MMSN

Analyse numérique :

Chapitre 4 :

Méthodes numériques pour résoudre des équations non linéaires

Cours 12-13-14

GM 3 Année 2018 - 2019

Contact : A. Tonnoir

antoine.tonnoir@insa-rouen.fr

Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Remarque :

Soulignons que si $p > n$, a priori le problème n'admet pas de solution dans le cas général et on s'intéressera au problème de minimisation :

$$\text{Trouver } \min_{\underline{x} \in \mathbb{R}^n} \|\underline{F}(\underline{x})\|$$

Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

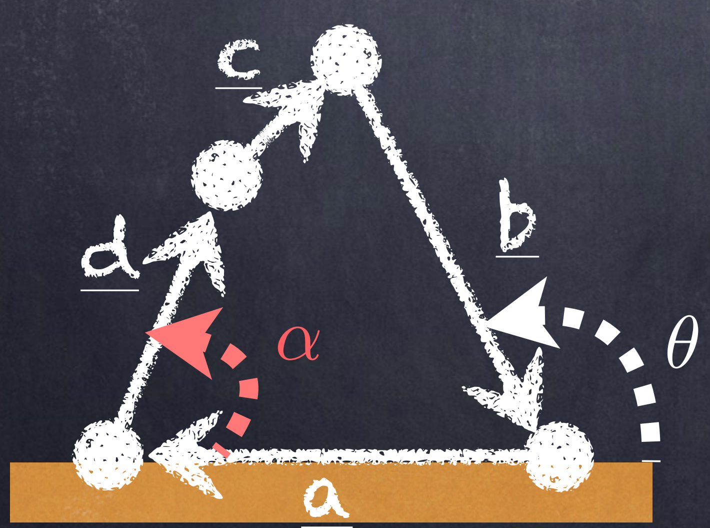
Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Exemples (calcul structure) :



Connaissant les longueurs de $\underline{a}$, $\underline{b}$, $\underline{c}$, $\underline{d}$, l'angle θ et l'égalité vectorielle :

$$\underline{a} + \underline{b} + \underline{c} + \underline{d} = \underline{0}$$

Déduire l'angle α

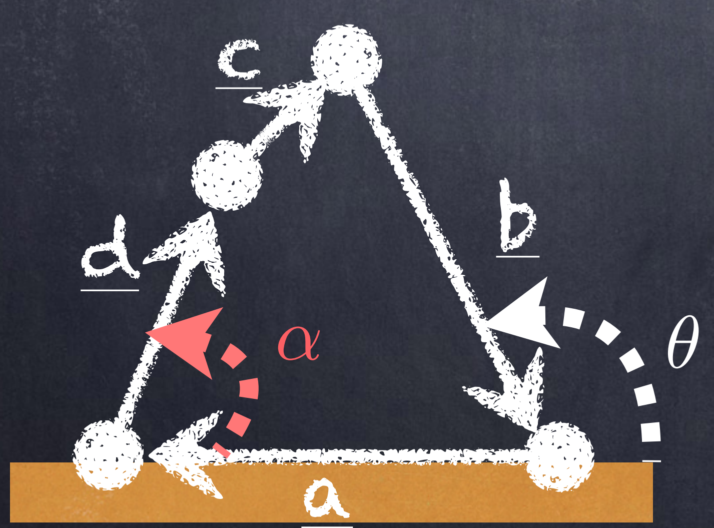
Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Exemples (calcul structure) :



Connaissant les longueurs de $\underline{a}$, $\underline{b}$, $\underline{c}$, $\underline{d}$, l'angle θ et l'égalité vectorielle :

$$\underline{a} + \underline{b} + \underline{c} + \underline{d} = \underline{0}$$

$$\Rightarrow \quad \underline{c} \cdot \underline{c} = (\underline{a} + \underline{b} + \underline{d}) \cdot (\underline{a} + \underline{b} + \underline{d})$$

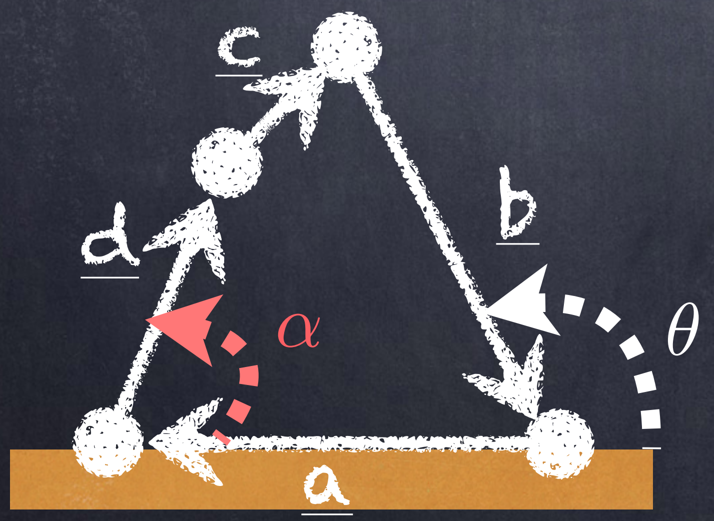
Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Exemples (calcul structure) :



Connaissant les longueurs de $\underline{a}$, $\underline{b}$, $\underline{c}$, $\underline{d}$, l'angle θ et l'égalité vectorielle :

$$\underline{a} + \underline{b} + \underline{c} + \underline{d} = \underline{0}$$

$$\Rightarrow \quad |\underline{c}|^2 = |\underline{a}|^2 + |\underline{b}|^2 + |\underline{d}|^2 - 2|\underline{a}||\underline{b}|\cos(\pi - \theta)$$
$$- 2|\underline{a}||\underline{d}|\cos(\alpha) - 2|\underline{b}||\underline{d}|\cos(\alpha - \theta)$$

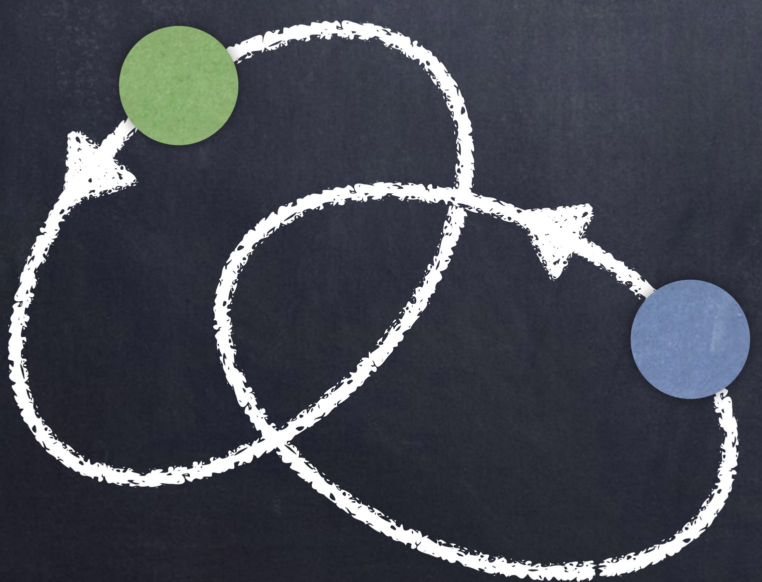
Introduction

Dans ce chapitre, on s'intéressera au problème non linéaire suivant :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

où a priori la fonction $\underline{F}$ peut être vectorielle, i.e. à valeurs dans $\mathbb{R}^p$

Exemples (intersection trajectoires) :



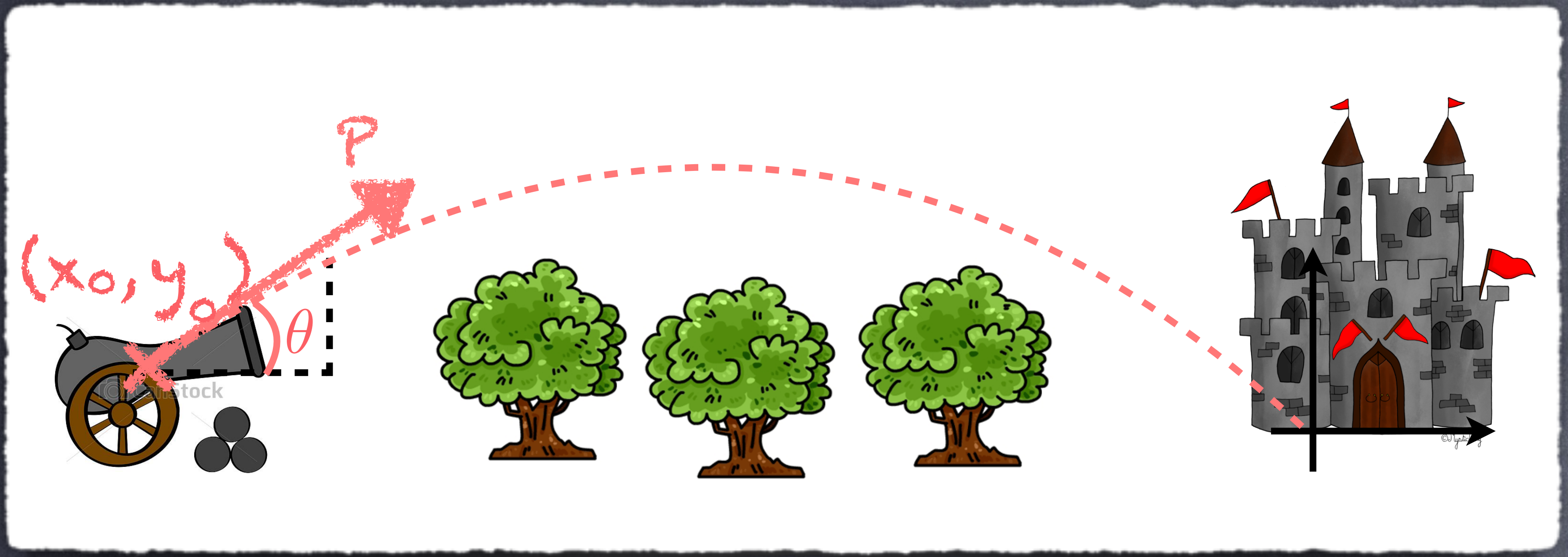
On souhaite déterminer l'instant de collision entre deux objets :

$$\underline{x}_1(t) = (R_1 \cos(\omega_1(t - t_0^1)),\ R_1 \sin(\omega_1(t - t_0^1)))$$

$$\underline{x}_2(t) = (R_2 \cos(\omega_2(t - t_0^2)),\ R_2 \sin(\omega_2(t - t_0^2))) + (x_2,\ y_2)$$

Introduction

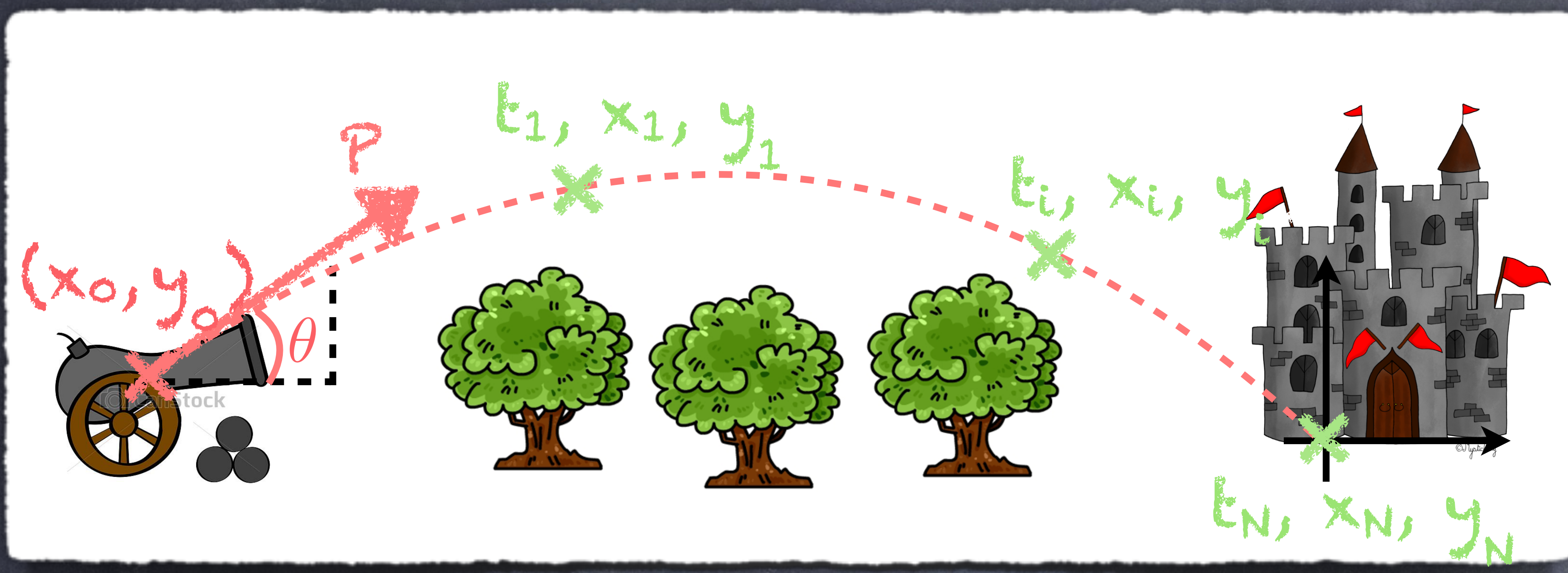
Exemples (identification paramètres) :



Paramètres inconnus : x_0, y_0, P, θ

Introduction

Exemples (identification paramètres) :



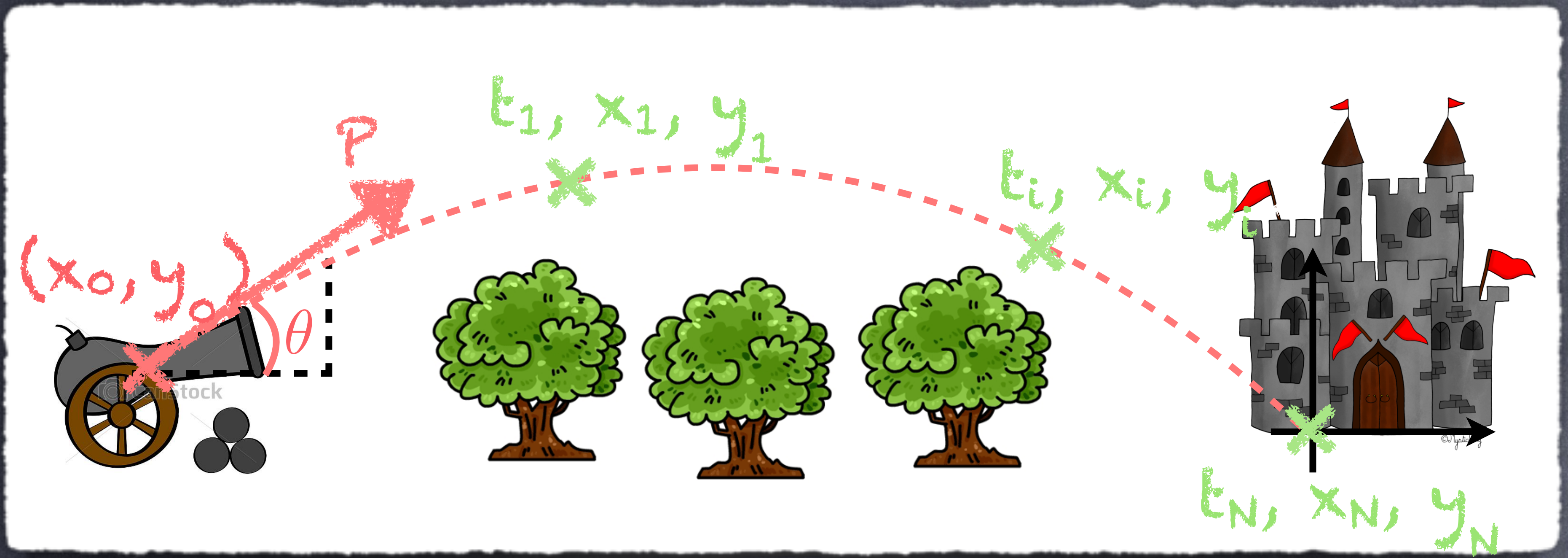
Paramètres inconnus : x_0, y_0, P, θ

Données : $(t_i, x_i, y_i)_{i=1,N}$

À l'aide des mesures $(t_i, x_i, y_i)_{i=1,N}$ on souhaite pouvoir identifier les paramètres : x_0, y_0, P, θ

Introduction

Exemples (identification paramètres) :



Paramètres inconnus : x_0, y_0, P, θ

Données : $\left(t_i, x_i, y_i\right)_{i=1,N}$

D'après le Principe Fondamentale de la dynamique, on a :

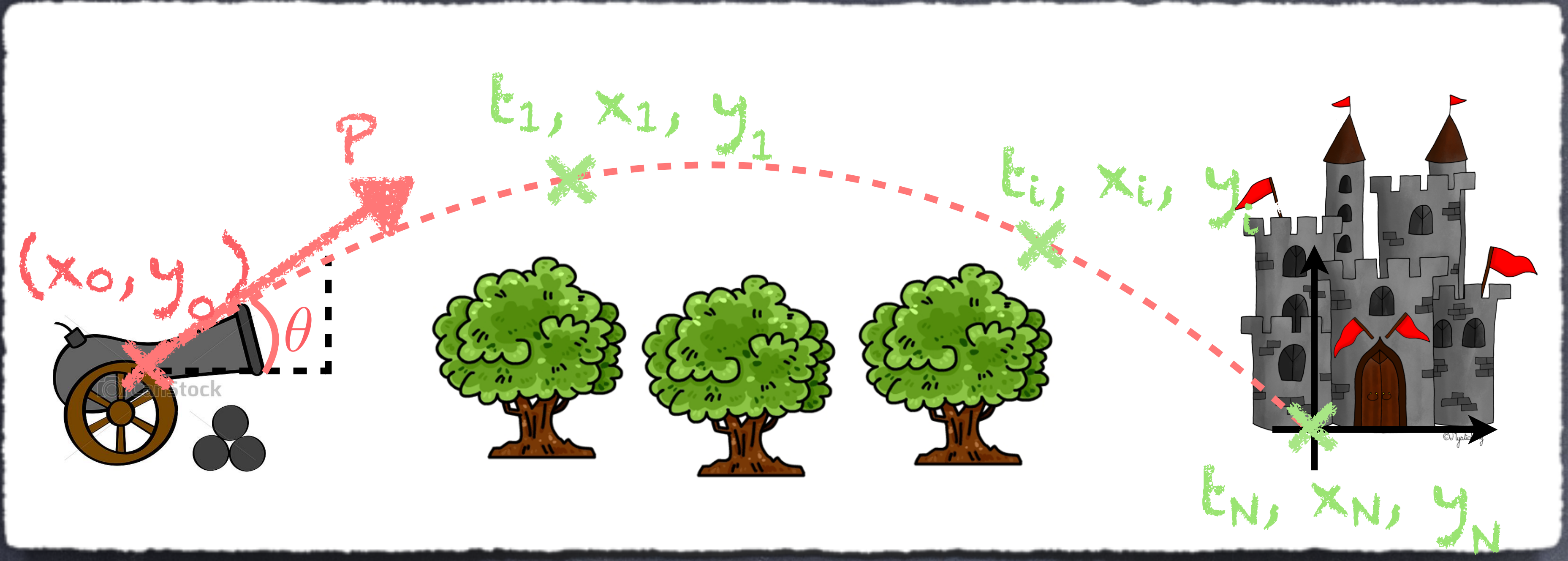
$$x(t) = P\cos(\theta)t + x_0 \quad \text{et} \quad y(t) = -\frac{g}{m}t^2 + P\sin(\theta)t + y_0$$

On cherche les « meilleurs » paramètres pour vérifier

$$\forall i \in [1, N], \quad \left(x(t_i), y(t_i)\right) \sim \left(x_i, y_i\right)$$

Introduction

Exemples (identification paramètres) :



Paramètres inconnus : $x_0, \ y_0, \ P, \ \theta$

Données : $\left(t_i, \ x_i, \ y_i\right)_{i=1,N}$

Idée : Chercher à minimiser la fonction

$$J(x_0, y_0, P, \theta) = \sum_{i=1}^{N} (x(t_i) - x_i)^2 + (y(t_i) - y_i)^2$$

Remarque :

J n'est rien d'autre que la norme 2 du vecteur formé par la différence entres les données et les « prédictions »

Au programme (Chapitre 4)

Objectif :

Étudier des **méthodes numériques** pour résoudre le **problème non linéaire :**

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

Plan :

I. Les équations non linéaires

II. Le cas des polynômes

Au programme (Chapitre 4)

Objectif :

Étudier des méthodes numériques pour résoudre le problème non linéaire :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

Plan :

I. Les équations non linéaires

- a) Les équations de point fixe
- b) La méthode de Newton
- c) La méthode de la fausse position

II. Le cas des polynômes

I. a) Les équations de point fixe

Considérons l'équation de point fixe suivante :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{G}(\underline{x}) = \underline{x}$$

où $G(\underline{x})$ est une fonction de $\mathbb{R}^n$ dans lui-même.

Remarque :

On peut facilement faire le lien entre l'équation de point fixe ci-dessus et le problème initial :

$$\underline{G}(\underline{x}) = \underline{x} \iff \underline{G}(\underline{x}) - \underline{x} = \underline{0}$$

où de manière plus général

$$\underline{G}(\underline{x}) = \underline{x} \iff \underline{H}(x)\left(\underline{G}(\underline{x}) - \underline{x}\right) = \underline{0}$$

où on suppose $\underline{H}(\underline{x})$ inversible.

I. a) Les équations de point fixe

Considérons l'équation de point fixe suivante :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{G}(\underline{x}) = \underline{x}$$

où $G(\underline{x})$ est une fonction de $\mathbb{R}^n$ dans lui-même.

1.1 Définition

On dit que l'application $\underline{G}$ est contractante ssi il existe une constante K t.q. $0 < K < 1$ et

$$\forall \left(\underline{x}, \underline{y}\right) \in \mathbb{R}^n, \quad \|G(\underline{x}) - G(\underline{y})\| \leq K\|\underline{x} - \underline{y}\|$$

I. a) Les équations de point fixe

Considérons l'équation de point fixe suivante :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{G}(\underline{x}) = \underline{x}$$

où $G(\underline{x})$ est une fonction de $\mathbb{R}^n$ dans lui-même.

