Nb: On rédigera les exercices sur des feuilles séparées. On pourra admettre tout résultat intermédiaire afin de poursuivre la résolution d'un exercice.

Exercice 1

Le but de cet exercice est d'approximer la fonction logarithme népérien, notée ln, par différentes méthodes numériques.

1.1 *Polynôme d'interpolation*

a) Donner la forme de Newton du polynôme d'interpolation de la fonction ln sur le support $\{1, 2\}$.

b) On considère la fonction erreur e_1, définie par $e_1(x) = \ln(x) - p(x)$.

Étudier les variations de e_1 sur $[1, 2]$.

En déduire que l'erreur est maximale, en valeur absolue, pour $x = 1/\ln(2)$; donner la valeur maximale de e_1 sur $[1, 2]$.

1.2 *Méthode du point milieu*

Pour la suite du problème on rappelle que la fonction ln est définie par:

$$\ln(x) = \int_1^x \frac{1}{t} dt$$

Nb : On remarquera ici que la variable d'intégration est t et non x; la variable x, elle, définit le segment d'intégration !

a) Montrer en utilisant la méthode d'intégration du point milieu, que :

$$\int_1^x \frac{1}{t} dt \simeq 2\frac{x-1}{x+1}$$

b) Étudier les variations de la fonction erreur e_M, définie sur $[1, 2]$ pour cette méthode par :

$$e_M(x) = \ln(x) - 2\frac{x-1}{x+1}.$$

c) En déduire la valeur maximale de e_M sur $[1, 2]$.

1.3 *Intégration gaussienne*

Maintenant on cherche à évaluer $\ln(x) = \int_1^x \frac{1}{t} dt$ par une intégration de Gauss-Legendre à deux points.

a) Montrer en précisant le changement de variable à réaliser que :

$$\int_1^x \frac{1}{t} dt = \int_{-1}^{1} \frac{x-1}{(x-1)u + x + 1} du$$

b) En déduire, en majorant l'erreur de méthode e_G commise dans l'évaluation de I, que cette méthode donne une approximation de $\ln(x)$ à 10^{-2} près sur $[1, 2]$.

.../...

1.4 *Discussion*

a) *Remarques générales :*

- La différence de précision entre les méthodes **1.2** et **1.3** est-elle surprenante ?
- Au vu des calculs effectués, par quel type d'objets mathématiques a-t-on approché $\ln(x)$ dans les parties **1.2** et **1.3** ? Même question pour la partie **1.1**.

b) *Généralisation*

- Si on veut une précision d'ordre ε en utilisant la méthode de la partie **1.3**, comment opérer ? Fournir le plan de mise en oeuvre.

Exercice 2

L'objet de l'exercice est l'étude de la méthode de Müller, extension de la méthode de la sécante utilisée dans le cadre de la résolution d'équations non linéaires.

Soit f une fonction numérique qu'on suppose définie sur $\mathbb{R}$ pour des raisons de simplicité. Soit i un entier naturel supérieur ou égal à 2; on considère trois réels notés x_i, x_{i-1} et x_{i-2}. On fournit en page 4 une représentation graphique générique de la situation où figurent f et son polynôme d'interpolation sur le support $\{x_i, x_{i-1}, x_{i-2}\}$. On considère l'équation (E) : $f(x) = 0$ et on note l la solution cherchée.

2.1 *Etude mathématique préalable*

a) *Interpolation*

Fournir l'écriture de Newton du polynôme interpolateur p de f sur le support $\{x_i, x_{i-1}, x_{i-2}\}$ en fonction des différences divisées de f et des points du support.

b) *Transformation de l'écriture de p*

- Montrer que pour tout x réel on a:

$$(x - x_i)(x - x_{i-1}) = (x - x_i)^2 + (x - x_i)(x_i - x_{i-1}).$$

- En déduire que pour tout x réel on peut écrire:

$$p(x) = f(x_i) + (x - x_i)c_i + f[x_i, x_{i-1}, x_{i-2}](x - x_i)^2$$

sachant que

$$c_i = f[x_i, x_{i-1}] + f[x_i, x_{i-1}, x_{i-2}](x_i - x_{i-1}).$$

.../...

c) Soit (E') l'équation approchée de (E), définie par: $(E') : p(x) = 0$.

