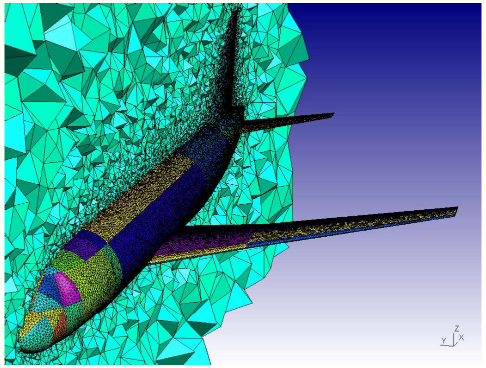
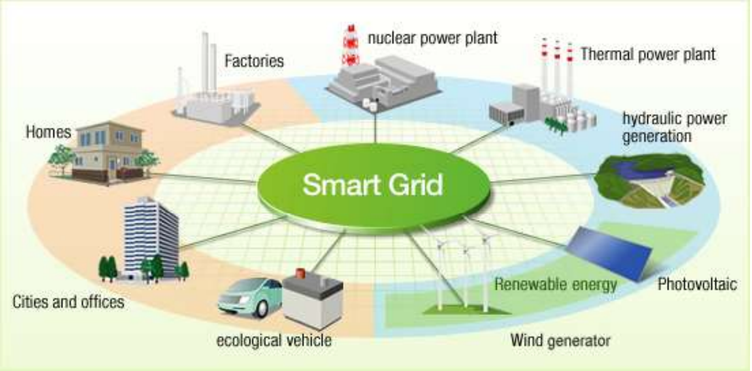
Méthodes Numériques





1 Ch1. Résolution des équations non linéaires

2 Ch2. Interpolation polynômiale

3 Ch3. Dérivation et intégration numériques

4 Ch4. Schémas numériques pour les EDO

5 Ch5. Analyse des erreurs

6 Ch6. Méthode des différences finies

- **Position du problème** :

  Soit $f \in C[a,b]$ une fonction donnée.
  Trouver une ou plusieures solutions de l'équation $f(x) = 0$.

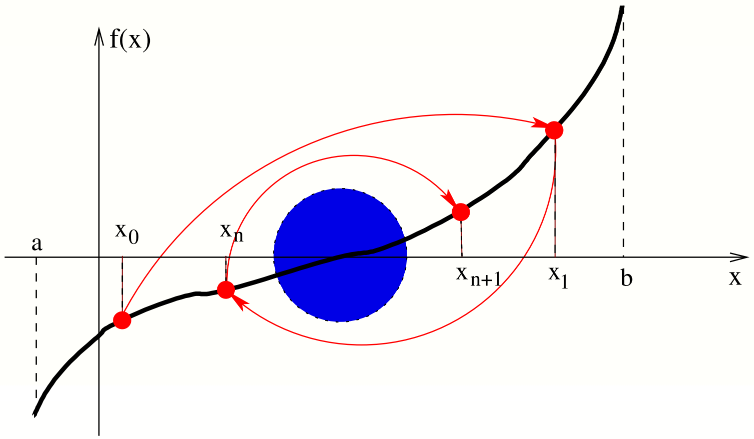
- **Motivation** : *Il n'existe pas de solution analytique.* $\Longrightarrow$ *Approximation numérique !*

- **Principe des approximation successives** :

  Construire une suite $(x_n)_{n\in\mathbb{N}}$

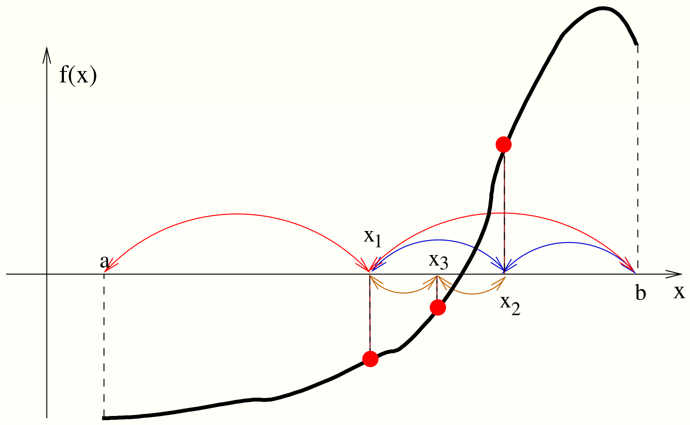
  tel que $x_n \xrightarrow[n\to\infty]{} x^*$,

  où x^* *est une racine de* $f(x^*) = 0$.



