Mini Projet

Nicolas BOUTON

November 26, 2020

1. Nous avons le système suivant :

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

$$\begin{aligned} &u(t) = ke^{-t} \\ &u(t=0) = 1 = ke^0 = k \\ &u(t) = e^{-t} \end{aligned}$$

2. Appliquons la méthode d'Euler explicite :

$$\begin{aligned} u_{m+1} &= u_m + \Delta t f(u_m) \\ &= u_m + \Delta t(-u_m) \\ &= u_m(1 - \Delta t) \end{aligned}$$

Nous remarquons que c'est une suite géométique de raison $1-\Delta t$. Donc on en déduit que $\forall m \in [0, n], u_m = (1 - \Delta t)^m$

3. Calculons la solution exacte en $t = 1$:

$$u(1) = e^{-1} \approx 0,37$$

Maintenant essayons de voir la convergence de la solution trouvé avec la méthode d'Euler explicite :

On pose $\Delta t = \frac{1}{n}$

$$\begin{aligned} u_n &= (1 - \Delta t)^n \\ &= \left(1 - \frac{1}{n}\right)^n \end{aligned}$$

$$\lim_{\Delta t \to 0} u_n = \lim_{n \to +\infty} u_n = \lim_{n \to +\infty} \left(1 - \frac{1}{n}\right)^n = \lim_{n \to +\infty} 1^n = 0$$

Donc on en conclu que lorsque Δt tend vers 0, alors u_m converge vers 0.

4. Appliquons le développenment limité de taylor :

$$\begin{aligned}
u(t_m) &= u(t_{m+1}) + u'(t_{m+1})(t_m - t_{m+1}) + O(t_m - t_{m+1}) \\
&= u(t_{m+1}) + u'(t_{m+1})(-\Delta t) + O(-\delta t) \\
&= u(t_{m+1}) - u'(t_{m+1})\Delta t \\
u'(t_{m+1}) &= \frac{u(t_{m+1}) - u(t_m)}{\Delta t}
\end{aligned}$$

Et on a $u'(t_{m+1}) = f(u(t_{m+1})) = -u(t_{m+1})$

Donc

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} = u'(t_{m+1}) \approx f(u(t_{m+1}))$$

5. Le schéma d'euler implicite :

$$u(t_{m+1}) = u(t_m) + \Delta t f(u(t_{m+1}))$$

6. D'après la question précédentes on a :

$$\begin{aligned}
u(t_{m+1}) &= u(t_m) + \Delta t f(u(t_{m+1})) \\
u(t_m) &= u(t_{m+1}) - \Delta t f(u(t_{m+1})) \\
u(t_m) &= u(t_{m+1}) + \Delta t u(t_{m+1}) \\
u(t_m) &= u(t_{m+1})(1 + \Delta t) \\
u(t_{m+1}) &= \frac{u(t_m)}{(1 + \Delta t)} \\
u(t_m) &= \frac{u(t_{m-1})}{(1 + \Delta t)} \\
u(t_m) &= \frac{1}{(1 + \Delta t)^m}
\end{aligned}$$

7. Partons d'Euler explicite :

$$u_{m+1} = u_m + \Delta t f(u_m)$$
$$\frac{u_{m+1} - u_m}{\Delta t} = f(u_m)$$

or on a :

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} = f(u(t_{m+1}))$$

Donc

$$\frac{u(t_{m+1}) - u(t_m)}{\Delta t} = \frac{f(u(t_{m+1})) + f(u(t_m))}{2}$$

8. D'après la question précédentes on a :

$$\begin{aligned}
\frac{u(t_{m+1}) - u(t_m)}{\Delta t} &= \frac{f(u(t_{m+1})) + f(u(t_m))}{2} \\
u(t_{m+1}) - u(t_m) &= \frac{\Delta t f(u(t_{m+1})) + \Delta t f(u(t_m))}{2} \\
u(t_{m+1}) - \frac{1}{2}\Delta t f(u(t_{m+1})) &= u(t_m) + \frac{1}{2}\Delta t f(u(t_m))
\end{aligned}$$

