Arts et Métiers Sciences et Technologies

Année 2021-2022

Méthodes Numériques - TP 4

Ruding LOU

ruding.lou@ensam.eu

Résolution itérative d'équations

- Veuillez respecter le **nommage de dossier** à créer pour déposer vos fichiers : **NOM_Prenom_TP4_ex1.py**
- Respecter les consignes générales (Annexe. 1).

1. Utiliser les **méthodes itératives** pour résoudre une équation **non-linéaire**.
   - On souhaite approcher la valeur de $\sqrt{c}$ où c est un nombre positif introduit par l'utilisateur. Ce dernier revient à résoudre une équation $f(x) = x^2 - c = 0$. Implémenter la méthode itérative de **Newton** et tester votre programme pour résoudre l'équation en utilisant le point de départ $x_0 = c$ et avec quelques itérations pour atteindre une erreur $|x_n - x_{n-1}| \leq 10^{-3}$. Tracer l'illustration graphique des itérations (Annexe. 2).
   - Même question pour l'algorithme de **Quasi-Newton**. Au démarrage la méthode nécessite deux points d'où on ajoute un point auxiliaire x_1 au voisinage du point de départ x_0. Si on prend $x_0 = c$ et un point voisin $x_1 = c + 0.1$ on obtiendra des itérations illustrées dans la figure (Annexe. 3).
2. Utiliser les **méthodes itératives** pour résoudre un système **non-linéaire**: $\begin{cases} e^{x-1} - x + 1 & = y \\ \frac{(x-0)^2}{5^2} + \frac{(y-4)^2}{1^2} & = 1 \end{cases}$
   - Afin d'illustrer graphiquement les deux équations, tracer les deux courbes (Annexe. 4) :

$$\boxed{y = e^{x-1} - x + 1} \quad x \in [-6,\ 4] \qquad \text{et} \qquad \boxed{y = \begin{cases} \frac{1}{5}\left(20 - \sqrt{25 - x^2}\right) \\ \frac{1}{5}\left(20 + \sqrt{25 - x^2}\right) \end{cases}} \quad x \in [-5,\ 5]$$

   - Implémenter l'algorithme de **Newton** et tester votre programme pour résoudre le système en prenant comme points de départ : $[\ (6,8), (6,1), (-2,8), (-1,1)\ ]$ et avec quelques itérations pour atteindre une erreur $\left\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)}\right\|_2 \leq 10^{-3}$. Tracer l'illustration graphique des itérations (Annexe. 5). Veuillez utiliser la méthode de **Gauss** pour résoudre un système linéaire.
   - Même question pour l'algorithme de **Quasi-Newton**. Au démarrage la méthode nécessite deux points d'où on ajoute un point auxiliaire au voisinage du point de départ. Si on reprend les points de départs proposés dans l'exercice précédent on aura les suites d'itérations :

$$\left| \begin{array}{l} (\ 6.1,\ 8.1),\ (\ \ 6,\ 8),\ \ldots \\ (\ 6.1,\ 1.1),\ (\ \ 6,\ 1),\ \ldots \\ (-1.9,\ 8.1),\ (-2,\ 8),\ \ldots \\ (-0.9,\ 1.1),\ (-1,\ 1),\ \ldots \end{array} \right.$$

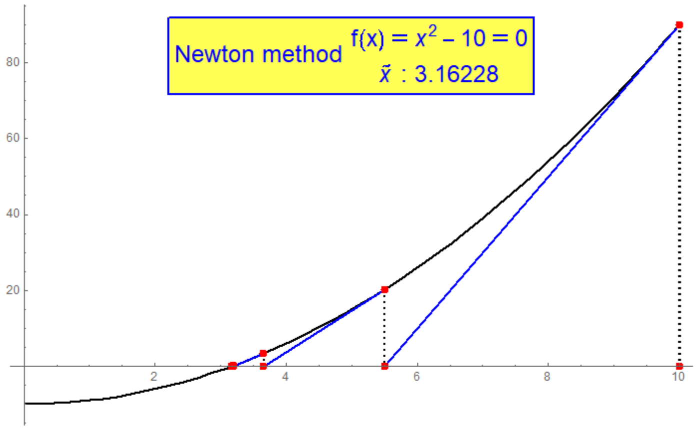
(Annexe. 6)

Annexe. 1

L'utilisateur **lance le programme** et **se laisse guider** dans la console :