§1.1. Méthode de Dichotomie

(1) **Idée de la méthode** : Théorème de valeur intermédiaire



- Questions à étudier :
  - *x_n converge ?*
  - *si $x_n \to x^*$, x^* est une racine ?*
  - *Comment arrêter les itérations afin de trouver une valeur approchée de x^* à une précision près ?*

(2) Algorithme de Dichotomie :

Entrées : $\begin{cases} \text{Bornes } a \text{ et } b \text{ de l'intervalle ; Tolérance } \varepsilon \text{ ;} \\ \text{Nombre maximum d'itérations } Nmax. \end{cases}$

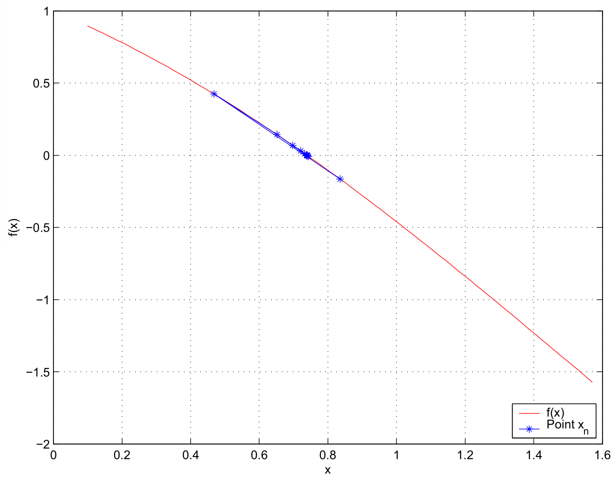
Sorties : la solution approchée x ou un message d'erreur

Algorithme :

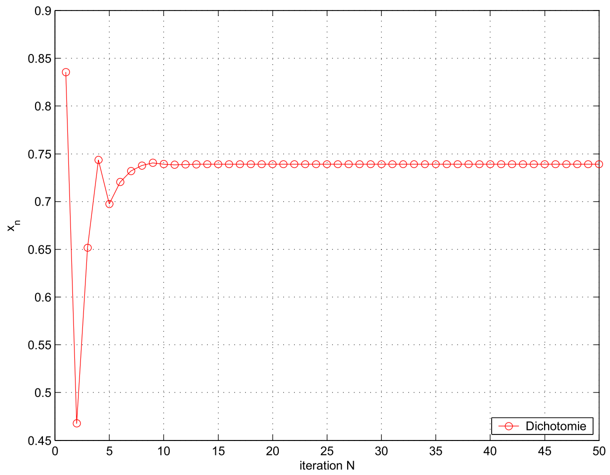
1. k=1
2. Tant que $k \leq Nmax$, répéter les étapes 3 à 6 :
   3. Calculer $x = (b + a)/2$.
   4. Si le test d'arrêt est réalisé (ex. $|f(x)| < \varepsilon$ ou $|b - a|/2 < \varepsilon$), alors retourner x et arrêter l'algorithme.
   5. $k = k + 1$.
   6. Si $f(a) \cdot f(x) > 0$, alors $a = x$;
      sinon, $b = x$.
7. Afficher un message afin de signaler que le nombre maximum d'itérations atteint. Arrêter.

(3) Exemples d'application :

- *Exemple 1* : $\boxed{\cos x - x = 0}$
  - l'intervalle $[a, b] = [0, \pi/2]$
  - Résultats de l'algorithme de Dichotomie :



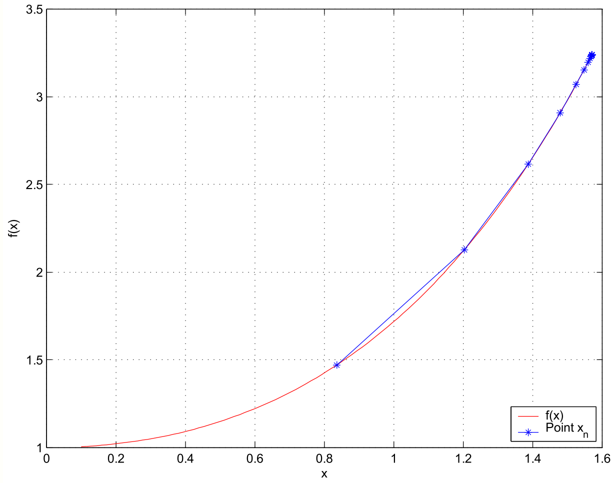
Position des points $(x_n, f(x_n))$.



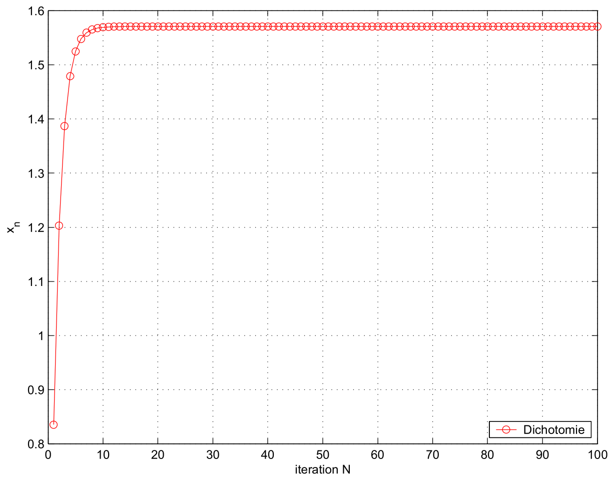
Comportement de x_n.

