Recherche de zéros de fonctions

Le but de ce TP est le suivant : étant donnée une fonction $f : [a,b] \longrightarrow \mathbf{R}$, on cherche un point $x \in]a,b[$ tel que $f(x) = 0$, qu'on appelle un zéro ou une racine de f. Plus exactement, on cherchera une valeur approchée de x (puisque Scilab ne sait pas faire de calculs exacts, et nous non plus en général...).

Une condition suffisante pour que l'intervalle $[a,b]$ contienne un zéro de f est que f soit continue et vérifie $f(a)\cdot f(b) < 0$ (pourquoi ?). On essaiera donc généralement de se placer dans ce cas de figure.

Toutes les fonctions `Scilab` que vous aurez à écrire prennent la fonction f comme paramètre. Nous pourrons les utiliser avec les fonctions `Scilab` usuelles, par exemple `exp()` ou `cos()`, ou définir de nouvelles fonctions, grâce à la commande `deff`. Vous pouvez commencer en testant l'exemple suivant :

```
deff('[y]=f1(x)','y=x-exp(-x)')
f1(1)
X=[-3:0.05:3]
plot2d(X,f1(X))
```

Les méthodes mises en œuvre reposent toutes sur la construction de suites convergentes vers un zéro. Un point crucial est de décider au bout de combien d'itérations on s'arrête. Il y a deux critères possibles dans ce cas : on peut s'arrêter quand la suite a l'air de converger, c'est à dire que deux termes consécutifs sont proches, à une précision donnée par l'utilisateur, ou bien quand la fonction f évaluée au point trouvée est assez proche de zéro, avec une tolérance fixée par l'utilisateur. On doit toujours aussi prévoir un nombre maximal d'itérations pour forcer la fonction à s'arrêter même quand l'algorithme ne converge pas.

Méthode de dichotomie. Écrire une fonction `dicho(f,a,b,tol,prec,nmax)` qui effectue la dichotomie sur la fonction `f` avec comme valeurs initiales `a` et `b`. La fonction `dicho` renvoie le vecteur `[n,an,bn]` et s'arrête :
— au bout de `nmax` itérations ;
— dès que `|bn-an|<prec` ;
— dès que `|f((an+bn)/2)|<tol`.

Tester sur

(i) $f : x \longmapsto x - \exp(-x)$

(ii) $f : x \longmapsto x^2 - 2$

(en choisissant bien l'intervalle $[a,b]$).

Méthode de la corde. Soit $a < x_0 < b$. On définit la suite

$$x_{k+1} = x_k - \frac{b-a}{f(b)-f(a)} f(x_k)$$

Faire un dessin et donner une interprétation géométrique.
Écrire une fonction `corde(f,a,b,x0,prec,nmax)` qui effectue la méthode de la corde sur la fonction `f`. La fonction `corde` renvoie le vecteur `[n,xn]` et s'arrête :
— au bout de `nmax` itérations ;
— dès que `|x(n+1)-xn|<prec`.

Méthode de la sécante. Soit x_0 et x_1 deux valeurs initiales. On définit la suite

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

Faire un dessin et donner une interprétation géométrique.

Écrire une fonction `secante(f,x1,x0,prec,nmax)` qui effectue la méthode de la sécante sur la fonction `f`. La fonction `secante` renvoie le vecteur `[n,xn]` et s'arrête :

- — au bout de `nmax` itérations ;
- — dès que `|x(n+1)-xn|<prec`.

Méthode de Newton. Soit $a < x_0 < b$. On définit la suite

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

Faire un dessin et donner une interprétation géométrique.

Écrire une fonction `newton(f,df,x0,prec,nmax)` qui effectue la méthode de Newton sur la fonction `f` : `df` contient la dérivée de `f`. La fonction `newton` renvoie le vecteur `[n,xn]` et s'arrête :

- — au bout de `nmax` itérations ;
- — dès que `|x(n+1)-xn|<prec`.

Le critère d'arrêt `|x(n+1)-xn|<prec` n'est pas forcément optimal. On lui préfère généralement `|f(xn)|<tol` où `tol` est la tolérance. Programmer une nouvelle fonction `newton2(f,df,x0,tol,nmax)`.

Test. On pose $f(x) = \cos^2(2x) - x^2$ sur l'intervalle $]0, 1.5[$, ayant pour racine $\alpha \approx 0.5149$. Pour toutes les méthodes, la précision est fixée à 10^{-10}, $x_0 = 0.75$ (si on a besoin de x_0) et $x1 = 1$ pour la méthode de la sécante. Comparer les vitesses de convergence des 4 méthodes (`dicho, corde, secante, newton`). Établir un classement.

En faisant varier `nmax` ou `prec`, donner pour chaque méthode une valeur expérimentale pour avoir une précision inférieure ou égale à 10^{-10}.