Exercices sur les fonctions

Exercice 1

On sait que si f est une fonction continue strictement monotone sur un intervalle $[a;b]$ et que $f(a)$ et $f(b)$ sont de signes contraires, l'équation $f(x) = 0$ admet une racine unique α appartenant à $]a;b[$.

Par **dichotomie**, on peut obtenir une suite de nombres convergeant vers α. L'algorithme en est le suivant :

- on connaît au départ : - la fonction f (supposée continue strictement monotone),
  - les réels a et b tels que $f(a)\times f(b)$ soit négatif,
  - une constante epsilon indiquant la précision attendue.
- la variable auxiliaire c désignera la demi-somme de a et b.
- algorithme :

```
début
      tantque (a+b)/2>epsilon
        debut
          c  <--  (a+b)/2;  //afficher(c) ici si on désire toutes les approximations
          si  f(a)*f(c)<0 alors  b <-- c sinon  a <-- c;
        fin
      afficher((a+b)/2);  //ou retourner((a+b)/2) s'il s'agit d'une fonction
fin
```

1) Ecrire une fonction [y]=f(x) permettant la déclaration de la fonction f définie par cos(x) - x
2) Ecrire une *fonction* **DICHO** réalisant l'algorithme précédent, a et b étant reçus en paramètres entrants et c comme paramètre de sortie.

Ecrire la ou les instructions sous Scilab permettant d'obtenir une solution de $\cos(x) = x$ à 10^{-10} près.

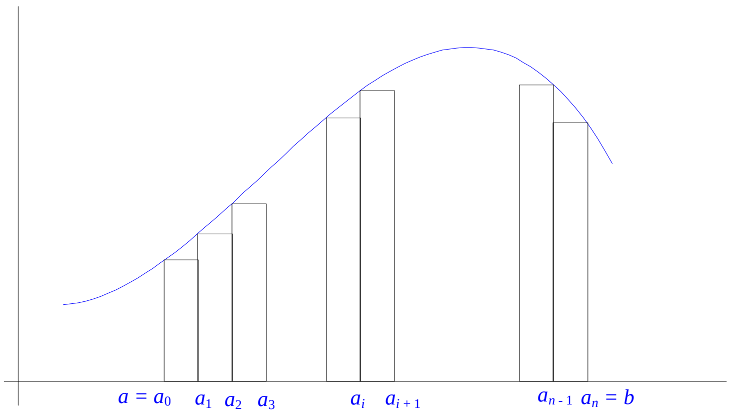
Exercice 2

Calculs approchés d'intégrales

Si f est continue (ou continue par morceaux) sur $[a;b]$, alors f admet des primitives sur $[a;b]$ (ou est intégrable sur $[a;b]$) mais il se peut qu'on ne sache pas calculer $\int_a^b f(t)dt$.

<u>Méthode</u> : On subdivise $[a;b]$ en n segments de longueur $\frac{b-a}{n}$: $a_0 = a$; $a_1 = a_0 + \frac{b-a}{n}$; $a_2 = a_1 + \frac{b-a}{n}$; … ; $a_i = a_{i-1} + \frac{b-a}{n}$; … ; $a_n = b$.

Dans ce cas, $I = \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} f(t)dt$



On remplace $\int_{a_i}^{a_{i+1}} f(t)dt$ par une valeur approchée obtenue en prenant pour f une fonction approximant f et dont on sait calculer l'intégrale sur $[a_i;a_{i+1}]$

Méthode des rectangles : On remplace $f(t)$ sur $[a_i;a_{i+1}]$ par $f(a_i)$

On obtient alors comme valeur approchée de l'intégrale $S_n = \frac{b - a}{n} \sum_{i=0}^{n-1} f(a_i)$

Programmation :
On dispose de la fonction suivante permettant d'obtenir les images d'un réel x par une fonction

```
function [y]=f(x)
  y = 1/(1+x*x);
```

Ecrire la fonction **[s]=somme(a,b)** permettant de calculer la somme des aires des rectangles après avoir demandé à l'utilisateur le nombre d'intervalles de la subdivision.

Exercice 3

1) Ecrire une *fonction* **PPDP** renvoyant le **P**lus **P**etit **D**iviseur **P**remier d'un entier au moins égal à 2 reçu en paramètre.
2) Ecrire une *fonction* **DECOMPOSER** affichant à l'écran la décomposition en facteurs premiers d'un entier au moins égal à 2 reçu en paramètre. Cette procédure devra appeler la fonction précédente. L'affichage devra ressembler à ceci :

360 = 2 * 2 * 2 * 3 * 3 * 5   ou à : 360 = 2^3 * 3^2 * 5.

3) Ecrire une *fonction* **SAISIR** demandant à l'utilisateur un entier au moins égal à 2 (et répétant la saisie jusqu'à ce que cette condition soit satisfaite), puis renvoyant cet entier dans un paramètre (passé bien sûr par adresse).
4) Vous pourrez réaliser le programme suivant sous Scilab (après avoir chargé le fichier de fonctions) :

```
a = SAISIR()
DECOMPOSER(a)
```