  - Vérification : $f(x^*) \simeq 0$?

- *Exemple 2* : $\boxed{e^x - x = 0}$
  - l'intervalle $[a, b] = [0, \pi/2]$
  - Résultats de l'algorithme de Dichotomie :



Position des points $(x_n, f(x_n))$.



Comportement de x_n.

  - Vérification : $f(x^*) \simeq 0$?
  - *Remarques : Pourquoi cet échec ?*

(4) Problème sur la convergence et sur les erreurs :

Théorème 1.1 (de convergence)

Soit $f \in C[a,b]$ une fonction donnée telle que $f(a) \cdot f(b) < 0$. L'algorithme de Dichotomie génère une suite $(x_n)_{n\in\mathbb{N}}$ qui converge vers une racine x^* de f avec l'erreur à n-ième itération donné par

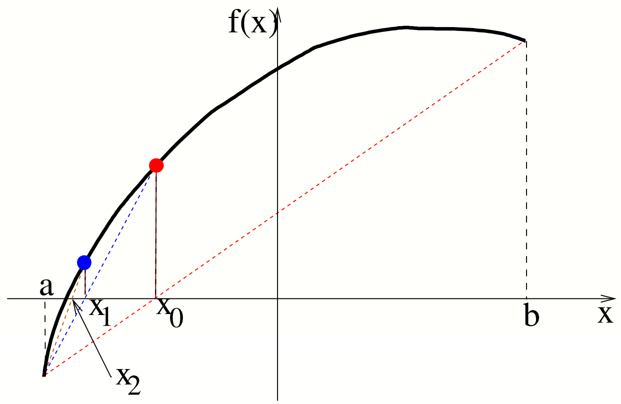
$$\left|x_n - x^*\right| \leq \frac{b-a}{2^n} \qquad \textit{pour tout } n \geq 1.$$

(5) Remarques :

- "$f(a) \cdot f(b) < 0$" restreint le domaine d'application.
- La vitesse de convergence : pas très rapide.
- Le choix de test d'arrêt non unique :
  - ⋆ $|x_{n+1} - x_n| < \varepsilon$;
  - ⋆ $|f(x_n)| < \varepsilon$;
  - ⋆ $\frac{|x_{n+1}-x_n|}{|x_{n+1}|} < \varepsilon$.
- Pouvez-vous "inventer" des autres méthodes ?

On trace la droite passant par les points $(a, f(a))$ et $(b, f(b))$ au lieu de couper au milieu ?

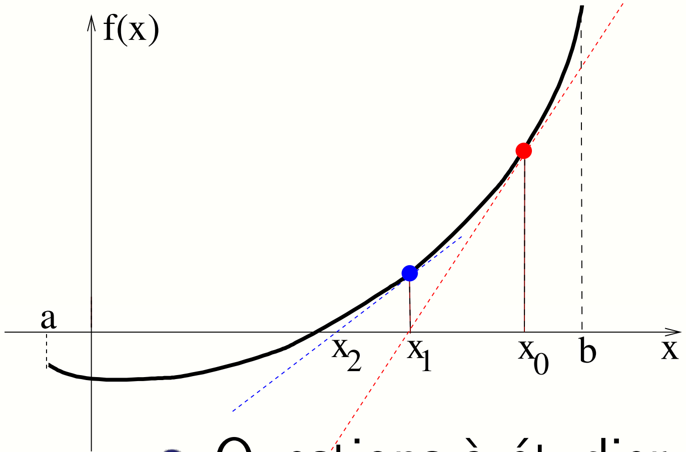
$\Longrightarrow$ **Méthode de Lagrange**.



§1.2. Méthode de Newton

(1) Idée de la méthode :

- Point de vue géométrique.



- Point de vue algèbrique.