1.1 Définition

On dit que l'application $\underline{G}$ est contractante ssi il existe une constante K t.q. $0 < K < 1$ et

$$\forall \left(\underline{x}, \underline{y}\right) \in \mathbb{R}^n, \quad \|G(\underline{x}) - G(\underline{y})\| \leq K\|\underline{x} - \underline{y}\|$$

1.2 Théorème (Point fixe)

L'application $\underline{G}$ admet un unique point fixe x^* et la suite $\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$ converge vers ce point fixe.

Preuve : au (vrai) tableau !

I. a) Les équations de point fixe

1.3 Définition

On dit que l'application $\underline{G}$ est différentiable au point $\underline{x}_0 \in \mathbb{R}^n$ s'il existe une application linéaire $d_{\underline{x}_0}\underline{G}$ vérifiant :

- $\forall \underline{h} \in \mathbb{R}^n, \quad \underline{G}(\underline{x}_0 + \underline{h}) = \underline{G}(\underline{x}_0) + d_{\underline{x}_0}\underline{G}(\underline{h}) + \underline{\Phi}(\underline{x}_0, \underline{h})$

- $\lim\limits_{\underline{h} \to \underline{0}} \dfrac{\|\underline{\Phi}(\underline{x}_0, \underline{h})\|}{\|\underline{h}\|} = 0$

I. a) Les équations de point fixe

1.3 Définition

On dit que l'application $\underline{G}$ est différentiable au point $\underline{x}_0 \in \mathbb{R}^n$ s'il existe une application linéaire $d_{\underline{x}_0}G$ vérifiant :

- $\forall \underline{h} \in \mathbb{R}^n, \quad \underline{G}(\underline{x}_0 + \underline{h}) = \underline{G}(\underline{x}_0) + d_{\underline{x}_0}G(\underline{h}) + \underline{\Phi}(\underline{x}_0, \underline{h})$

- $\lim\limits_{\underline{h} \to \underline{0}} \dfrac{\|\underline{\Phi}(\underline{x}_0, \underline{h})\|}{\|\underline{h}\|} = 0$

Remarque :

L'application linéaire $d_{\underline{x}_0}G : \mathbb{R}^n \to \mathbb{R}^n$ peut être représentée par sa matrice Jacobienne.

De plus, cette application est appelée la différentielle et elle généralise la notion de dérivée.

I. a) Les équations de point fixe

1.3 Définition

On dit que l'application $\underline{G}$ est différentiable au point $\underline{x}_0 \in \mathbb{R}^n$ s'il existe une application linéaire $d_{\underline{x}_0}G$ vérifiant :

- $\forall \underline{h} \in \mathbb{R}^n, \quad \underline{G}(\underline{x}_0 + \underline{h}) = \underline{G}(\underline{x}_0) + d_{\underline{x}_0}G(\underline{h}) + \underline{\Phi}(\underline{x}_0, \underline{h})$

- $\lim\limits_{\underline{h} \to \underline{0}} \dfrac{\|\underline{\Phi}(\underline{x}_0, \underline{h})\|}{\|\underline{h}\|} = 0$

Remarque 2 :

Cette définition (ainsi que le théorème de Point Fixe) se généralise au cas d'application G d'un espace de Banach dans lui-même.

I. a) Les équations de point fixe

1.3 Définition

On dit que l'application $\underline{G}$ est différentiable au point $\underline{x}_0 \in \mathbb{R}^n$ s'il existe une application linéaire $d_{\underline{x}_0}\underline{G}$ vérifiant :

- $\forall \underline{h} \in \mathbb{R}^n, \quad \underline{G}(\underline{x}_0 + \underline{h}) = \underline{G}(\underline{x}_0) + d_{\underline{x}_0}\underline{G}(\underline{h}) + \underline{\Phi}(\underline{x}_0, \underline{h})$

- $\lim\limits_{\underline{h} \to \underline{0}} \dfrac{\|\underline{\Phi}(\underline{x}_0, \underline{h})\|}{\|\underline{h}\|} = 0$

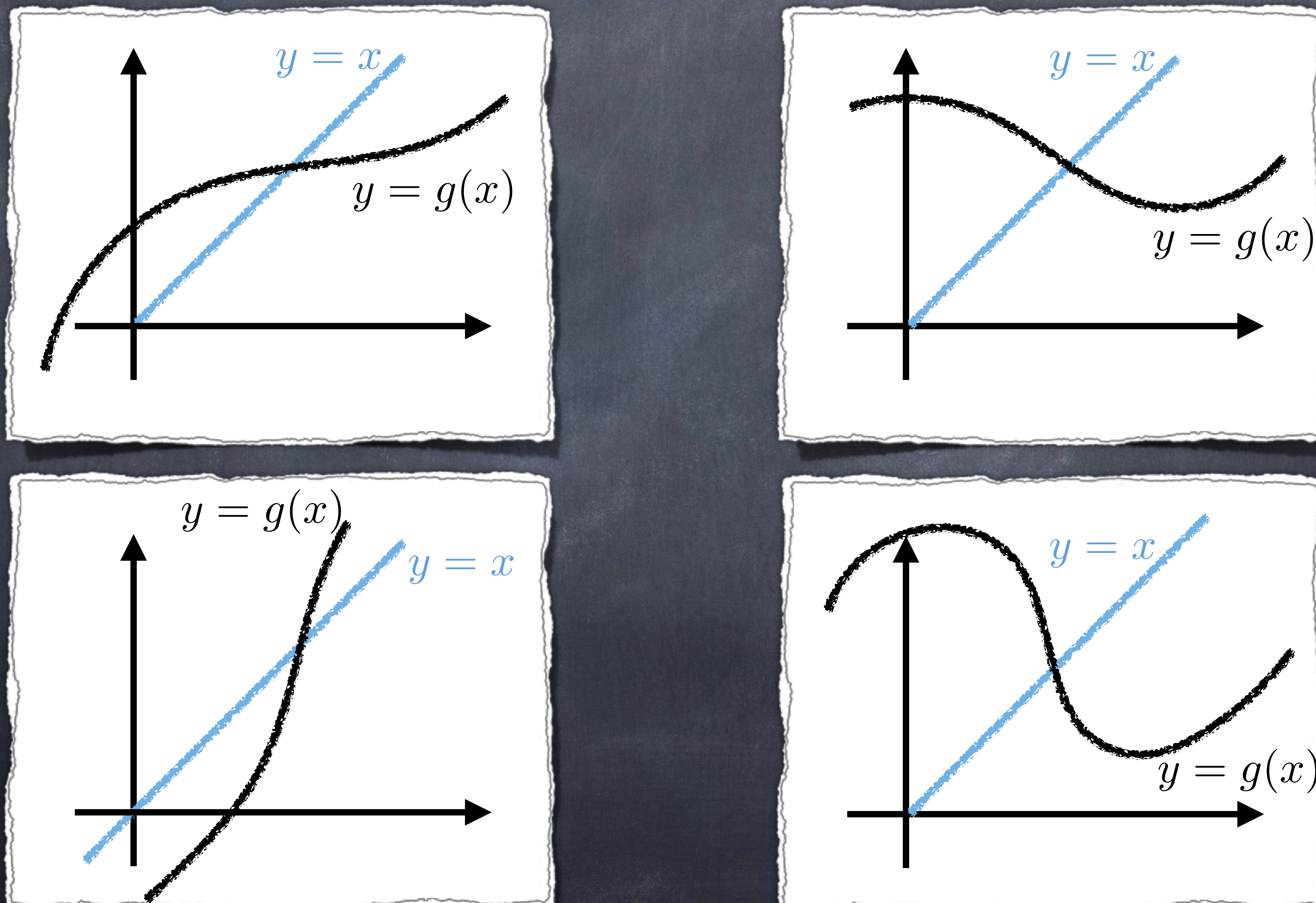
1.4 Théorème (Point fixe, le retour)

Si l'application $\underline{G}$ admet un point fixe $\underline{x}^* \in \mathbb{R}^n$, est différentiable en ce point et vérifie $\|d_{\underline{x}^*}\underline{G}\| < 1$, alors il existe un voisinage V t.q. $\forall \underline{x}_0 \in V$, la suite $\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$ converge vers le point fixe.

Preuve : au (vrai) tableau !

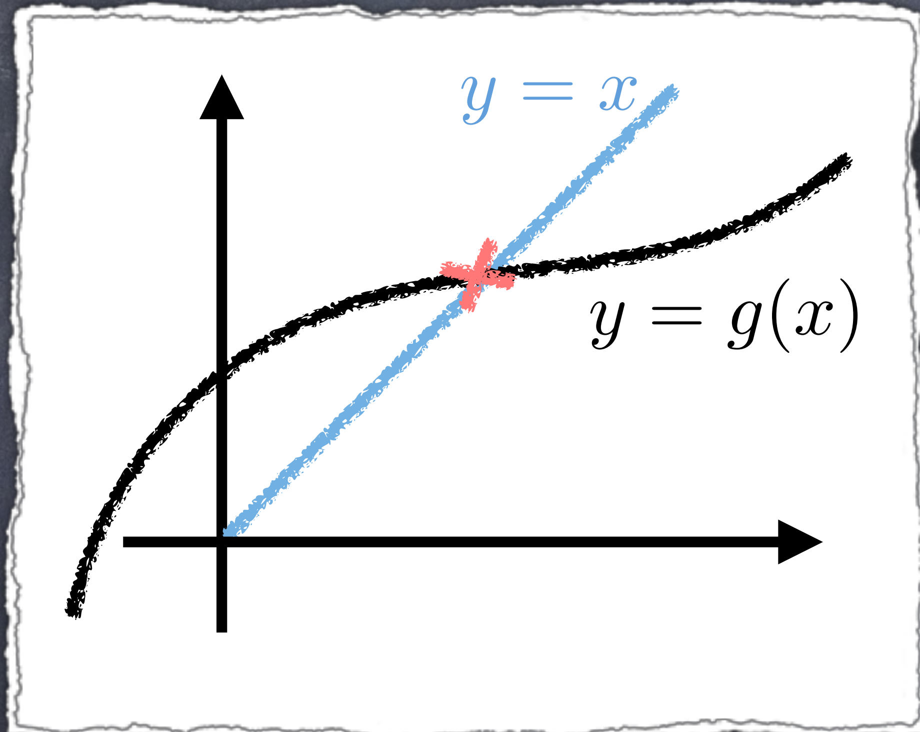
I. a) Les équations de point fixe

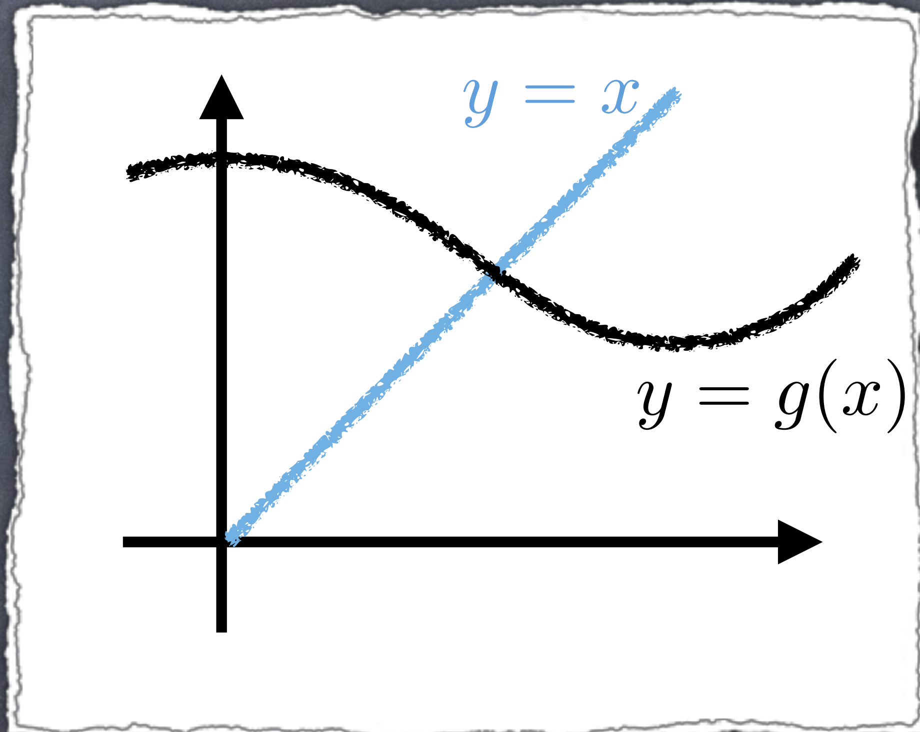
Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$

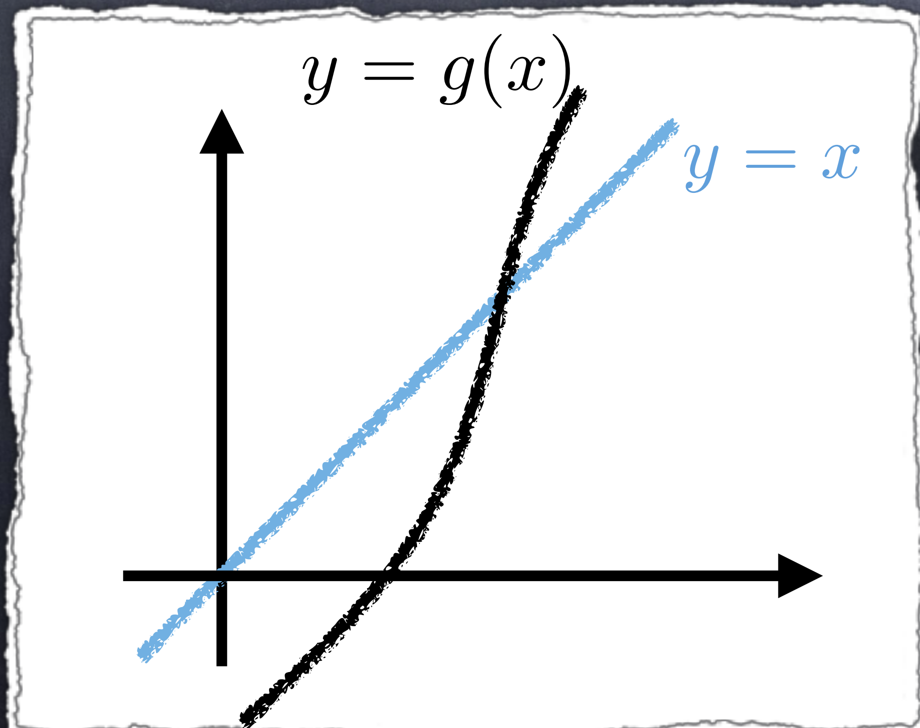


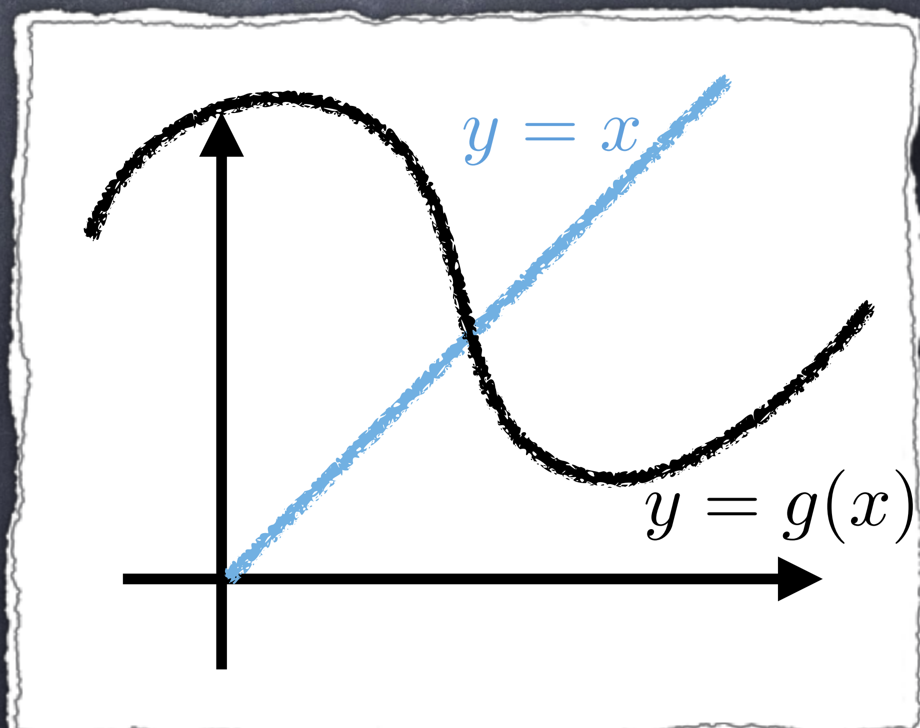
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



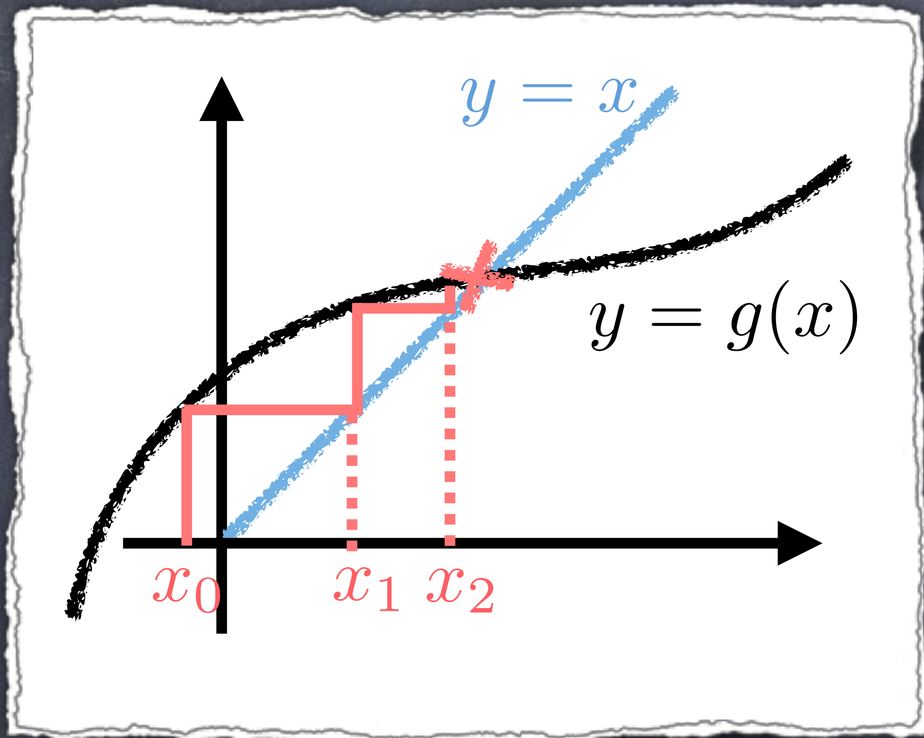


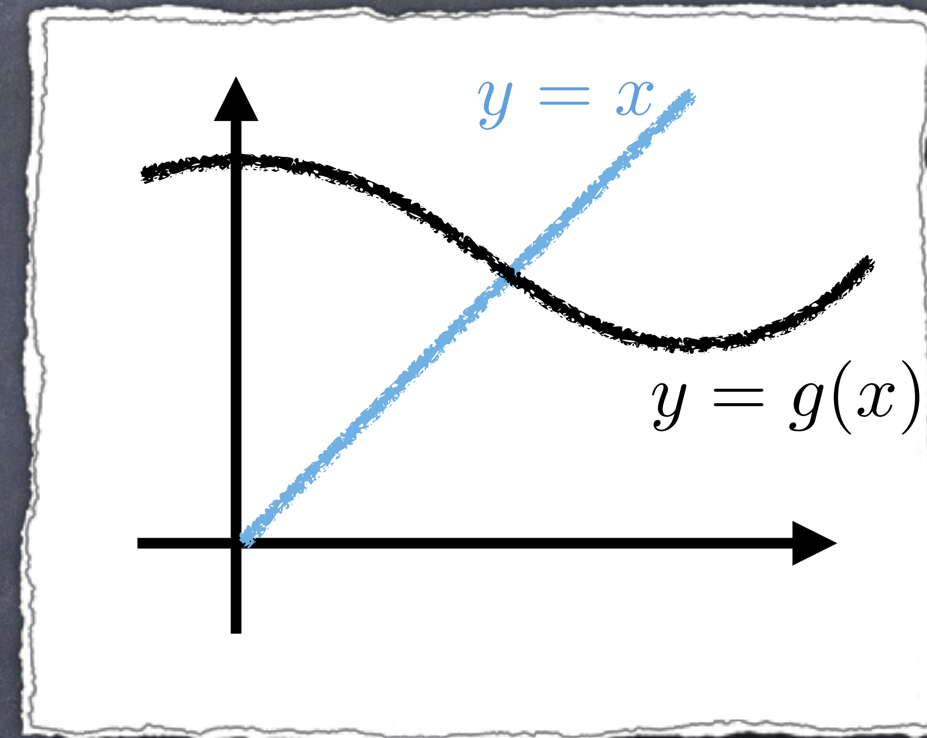


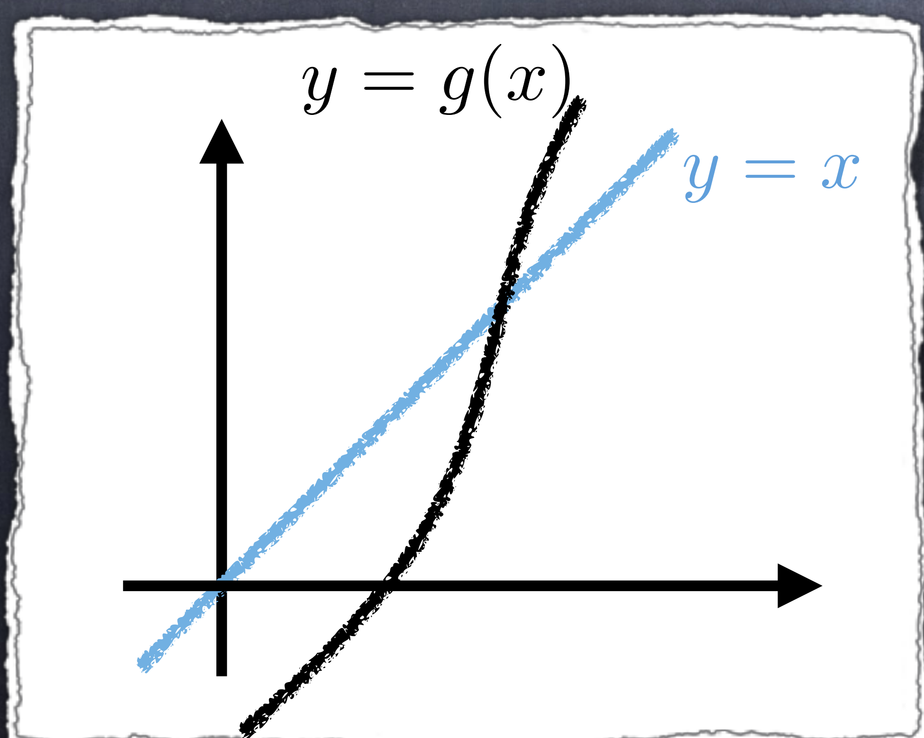


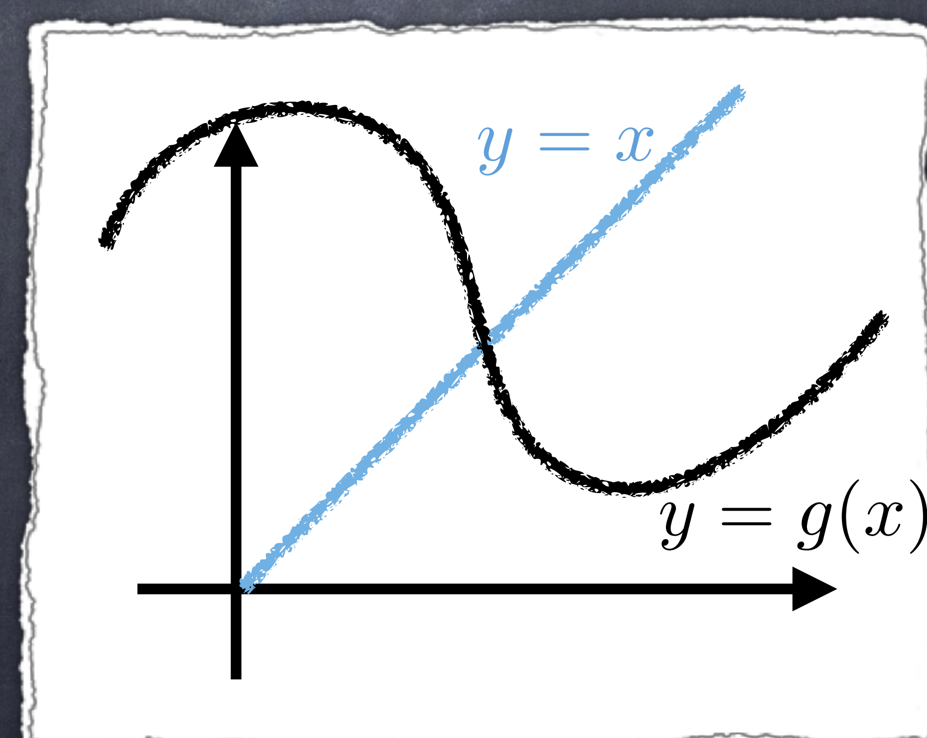
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



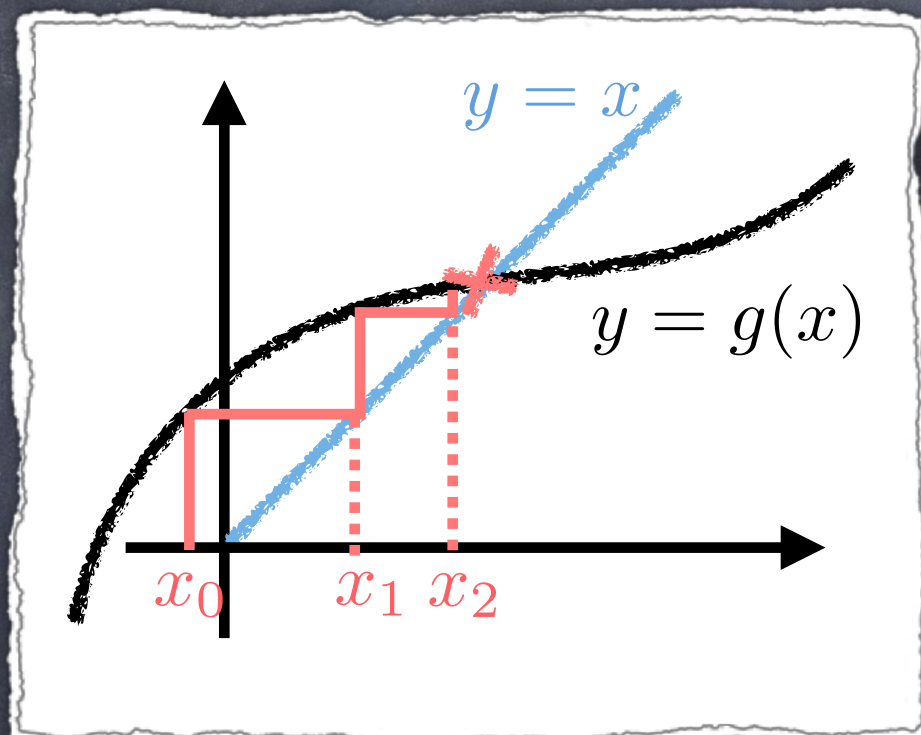


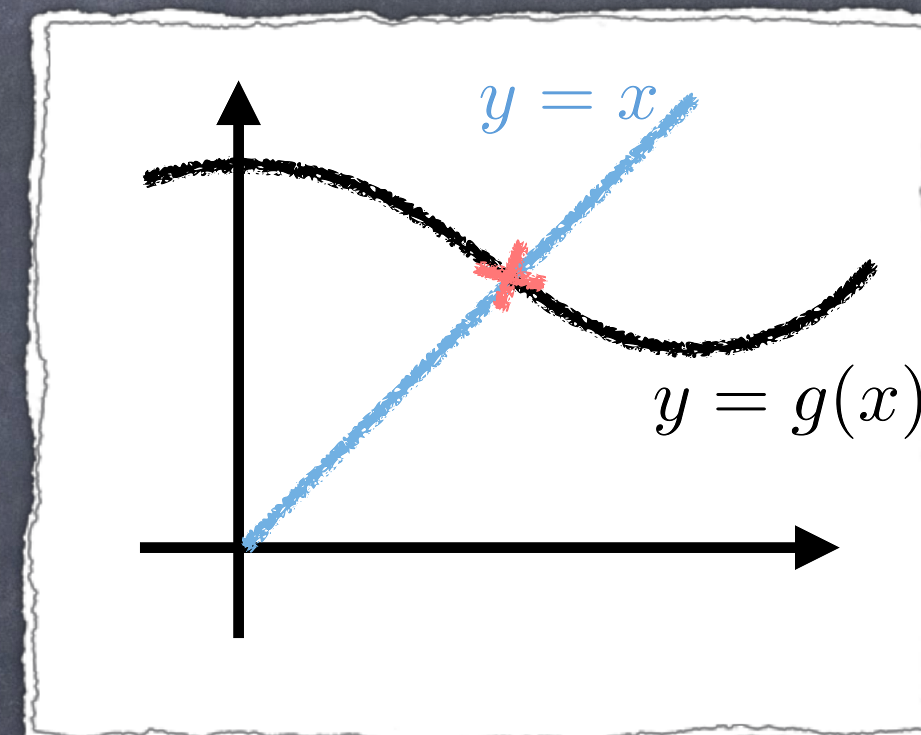


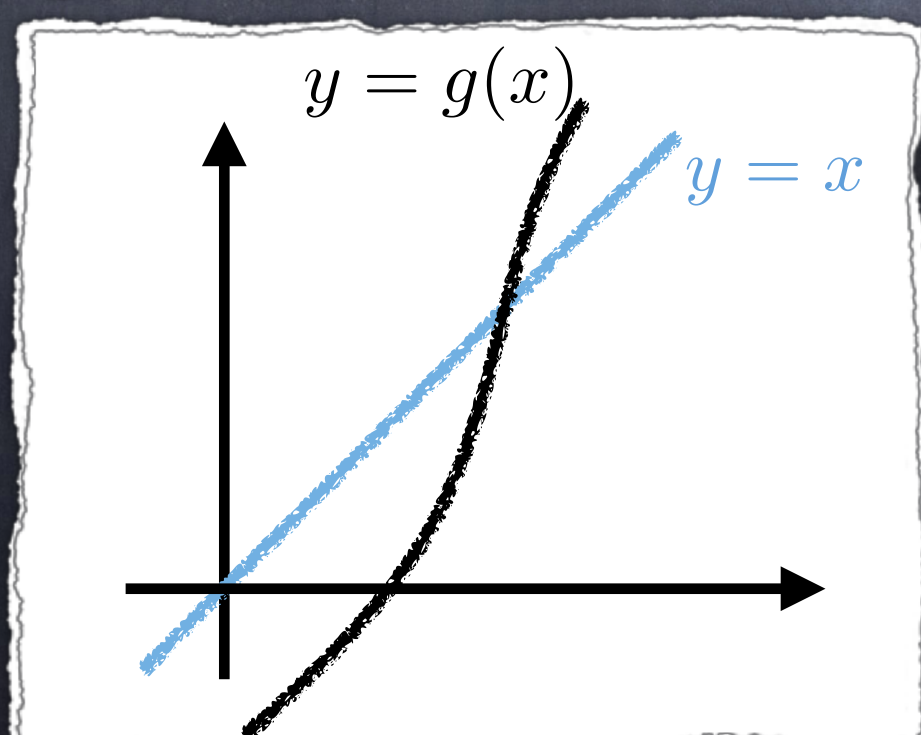


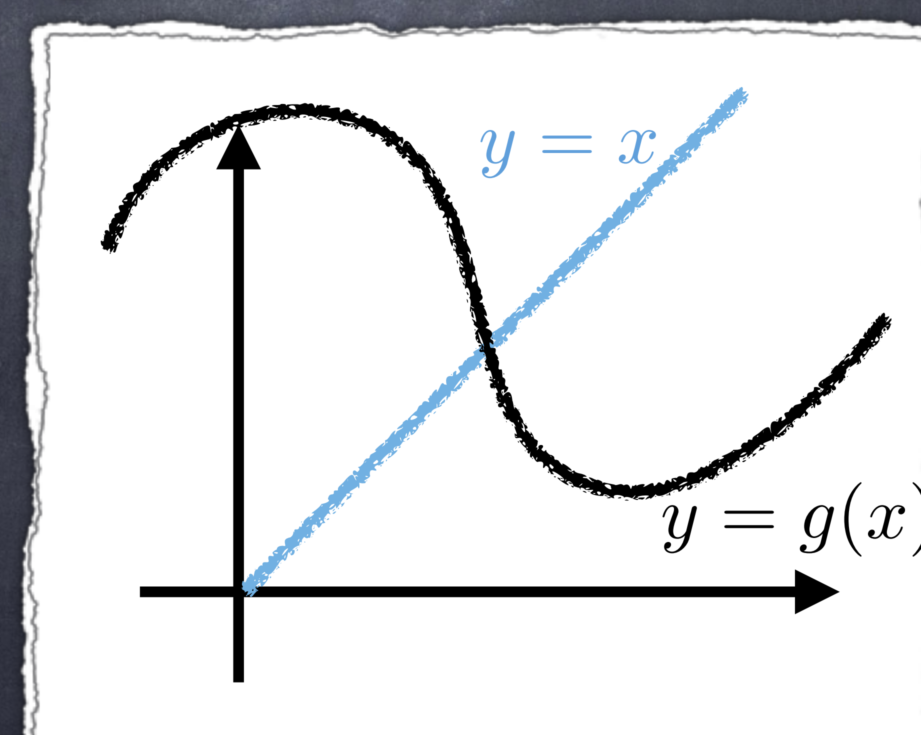
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



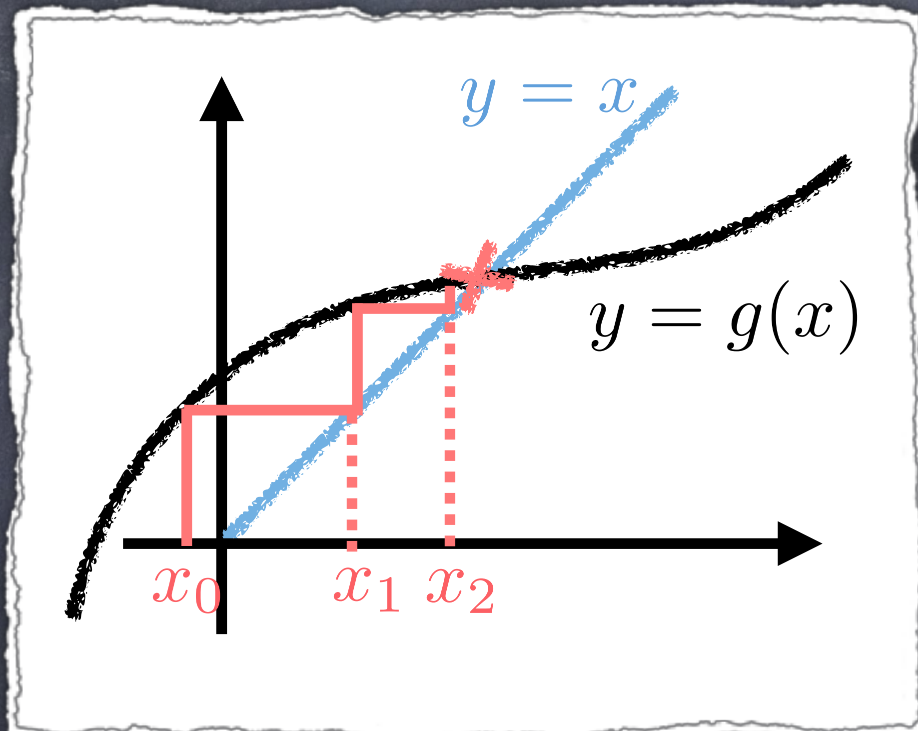


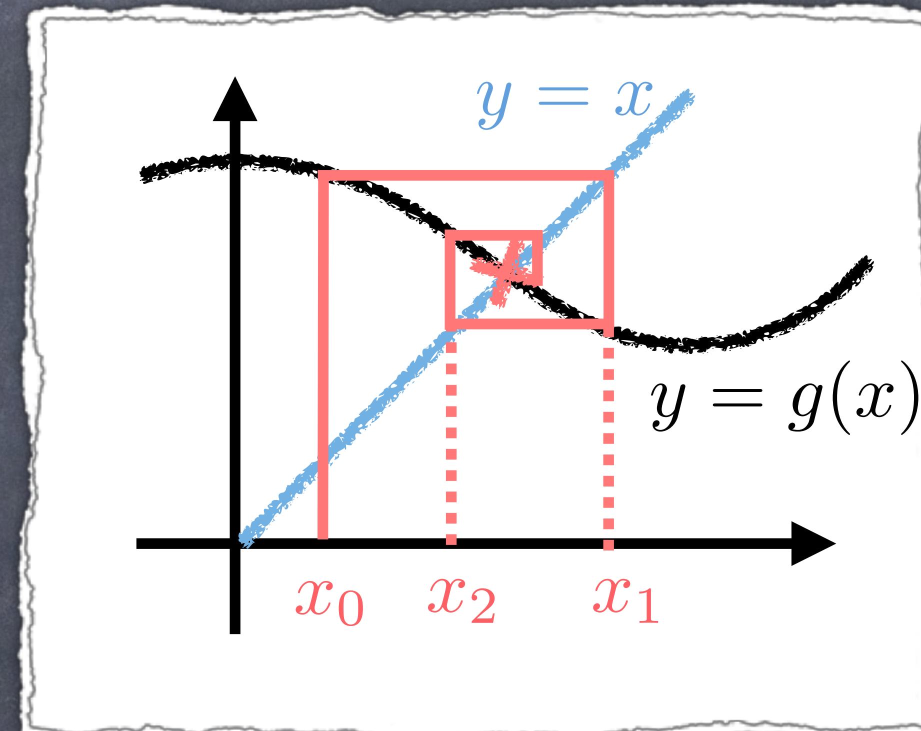


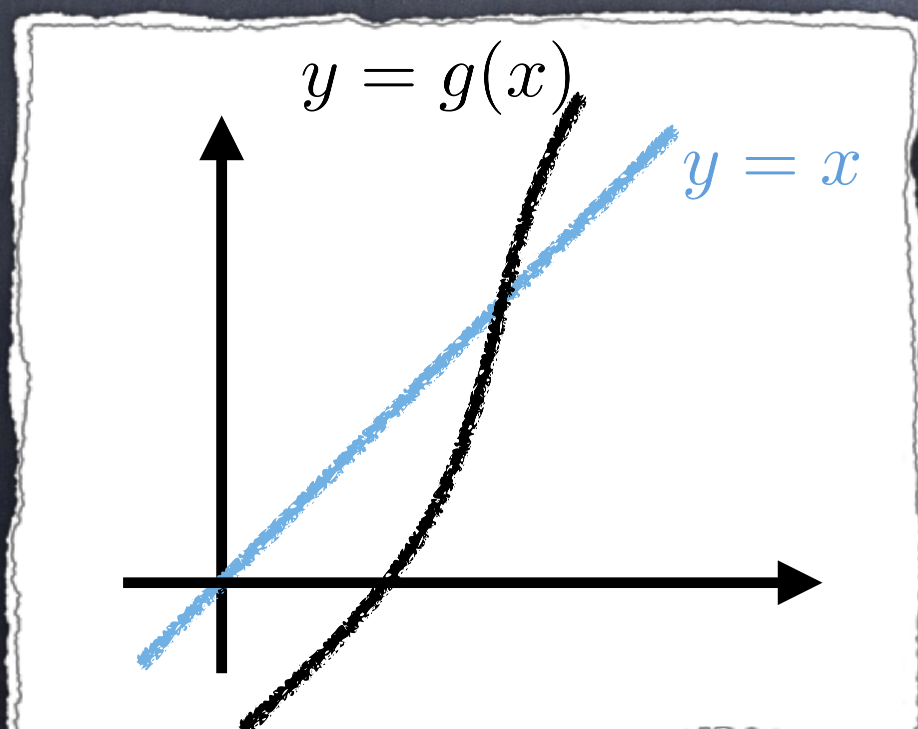


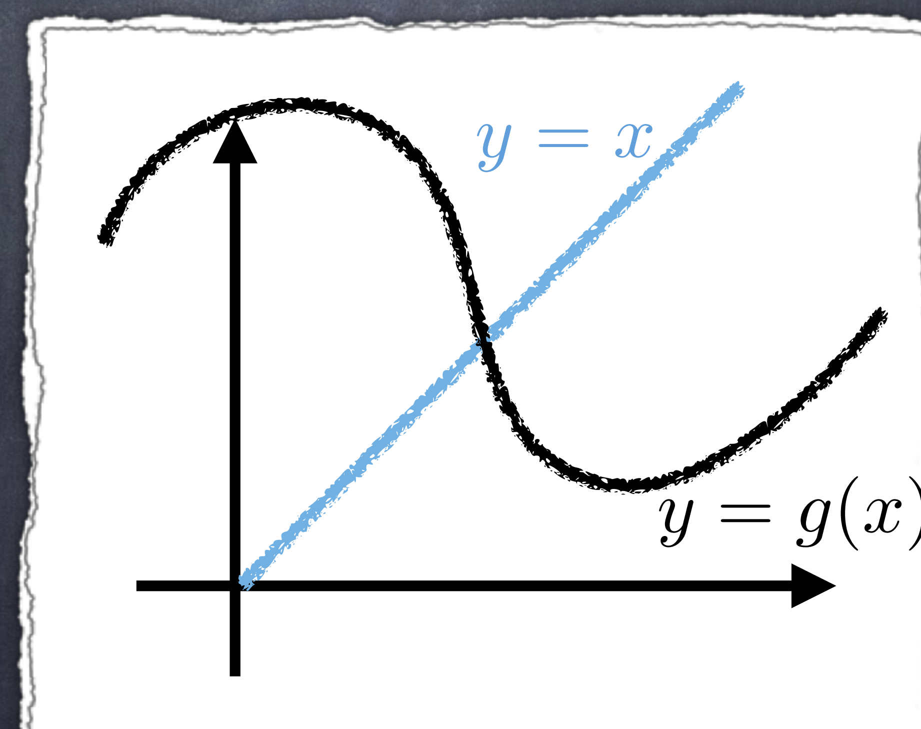
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



