 1);
    C = zeros(N + 1);

    // Calcul h
    h = 1 / N;
    h2 = h * h;

    // Initialise u0 et u1
    U(1) = alpha;
    U(2) = U(1) + h * beta;

    // Calcul toutes les fonction f
    for i = 1:N+1
        ti = i * h;
        B(i) = foncf(ti);
    end

    // Calcul toutes les coefficients c
    for i = 1:N+1
        ti = i * h;
        C(i) = cofc(ti);
    end

    // Initilaise la matrice A
    // Si i = j, alors a(i, j) = (2 / h2) + C(i);
    // Sinon si i = j+1 ou i = j-1, alors a(i, j) = -(1 / h2);
    for i = 1:N+1
        for j = 1:N+1
            if (i == j)
              A(i, j) = (2 / h2) + C(i);
            elseif (i == j-1 || i == j+1)
              A(i, j) = -(1 / h2);
            end
        end
    end

    // Calcul le résultat final
    U = inv(A) * B;

endfunction

//*******************************************************
```

```
// Fonction 0
function [y] = zer(a)
    y = 0;
endfunction

//*****************************************************

// Calcul tous les Ui de 0 à N
function [U] = partie2(N, cofc, foncf)

    // Initialise les tableaux
    U = zeros(N + 1);
    A = zeros(N + 1, N + 1);
    B = zeros(N + 1);
    C = zeros(N + 1);

    // Calcul h
    h = 1 / N;
    h2 = h * h;

    // Calcul toutes les fonction f
    for i = 1:N+1
        ti = i * h;
        B(i) = foncf(ti);
    end

    // Calcul toutes les coefficients c
    for i = 1:N+1
        ti = i * h;
        C(i) = cofc(ti);
    end

    // Initilaise la matrice A
    // Si i = j, alors a(i, j) = (2 / h2) + C(i);
    // Sinon si i = j+1 ou i = j-1, alors a(i, j) = -(1 / h2);
    for i = 1:N+1
        for j = 1:N+1
            if (i == j)
                A(i, j) = (2 / h2) + C(i);
        elseif (i == j-1 || i == j+1)
                A(i, j) = -(1 / h2);
        end
        end
    end

    // Calcul le résultat final
    U = inv(A) * B;

endfunction
```

src/exo6.sci