- Combien l'équation (E') admet-elle de solutions dans $\mathbb{C}$ en général ? Existe-t-il des exceptions ?
- On pose $X = x - x_i$. On considère l'équation $(E") : P(X) = 0$, où $P(X)$ est donnée par:

$$P(X) = f(x_i) + c_i X + f[x_i, x_{i-1}, x_{i-2}] X^2$$

  Montrer que si l'on souhaite construire par ce procédé une suite $(x_i)_{i\in\mathbb{N}}$ dont on espère qu'elle converge vers l, on devra :

  – définir x_{i+1} comme la racine convenable de l'équation (E');
  – et par conséquent poser

$$x_{i+1} = x_i + X_1,$$

  où X_1 désigne la solution de $(E")$ de plus petit module.

2.2 *Algorithme de Müller*

Un certain nombre de sous modules de l'algorithme proposé ci-dessous, sont seulement nommés, sans être précisément décrits Une fois complété, cet algorithme pourra être implémenté sous langage et opérationnel!

On demande, grâce à l'étude mathématique préalable, de fournir le détail des modules suivants:

Détermination de $(E")$, *Calcul de* X_1 et *Définition de* $vect(i+1)$

en identifiant nettement les outils et l'environnement nécessaires.

****

$müller\,(x_2, x_1, x_0, nmax \to vect)$

1. En-tête
   - Entrées:
     – x_2, x_1, x_0 valeurs initiales de la racine cherchée;
     – $nmax$: nombre d'itérés de Müller demandés ($nmax \geq 3$)
   - Sorties: Vecteur des $nmax$ premiers itérés de Müller.
2. Corps d'algorithme
   (a) *Initialisations:*
   - $\{$*déclaration de vect* comme vecteur de $nmax$ complexes;
   - $vect(1) \leftarrow x_0$;$vect(2) \leftarrow x_1$;$vect(3) \leftarrow x_2$ $\}$

   (b) *faire pour* $i \leftarrow 3$ *jusqu'à* $nmax - 1$
      1. *Détermination de* $(E")$;
      2. *Calcul de* X_1;
      3. *Définition de* $vect(i+1)$.

      fin de faire en i
3. Fin de fonction $müller$ ( )

****

2.3 On suppose ici que la fonction f est polynôme.

a) La méthode de Newton permet-elle de déterminer des racines multiples, des racines complexes? Pourquoi?

b) Mêmes questions pour l'algorithme de Müller.

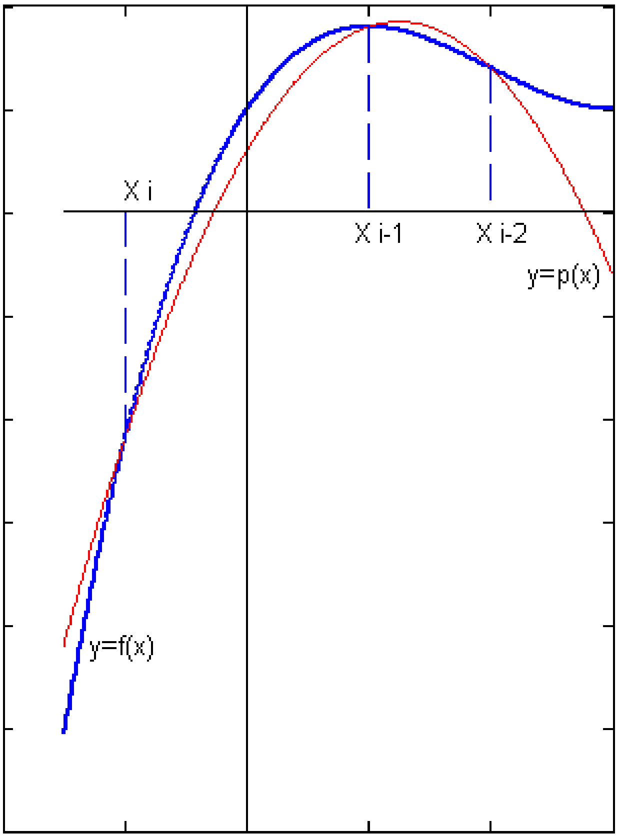

Figure 1: