Université Paris I, Panthéon - Sorbonne

Deuxième Année Licence M.I.A.S.H.S. 2016 – 2017

TP de Méthodes Numériques n^0 5 : Solutions approchées de l'équation $f(x) = 0$

Dans ce TP, on met en place deux méthodes permettant de résoudre numériquement l'équation $f(x) = 0$.

Un problème de résolution d'équation

Dans la suite nous commençons par considérer l'exemple de fonction f définie par:

$$f_1(x) = \sin(2x) - x$$

Étudier théoriquement les solutions de l'équation $f(x) = 0$ dans ce cas où $f = f_1$ (utiliser par exemple un tableau de variations...). En déduire que l'équation $f_1(x) = 0$ possède 2 solutions positives ou nulles, une que l'on peut donner théoriquement et notée x_0, l'autre notée x_1 a priori inconnue. Vérifier en traçant le graphe de f sur $[0, 1]$. Déterminer une valeur approchée à 0.1 près de x_1. Nous allons commencer par donner une approximation aussi fine que désirée de x_1 suivant le même principe que celui que vous avez utilisé pour donner cette première valeur approchée.

Méthode d'approximation par dichotomie

On suppose que $f : \mathbf{R} \to \mathbf{R}$ est une fonction continue sur un voisinage de x_1 (ce qui est le cas pour $f = f_1$), unique solution de l'équation $f(x) = 0$ dans ce voisinage. On suppose qu'il existe $[a_0, b_0]$ dans ce voisinage tel que $x_1 \in [a_0, b_0]$ et $f(a_0)f(b_0) < 0$. On définit alors les suites (a_n) et (b_n) telles que:

$$\begin{array}{lll} a_{n+1} = \frac{1}{2}(a_n + b_n) \quad b_{n+1} = b_n & \text{si} & f(a_n)f\big(\frac{1}{2}(a_n + b_n)\big) \geq 0 \\ b_{n+1} = \frac{1}{2}(a_n + b_n) \quad a_{n+1} = a_n & \text{si} & f(b_n)f\big(\frac{1}{2}(a_n + b_n)\big) \geq 0 \end{array}$$

En gros on se rapproche de plus en plus de x_1 en découpant en 2 chaque intervalle précédent. Ainsi à l'étape n, la taille de l'intervalle est $b_n - a_n = 2^{-n}(b - a)$, ce qui mesure aussi la qualité de l'approximation de x_1.

Programmer cette méthode pour la fonction f_1 considérée plus haut et sur un intervalle $[a_0, b_0]$ contenant x_1 et ne contenant pas x_0. En combien d'étapes avez vous obtenu une approximation à 10^{-10} près en partant de la valeur approchée à 10^{-1} obtenue plus haut?

> Un résultat vu en cours nous apprend qu'avec cette méthode la qualité de l'approximation est donnée par $b_n - a_n = 2^{-n}(b_0 - a_0)$, donc si $b_0 - a_0 = 1$, pour une approximation à 10^{-p} près, il faut $p\ln(10)/\ln(2)$ étapes.

Vérifier numériquement ce résultat pour f_1.

Méthode de la sécante

On se place dans le cadre précédent, c'est-à-dire que l'on suppose qu'il existe $u_0 < u_1$ tel que $x_1 \in]u_0, u_1[$ soit l'unique solution de $f(x) = 0$ dans $]u_0, u_1[$ (donc $f(u_0)f(u_1) < 0$). On va alors définir u_2 de la manière suivante: u_2 est le point d'intersection entre la droite joignant $(u_0, f(u_0))$ et $(u_1, f(u_1))$ et l'axe des abscisses (faire un dessin!). D'où le nom de méthode de la sécante... Montrer qu'alors $u_2 = u_1 - f(u_1)\,\frac{u_1 - u_0}{f(u_1) - f(u_0)}$.
On peut généraliser cette démarche et ainsi définir la suite $(u_n)_n$ telle que

$$u_{n+1} = u_n - f(u_n)\,\frac{u_n - u_{n-1}}{f(u_n) - f(u_{n-1})} \quad \text{pour } n \geq 1.$$

Programmer cette méthode d'approximation et appliquer la à la fonction f_1 et l'obtention d'une valeur approchée de x_1. En combien d'étapes avez vous obtenu une approximation à 10^{-10} près en partant de la valeur approchée à 10^{-1} obtenue plus haut? Notons que cette méthode ne nécessite pas de condition de dérivation sur f. Cependant, on peut mesurer la qualité de l'approximation en rajoutant des hypothèses de dérivation d'ordre 2.

