

## Méthode du point fixe

Le principe de la méthode du point fixe consiste à transformer, la fonction  $(x) = 0$ ,  $f : [a, b] \rightarrow R$ , en une fonction  $\varphi(x) = x$ . La fonction  $\varphi : [a, b] \rightarrow R$ , est construite de façon à ce que  $\varphi(\alpha) = \alpha$  quand  $(\alpha) = 0$ . Trouver la racine de  $(x)$ , se résume donc à déterminer un  $\alpha \in [a, b]$  tel que :

$$\alpha = \varphi(\alpha)$$

Dans le cas où un tel point existe, il sera qualifié de point fixe de  $\varphi$  et cette dernière est dite fonction d'itération. Le schéma numérique de cette méthode est donné par :

$$x^{(k+1)} = \varphi(x^{(k)}) \quad \text{pour } k \geq 0$$

On rappelle que le vecteur erreur  $e_n$  est calculé à partir de :  $e_n = |x_n - x_{app}|$ . Avec,  $x_{app}$  est la solution approchée, de la valeur exacte, déterminée avec une tolérance fixée préalablement.  $n$ , étant le nombre d'itérations. Par ailleurs, l'estimation de l'erreur servira, entre autre, à comparer la vitesse de convergence pour des méthodes numériques différentes. Sur le plan pratique, l'erreur est représentée graphiquement en traçant  $e_{n+1}$  en fonction de  $e_n$  avec une échelle logarithmique. Ainsi, l'ordre noté  $p$ , d'une méthode numérique s'obtient à partir de :

$$|e_{n+1}| \approx A |e_n|^p \implies \log |e_{n+1}| \approx p \log |e_n| + \log A$$

Ainsi l'ordre,  $p$ , est quantifié via la pente de l'équation ci-dessus. On en déduira que :

1. Si  $p = 1 \Rightarrow x_n$  converge linéairement vers la solution approchée. Dans ce cas on gagne la même quantité de précision à chaque itération.
2. Si  $p = 2 \Rightarrow x_n$  converge quadratiquement vers la solution approchée. Dans ce cas on gagne le double de précision à chaque itération.
3. Si  $p = 3 \Rightarrow x_n$  converge cubiquement vers la solution approchée. Dans ce cas on gagne le triple de précision à chaque itération.

D'un point de vue pratique, et pour un  $n$  suffisamment élevé, la vitesse de convergence d'une méthode itérative est évaluée au moyen de la relation :

$$K_p(x, n) = \frac{x_{n+2} - x_{n+1}}{(x_{n+1} - x_n)^p}$$

Tenant compte de cette équation, il vient que plus  $K_p(x, n)$  tend vers zéro plus la vitesse de convergence de la méthode est élevée.

### Exercice 01 : Dans cet exercice

Dans cet exercice, il est demandé de trouver la racine de la fonction  $f_1(x) = x - \cos(x)$ , en utilisant la méthode du point fixe.

1. Tracer la fonction  $f_1(x)$  sur l'intervalle  $[-1/2 \quad 3]$ .
2. Écrire un programme Matlab permettant l'implémentation du schéma numérique de cette méthode.
3. Afficher, sur le même graphe, la fonction  $f_1(x)$  et la solution approchée.
4. Afficher le graphe représentant le nombre d'approximations successives  $(k+1)$  en fonction du nombre d'itérations.
5. Tracer l'évolution de l'erreur en fonction du nombre d'itérations.
6. Tracer l'erreur  $\ln|e_{n+1}|$  en fonction de  $\ln|e_n|$  et déterminer l'ordre de la méthode numérique. Calculer la vitesse de convergence.

Appliquez le même algorithme pour résoudre l'équation :  $f_2(x) = x + \exp(x) + 1$  avec :  $x \in [-2 \quad 3/2]$ .

**On donne :** tolérance =  $10^{-6}$ , les valeurs initiales sont  $x_0 = 0.8$  pour  $f_1(x)$  et  $x_0 = -1/5$  pour  $f_2(x)$ .



```

Color','k','LineWidth',1) ; axis([0 3 -1 1.5])
37
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AFFICHAGE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 text('Interpreter','latex',...
40 'String','$ x = \cos(x) $','Position',[2 -1/3],'FontSize',15)
41 text('Interpreter','latex','String','$ y = x $',...
42 'Position',[1.2 1],'FontSize',15) ; text(sol,2*sol,['\alpha = ',
    num2str(sol)],...
43 'Position',[0.8 -1/4],'BackgroundColor',[1 1 1]);
44
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CALCUL D'ERRRUR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 err = abs(xn - sol); error_1 = err(1:end-1) ; error_2 = err(2: end);
47
48 figure('color', [1 1 1]) ; loglog(error_1(1:end-4),error_2(1:end-4),'
    +')
49 ylabel('Ln |e_{n+1}|') ; xlabel('Ln |e_n|')
50
51 cutoff = 7; % ATTENTION AU CHOIX DE LA NIEME ITERATION
52
53 pente = (log(error_2(end-cutoff)) - log(error_2(end-(cutoff+1))))/(
    log(error_1(end-cutoff)) - log(error_1(end-(cutoff+1)))); %
    VITESSE DE CONVERGENCE log|e_{n+1}| = log|e_n|
54 pente = round(pente);
55
56 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VITESSE DE CONVERGENCE (CV) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
57 n = 20; % ATTENTION AU CHOIX DE LA NIEME ITERATION
58 vitesse_CV = (xn(n+2) - xn(n+1))/(xn(n+1) - xn(n));
59
60 msg1 = gtext(strcat('CETTE METHODE EST D'ORDRE : ', num2str(pente)))
    ;% CLIQUER SUR LA FIGURE POUR AFFICHER msg1
61
62 msg2 = gtext(strcat('LA VITESSE DE CONVERGENCE VAUT : ', num2str(
    vitesse_CV)));
63 % CLIQUER SUR LA FIGURE POUR AFFICHER msg2

```