Université Paris Descartes
UFR de Mathématiques et Informatique
45, rue des Saints-Pères 75270 Paris cedex 06

MASTER 1

Programmation

Résolution d'équations non linéaires - Scilab

Exercice 1 (méthodes de dichotomie et de Newton–Raphson, *d'après A. Quarteroni*). Dans cet exercice, on souhaite utiliser sur des exemples différentes méthodes d'approximation d'un zéro d'une fonction.

1. On considère tout d'abord la fonction $g(x) = \frac{x}{2} - \sin(x) + \frac{\pi}{6} - \frac{\sqrt{3}}{2}$ sur l'intervalle $\left[-\frac{\pi}{2}, \pi\right]$, en observant qu'elle y possède deux zéros.

   a. Définir la fonction $y = \mathtt{g}(x)$ dans Scilab.

   b. Tracer le graphe de la fonction g sur $\left[-\frac{\pi}{2}, \pi\right]$ avec les commandes

   ```
   x=linspace(-%pi/2,%pi,100);
   plot2d(x,g(x));
   ```

   Expliquer pourquoi la méthode de dichotomie ne peut être utilisée que pour approcher l'un des deux zéros de g, que l'on notera ξ dans la suite.

   c. Ecrire une fonction `[zero,iter,res,inc]=dichotomie(f,a,b,tol,nmax)` implémentant la méthode de dichotomie pour l'approximation d'un zéro d'une fonction f donnée, compris dans un intervalle $[a, b]$ tel que $f(a)f(b) < 0$. Les autres paramètres d'entrée `tol` et `nmax` de la fonction `dichotomie` sont respectivement la tolérance pour le critère d'arrêt de la méthode et le nombre maximum d'itérations à effectuer. Les paramètres de sortie `zero`, `iter`, `res` étant pour leur part l'approximation du zéro obtenue, le nombre d'itérations nécessaire au calcul de cette approximation, la valeur de la fonction f en ce point. Le paramètre de sortie `inc` est un vecteur contenant la suite des valeurs absolues des différences entre deux approximations successives (dite suite des incréments). Si on note $x^{(k)}$ l'approximation du zéro à la k^{ieme} itération de la dichotomie, $\mathtt{inc}(k)$ doit donc contenir la valeur de l'incrément $|x^{(k)} - x^{(k-1)}|$.

   d. Utiliser la fonction `dichotomie` sur la fonction g de la question a) pour calculer une approximation de ξ avec une tolérance égale à 10^{-10} pour le critère d'arrêt à partir du choix d'un intervalle $[a, b]$ convenable.

   e. Au moyen de la commande `plot2d("nl",inc)`, tracer le graphe de la suite des incréments $|x^{(k+1)} - x^{(k)}|$ (en fonction de k) avec une échelle semilogarithmique et déterminer la loi selon laquelle ces quantités tendent vers 0 quand k tend vers l'infini.

   f. On souhaite maintenant utiliser la méthode de Newton pour approcher les zéros d'une fonction f. On rappelle qu'on part d'un point x_0 dans l'ensemble de définition de la fonction et qu'on construit la suite :

   $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{1}$$

   Le point x_{k+1} est en fait l'intersection de la droite $y = f(x_k) + f'(x_k)(x - x_k)$ avec l'axe des abcisses. Pour appliquer la méthode de Newton dans Scilab à une fonction f donnée, il faudra non seulement définir f dans Scilab (comme on l'a fait pour g dans la question a.), mais également définir dans Scilab la fonction df qui correspond à la dérivée de f.

   Ecrire une fonction `[zero, iter,res,inc]=newton(f,df,x0,tol,nmax)` qui implémente la méthode de Newton–Raphson pour l'approximation d'un zéro d'une fonction dérivable

f donnée. Les paramètres d'entrée `df`, `x0`, `tol` et `nmax` représentent respectivement le nom de la fonction correspondant à la fonction f', l'initialisation de la méthode, la tolérance pour le critère d'arrêt de la méthode et le nombre maximum d'itérations à effectuer. En sortie, les paramètres sont identiques à ceux de la fonction `dichotomie`.

g. Calculer la dérivée g' de la fonction g définie en a., et définir une fonction $y = \mathtt{dg}(x)$ dans Scilab qui correspond à cette dérivée. Calculer des approximations des deux zéros ξ et ζ de la fonction g avec la méthode de Newton–Raphson, en prenant une tolérance égale à 10^{-10} pour le critère d'arrêt et comme initialisation le point π pour ξ et $-\frac{\pi}{2}$ pour ζ. Comparer les nombres d'itérations effectuées pour obtenir une approximation de chacun des zéros. Pourquoi sont-ils très différents ? Comparer également les graphes des suites des incréments obtenus avec la commande `plot2d("nl",inc)`.

h. On cherche à réduire le nombre d'itérations nécessaires à l'obtention d'une approximation du zéro négatif ζ de la fonction g. La méthode de Newton–Raphson modifiée, basée sur la modification suivante de la relation de récurrence de la méthode de Newton–Raphson

$$x^{(k+1)} = x^{(k)} - 2\frac{f(x^{(k)})}{f'(x^{(k)})},$$

a une convergence quadratique si $f'(\zeta) = 0$. Implémenter cette méthode dans une fonction `modnewton`. Appliquer la sur la fonction g et observer combien d'itérations sont nécessaires pour qu'elle fournisse une approximation de ζ avec une tolérance égale à 10^{-10} pour le critère d'arrêt.

2. On considère à présent la fonction $g(x) = x + e^{-20\,x^2}\cos(x)$, dont on veut approcher les zéros par la méthode de Newton–Raphson.

   a. Définir une première fonction Scilab pour g, puis une seconde pour sa dérivée g'.

   b. Utiliser la fonction `newton` pour essayer d'approcher d'un zéro de g en prenant $x^{(0)} = 0$ pour initialisation et une tolérance égale à 10^{-10} pour le critère d'arrêt.

   c. Tracer le graphe de g sur l'intervalle $[-1, 1]$ et tenter de donner une explication qualitative du fait la méthode de Newton–Raphson ne converge pas avec l'initialisation précédente.

   d. Appliquer la méthode de dichotomie à la fonction g sur l'intervalle $[-1, 1]$ pour la recherche d'un zéro de g.

3. Renommer et modifier la fonction `dichotomie` pour obtenir, sur le même modèle, une fonction `regulafalsi` implémentant la méthode de la fausse position [1]. De la même manière, écrire une fonction qui implémente la méthode de la sécante [2] à partir de la fonction `newton`.

Exercice 2 (calcul de $\sqrt{2}$). Dans cet exercice, on cherche à calculer une approximation de $\sqrt{2}$ de diverses façons.

1. On peut tout d'abord obtenir une valeur approchée de $\sqrt{2}$ en cherchant la racine positive du polynôme $f(x) = x^2 - 2$. Pour cela, appliquer successivement à f les méthodes de dichotomie, de la fausse position, de Newton–Raphson et de la sécante sur l'intervalle $[1, 2]$ et déterminer celles qui convergent.

---

1. On rappelle que, pour cette méthode, l'approximation du zéro à l'itération k $x^{(k)}$ est donnée par l'abscisse de l'intersection de la droite passant par les points $(a^{(k)}, f(a^{(k)}))$ et $(b^{(k)}, f(b^{(k)}))$ avec l'axe des abscisses, c'est-à-dire

$$x^{(k)} = a^{(k)} - \frac{a^{(k)} - b^{(k)}}{f(a^{(k)}) - f(b^{(k)})} f(a^{(k)}).$$

2. On rappelle que la méthode de la sécante peut être obtenue à partir de la méthode de Newton–Raphson en remplaçant la quantité $f'\left(x^{(k)}\right)$ apparaissant dans la relation de récurrence par le quotient $\frac{f\left(x^{(k)}\right)-f\left(x^{(k-1)}\right)}{x^{(k)}-x^{(k)-1}}$. L'initialisation de cette méthode nécessite donc deux points distincts, $x^{(-1)}$ et $x^{(0)}$, si possible proches du zéro recherché.