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \circ(|x - x_0|).$$

Si $|x^* - x_0|$ est assez petit, on a

$$0 = f(x^*) \simeq f(x_0) + f'(x_0)(x^* - x_0).$$

D'où

$$x^* \simeq x_0 - \frac{f(x_0)}{f'(x_0)}.$$

- Questions à étudier :
  - *x_n converge vers une racine x^* ?*
  - *estimation des erreurs $|x_n - x^*|$?*

(2) Algorithme de Newton :

Entrées : $\begin{cases} \text{Approximation initiale } x_0 \text{ ; Tolérance } \varepsilon \text{ ;} \\ \text{Nombre maximum d'itérations } Nmax. \end{cases}$

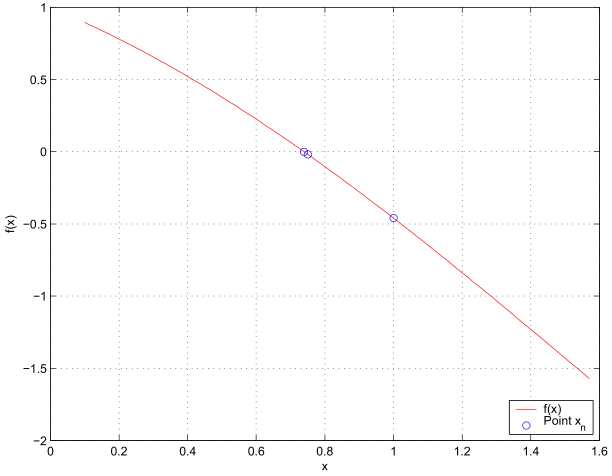
Sorties : la solution approchée x ou un message d'erreur

Algorithme :

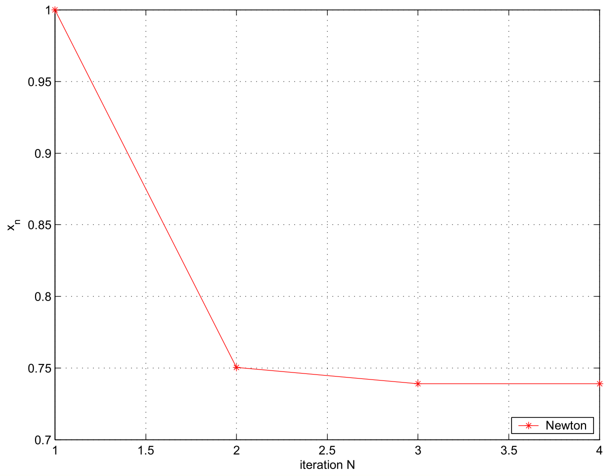
1. k=1
2. Tant que $k \leq Nmax$, répéter les étapes 3 à 6 :
    3. Calculer $x = x_0 - f(x_0)/f'(x_0)$.
    4. Si le test d'arrêt est réalisé (ex. $|x - x_0| < \varepsilon$), alors retourner x et arrêter l'algorithme.
    5. $k = k + 1$.
    6. Préparer l'itération prochaine : $x_0 = x$
7. Afficher un message afin de signaler que le nombre maximum d'itérations atteint. Arrêter.

(3) Exemples d'application :

- *Exemple 1* : $\boxed{\cos x - x = 0}$
  - L'approximation initiale $x_0 = 1$.
  - Résultats de l'algorithme de Newton :



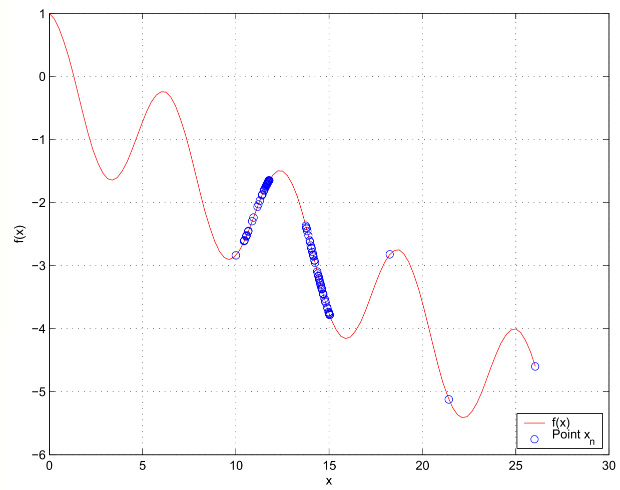
Position des points $(x_n, f(x_n))$.



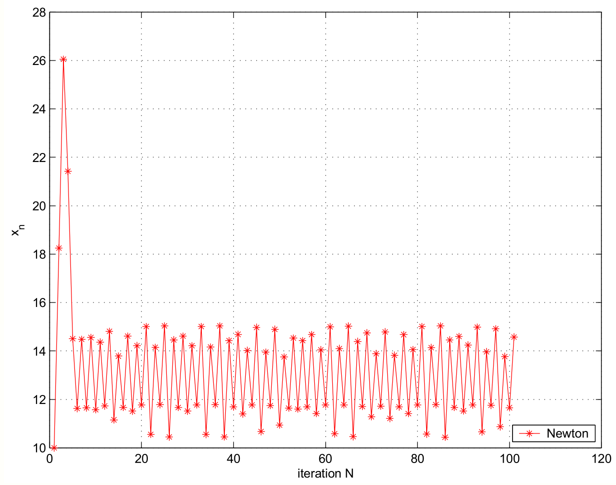
Comportement de x_n.

  - Vérification : $f(x^*) \simeq 0$?

