

TP n° 2

Buts : E/S, variables, boucles, if, limites du calcul numérique
Durée : 1 semaine

Rappel: ce TP **doit être préparé** avant la séance en écrivant l'algorithme ou le programme en C sur une feuille **manuscrite**. Vous pouvez aussi taper les fichiers correspondants sur machine pour gagner du temps pendant la séance de TP.

Rappel: voir le document « comment développer un programme en TD sur le site tdinfo »

Pour réaliser vos programmes sur machine, il faut aller dans le répertoire concerné, éditer le fichier source C, compiler (et corriger si nécessaire) puis exécuter le programme. Si les tests ne sont pas tous corrects, il faut corriger et compiler et tester etc Faites un répertoire par TD. Cela correspond à :

Ouvrir un terminal (Menu Outils de système ou Menu accessoires selon la version)

1. Dans le terminal :
 - a. Aller dans votre répertoire tdinfo : `cd tdinfo`
 - b. Créer un répertoire pour le TD2 : `mkdir td2`
 - c. Aller dans le répertoire td2 : `cd td2`
 - d. Editer le fichier source C, nommé `eq1.c` : `gedit eq1.c &`
2. Dans l'éditeur,
 - a. taper votre programme
 - b. sauver votre programme
3. Dans le terminal :
 - a. Compiler votre programme : `gcc eq1.c -o eq1 -lm (-lm = library Math)`
 - b. Exécuter votre programme s'il n'y a pas d'erreur de compilation: `./eq1`
 - c. Sinon, corriger votre programme dans l'éditeur, sauver à nouveau (étape 2) et compiler dans le terminal. (étape 3)

1/ Système de 2 équations à 2 inconnues

Résoudre un système linéaire de 2 équations à 2 inconnues :
$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

Pour résoudre un tel système, il faut calculer son déterminant qui est $d = a_1b_2 - a_2b_1$.

S'il est nul, le système admet soit une infinité de solutions (les deux équations sont proportionnelles), soit aucune solution.

sinon les solutions sont données par

$$x = (c_1b_2 - c_2b_1) / d$$

$$y = (a_1c_2 - a_2c_1) / d$$

Travail à réaliser :

Ecrire le programme C résolvant un tel système. Votre programme doit lire les 6 paramètres sur deux lignes

a1 b1 c1

a2 b2 c2

Il doit effectuer ensuite les calculs nécessaires pour afficher les résultats sous la forme
Les solutions du système sont X=1 et Y=1

Attention : il existe des cas particulier très « tordus ». L'objectif du TD est avant tout de faire travailler les notions simples (E/S, structures conditionnelles), vous pouvez simplifier (pour ce TD) en excluant ces cas : les . Quelques exemples de systèmes à résoudre :

Ex1 : $2x+3y=5$ $x+y=2$
Ex2 : $2x+3y=2$ $x+y=1$
Ex3 : $2.4x+6.8y=19.67$ $-3.765x+8.3y=-9.7$
Ex4 : $2x+2y=4$ $x+y=2$
Ex5 : $2x+2y=5$ $x+y=2$
Ex6 : $2x+2y=5$ $0x+0y=0$
Ex7 : $2x+2y=5$ $0x+0y=1$

Ces exemples ne sont bien sûr pas exhaustifs. Vous avez sans doute pensé à tester certains d'entre eux, mais peut être pas tous. Ils sont ici pour rappeler que la notion de développement doit intégrer la **notion de jeu de tests permettant de valider un programme y compris dans les cas les moins prévisibles**.

2/ Nombre d'or et Fibonacci

Au XIII^e siècle, Fibonacci, appelé aussi Léonard de Pise, celui là même qui avait amené en occident les chiffres hindous au retour d'un long séjour à Alger, tenta de résoudre le problème de la prolifération des lapins. Il voulait se rendre compte de la puissance de cette prolifération, la formuler, alors qu'elle dépassait très vite l'imagination. La myxomatose lui vint alors en aide...

Il partit d'un fait très rare dans l'espèce humaine : que des oncles et des neveux soient de même âge. Dans ce cas, possédant initialement un couple de lapins, combien de couples obtient-on en douze mois si chaque couple engendre tous les mois un nouveau couple à compter du second mois de son existence ?

Il étudiait donc la suite : $U_n = U_{n-1} + U_{n-2}$ en supposant que $U_1 = 1$ et $U_0 = 0$.

Cette suite donne aussi accès au nombre d'Or, qui en architecture et dans les arts graphiques, donne le rapport entre la longueur et la largeur du rectangle le plus agréable à l'œil. Ce nombre est

la limite de la suite $V_n = U_n / U_{n-1}$ et vaut $\frac{1+\sqrt{5}}{2}$, soit 1,618 033 988 749 894 848 204 586 834 365 638 117 720 309 179 805 762 862 135 448 622 705 260 462 189 024 497 072 072 041.

Pour calculer une limite, il faut utiliser l'écart relatif entre 2 termes consécutifs et le comparer à une valeur faible, compatible avec la précision des nombres réels que vous utilisez (10^{-7} en float, 10^{-15} en double) c.-à-d. tester $|V_n - V_{n-1}| > \epsilon * (|V_n| + |V_{n-1}|)$. Attention, cela ne garantit pas que la limite est atteinte, mais simplement que la machine ne sait pas faire mieux !!!

Travail à réaliser:

1. Ecrire le programme qui affiche à l'écran les n premiers termes de la suite U_n , n > 1 étant donné au clavier. Ce programme fait intervenir une boucle. En plus de l'indice de boucle et du nombre n, vous ne devez utiliser que 3 variables pour calculer les différents termes U_n de la suite. Ces trois variables (a,b,c par exemple) représenteront u_2, u_1, u_0 au départ de la boucle

(itération 0), puis u_3, u_2, u_1 au tour de boucle suivant (itération 1), puis $u_4, u_3, u_2 \dots$ il suffit de modifier dans le bon ordre les valeurs de a et de b à la fin de la boucle, puisqu'il faut qu'au début de la 2^{ième} itération de la boucle, b contienne u_3 et a contienne u_2 de manière à ce que a+b donne u_4 .

Attention, regardez attentivement ce qui se passe autour des valeurs de U_{44} . Cette suite est très rapidement croissante et positive en théorie

2. Ecrire le programme qui calcule la limite de la suite V et donne le nombre d'or. La valeur absolue d'un réel x est obtenue par fabs(x).

Facultatif

1. Soit la série $S_n = \sum_{i=1}^n \frac{1}{i^2}$

Calculer les N premiers termes de cette série, dans le sens croissant des i et dans le sens décroissant, puis comparer les résultats. Attention : en C, l'écriture $1 / (i*i)$ réalise la division entière et il faut donc écrire $1.0 / (i*i)$ pour réaliser une division réelle.

Comment expliquez vous la différence ?

2. Faire un programme qui calcule Pi par : $\frac{\pi}{4} = 4 \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$

avec la série $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$

Quelle précision obtenez vous ?

3. Différences évanescences.

Soit x un nombre positif. La suite x_n définie par $x_0 = x$ et $x_{n+1} = \sqrt{\sqrt{x_n}}$ converge vers 1.

On définit alors $\rho_n = \frac{x_{n+1} - 1}{x_n - 1}$ qui converge vers $\frac{1}{4}$. Ecrire le programme calculant les

différentes valeurs des suites x_n et ρ_n afin de déterminer sa limite. Quelle observation pouvez vous faire ?