2. On peut également se servir de méthodes de point fixe, définies à partir des applications suivantes
$$g_1(x) = 2 + x - x^2,\ g_2(x) = \frac{2}{x} \text{ et } g_3(x) = \frac{x+2}{x+1},$$
considérées sur l'intervalle $[1, 2]$.
   a. Parmi les trois fonctions ci-dessus, lesquelles conduisent à une méthode de point fixe convergente ?
   b. Vérifier ces affirmations en calculant les 20 premiers termes des suites définies par les relations de récurrence
$$x^{(0)} = \frac{1}{2},\ x^{(k+1)} = g_i(x^{(k)}),\ k \in \mathbb{N},\ i \in \{1, 2, 3\}.$$

Exercice 3 (ordres de convergence). On considère les fonctions
$$f(x) = 3\cos(x) - 2\ln(x+1) - 1,\ g(x) = 2x - 1,\ h(x) = (2x-1)^3 \text{ et } k(x) = (2x-1)^5.$$

1. Calculer les dérivées de chacune de ces fonctions.
2. Comparer le nombre d'itérations nécessaires aux méthodes de dichotomie, de la fausse position, de Newton–Raphson et de la sécante pour obtenir le zéro, compris entre 0 et 1, de ces fonctions, en prenant une tolérance égale à 10^{-5} pour les critères d'arrêt des méthodes.
3. Pour chacune des fonctions f, g et h et pour chacune des méthodes utilisées dans la question précédente, représenter graphiquement les premiers termes de la suite $\left(\frac{\ln|x^{(k+1)}-\xi|}{\ln|x^{(k)}-\xi|}\right)_{k\in\mathbb{N}}$ en fonction de k, où ξ est le zéro de la fonction considérée[3] et $\left(x^{(k)}\right)_{k\in\mathbb{N}}$ est la suite des approximations de ξ produites par la méthode considérée. Déterminer à partir de ces graphes les ordres de convergence respectifs des méthodes.

Exercice 4 (bassins de convergence de la méthode de Newton–Raphson). On s'intéresse à la recherche des solutions complexes de l'équation $z^3 = 1$ par la méthode de Newton–Raphson. On considère pour cela la fonction d'une variable complexe $f(z) = z^3 - 1$, qui s'annule en chaque point z du plan complexe tel que $z^3 = 1$.

1. Définir deux fonctions Scilab `f` et `df` renvoyant respectivement les valeurs de f et de f' en un point quelconque de $\mathbb{C}$.
2. Pour tout entier $n \geq 2$, on définit une grille de pas $h = \frac{3}{n-1}$ couvrant le carré $[-1,5, 1,5] \times [-1,5\mathrm{i}, 1,5\mathrm{i}]$. Écrire un programme résolvant, pour une valeur donnée de n, l'équation $f(z) = 0$ avec une tolérance égale à 10^{-4} par la méthode de Newton–Raphson[4] initialisée successivement en chaque point de la grille $z_{ij} = -1,5(1+\mathrm{i}) + (i + j\mathrm{i})h$, $0 \leq i, j \leq n$. Pour chaque couple (i, j), stocker dans le tableau à deux dimensions `nrac` le numéro k ($k = 1$, 2 ou 3) de la racine cubique complexe de l'unité $e^{i\frac{2k\pi}{3}}$ vers laquelle la méthode aura convergée à partir de z_{ij} (en posant $k = 4$ lorsque la méthode n'a pas convergé après `nmax=100` itérations) et dans le tableau `niter` le nombre d'itérations nécessaires pour atteindre la convergence (en stockant le nombre maximal d'itérations autorisées `nmax` en l'absence de convergence).
3. Afficher une représentation de chacun des tableaux `nrac` et `niter` obtenus pour $n = 100$.
4. Refaire des tracés pour des pas de grille plus petits (*i.e.*, de plus grandes valeurs de n). Que dire des « frontières » des trois bassins de convergence de la méthode ?

3. On pourra éventuellement obtenir une approximation de ξ en utilisant la commande `fsolve`.
4. On pourra pour cela utiliser la fonction `newton` écrite à l'exercice 1.