- *Exemple 2* : $\boxed{\cos x - 0.2x = 0}$
  - L'approximation initiale $x_0 = 10$
  - Résultats de l'algorithme de Newton :



Position des points $(x_n, f(x_n))$.



Comportement de x_n.

  - Vérification : $f(x^*) \simeq 0$?
  - *Remarques : Pourquoi cet échec ?*

(4) Problème sur la convergence et sur les erreurs :

Théorème 1.2 (de convergence)

Soit $f \in C^2[a,b]$. On suppose que f admet une racine x^* et que $f'(x^*) \neq 0$. Alors, il existe $\delta > 0$, tel que, pour tous $x_0 \in [x^* - \delta, x^* + \delta]$, on a $(x_n)_{n\in\mathbb{N}}$, la suite générée par la méthode de Newton avec l'approximation initiale x_0, converge vers la racine x^* de f.

Remarques :

- Pourquoi $f \in C^2[a,b]$? Pourquoi $f'(x^*) \neq 0$?
- Comment choisir l'approximation initiale x_0 en pratique?
- Estimation des erreurs : $|x_n - x^*| \leq \frac{1}{M}\big(M|x_0 - x^*|\big)^{2^n}$,

  où $M = \displaystyle\max_{x\in[x^*-\delta, x^*+\delta]} \left|\frac{f''(x)}{f'(x)}\right|$.

(5) Remarques :

- Avantage : converge assez rapide.
- Inconvenient : choix de l'approximation initiale x_0.
- Inconvenient : calculer $f'(x_n)$ à chaque itération.
- Une variante : **Méthode de la sécante**
  - — Deux approximations initiales x_0 et x_1 à choisir.
  - — $f'(x_n)$ est remplacé par $\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$.

Remarques :

- Que faire pour un système des équations non linéaires ?

$$\left\{ \begin{array}{rcl} f_1(x_1, x_2, \cdots, x_n) & = & 0, \\ f_2(x_1, x_2, \cdots, x_n) & = & 0, \\ \vdots & & \vdots \\ f_n(x_1, x_2, \cdots, x_n) & = & 0. \end{array} \right.$$

- Ecriture en formulation vectorielle $\mathbf{F}(\mathbf{X}) = \mathbf{0}$, où
  $\mathbf{X} = (x_1, x_2, \cdots, x_n)^t \in \mathbb{R}^n$ et
  $\mathbf{F} = (f_1, f_2, \cdots, f_n)^t : \mathbb{R}^n \longrightarrow \mathbb{R}^n$.

Remarques :

- On reprend l'idée de la méthode de Newton
  Dévéloppement limité pour fonctions vectorielle :
  $$\mathbf{F}(\mathbf{X}) = \mathbf{F}(\mathbf{X}^{(0)}) + J_F(\mathbf{X}^{(0)})(\mathbf{X} - \mathbf{X}^{(0)}) + \circ(\|\mathbf{X} - \mathbf{X}^{(0)}\|),$$
  où $J_F(\mathbf{X}) = \left(\frac{\partial f_i}{\partial x_j}\right)_{i,j=1,\cdots,n}$ la matrice Jacobian de $\mathbf{F}$.

  Si $\|\mathbf{X}^* - \mathbf{X}^{(0)}\|$ est assez petit, on a
  $$\mathbf{0} = \mathbf{F}(\mathbf{X}^*) \simeq \mathbf{F}(\mathbf{X}^{(0)}) + J_F(\mathbf{X}^{(0)})(\mathbf{X}^* - \mathbf{X}^{(0)}).$$
  D'où
  $$\mathbf{X}^* \simeq \mathbf{X}^{(0)} - (J_F(\mathbf{X}^{(0)}))^{-1}\mathbf{F}(\mathbf{X}^{(0)}).$$
- D'où la méthodes de Newton (version vectorielle) :
  $$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} - (J_F(\mathbf{X}^{(n)}))^{-1}\mathbf{F}(\mathbf{X}^{(n)})$$

§1.3. Méthode du point fixe

(1) Idée de la méthode :

- Généralisation de l'algo. de Newton

$$x_{n+1} = \boxed{x_n - \frac{f(x_n)}{f'(x_n)}} \overset{\text{noté}}{=} g(x_n).$$

- Peut-on utiliser d'autre fonction g ?

$\Longrightarrow$ Propriétés de g ?

$\Longrightarrow$ l'équivalence entre $\boxed{f(x) = 0}$ et $\boxed{x = g(x)}$?