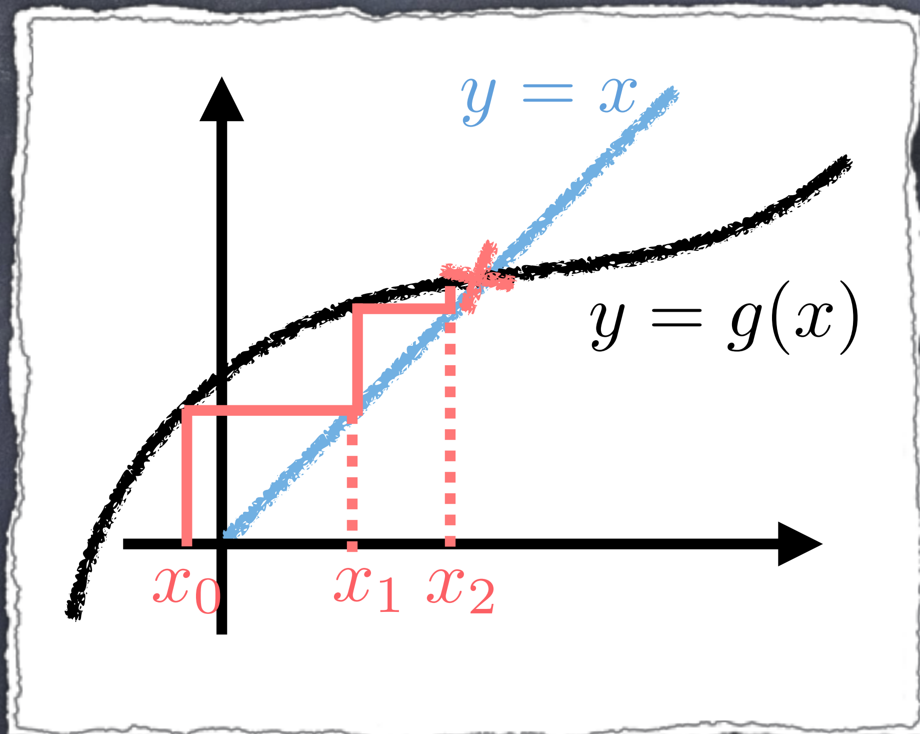


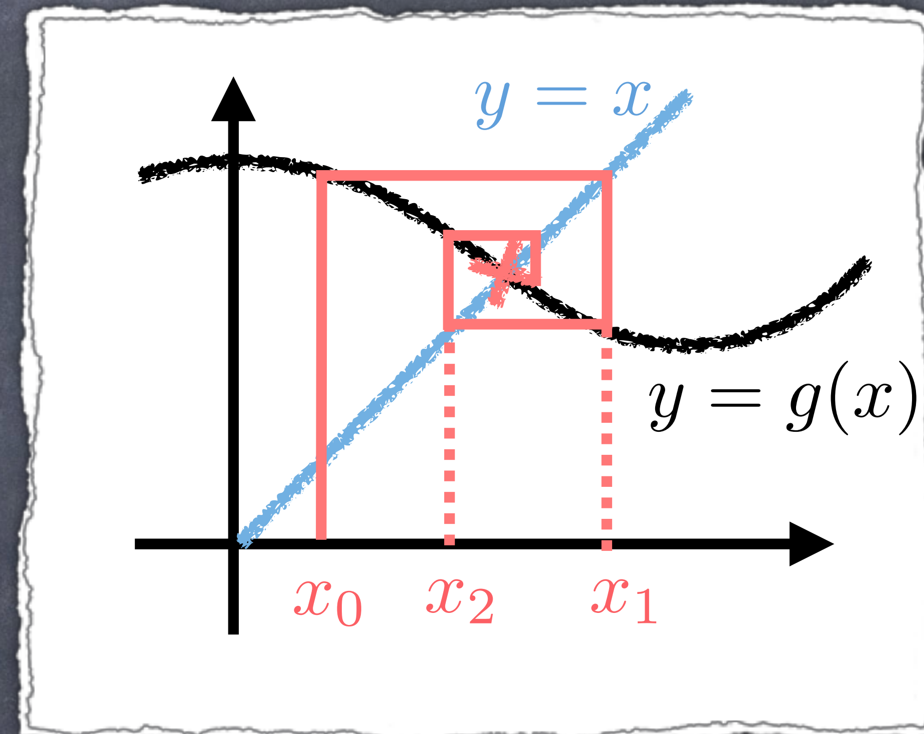


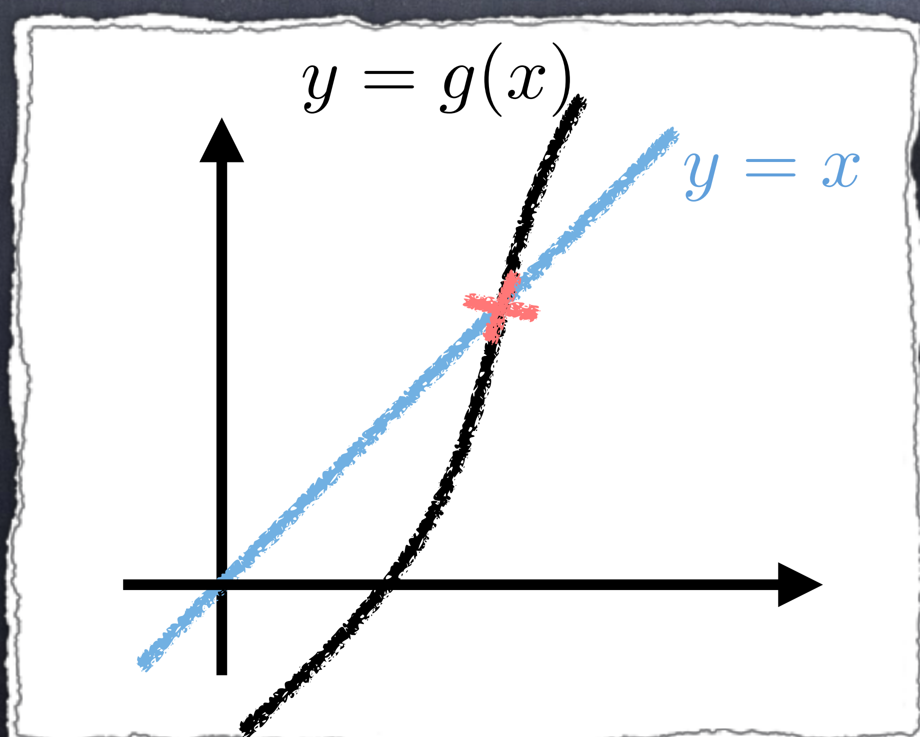


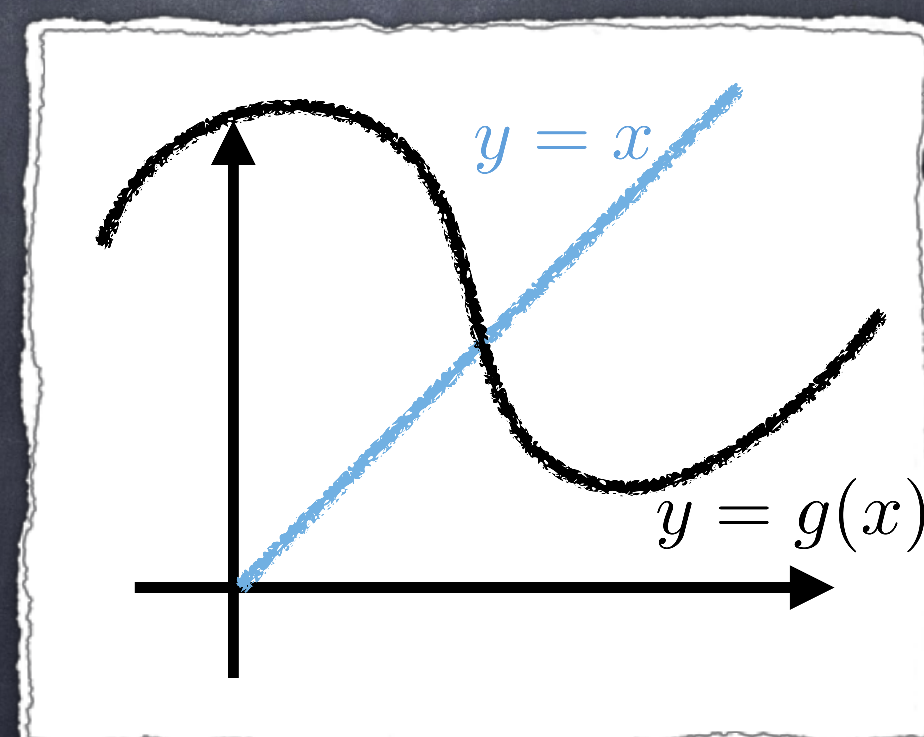
I. a) Les équations de point fixe

Quelques illustrations : $G:\mathbb{R}\to\mathbb{R}$



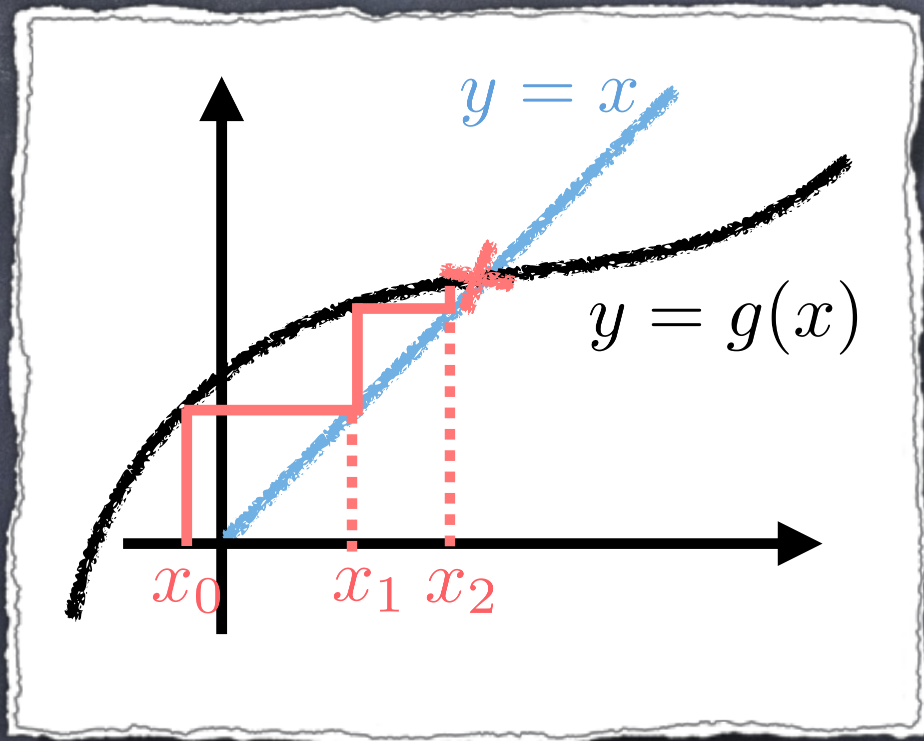


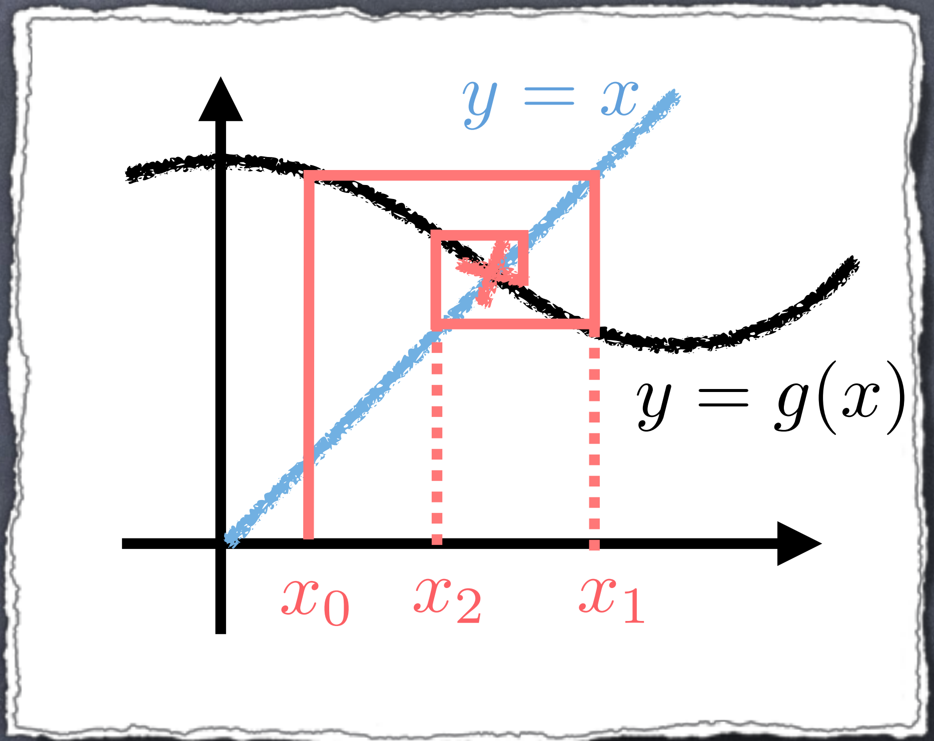


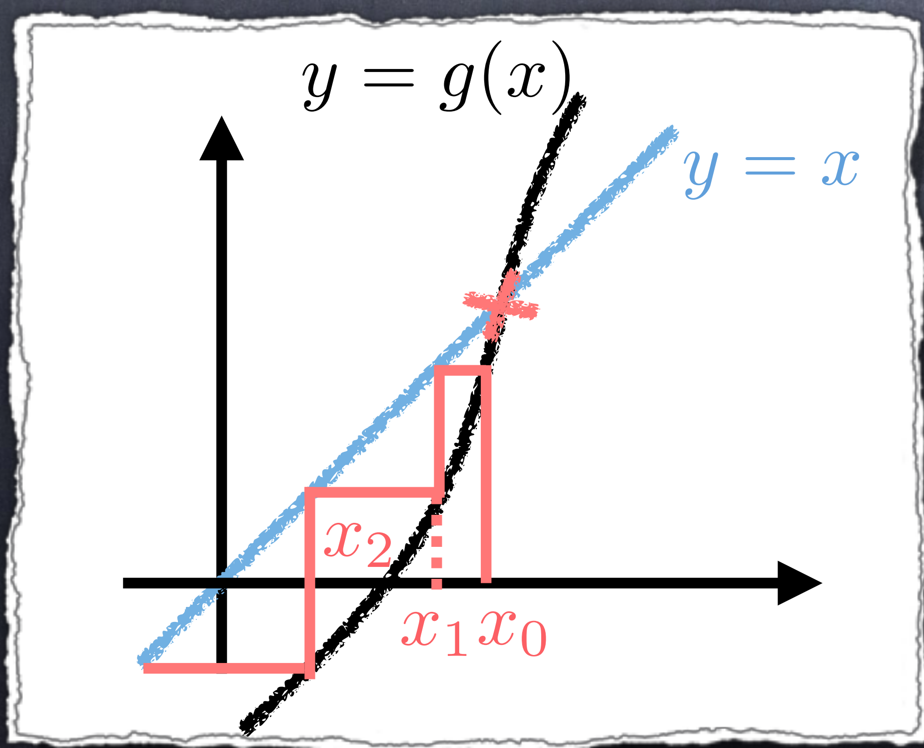


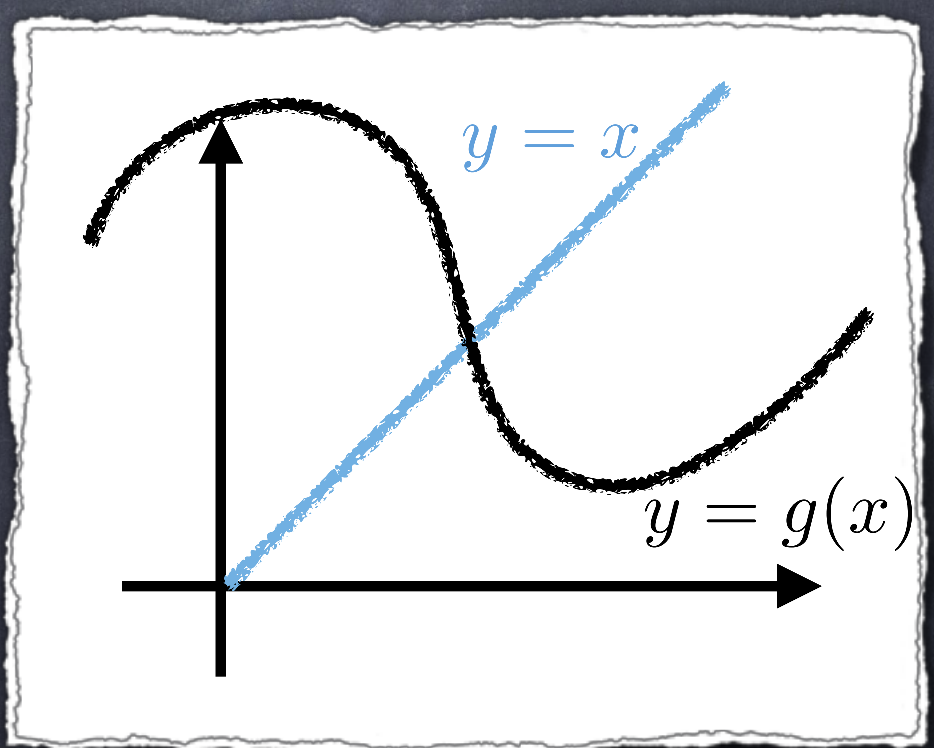
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



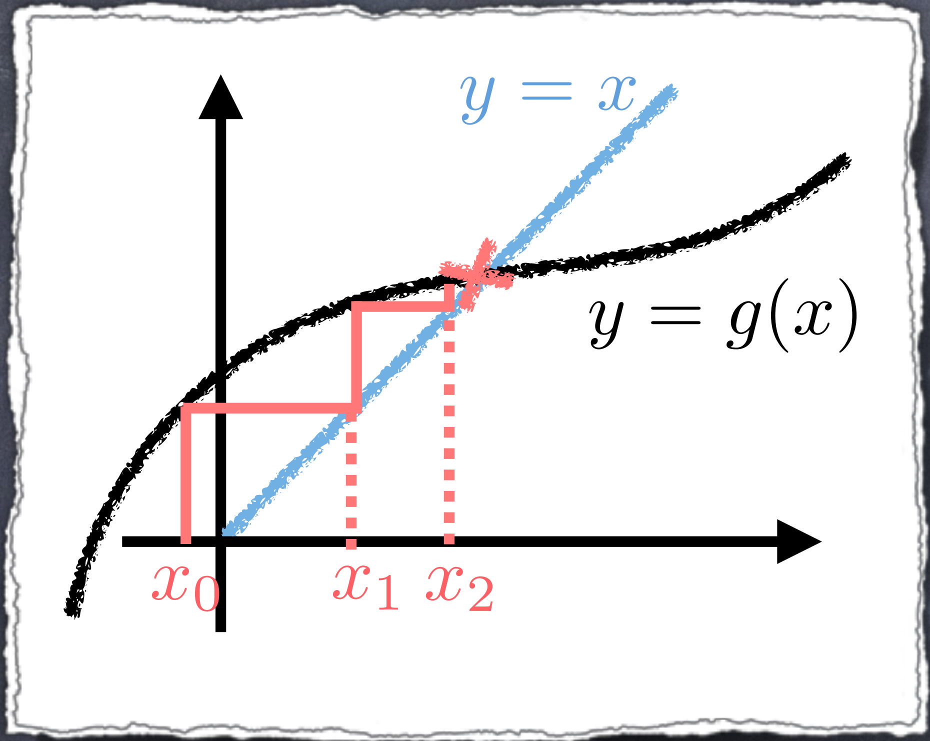


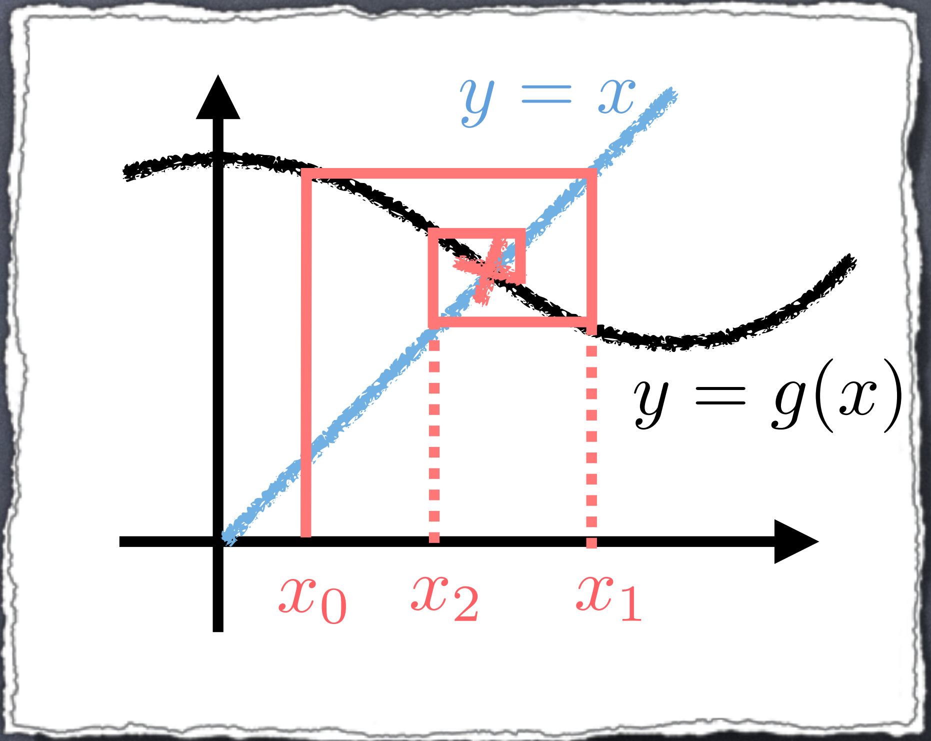


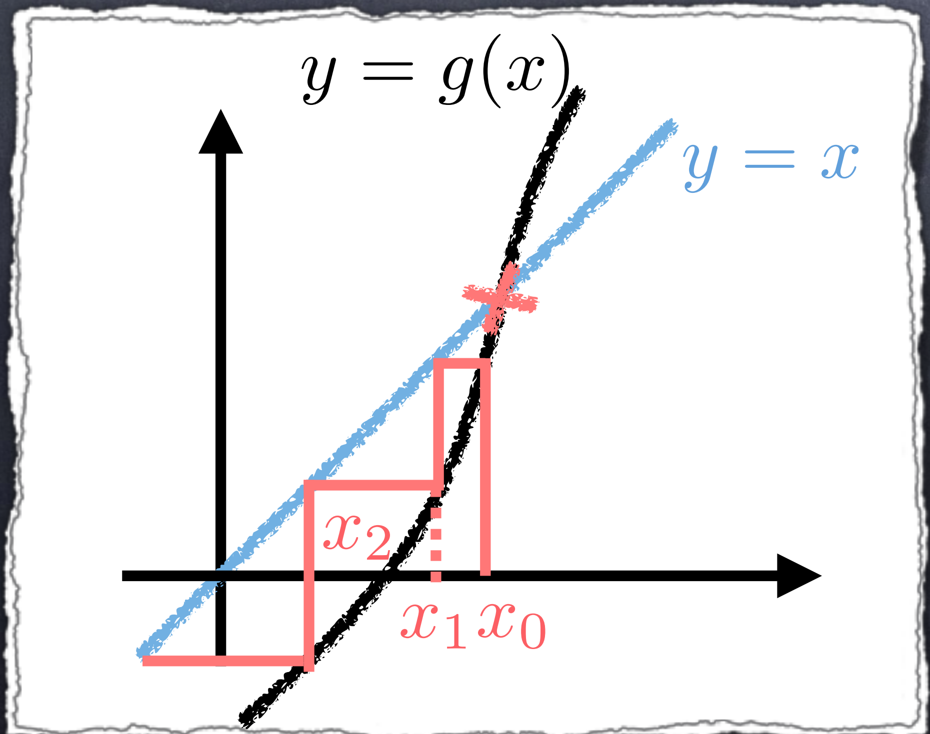


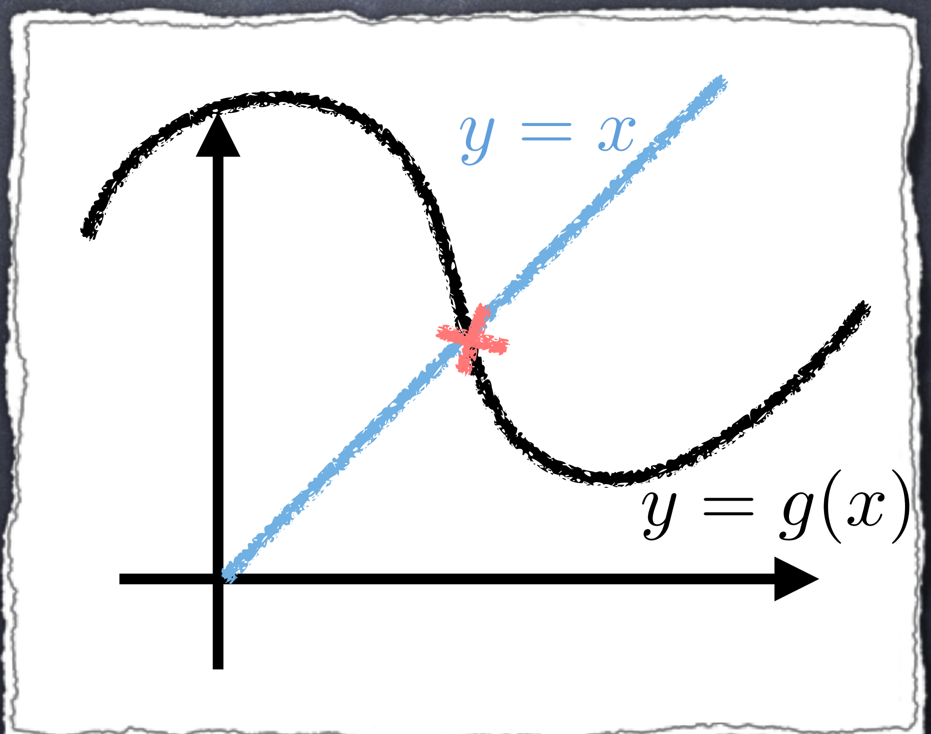
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$