9. D'après la question précédentes on a :

$$\begin{aligned}
u(t_{m+1}) - \frac{1}{2}\Delta t f(u(t_{m+1})) &= u(t_m) + \frac{1}{2}\Delta t f(u(t_m)) \\
u(t_{m+1}) + \frac{1}{2}\Delta t u(t_{m+1}) &= u(t_m) - \frac{1}{2}\Delta t u(t_m) \\
u(t_{m+1}) \left(1 + \frac{1}{2}\Delta t\right) &= u(t_m) \left(1 - \frac{1}{2}\Delta t\right) \\
u(t_{m+1}) &= u(t_m) \left(\frac{1 - \frac{1}{2}\Delta t}{1 + \frac{1}{2}\Delta t}\right) \\
u(t_{m+1}) &= u(t_m) \left(\frac{1 - \frac{1}{2}\Delta t}{1 + \frac{1}{2}\Delta t}\right) \\
u(t_{m+1}) &= u(t_m) \left(\frac{\frac{2-\Delta t}{2}}{\frac{2+\Delta t}{2}}\right) \\
u(t_{m+1}) &= u(t_m) \left(\frac{2 - \Delta t}{2 + \Delta t}\right)
\end{aligned}$$

Donc

$$u_{m+1} = u_m \left(\frac{2-\Delta t}{2+\Delta t}\right)$$
$$u_m = u_{m-1} \left(\frac{2-\Delta t}{2+\Delta t}\right)$$
$$u_m = \left(\frac{2-\Delta t}{2+\Delta t}\right)^m$$

10. Calculons la solutionà $t=1$:

$$u(1) = e^{-1} \approx 0,37$$

Supposons que Δt est petit, c'est-à-dire $\Delta t = \frac{1}{n}$

Schéma d'Euler explicite :

$$u_n = \left(1 - \frac{1}{n}\right)^n$$

Schéma d'Euler implicite :

$$u_n = \frac{1}{\left(1+\frac{1}{n}\right)^n}$$

Schéma numérique :

$$u_n = \left(\frac{2-\frac{1}{n}}{2+\frac{1}{n}}\right)^n$$

Ces schéma sont implémente dans **main.f90**.

```fortran

! Calcul Euler Explicite
function euler_explicite(n)

  real :: euler_explicite
  real :: n

  euler_explicite = (1 - (1 / n)) ** n
  return

end function euler_explicite

! Calcul Euler Implicite
function euler_implicite(n)

  real :: euler_implicite
  real :: n

  euler_implicite = 1 / ((1 + (1 / n)) ** n)
  return

end function euler_implicite

! Calcul numerique
function numerique(n)

  real :: numerique
  real :: n

  numerique = ( (2 - (1 / n)) / (2 + (1 / n)) ) ** n
  return

end function numerique

program main

  implicit none
  real :: euler_explicite
  real :: euler_implicite
  real :: numerique
  real :: n1, n2
  real :: ee1, ei1, num1
  real :: ee2, ei2, num2
  real :: dee, dei, dnum

  ! Display choice
  print *, "Choose n1 :"
  read (*, *) n1
  print *, "Choose n2 :"
  read (*, *) n2
```

```

  ! Compute scheme with n2
  ee1 = euler_explicite(n1)
  ei1 = euler_implicite(n1)
  num1 = numerique(n1)

  ! Display result of scheme with n1
  print *, ""
  print *, "Avec n = " , n1
  print *, "euler explicite : ", ee1
  print *, "euler implicite : ", ei1
  print *, "numerique       : ", num1

  ! Compute scheme with n2
  ee2 = euler_explicite(n2)
  ei2 = euler_implicite(n2)
  num2 = numerique(n2)

  ! Display result of scheme with n2
  print *, ""
  print *, "Avec n = " , n2
  print *, "euler explicite : ", ee2
  print *, "euler implicite : ", ei2
  print *, "numerique       : ", num2

  ! Compute delta
  dee = abs(ee1 - ee2)
  dei = abs(ei1 - ei2)
  dnum = abs(num1 - num2)

  ! Display delta
  print *, ""
  print *, "Delta :"
  print *, "delta euler explicite : ", dei
  print *, "delta euler implicite : ", dee
  print *, "delta numerique       : ", dnum

end program main
```

src/main.f90

En exécutant ce code on obtient :

```
 Choose n1 :
10
 Choose n2 :
100

 Avec n =     10.0000000
 euler explicite :    0.348678350
 euler implicite :    0.385543197
 numerique       :    0.367572576
```