- Questions à étudier :
  - x_n *converge vers une racine* x^* *?*
  - *estimation des erreurs* $|x_n - x^*|$ *?*

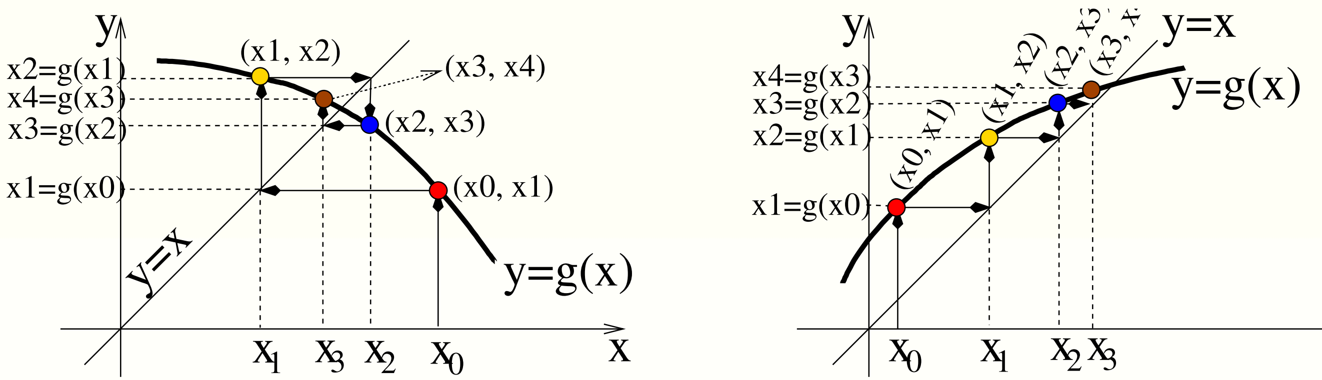
(2) **Algorithme du point fixe** :

Entrées : $\begin{cases} \text{Approximation initiale } x_0 \text{ ; Tolérance } \varepsilon \text{ ;} \\ \text{Nombre maximum d'itérations } Nmax. \end{cases}$

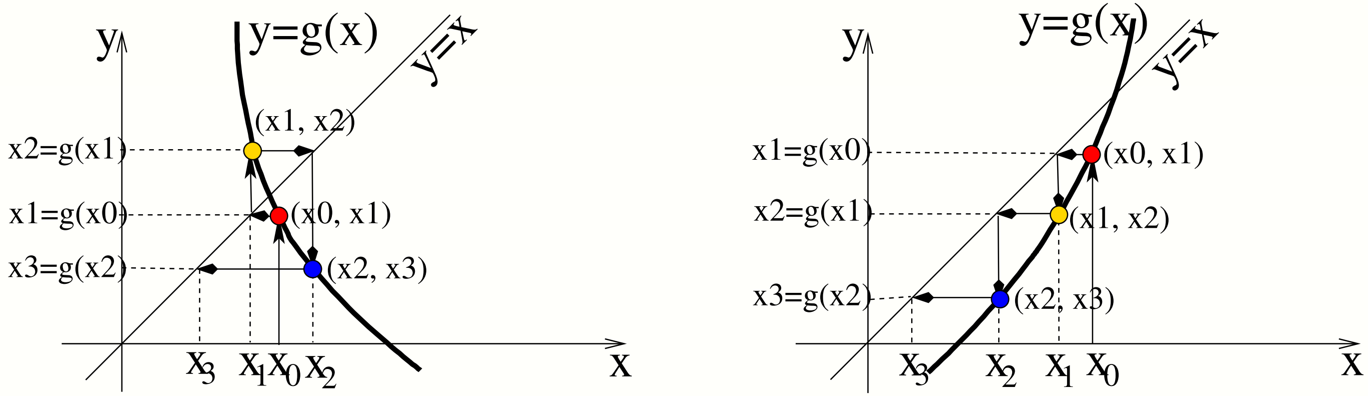
Sorties : la solution approchée x ou un message d'erreur

Algorithme :

1. k=1
2. Tant que $k \leq Nmax$, répéter les étapes 3 à 6 :
   3. Calculer $x = g(x_0)$.
   4. Si le test d'arrêt est réalisé (ex. $|x - x_0| < \varepsilon$), alors retourner x et arrêter l'algorithme.
   5. $k = k + 1$.
   6. Préparer l'itération prochaine : $x_0 = x$
7. Afficher un message afin de signaler que le nombre maximum d'itérations atteint. Arrêter.