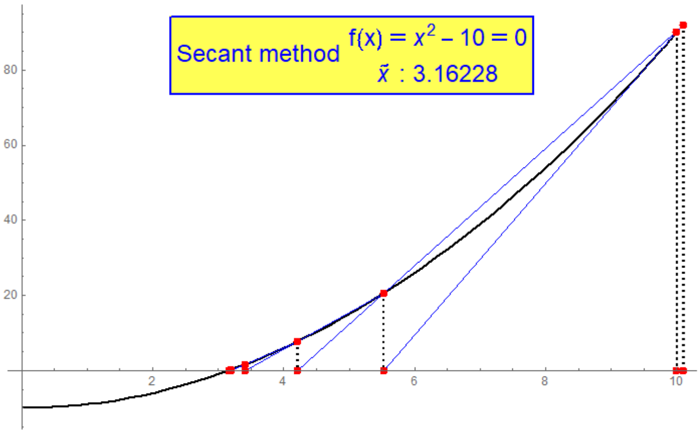
- demander à l'utilisateur de saisir un nombre positif *c* si cela concerne l'exercice ;
- demander à l'utilisateur de saisir le point de départ si cela concerne l'exercice ;
- afficher les valeurs calculées des itérations intermédiaires si cela concerne l'exercice ;
- afficher le résultat final de calcul et la comparaison avec la solution de référence ;
- tracer la figure des itérations si cela concerne l'exercice ;

Annexe. 2



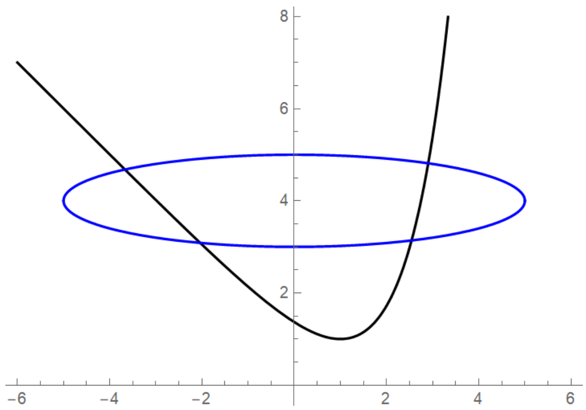
La fonction f : la courbe noire,
la solution recherchée: le point rouge d'intersection entre f et l'abscisse
la valeur initiale (x = 10) : le point rouge sur l'abscisse
Les pentes calculées à partir des points rouges sur la courbe f : les droites bleues
Les valeurs calculées : les points rouges d'intersection entre les pentes et l'abscisse

Annexe. 3

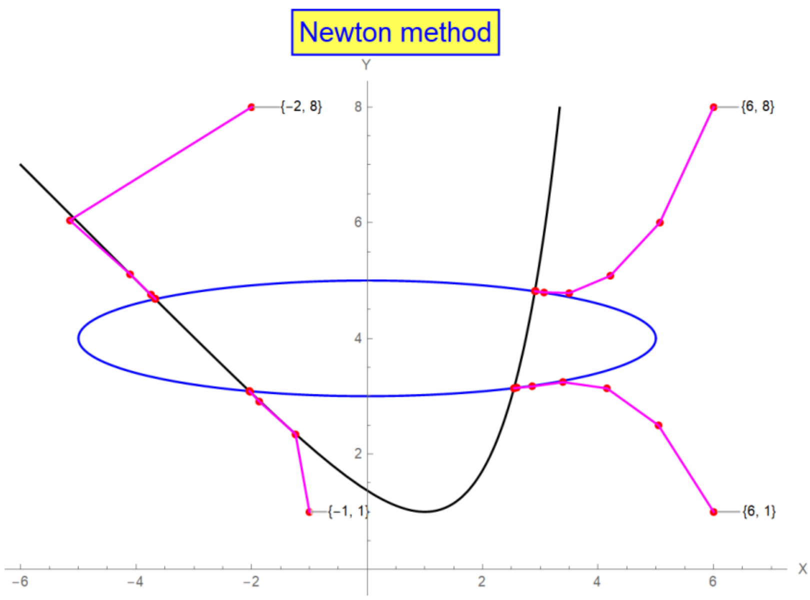


La fonction f : la courbe noire,
la solution recherchée: le point rouge d'intersection entre f et l'abscisse
les deux valeurs initiales (x=10 et x=10.1) : les points rouges sur l'abscisse
Les sécantes calculées à partir des points rouges sur la courbe f : les droites bleues
Les valeurs calculées : les points rouges d'intersection entre les sécantes et l'abscisse

Annexe. 4



Annexe. 5



Annexe. 6