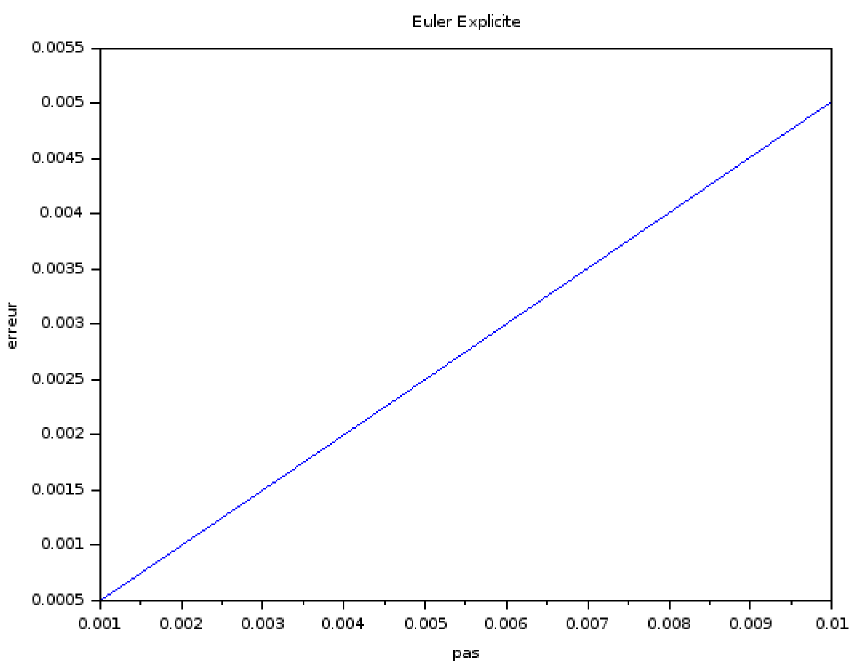
```

Avec n =      100.000000
euler explicite :    0.366032690
euler implicite :    0.369711548
numerique       :    0.367877424

Delta :
delta euler explicite :     1.58316493E-02
delta euler implicite :     1.73543394E-02
delta numerique       :     3.04847956E-04
```

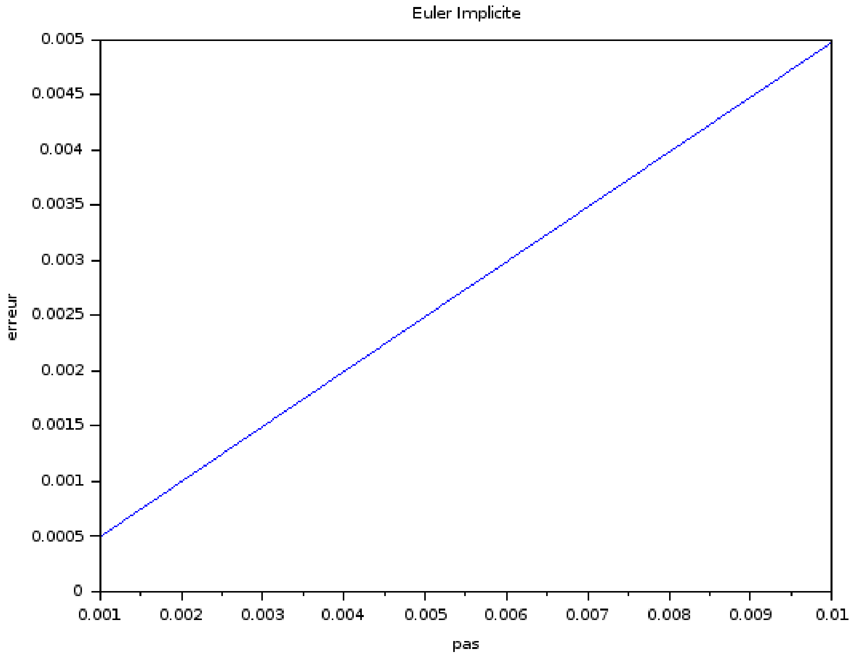
result.txt

Ici on a choisis $n_1 = 10$ et $n_2 = 100$. On voit que pour les schémas d'Euler le delta est de l'ordre de 10^{-2} et que pour le schéma numérique il est de l'ordre de 10^{-4}.