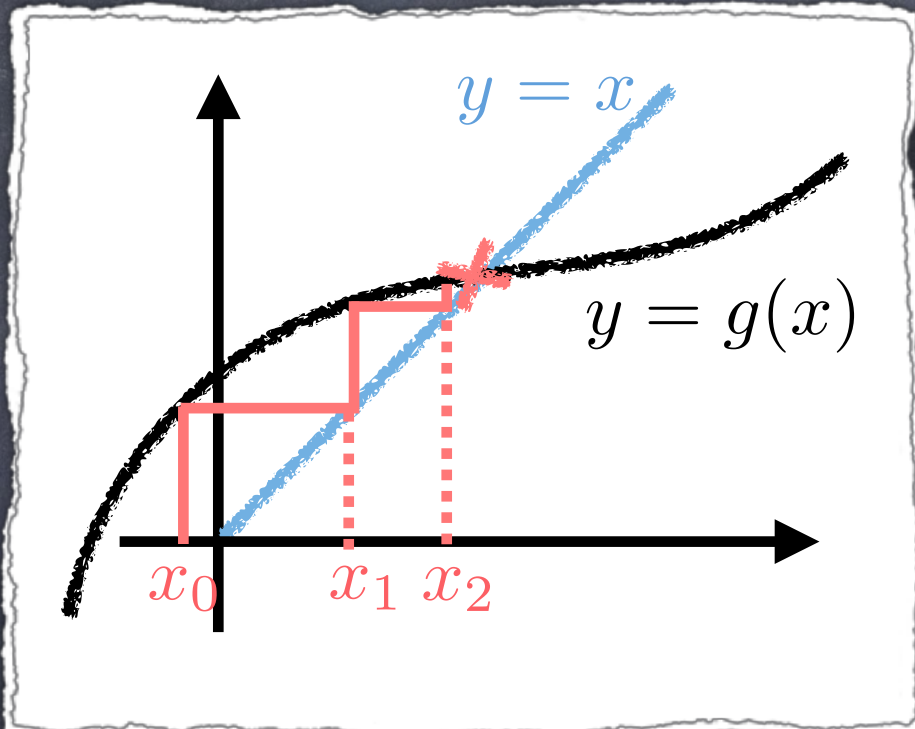


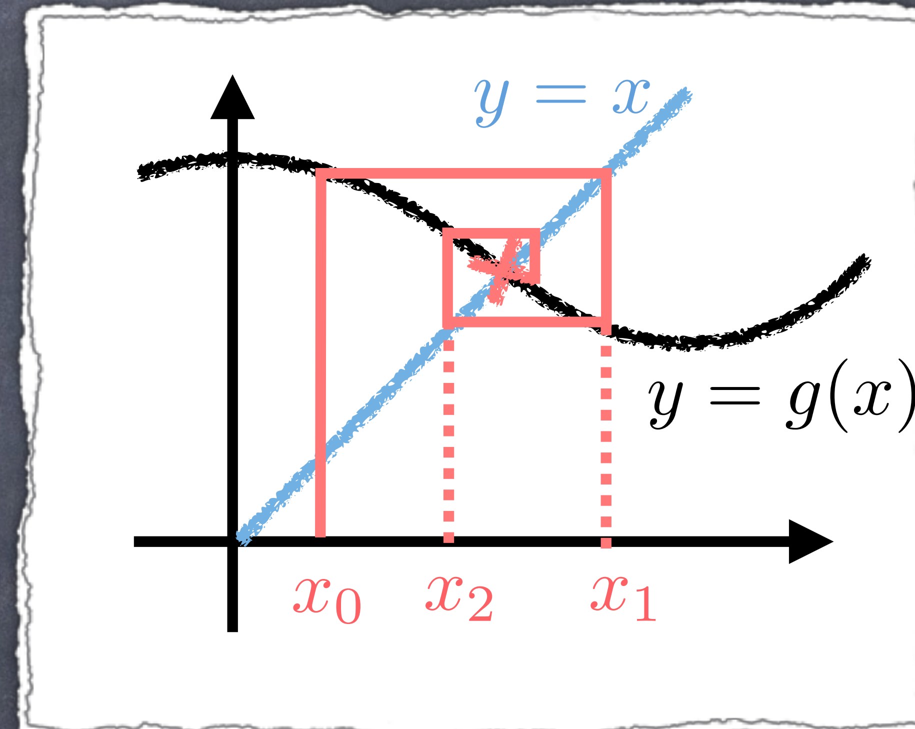


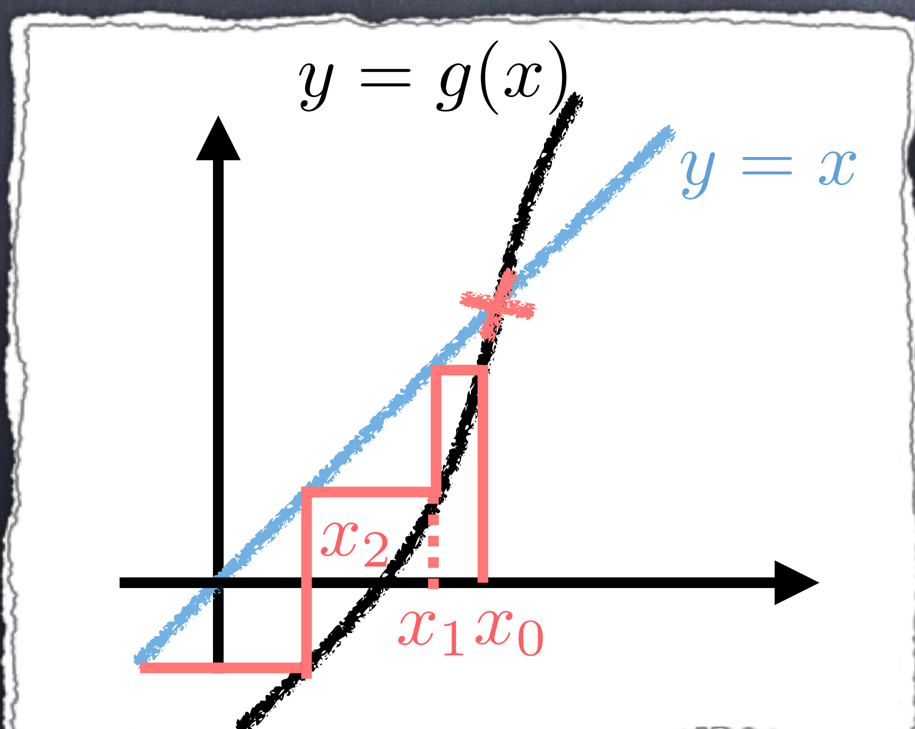


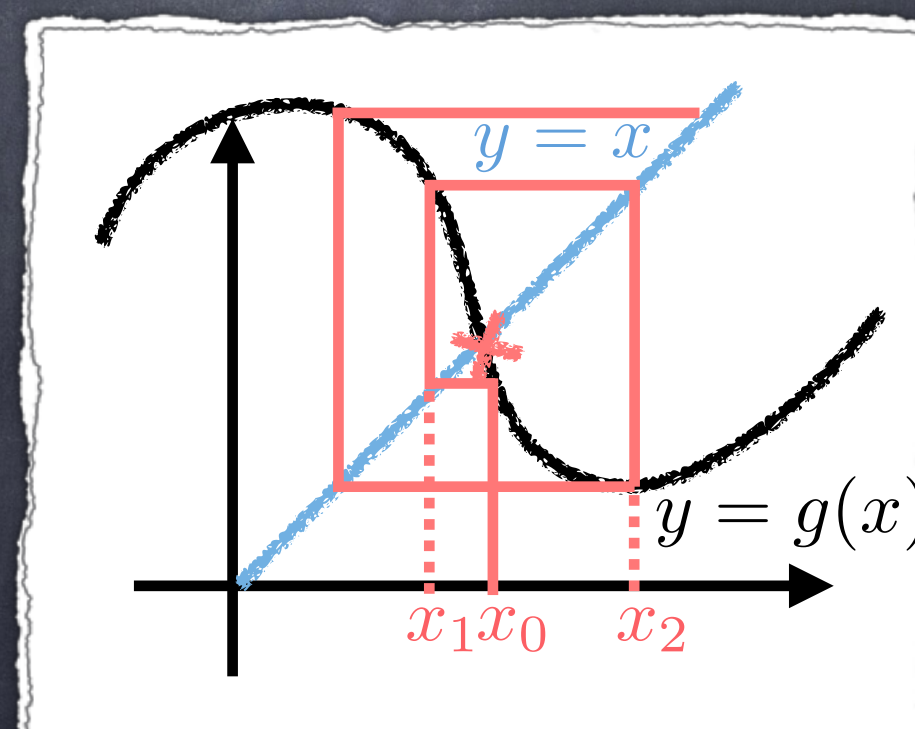
I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$





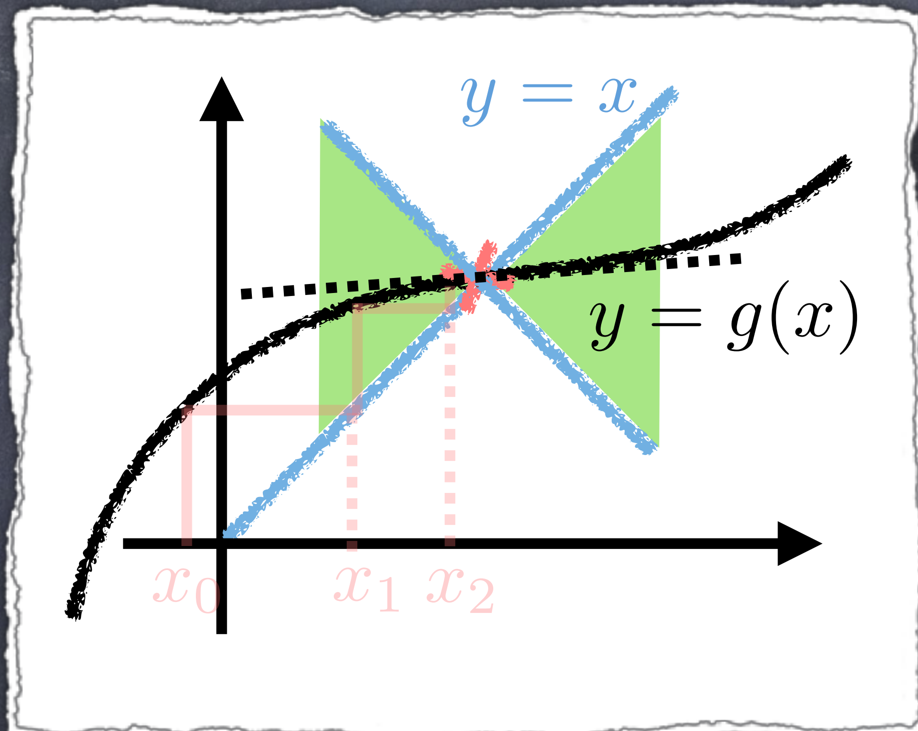


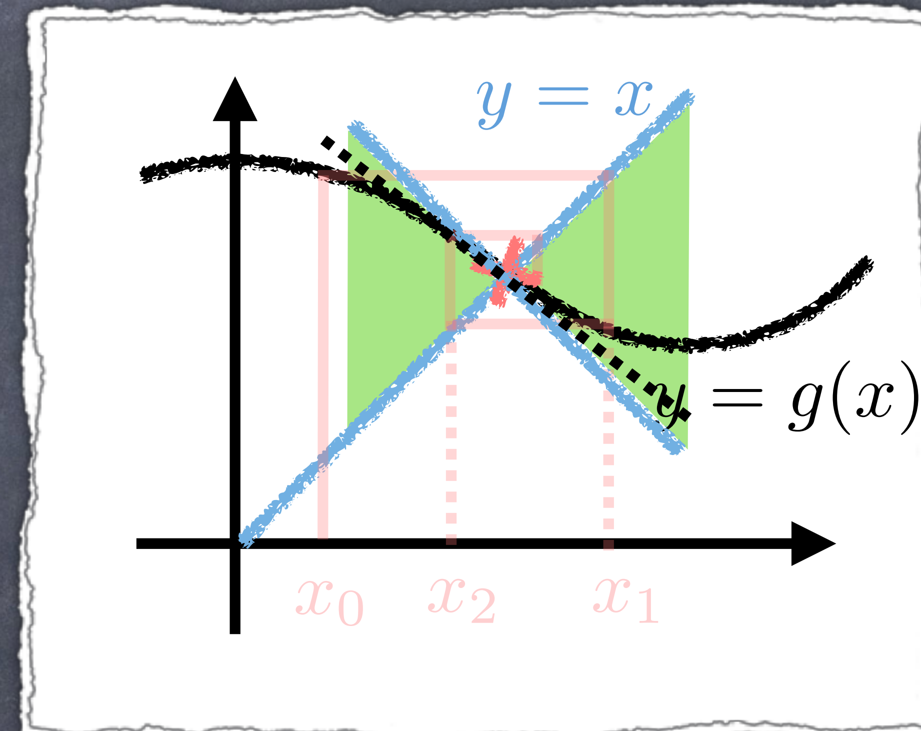


I. a) Les équations de point fixe

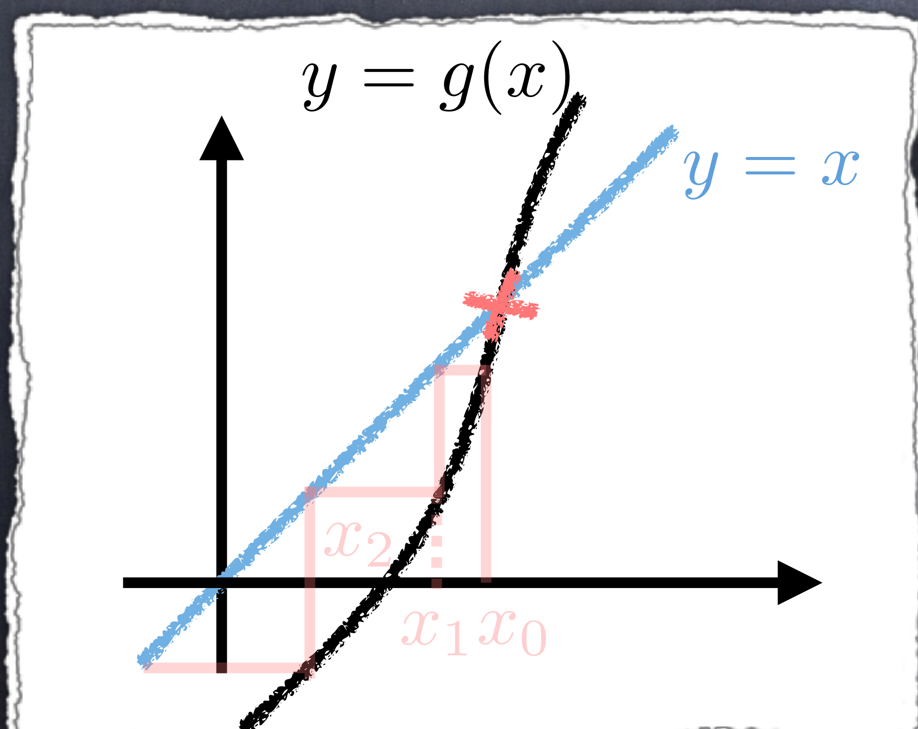
Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$