> Un résultat vu en cours nous apprend qu'avec cette méthode, si f est de classe $\mathcal{C}^2$ sur $[u_0, u_1]$, la qualité de l'approximation est donnée par $|u_n - x_1| \leq \frac{1}{C} \exp(-K\,\phi^n)$ lorsque $K = \ln(C(|u_0 - x_1|)/\phi - \ln(C(|u_1 - x_1|) > 0$, avec $\phi = (\sqrt{5}+1)/2$ (le nombre d'or), $C = \frac{1}{2}\,M_2/m_1$ où $m_1 = \inf_{x\in[u_0,u_1]} |f'(x)|$ et $M_2 = \sup_{x\in[u_0,u_1]} |f''(x)|$. Pour une approximation à 10^{-p} près, il faut $\ln\left(\frac{p\,\ln(10) - \ln C}{K}\right)/\ln(\phi)$ étapes.

Vérifier numériquement ce résultat pour f_1.

Méthode de Newton-Raphson

Si maintenant on suppose que $f : \mathbf{R} \to \mathbf{R}$ est une fonction de classe $\mathcal{C}^1$ sur un voisinage I de x_1, unique solution de l'équation $f(x_1) = 0$ dans ce voisinage, on voit avec la méthode de la sécante que pour n suffisamment grand, u_n se rapproche de x_1 et donc $\frac{u_n - u_{n-1}}{f(u_n) - f(u_{n-1})} \sim 1/f'(u_n)$. Aussi peut-on définir la suite (v_n) telle que:

$$v_{n+1} = v_n - \frac{f(v_n)}{f'(v_n)} \quad \text{pour } n \in \mathbf{N}.$$

Programmer cette méthode d'approximation et appliquer la à la fonction f_1 et l'obtention d'une valeur approchée de x_1. En combien d'étapes avez vous obtenu une approximation à 10^{-10} près en partant de la valeur approchée à 10^{-1} obtenue plus haut?

> Un résultat vu en cours nous apprend qu'avec cette méthode, si f est de classe $\mathcal{C}^2$ sur I contenant v_0 et x_1, la qualité de l'approximation est donnée par $|v_n - x_1| \leq \big(C\,|v_0 - x_1|\big)^{2^n}/C$ si v_0 est suffisamment proche de x_1 (si $C\,|v_0 - x_1| < 1$) avec $C = \frac{M_2}{2m_1}$ où $m_1 = \inf_{x\in I} |f'(x)|$ et $M_2 = \sup_{x\in I} |f''(x)|$. Pour une approximation à 10^{-p} près, il faut $\ln\left(\frac{\ln C - p\,\ln(10)}{\ln(C|v_1 - x_1|)}\right)/\ln(2)$ étapes.

Vérifier numériquement ce résultat pour f_1.

Exercices

1. Reprendre les 3 méthodes proposées pour déterminer une valeur approchée à 10^{-10} près de $\sqrt{2}$, puis de $10^{1/5}$.
2. On peut encore accélérer la convergence de la méthode de Newton-Raphson en utilisant une approximation polynomiale de f plutôt qu'une approximation linéaire de f. On considérera alors

$$w_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)} - \frac{f^2(w_n) f''(w_n)}{4 f'(w_n)^3} \quad \text{pour } n \in \mathbf{N}.$$

   Programmer et utiliser cette nouvelle suite pour estimer x_1 puis $\sqrt{2}$. Conclusion quant à la rapidité de cette nouvelle méthode?