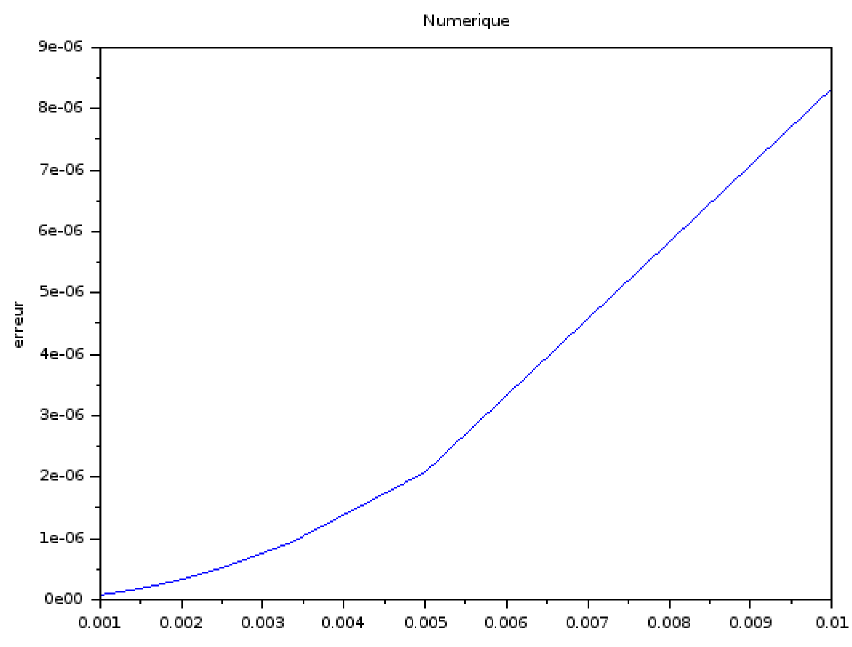
Donc on peut supposer que si les erreurs absolus des schémas d'Euler sont de l'ordre de $k\Delta t$ alors l'erreur absolue du schéma numérique est de l'ordre de $k\Delta t^2$.

Le programme **etude_1.sci** permet de générer les graphes suivants :





On peut voir que les erreurs relatives des schémas d'Euler suivent un ordre **linéaire** donc $k\Delta t$.



On peut voir que l'erreur relative du schéma numérique suit un ordre **quadratique** donc $k\Delta t^2$.

11. Solution exacte :

$$\begin{cases} u'(t) = -u^2(t) \\ u(0) = 1 \end{cases}$$

On trouve :

$$u(t) = -\frac{1}{-t+k}$$

Et on a :

$$u(0) = 1 = -\frac{1}{k}$$

Donc

$$k = -1$$

Et donc

$$u(t) = \frac{1}{t+1}$$

Euler explicite :

$$\begin{aligned} u_{m+1} &= u_m + \Delta t f(u_m) \\ &= u_m + \Delta t(-u_m^2) \\ &= u_m - \Delta t u_m^2 \\ &= u_m(1 - \Delta t u_m) \end{aligned}$$

Donc $u_m = u_{m-1}(1 - \Delta t u_{m-1})$

Euler implicite :

$$\begin{aligned} u_{m+1} &= u_m + \Delta t f(u_{m+1}) \\ &= u_m + \Delta t(-u_{m+1}^2) \\ &= u_m - \Delta t u_{m+1}^2 \\ \Delta t u_{m+1}^2 + u_{m+1} - u_m &= 0 \end{aligned}$$

Calculons le discriminant :