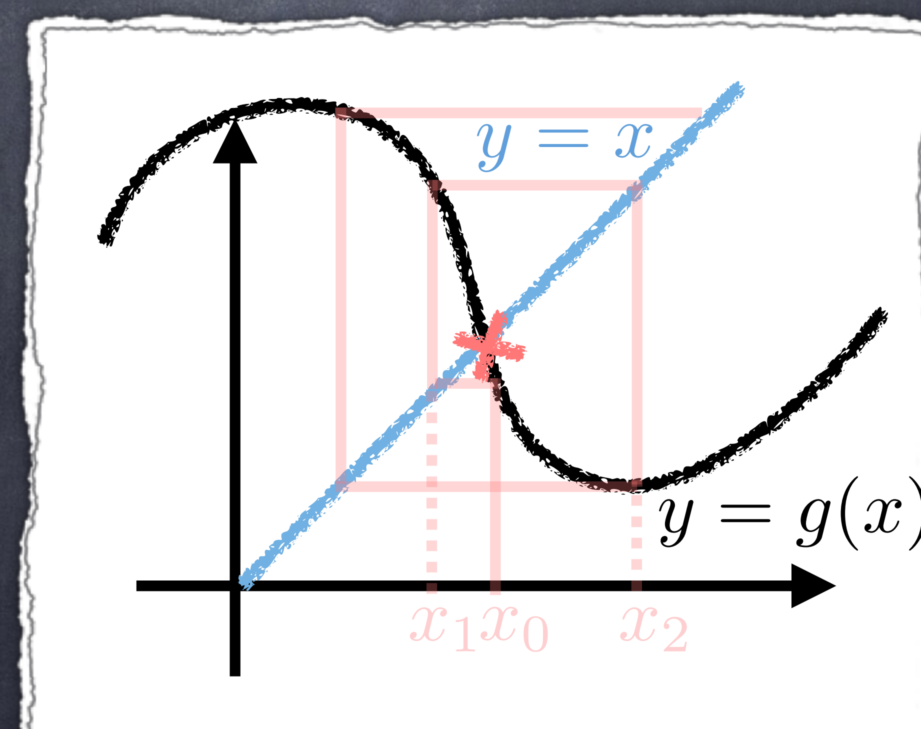
Pts Attractifs $\|d_{x^*}G\| < 1$





Pts répulsifs $\|d_{x^*}G\| > 1$



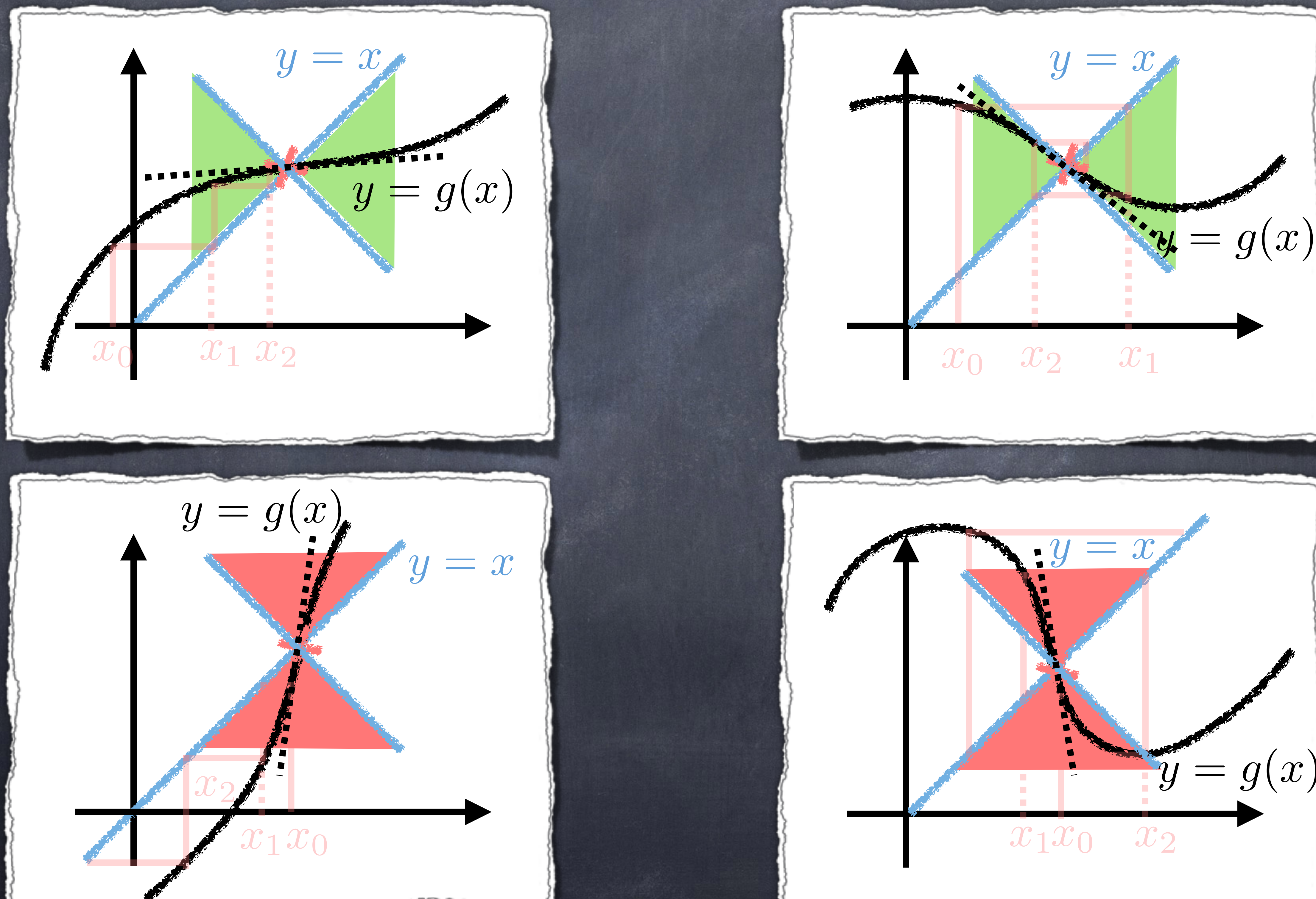


I. a) Les équations de point fixe

Quelques illustrations : $G : \mathbb{R} \to \mathbb{R}$

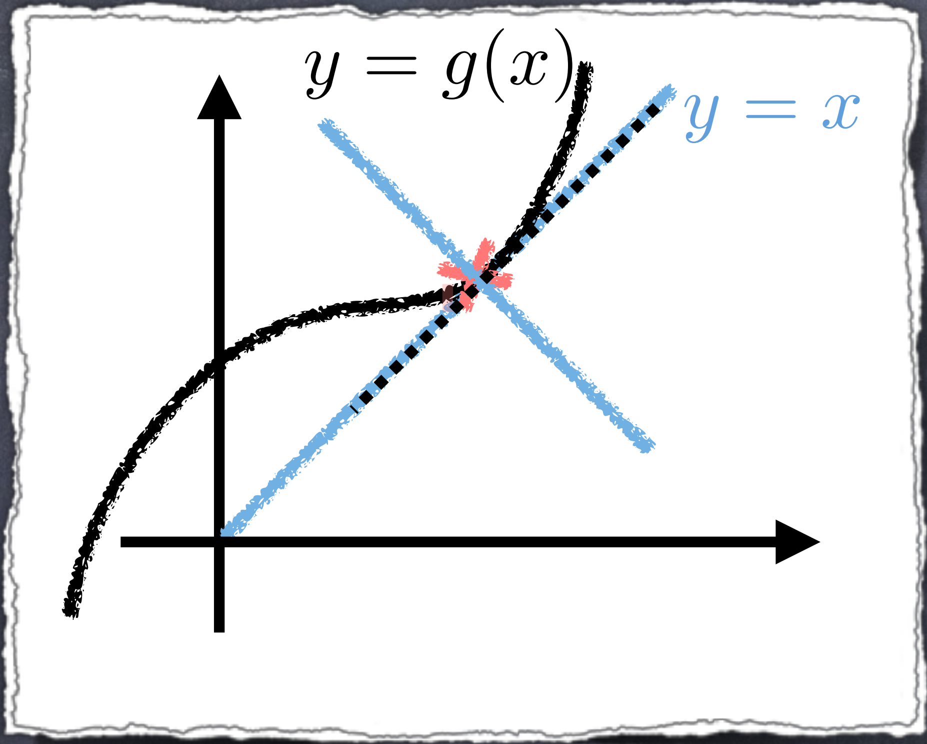
Pts Attractifs $\|d_{x^*}G\| < 1$

Pts répulsifs $\|d_{x^*}G\| > 1$



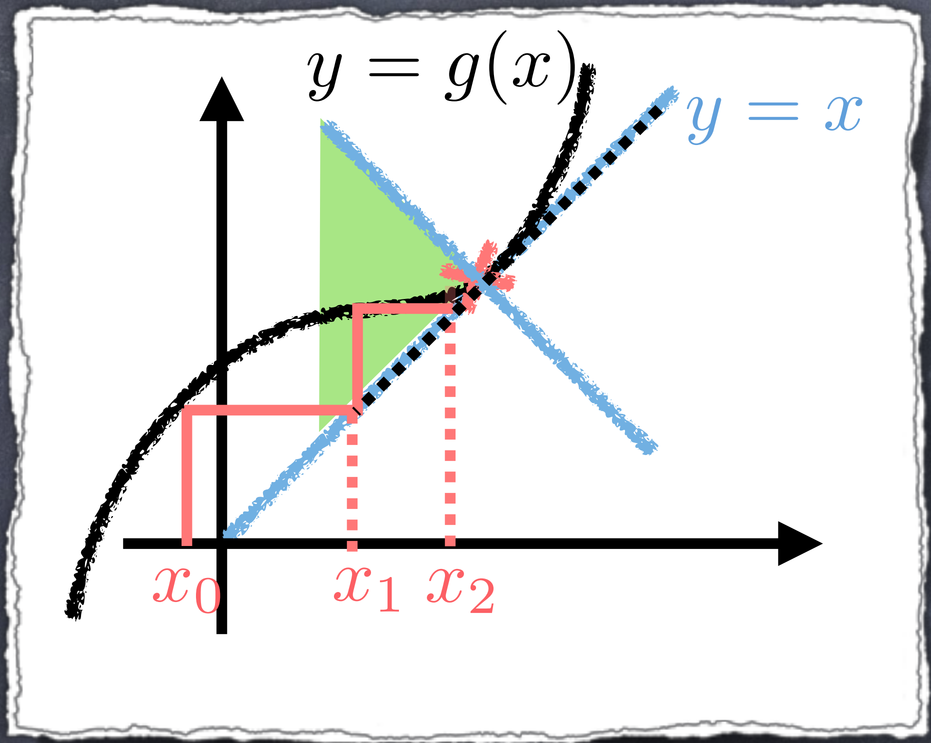
I. a) Les équations de point fixe

Un autre exemple… attractif ou répulsif ?



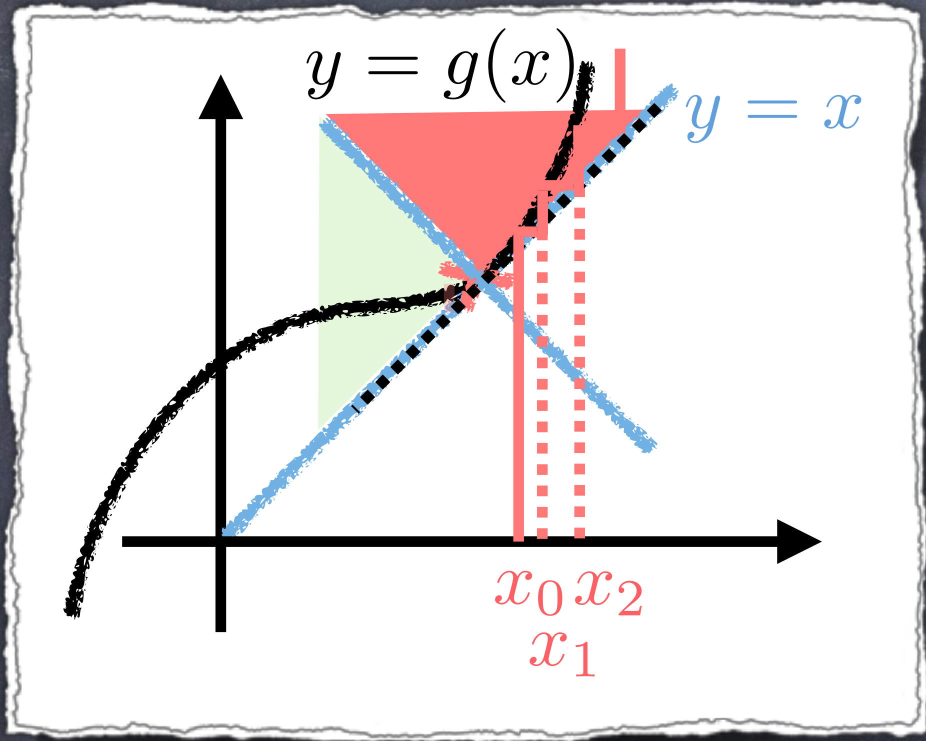
I. a) Les équations de point fixe

Un autre exemple... attractif ou répulsif ?



I. a) Les équations de point fixe

Un autre exemple... attractif ou répulsif ?



Dans cet exemple, tout dépend du point de départ !

I. a) Les équations de point fixe

Algorithme de point fixe :

$\underline{x}_1 \leftarrow \underline{G}(\underline{x}_0)$

Tant que $\|\underline{x}_0 - \underline{x}_1\| \geq \varepsilon$

$\quad \underline{x}_0 \leftarrow \underline{x}_1$

$\quad \underline{x}_1 \leftarrow \underline{G}(\underline{x}_0)$

Équivalent mathématiques

$\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$

I. a) Les équations de point fixe

Algorithme de point fixe :

$\underline{x}_1 \leftarrow \underline{G}(\underline{x}_0)$

Tant que $\|\underline{x}_0 - \underline{x}_1\| \geq \varepsilon$

- $\underline{x}_0 \leftarrow \underline{x}_1$
- $\underline{x}_1 \leftarrow \underline{G}(\underline{x}_0)$

Équivalent mathématiques

$$\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$$

Remarque :

Lorsque le test de convergence est satisfait, on a alors :

$$\begin{aligned} \|\underline{x}_n - \underline{x}^*\| &\leq \|\underline{x}_n - \underline{x}_{n+1}\| + \|\underline{x}_{n+1} - \underline{x}^*\| \\ &\leq \epsilon + \|\underline{G}(\underline{x}_n) - \underline{G}(\underline{x}^*)\| \\ &\leq \epsilon + K\|\underline{x}_n - \underline{x}^*\| \end{aligned}$$

d'où : $$\|\underline{x}_n - \underline{x}^*\| \leq \frac{\epsilon}{1 - K} \qquad (0 < K < 1)$$

I. a) Les équations de point fixe

1.5 Définition (ordre d'une suite)

On dit que la suite $(\underline{x}_n)_n$ est d'ordre r s'il existe une constante C > 0 t.q. :

$$\lim_{n\to+\infty} \frac{\|\underline{x}_{n+1} - \underline{x}^*\|}{\|\underline{x}_n - \underline{x}^*\|^r} = C$$

Remarque :

Cela revient à dire $\|\underline{x}_{n+1} - \underline{x}^*\| \sim C\|\underline{x}_n - \underline{x}^*\|^r$

Cette notion permet de « mesurer » la vitesse de convergence d'une suite (quand elle converge).

I. a) Les équations de point fixe

1.5 Définition (ordre d'une suite)

On dit que la suite $(\underline{x}_n)_n$ **est d'ordre r** s'il existe une constante $C > 0$ t.q. :

$$\lim_{n\to+\infty} \frac{\|\underline{x}_{n+1} - \underline{x}^*\|}{\|\underline{x}_n - \underline{x}^*\|^r} = C$$

Si les hypothèses du Thm. de point fixe 1.4 sont vérifiées, on a alors :

$$\underline{x}_{n+1} = G(\underline{x}^* + \underline{x}_n - \underline{x}^*)$$

$$= \underline{x}^* + d_{\underline{x}^*}\underline{G}(\underline{x}_n - \underline{x}^*) + \underline{\Phi}(\underline{x}^*, \underline{x}_n - \underline{x}^*)$$

d'où : $$\|\underline{x}_{n+1} - \underline{x}^*\| = \|(d_{\underline{x}^*}G)(\underline{x}_n - \underline{x}^*) + \underline{\Phi}(\underline{x}^*, \underline{x}_n - \underline{x}^*)\|$$

et dont on déduit que la suite est au moins d'ordre 1

I. a) Les équations de point fixe

1.5 Définition (ordre d'une suite)

On dit que la suite $(\underline{x}_n)_n$ est d'ordre r s'il existe une constante $C > 0$ t.q. :

$$\lim_{n \to +\infty} \frac{\|\underline{x}_{n+1} - \underline{x}^*\|}{\|\underline{x}_n - \underline{x}^*\|^r} = C$$

1.6 Théorème

Si l'application $\underline{G}$ admet un point fixe $x^* \in \mathbb{R}^n$, est r fois continûment différentiable et vérifie :

$$d_{\underline{x}^*}\underline{G} = \cdots = d_{x^*}^{r-1}\underline{G} = \underline{0} \quad \text{et} \quad d_{x^*}^{r}\underline{G} \neq \underline{0}$$

alors il existe un voisinage V t.q. $\forall \underline{x}_0 \in V$, la suite générée par $\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$ converge à l'ordre r vers le point fixe.

I. a) Les équations de point fixe

Preuve : (Cas des fonctions de $\mathbb{R} \to \mathbb{R}$)

On sait que la suite converge car $|G'(x_*)| = 0 < 1$ (cf. Thm 1.4)

I. a) Les équations de point fixe

Preuve : (Cas des fonctions de $\mathbb{R} \to \mathbb{R}$)

On sait que la suite converge car $|G'(x_*)| = 0 < 1$ (cf. Thm 1.4)

En utilisant le développement de Taylor au voisinage du point fixe, on a :

$$
\begin{aligned}
x_{n+1} &= G(x_* + (x_n - x_*)) \\
&= G(x_*) + \sum_{i=1}^{r-1} G^{(i)}(x_*) \frac{(x_n - x_*)^i}{i!} + G^{(r)}(\tilde{x}) \frac{(x_n - x_*)^r}{r!}
\end{aligned}
$$

où $\tilde{x} \in [x_n, x_*]$ (ou $[x_*, x_n]$ si $x_* < x_n$).

I. a) Les équations de point fixe

Preuve : (Cas des fonctions de $\mathbb{R} \to \mathbb{R}$)

On sait que la suite converge car $|G'(x_*)| = 0 < 1$ (cf. Thm 1.4)

En utilisant le développement de Taylor au voisinage du point fixe, on a :

$$x_{n+1} = G(x_* + (x_n - x_*))$$

$$= G(x_*) + \sum_{i=1}^{r-1} G^{(i)}(x_*) \frac{(x_n - x_*)^i}{i!} + G^{(r)}(\tilde{x}) \frac{(x_n - x_*)^r}{r!}$$

où $\tilde{x} \in [x_n, x_*]$ (ou $[x_*, x_n]$ si $x_* < x_n$). On déduit alors :

$$\frac{|x_{n+1} - x_*|}{|x_n - x_*|^r} = \frac{|G^{(r)}(\tilde{x})|}{r!} \xrightarrow[n \to +\infty]{} \frac{|G^{(r)}(x^*)|}{r!}$$

ce qui prouve le résultat souhaité. $\blacksquare$

I. b) La méthode de Newton

Revenons à notre problème initial :

$$\text{Trouver } x \in \mathbb{R} \text{ t.q. } \quad F(x) = 0$$

dans le cas particulier où $F : \mathbb{R} \to \mathbb{R}$

Si F est différentiable (dérivable) en x, on a alors :

$$\begin{aligned} F(x+h) &= F(x) + (d_x F)(h) + \Phi(x,h) \\ &\simeq F(x) + (d_x F)(h) \end{aligned}$$

I. b) La méthode de Newton

Revenons à notre problème initial :

$$\text{Trouver } x \in \mathbb{R} \text{ t.q. } \quad F(x) = 0$$

dans le cas particulier où $F: \mathbb{R} \to \mathbb{R}$

Si F est différentiable (dérivable) en x, on a alors :

$$\boxed{F(x+h) \simeq F(x) + (d_x F)(h)}$$

Si x est une solution approchée de notre problème, on va chercher à l'aide de l'approximation ci-dessus h t.q. :

$$F(x+h) = 0 \quad \text{“}\Leftrightarrow\text{”} \quad F(x) + (d_x F)(h) = 0$$

$$\text{“}\Leftrightarrow\text{”} \quad h = \frac{-F(x)}{d_x F}$$

I. b) La méthode de Newton

Revenons à notre problème initial :

$$\text{Trouver } x \in \mathbb{R} \text{ t.q. } \quad F(x) = 0$$

dans le cas particulier où $F: \mathbb{R} \to \mathbb{R}$

Si F est différentiable (dérivable) en x, on a alors :

$$\boxed{F(x+h) \simeq F(x) + (d_x F)(h)}$$

Si x est une solution approchée de notre problème, on va chercher à l'aide de l'approximation ci-dessus h t.q. :

$$F(x+h) = 0 \quad \text{“}\Leftrightarrow\text{”} \quad F(x) + (d_x F)(h) = 0$$

$$\text{“}\Leftrightarrow\text{”} \quad h = \frac{-F(x)}{d_x F} \qquad \Rightarrow \quad x_2 = x + h$$

x_2 sera a priori une meilleure solution approchée.

I. b) La méthode de Newton

Méthode de Newton :

On construit la suite $(x_n)_n$ comme suit :

$$\begin{aligned} F(x_{n+1}) &= F(x_n + (x_{n+1} - x_n)) \\ &\simeq F(x_n) + (d_{x_n}F)(x_{n+1} - x_n) \end{aligned}$$

où x_{n+1} est choisi de sorte que l'approximation ci-dessus soit nulle :

$$F(x_n) + (d_{x_n}F)(x_{n+1} - x_n) = 0 \quad \Leftrightarrow \quad \boxed{x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)}$$

Remarques :

- La suite est bien définie ssi $(d_{x_n}F) \neq 0$
- Ce choix de x_{n+1} correspond à imposer

$$F(x_{n+1}) = \Phi(x_n, x_{n+1} - x_n)$$

I. b) La méthode de Newton

Méthode de Newton :

On génère la suite $(x_n)_n$ ainsi : $x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$

1.7 Proposition

x est solution de $F(x) = 0$ est équivalent à dire que c'est le point fixe de la fonction $G(x) = x - (d_xF)^{-1}F(x)$

Preuve : au (vrai) tableau ! ■

I. b) La méthode de Newton

Méthode de Newton :

On génère la suite $(x_n)_n$ ainsi : $x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$

1.7 Proposition

x est solution de $F(x) = 0$ est équivalent à dire que c'est le point fixe de la fonction $G(x) = x - (d_xF)^{-1}F(x)$

1.8 Corollaire

Si F est 3 fois continument différentiable, admet un 0 en x^* et $F'(x^*) \neq 0$, alors la méthode de Newton converge pour $x_0 \in V(x^*)$ et est d'ordre 2 au moins.

Preuve : au (vrai) tableau !

I. b) La méthode de Newton

Méthode de Newton :

$$x_1 \leftarrow x_0 - F(x_0)/F'(x_0)$$

Tant que $\|\underline{x}_0 - \underline{x}_1\| \geq \varepsilon$

$$x_0 \leftarrow x_1$$

$$x_1 \leftarrow x_0 - F(x_0)/F'(x_0)$$

Equivalent mathématiques :

$$x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$$

Remarque :

Le point x_{n+1} est définie en fait comme le point où la tangent de F en x_n

$$y = (d_{x_n}F)(x - x_n) + F(x_n)$$

s'annule !

I. b) La méthode de Newton

Méthode de Newton :

$x_1 \leftarrow x_0 - F(x_0)/F'(x_0)$

Tant que $\|\underline{x}_0 - \underline{x}_1\| \geq \varepsilon$

$\quad x_0 \leftarrow x_1$

$\quad x_1 \leftarrow x_0 - F(x_0)/F'(x_0)$

Equivalent mathématiques :

$$x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$$

Remarque 2 :

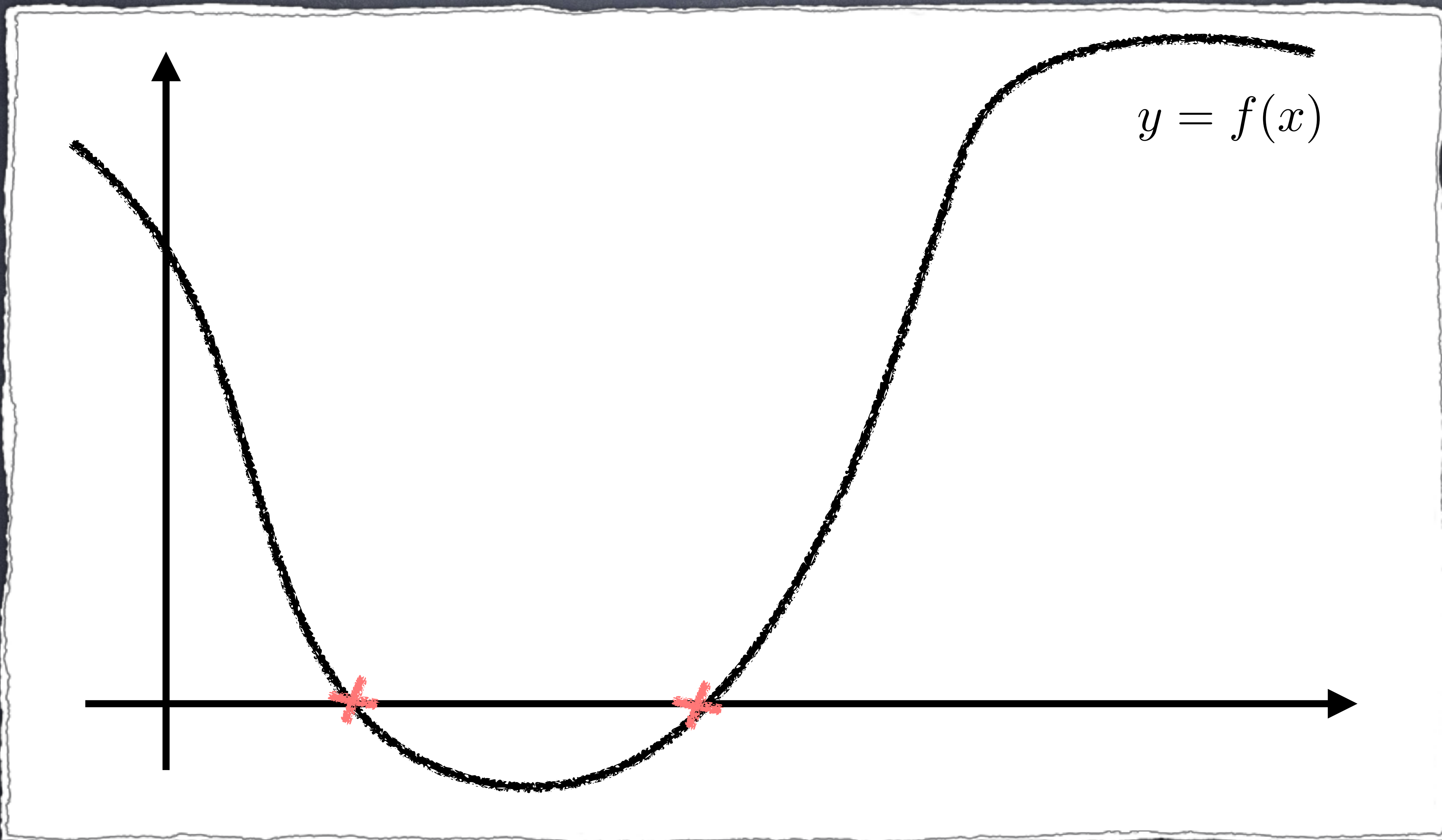
La méthode se généralise au cas de fonction $F : \mathbb{R}^n \to \mathbb{R}^n$

On doit alors résoudre à chaque itération le système linéaire :

$$(\underline{d}_{\underline{x}_n}\underline{F})\underline{x}_{n+1} = (\underline{d}_{\underline{x}_n}\underline{F})\underline{x}_n - \underline{F}(\underline{x}_n)$$

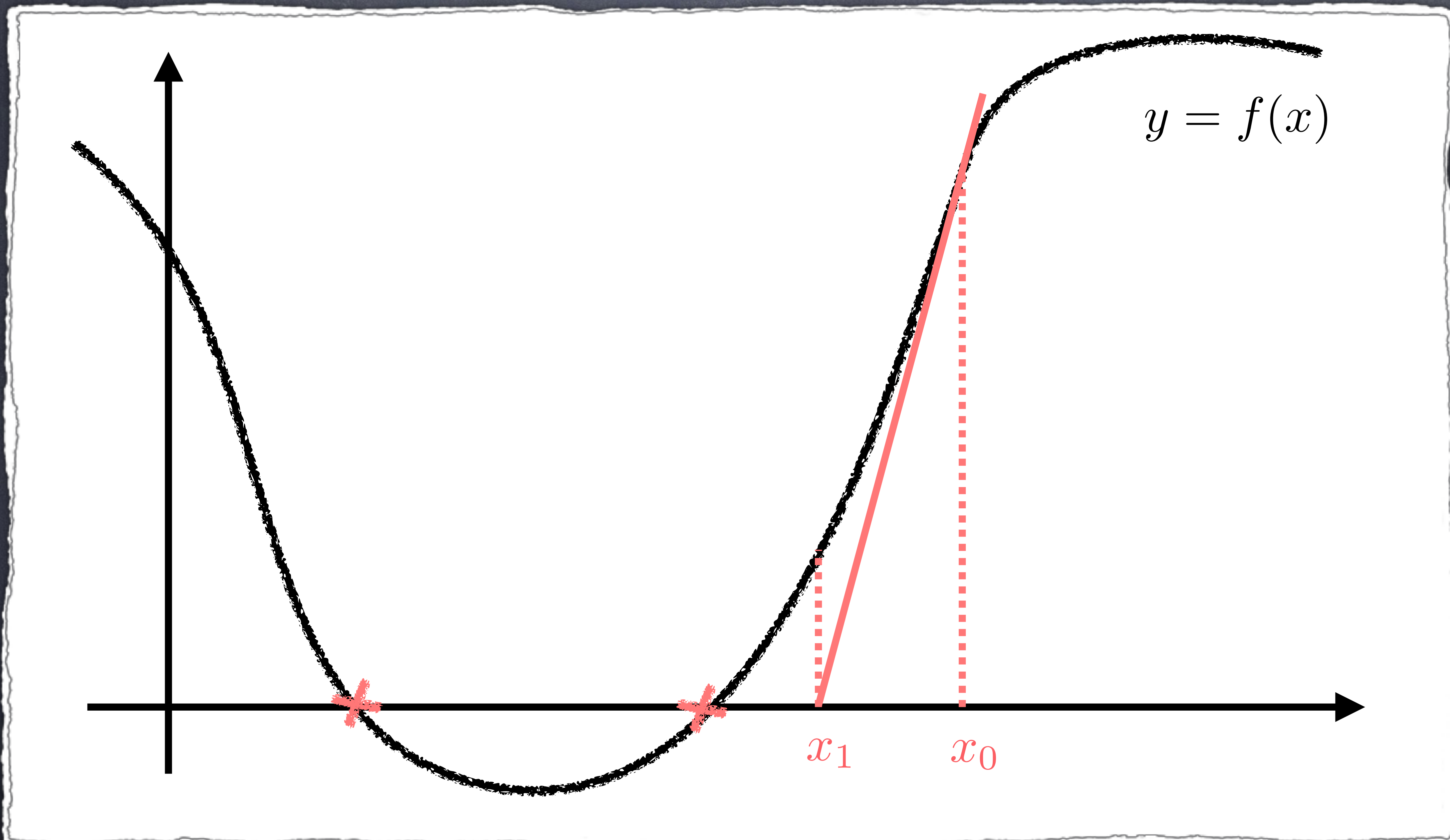
I. b) La méthode de Newton

Illustration de la méthode de Newton :



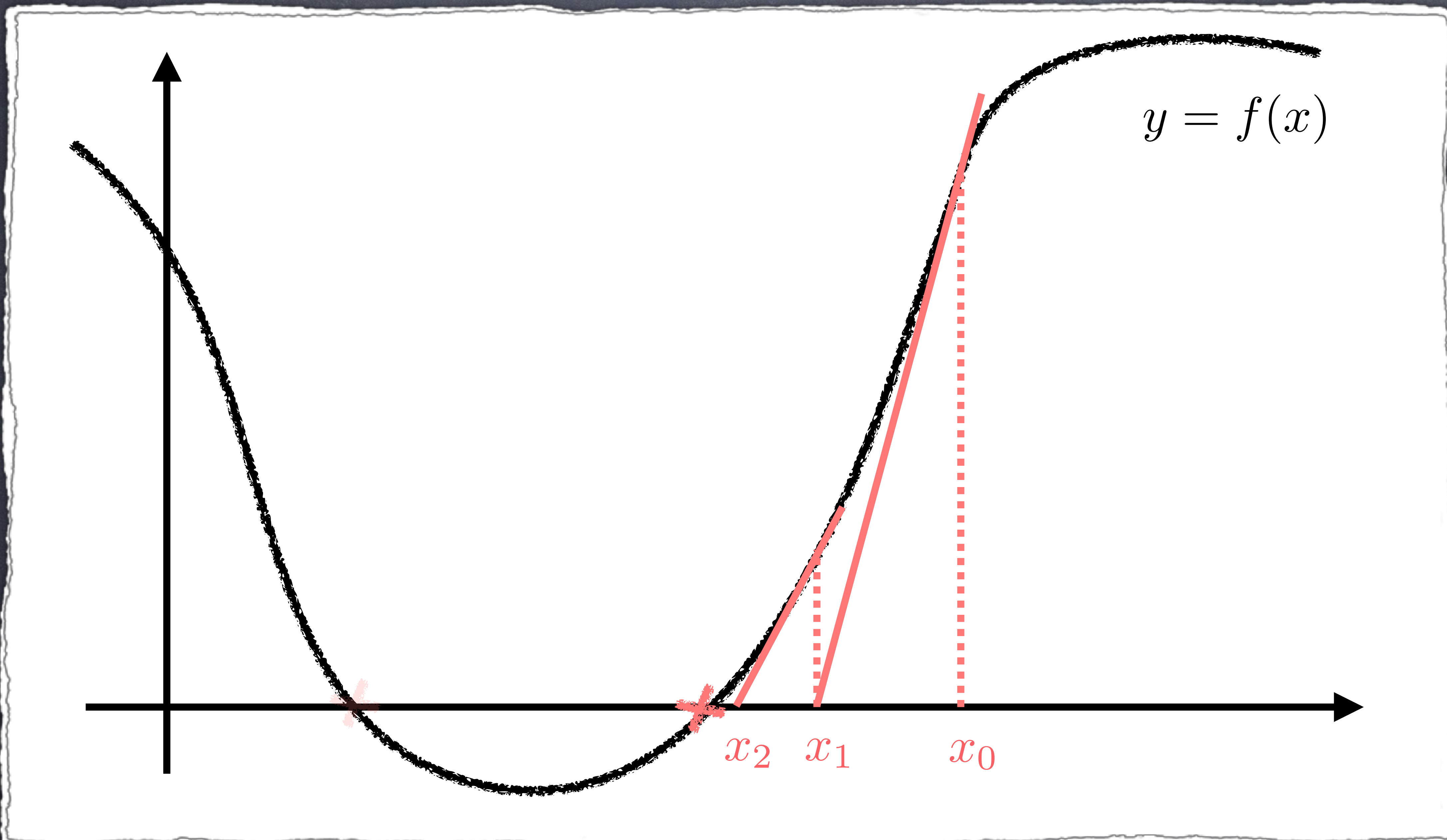
I. b) La méthode de Newton

Illustration de la méthode de Newton :



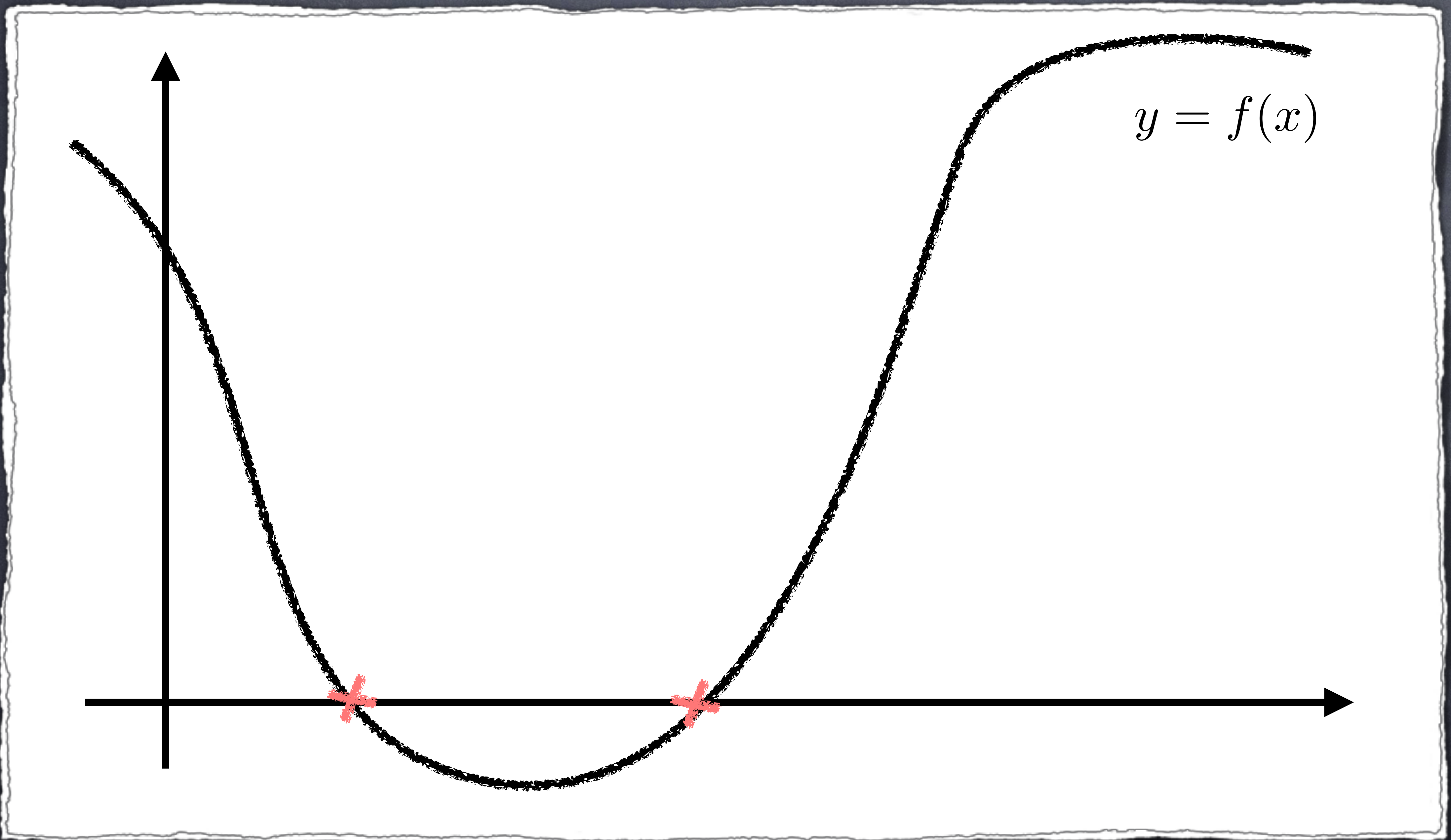
I. b) La méthode de Newton

Illustration de la méthode de Newton :



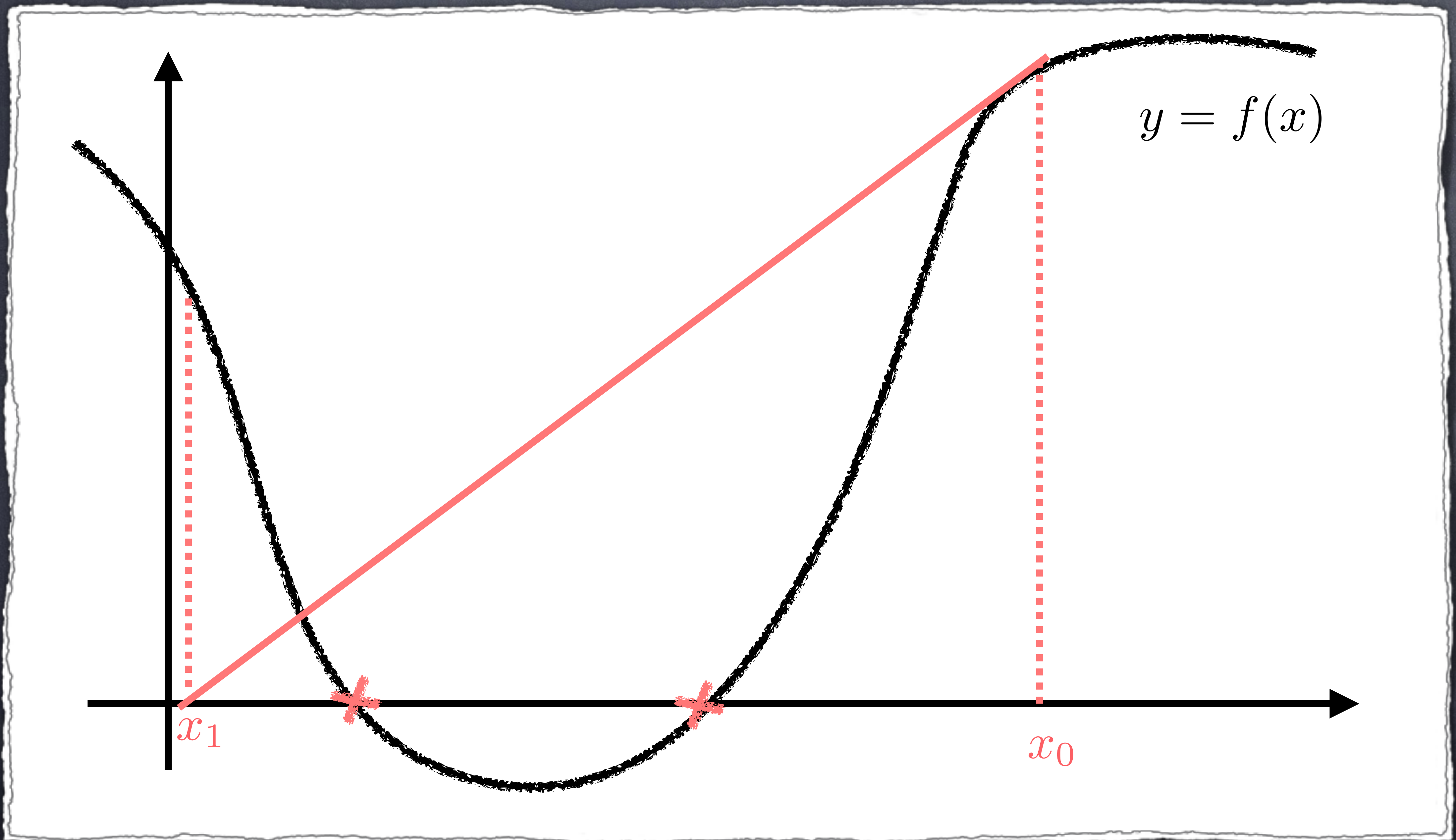
I. b) La méthode de Newton

Illustration de la méthode de Newton :



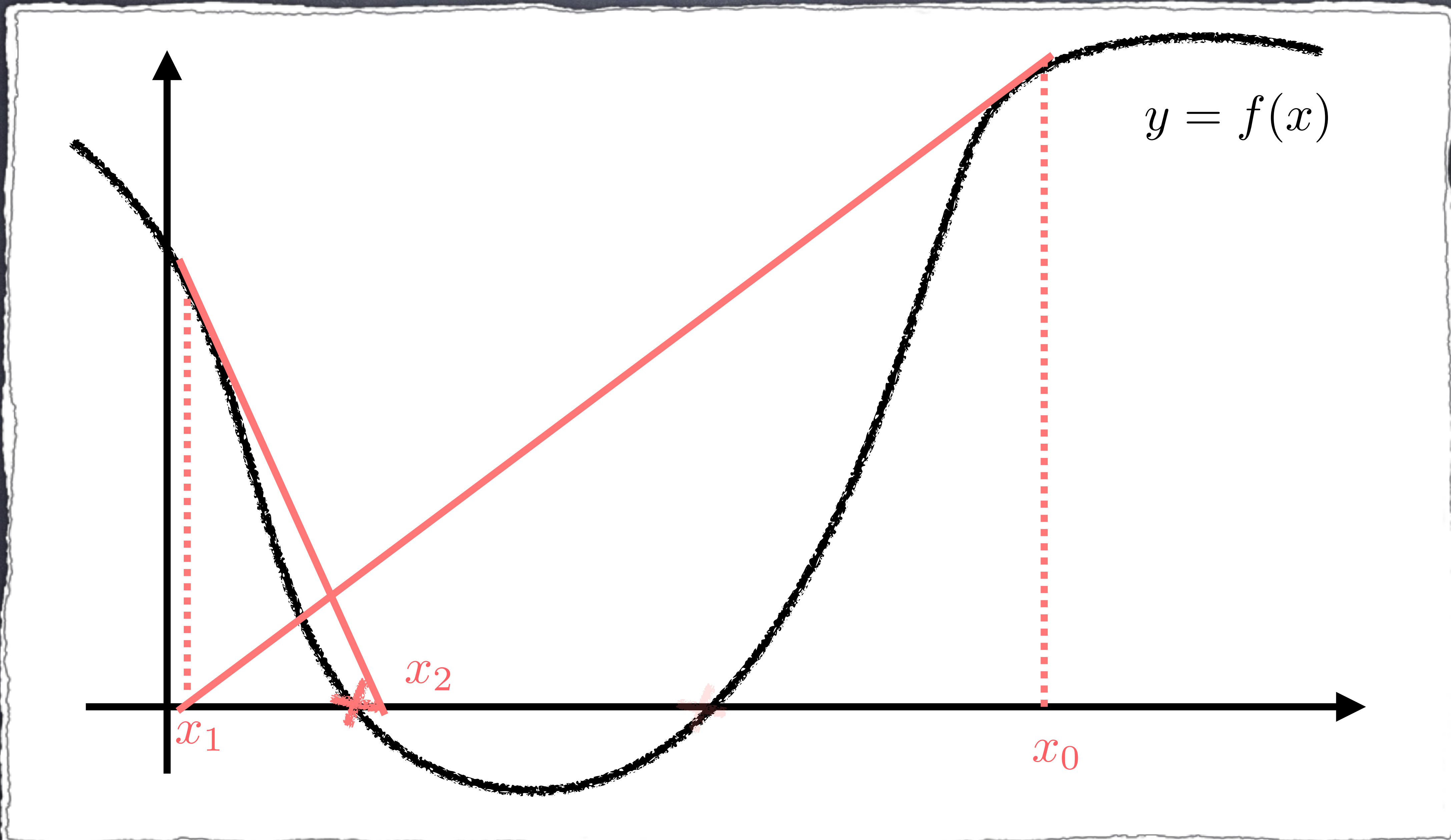
I. b) La méthode de Newton

Illustration de la méthode de Newton :



I. b) La méthode de Newton

Illustration de la méthode de Newton :



I. c) La méthode de la fausse position

Méthode de Newton :

On génère la suite $(x_n)_n$ ainsi : $x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$

Dans certaines situations, le calcul de $(d_{x_n}F)^{-1}$ est couteux voir impossible.

Une idée est alors d'utiliser l'approximation de la dérivée :

$$(d_{x_n}F) \simeq \frac{F(x_n) - F(x_{n-1})}{x_n - x_{n-1}}$$

On déduit alors la méthode de la fausse position

I. c) La méthode de la fausse position

Méthode de la fausse position :

On génère la suite $(x_n)_n$ ainsi :

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$$

I. c) La méthode de la fausse position

Méthode de la fausse position :

On génère la suite $(x_n)_n$ ainsi :

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$$

Remarques :

- À la différence de la méthode de Newton, il faut deux points pour initialiser la méthode de la fausse position.