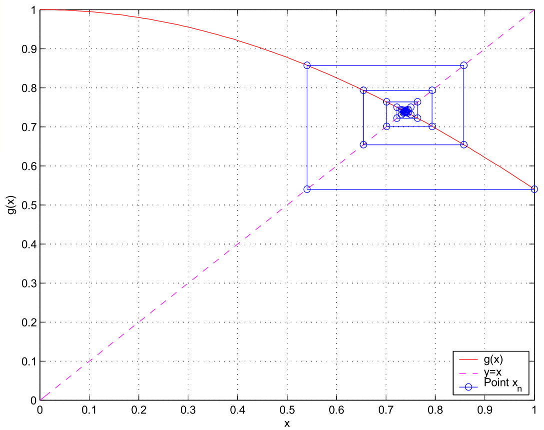
La méthode du point fixe marche bien pour certains cas,



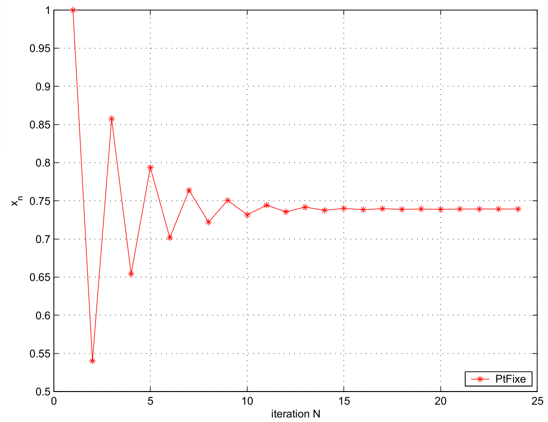
et ne marche pas du tout pour des autres cas.

(3) Exemples d'application :

- *Exemple 1* : $\boxed{\cos x - x = 0}$
  - Choisir la fonction g : ici, on prend $g(x) = \cos x$.
  - L'approximation initiale $x_0 = 1$
  - Résultats de l'algorithme du point fixe :



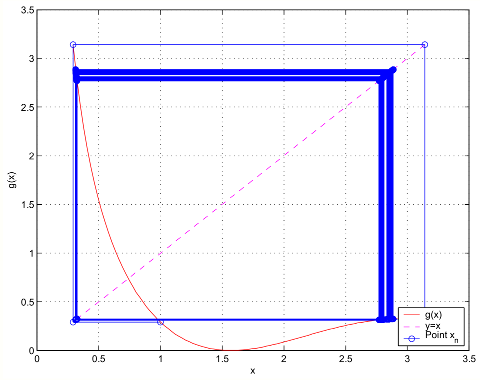
Position des points $(x_n, f(x_n))$.



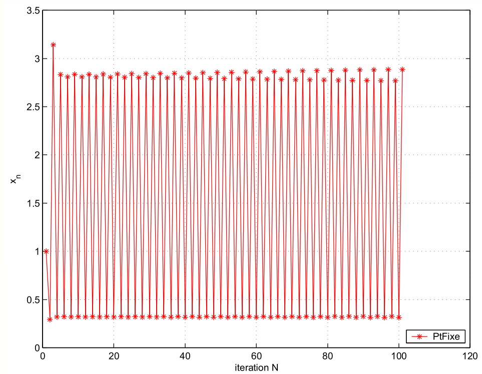
Comportement de x_n.

  - Vérification : $g(x^*) \simeq x^*$? (ou $f(x^*) \simeq 0$?)

- *Exemple 2* : même équation $\boxed{\cos x - x = 0}$
  - si on prend une autre formule équivalente $g(x) = \dfrac{\cos^2 x}{x}$
  - L'approximation initiale $x_0 = 1$.
  - Résultats de l'algorithme du point fixe :



Position des points $(x_n, f(x_n))$.



Comportement de x_n.

  - *Remarques :*
    - — Importance du choix de l'approximation initiale.
    - — Pourquoi cet échec ?
    - — Comment choisir la fonction g en pratique ?

(4) Problème sur la convergence et sur les erreurs :

Théorème 1.3 (du ponit fixe)

Soit g fonction définie sur l'intervalle $[a, b]$ qui vérifie les conditions :

(i) $x \in [a, b] \implies g(x) \in [a, b]$ *;*

(ii) g *est une fonction* **strictement contractante**, *i.e. il existe* $L \in \mathbb{R}$, $0 \leq L < 1$ *tel que,* $\forall x \in [a, b]$, $\forall y \in [a, b]$, *on ait*

$$|g(x) - g(y)| \leq L|x - y|.$$

Alors,

1. *il existe un point fixe unique* $x^* \in [a, b]$ *de* g, *i.e.* $x^* = g(x^*)$ *;*
2. *quel que soit* $x_0 \in [a, b]$, *la suite récurrente définie par* $x_{n+1} = g(x_n)$ *converge vers le point fixe* x^**;*
3. *on a l'estimation des erreurs suivante*

$$|x_n - x^*| \leq |x_1 - x_0| \frac{L^n}{1 - L}.$$

Proposition 1.4

Soit g fonction dérivable dans l'intervalle $[a,b]$. Si g' vérifie

$$\max_{x\in[a,b]} |g'(x)| = L < 1,$$

alors g est une fonction strictement contractante dans l'intervalle $[a,b]$.