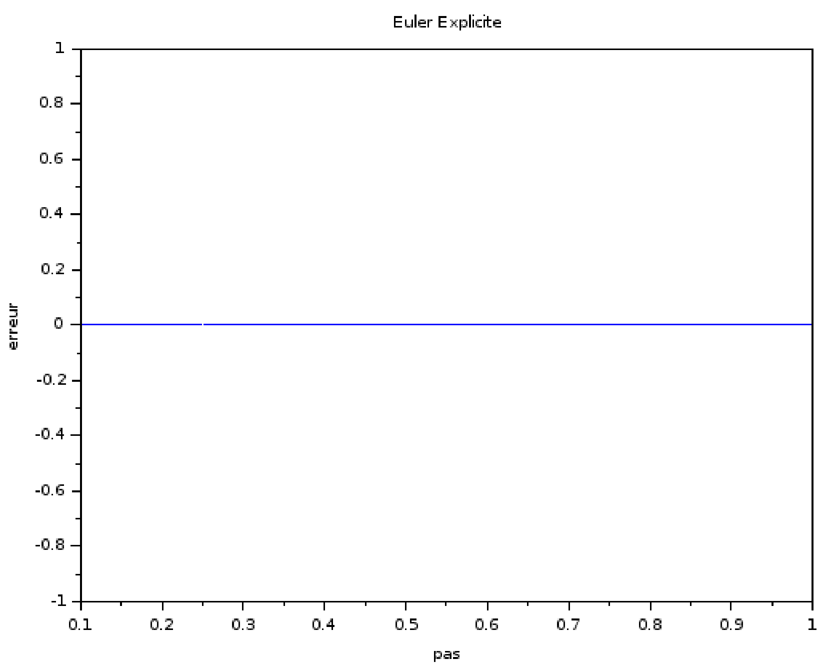
$$\begin{aligned}\Delta &= b^2 - 4ac \\ &= 1^2 - 4\Delta t(-u_m) \\ &= 4\Delta t u_m + 1\end{aligned}$$

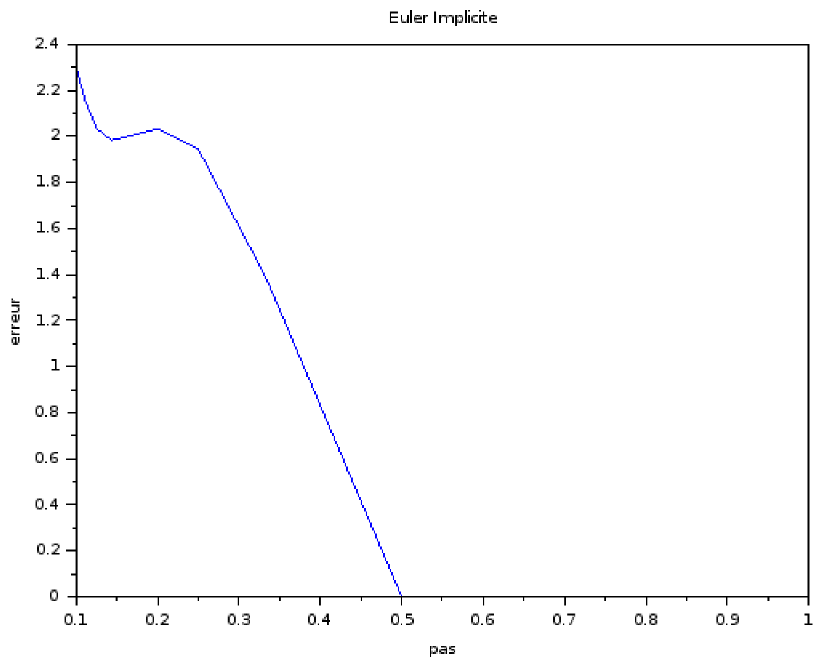
Le résustat de u_{m+1} est la solution poisitive :

$$\begin{aligned}u_{m+1} &= \frac{-b+\sqrt{\Delta}}{2a} \\ &= \frac{-1+\sqrt{4\Delta t u_m + 1}}{2\Delta t} \\ &= \frac{\sqrt{4\Delta t u_m + 1}-1}{2\Delta t}\end{aligned}$$

Donc $u_{m+1} = \frac{\sqrt{4\Delta t u_m+1}-1}{2\Delta t}$ et $u_m = \frac{\sqrt{4\Delta t u_{m-1}+1}-1}{2\Delta t}$

Le programme **etude_2.sci** permet de générer les graphes suivants :





J'ai sûrement dû faire une erreur car pour **euler explicite** j'ai une droite constante horizontale, c'est-à-dire que toutes les erreurs sont égales. Mais je l'ai fait avec des petites tailles car sinon **scilab** crashais car la récursion faisait appelle a trop de fonction.

12. Cette question est lié à la précédente donc de ce que je vois, si c'est bon, nous avons pas les mêmes conclusions que dans le cas linéaire initiale.