Fiche n^0 1. Résolution numérique d'EDOs en langage C

Exercice 1 – **Schémas d'Euler**
Etant donnés $a \in \mathbb{R}$ et $y_0 \in \mathbb{R}$, on souhaite résoudre de façon approchée l'équation différentielle ordinaire linéaire à coefficients constants suivante :

$$y' = ay \text{ avec } y(0) = y_0.$$

a) Quelle est la solution exacte de cette équation ? Discuter de sa monotonie en fonction de a et de y_0.

b)Ecrire les schémas d'Euler explicite et implicite correspondant (on notera Δt le pas de temps). Discuter, selon le signe de a, de la condition à vérifier sur Δt pour conserver le caractère monotone de la solution.

Réponses :

a) La solution exacte est une exponentielle : $y(t) = y_0 \exp(at)$. Sa dérivée étant $y'(t) = ay_0 \exp(at)$, elle est du signe de ay_0. La solution est donc strictement croissante si $ay_0 > 0$, constante si $ay_0 = 0$ et strictement décroissante sinon. Quoi qu'il en soit, elle est toujours monotone.

b) Le schéma d'Euler explicite s'écrit ici :

$$y^{(n+1)} = y^{(n)} + a\Delta t y^{(n)} = (1 + a\Delta t)y^{(n)},$$

avec $y^{(0)} = y_0$. Il s'agit donc d'une suite géométrique de raison $(1 + a\Delta t)$.

Le schéma d'Euler implicite s'écrit quant à lui, pour $a\Delta t \neq 1$:

$$y^{(n+1)} = y^{(n)} + a\Delta t y^{(n+1)} \quad soit \quad y^{(n+1)} = \frac{y^{(n)}}{(1 - a\Delta t)},$$

avec $y^{(0)} = y_0$. Il s'agit donc d'une suite géométrique de raison $(1 - a\Delta t)^{-1}$.

Pour répondre à la question posée, il suffit de se rendre compte que, pour un réel r donné, la suite $(r^n)_{n\in\mathbb{N}}$ est

- *croissante si $r > 1$,*
- *décroissante si $0 < r < 1$,*
- *constante si $r = 0$ ou $r = 1$ et*
- *oscillante si $r < 0$. Dans ce cas en effet r^n est positif une fois sur deux (pour n pair) et négatif une fois sur deux (pour n impair).*

Pour obtenir une suite monotone, il faut et il suffit donc d'avoir $r \geq 0$.

Pour Euler explicite, ceci se traduit par :

- *Si $a > 0$, alors $(1 + a\Delta t) \geq 1 \geq 0$, donc pas de condition requise sur Δt,*
- *Si $a < 0$, alors $(1 + a\Delta t) \geq 0$ si et seulement si $\Delta t \leq \frac{1}{|a|}$.*

Pour Euler implicite, ceci se traduit par :

- *Si $a > 0$, alors $(1 - a\Delta t) \geq 0$ si et seulement si $\Delta t \leq \frac{1}{a}$,*
- *Si $a < 0$, alors $(1 - a\Delta t) \geq 1 \geq 0$ donc pas de condition requise sur Δt.*

Exercice 2 – **Implémentation en Langage C - Principe de la compilation séparée**
a) Ecrire une fonction euler_imp *qui prendra comme arguments :*
`le nombre de pas de temps,`
`la valeur du pas de temps,`
`la valeur de la condition initiale,`
`le coefficient` *a* `de l'équation,`
et qui renverra comme valeur la valeur de $y(N\Delta t)$ approchée par l'algorithme d'Euler implicite.

On prendra soin d'écrire deux fichiers : l'un (euler_imp.h) avec la déclaration de la fonction (on dit aussi son prototype) sous la forme

```
---------------- euler_imp.h ----------------------

#ifndef EULER_IMP_H
#define EULER_IMP_H

double euler_imp(double, double, double, int);

#endif /* EULER_IMP_H */
```

et l'autre (euler_imp.c) qui implémentera effectivement cette fonction (c'est sa définition). Ce fichier aura la structure suivante

```
---------------- euler_imp.c ----------------------
#include "euler_imp.h"

double euler_imp(double coefficient, double pas_de_temps, double cond_init, int nombre_de_pas)
{

// A VOUS DE REMPLIR ICI

return ... ;
}
```

b) Idem, écrire une fonction euler_exp *avec les mêmes arguments. On utilisera aussi deux fichiers séparés, un .h et un .c. Attention à bien modifier les* `#ifndef` *et* `#define` *en haut du fichier .h*

c) Ecrire un programme principal main.c dans lequel on définira :

```
le temps final auquel on souhaite connaitre la valeur de y,
le nombre de pas de temps,
la valeur de la condition initiale,
le coefficient a de l'équation,
```

et qui appellera chacune des deux fonctions définies ci-dessus. Elle comparera ensuite les valeurs obtenues avec la solution exacte.

Ce programme principal devra débuter par

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "euler_imp.h"
#include "euler_exp.h"

int main()

{
 ...........
  return 0;
}
```

Pour compiler le programme, on utilisera un fichier Makefile sous la forme suivante :

```
----------------------- Makefile ---------------------------------
```

```
euler: main.o euler_imp.o euler_exp.o
<TAB>gcc -Wall -o euler main.o euler_imp.o euler_exp.o -lm

main.o: main.c euler_imp.h euler_exp.h
<TAB>gcc -Wall -c main.c

euler_imp.o: euler_imp.c euler_imp.h
<TAB>gcc -Wall -c euler_imp.c

euler_exp.o: euler_exp.c euler_exp.h
<TAB>gcc -Wall -c euler_exp.c

clean:
<TAB>   rm -f *.o euler
```

et la commande Unix `make` *sous la forme :* `make euler`

On obtiendra facilement les dépendances des .o en tapant la commande Unix `gcc -MM *.c`

Exercice 3 – **Amélioration du main pour ne pas recompiler à chaque changement**

La version précédente du main nous oblige à recompiler à chaque changement des valeurs des paramètres. C'est ennuyeux et chronophage.

On utilisera alors la possibilité de passer des arguments à l'exécutable au moment où on l'appelle. On utilisera un nouveau main sous la forme

```
---------------- main_arg.c -----------------------



#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "euler_imp.h"
#include "euler_exp.h"

int main(int argc, char *argv[])

{

    if (argc != 5) {

    printf("\nLe programme doit recevoir quatre arguments\n");

    return(EXIT_FAILURE);

    }

int N = atoi(argv[1]);
double temps_final = atof(argv[2]);
double a = atof(argv[3]);
double y0 = atof(argv[4]);
```

`argc` *est le nombre d'argument +1 passé à l'executable et* `argv[i]` *contient ces arguments. On utilisera la commande :* `./executable arg1 arg2 .....`

Exercice 4 **– Lecture des arguments dans un fichier**

On peut aussi vouloir faire lire les arguments dans un fichier pour ne pas avoir à les retaper à chaque exécution. Dans ce cas, on peut par exemple passer le fichier de données en argument de l'exécutable : `./executable fichier_de_donnees`*.*

```
---------------- main_fich.c -----------------------


#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "euler_imp.h"
#include "euler_exp.h"


int main(int argc, char *argv[])

{

    if (argc != 2) {

    printf("\nLe programme doit recevoir un seul argument\n");

    return(EXIT_FAILURE);

    }

int N ;
double temps_final;
double a ;
double y0 ;


FILE* fichier = NULL;

fichier = fopen(argv[1],"r");

    if (fichier == NULL)
    {
         printf("Impossible d'ouvrir le fichier %s \n",argv[1]);
         return(EXIT_FAILURE);
    }

fscanf(fichier, "%d %lf %lf %lf", &N , &temps_final, &a , &y0);
```

Le fichier `fichier_de_donnees` *devra comprendre : un entier, suivi de trois doubles.*