Remarques :

- $\boxed{|g'(x)| < 1 \text{ pour tout } x \in [a,b]}$ *ne suffit pas.*
  Contre-exemple : $g(x) = x + \frac{1}{x}$ et $[a,b] = [1,+\infty[$. On a
  $$|g'(x)| < 1, \quad \text{mais} \quad \sup_{x\in[a,b]} |g'(x)| = 1.$$
  Et l'équation $x = g(x)$ n'a pas de point fixe dans $[1,+\infty[$.
- *Soit x^* une solution de $x = g(x)$. Si $g \in C^1$ et si $|g'(x^*)| < 1$, alors on peut toujours trouver un intervalle $[a,b]$ contenant x^* pour lequel l'algorithme du point fixe converge vers x^*.*

§1.4. Ordre de convergence

(1) **Motivation** :

- Comparaison de résultats obtenus par différentes méthodes.
  $\Longrightarrow$ En plus de la convergence, on veut la rapidité !

- Un exemple : $x_n = 0.1^{2n}$ et $\tilde{x}_n = 0.99^{2^n}$.

n	1	2	5	10	12	13
x_n	10^{-2}	10^{-4}	10^{-10}	10^{-20}	10^{-24}	10^{-26}
$\tilde{x}_n$	0.98	0.96	0.72	$3.4\ 10^{-5}$	$1.3\ 10^{-18}$	$1.8\ 10^{-36}$

- Comment caractériser la vitesse de convergence de la suite $(x_n)_{n\in\mathbb{N}}$ générée par un algorithme donné ?

(2) **Définition 1.5 (Ordre de convergence)**

Soit $(x_n)_{n\in\mathbb{N}}$ une suite qui converge vers x^*.
On suppose que $x_n \neq x^*$ pour tout $n \in \mathbb{N}$.
On dit que la suite $(x_n)_{n\in\mathbb{N}}$ converge vers x^* à l'ordre α, si l'il existe $\alpha > 0$ et $\lambda > 0$ tels que

$$\lim_{n\to+\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^\alpha} = \lambda.$$

Remarques :

- *cas $\alpha = 1$: la suite converge linéairement ;*
  cas $\alpha = 2$: la suite converge quadratiquement.

- *Pour tout $\varepsilon > 0$, on a*

$$\lim_{n\to+\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^{\alpha+\varepsilon}} = +\infty \quad \textit{et} \quad \lim_{n\to+\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^{\alpha-\varepsilon}} = 0.$$

(3) Ordre d'une méthode du point fixe :

- Supposer que la méthode définie par $x_{n+1} = g(x_n)$.
- On note $e_n = x_n - x^*$ l'erreur à l'itération n.

  Théorème de Taylor $\Longrightarrow$

$$
\begin{aligned}
e_{n+1} &= x_{n+1} - x^* = g(x_n) - g(x^*) \\
&= g'(x^*)(x_n - x^*) + \frac{g''(x^*)}{2!}(x_n - x^*)^2 + \cdots \\
&\qquad + \frac{g^{(p)}(x^*)}{p!}(x_n - x^*)^p + o(|(x_n - x^*)|^p) \\
&= g'(x^*)e_n + \frac{g''(x^*)}{2!}e_n^2 + \cdots + \frac{g^{(p)}(x^*)}{p!}e_n^p + o(|e_n|^p)
\end{aligned}
$$

- L'ordre de cette méthode (noté α)

$$
\lim_{n\to+\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^\alpha} = \lim_{n\to+\infty} \frac{|g'(x^*)e_n + \frac{g''(x^*)}{2!}e_n^2 + \cdots}{|e_n|^\alpha}
$$

  Donc, α depend les valeurs $g'(x^*)$, $g''(x^*), \cdots g^{(p)}(x^*)$!

Théorème 1.6

On considère une équation $f(x) = 0$.
Si x^* est une racine simple (i.e. $f(x^*) = 0$ et $f'(x^*) \neq 0$),
alors la méthode de Newton est au moins d'ordre 2 en partant d'une approximation initiale x_0 convenablement choisie.

Remarques :

- *Méthode de Dichotomie : pas de notion d'ordre de convergence.*
- *Méthode du point fixe : déjà vu (page précédente).*
- *On peut accélérer la convergence pour une méthode donnée ?*
  Par exemple, on a une méthode $x_{n+1} = g(x_n)$ (t.q. $x_n \to x^*$).
  Si on utilise $\tilde{x}_{n+1} = g(g(\tilde{x}_n))$,

$$\tilde{x}_n \to x^*? \qquad \tilde{x}_n \text{ converge plus vite que } x_n?$$

Remarques :

- **Méthode de Steffensen** *(accélération de la convergence) :*
  - *Soit une méthode itérative d'ordre 1, engendrée par g.*
  - *On peut définir formellement une fonction G par*

$$G(x) = \frac{xg\Big(g(x)\Big) - \Big(g(x)\Big)^2}{g\Big(g(x)\Big) - 2g(x) + x}.$$

La méthode itérative engendrée par G est d'ordre 2 au moins.