I. c) La méthode de la fausse position

Méthode de la fausse position :

On génère la suite $(x_n)_n$ ainsi :

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$$

Remarques :

- À la différence de la méthode de Newton, il faut deux points pour initialiser la méthode de la fausse position.

- Le point x_{n+1} correspond au point où s'annule la droite définie par :

$$y = \frac{F(x_n) - F(x_{n-1})}{x_n - x_{n-1}} (x - x_n) + F(x_n)$$

I. c) La méthode de la fausse position

Méthode de la fausse position :

On génère la suite $(x_n)_n$ ainsi :

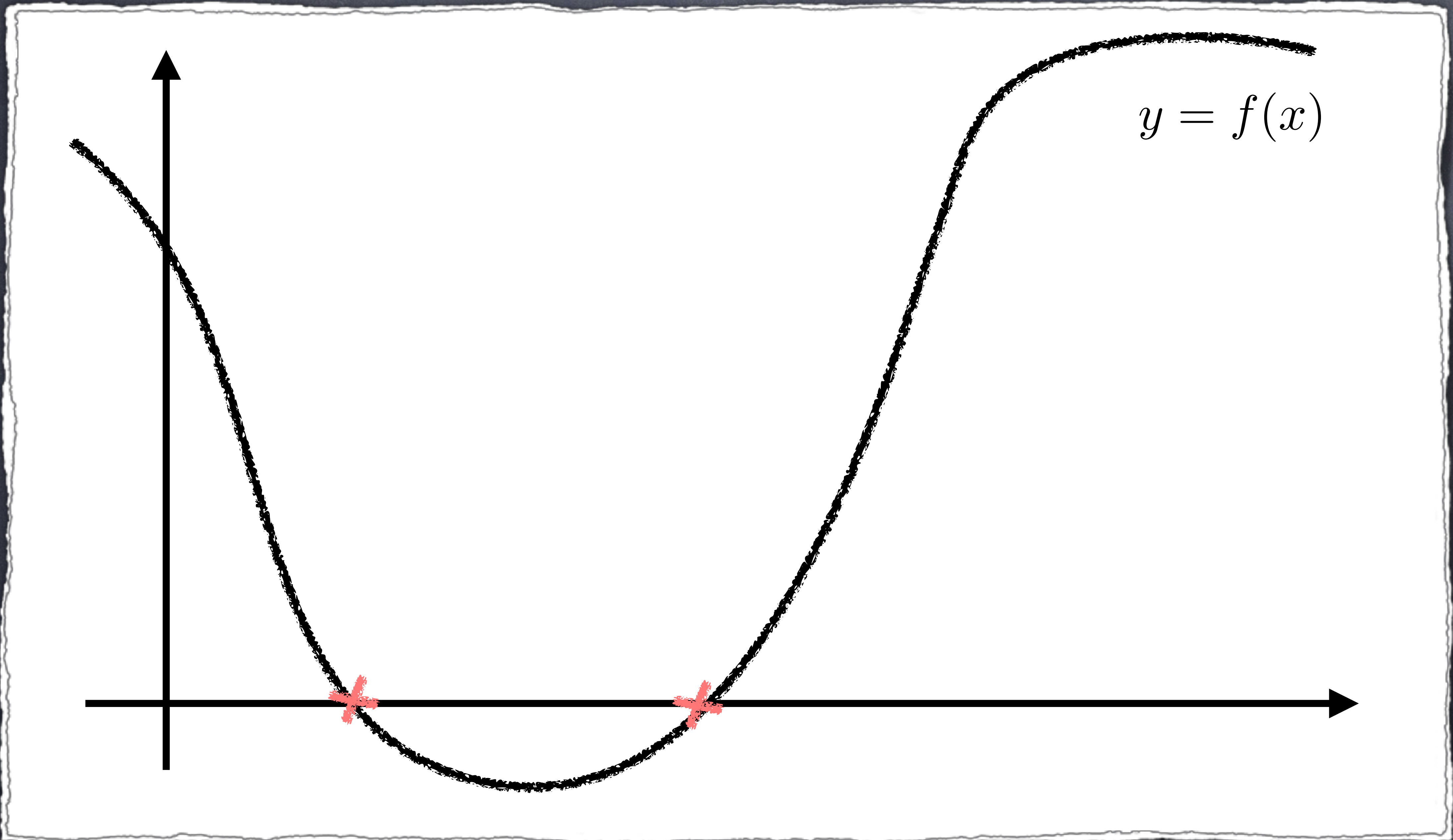
$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$$

1.9 Théorème (admis)

Si F est 2 fois continument différentiable, admet un 0 en x^* et $F'(x^*) \neq 0$, alors il existe un voisinage V dans lequel $\forall (x_0, x_1) \in V$ la méthode de F.P. converge et est d'ordre $(1+\sqrt{5})/2$ au moins.

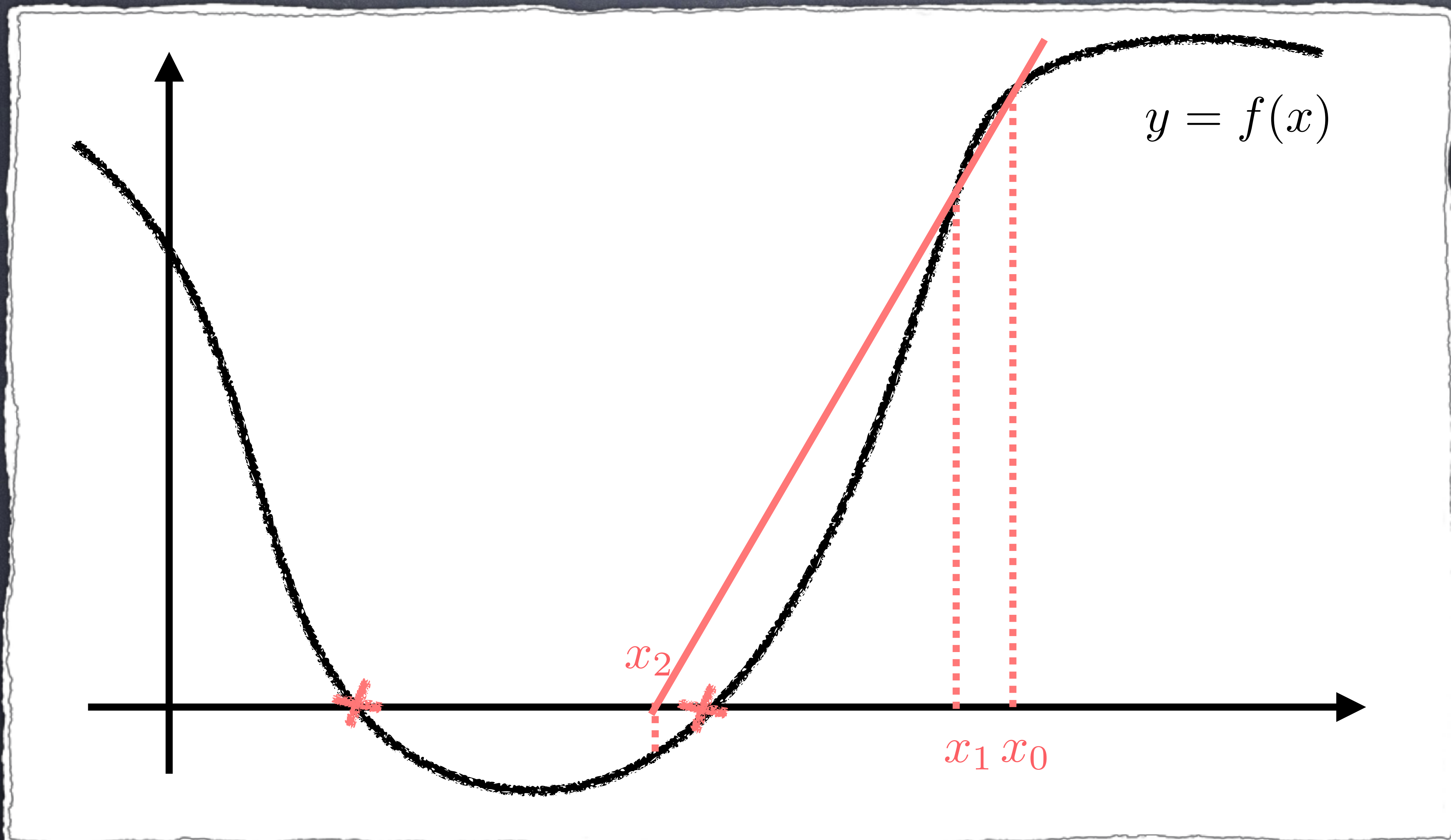
I. c) La méthode de la fausse position

Illustration de la méthode de la fausse position :



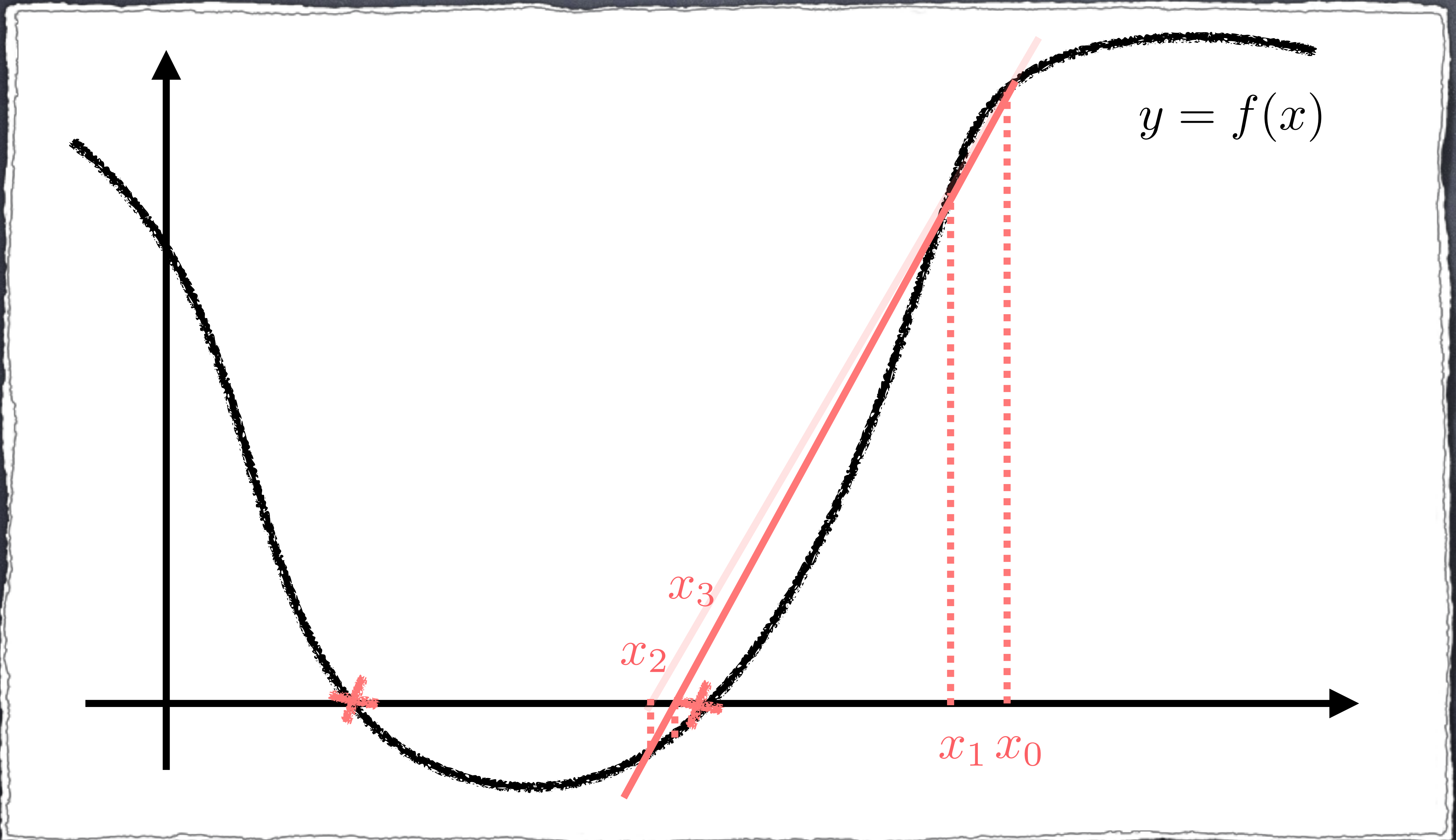
I. c) La méthode de la fausse position

Illustration de la méthode de la fausse position :



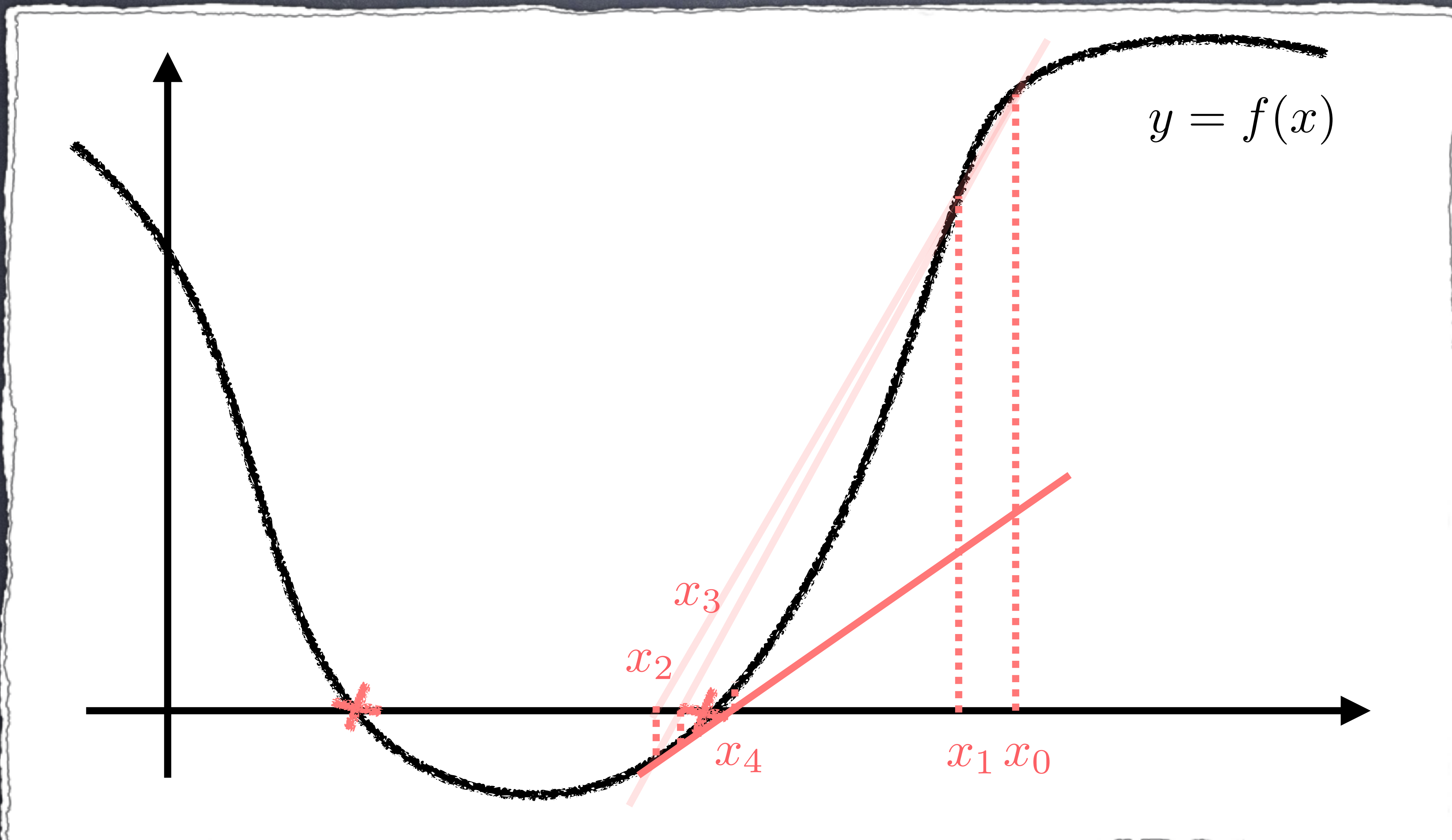
I. c) La méthode de la fausse position

Illustration de la méthode de la fausse position :



I. c) La méthode de la fausse position

Illustration de la méthode de la fausse position :



I. c) La méthode de la fausse position

Méthode de la fausse position :

Tant que $\|\underline{\mathsf{x}}_0 - \underline{\mathsf{x}}_1\| \geq \varepsilon$

$$\mathsf{x}_2 \leftarrow \mathsf{x}_1 - (\mathsf{x}_1 - \mathsf{x}_0) \times \mathsf{F}(\mathsf{x}_1)/(\mathsf{F}(\mathsf{x}_1) - \mathsf{F}(\mathsf{x}_0))$$

$$\mathsf{x}_0 \leftarrow \mathsf{x}_1$$

$$\mathsf{x}_1 \leftarrow \mathsf{x}_2$$

Equivalent mathématiques : $x_{n+1} = x_n - \dfrac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$

I. c) La méthode de la fausse position

Méthode de la fausse position :

Tant que $\|\underline{\mathsf{x}}_0 - \underline{\mathsf{x}}_1\| \geq \varepsilon$

$\mathsf{x}_2 \leftarrow \mathsf{x}_1 - (\mathsf{x}_1 - \mathsf{x}_0) \times \mathsf{F}(\mathsf{x}_1)/(\mathsf{F}(\mathsf{x}_1) - \mathsf{F}(\mathsf{x}_0))$

$\mathsf{x}_0 \leftarrow \mathsf{x}_1$

$\mathsf{x}_1 \leftarrow \mathsf{x}_2$

Equivalent mathématiques : $x_{n+1} = x_n - \dfrac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$

Remarque :

Pour la mise en oeuvre de la méthode, en particulier si le calcul de F(x) est couteux, il est intéressant de remarquer qu'à chaque étape une seule évaluation de F est nécessaire...

I. c) La méthode de la fausse position

Remarques de conclusion

- On est assuré de la convergence des méthodes de Newton et de F.P. seulement lorsque l'initialisation est dans un voisinage proche de la solution… Il faut donc avoir a priori une bonne solution approchée !

- Lorsque la racine est double, i.e. $F(x_*) = d_{x_*}F = 0$ les méthodes de Newton et F.P. convergent mais à un ordre plus petit (i.e. moins vite, cf TD !)

- On peut adapter la méthode de Newton pour résoudre des problèmes de minimisation (cf TD !).

Au programme (Chapitre 4)

Objectif :

Étudier des méthodes numériques pour résoudre le problème non linéaire :

$$\text{Trouver } \underline{x} \in \mathbb{R}^n \text{ t.q. } \underline{F}(\underline{x}) = \underline{0}$$

Plan :

I. Les équations non linéaires

II. Le cas des polynômes

a) L'algorithme de Horner

b) La méthode de Bairstow

c) Les polynômes orthogonaux

II. Le cas des polynômes

On cherche ici à déterminer les zéros (ou racines) d'un polynôme, i.e. :

$$\text{Trouver } x \in \mathbb{C} \text{ t.q. } P(x) = 0$$

où : $P(x) = \sum_{i=0}^{d} a_i x^i$ et $a_d \neq 0$

Remarque :

Évidemment, les méthodes de Newton et de F.P. peuvent s'appliquer au cas des polynômes. Néanmoins, ces méthodes ne permettent pas d'obtenir toutes les racines du polynômes.

II. Le cas des polynômes

On cherche ici à déterminer les zéros (ou racines) d'un polynôme, i.e. :

$$\text{Trouver } x \in \mathbb{C} \text{ t.q. } P(x) = 0$$

où : $P(x) = \sum_{i=0}^{d} a_i x^i$ et $a_d \neq 0$

2.1 Théorème Fondamentale (admis)

Si $d > 0$, alors le polynôme admet au moins une racine complexe. Par conséquent, tout polynôme de degré d admet d racines complexes (possiblement multiples).

II. Le cas des polynômes

On cherche ici à déterminer les zéros (ou racines) d'un polynôme, i.e. :

$$\text{Trouver } x \in \mathbb{C} \text{ t.q. } P(x) = 0$$

où : $P(x) = \sum_{i=0}^{d} a_i x^i$ et $a_d \neq 0$

Depuis la théorie de Galois, on sait que les équations polynomiales sont résolubles (on peut déterminer les racines) de manière générale si et seulement si le degré du polynôme est inférieur strictement à 5…

Cela motive d'autant plus l'utilisation de méthodes numériques pour déterminer les racines de P.

II. a) Algorithme de Horner

Voyons tout d'abord comment évaluer un polynôme $P(x)$ en un point s donné :

$$\begin{aligned} P(s) &= a_0 + a_1 s + \cdots + a_d s^d \\ &= a_0 + s\left(a_1 + a_2 s + \cdots + a_d s^{d-1}\right) \\ &= a_0 + s\left(a_1 + s\left(a_2 + \cdots + a_d s^{d-2}\right)\right) \\ &= a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(a_{d-1} + s a_d\right) \cdots \right)\right)\right) \end{aligned}$$

II. a) Algorithme de Horner

Voyons tout d'abord comment évaluer un polynôme P(x) en un point s donné :

$$
\begin{aligned}
P(s) &= a_0 + a_1 s + \cdots + a_d s^d \\
&= a_0 + s\left(a_1 + a_2 s + \cdots + a_d s^{d-1}\right) \\
&= a_0 + s\left(a_1 + s\left(a_2 + \cdots + a_d s^{d-2}\right)\right) \\
&= a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(a_{d-1} + s a_d\right) \cdots \right)\right)\right)
\end{aligned}
$$

L'idée est de factoriser autant que possible s afin de réduire le cout de calcul en évitant de calculer tous les s^i

II. a) Algorithme de Horner

Voyons tout d'abord comment évaluer un polynôme P(x) en un point s donné :

$$P(s) = a_0 + s\,(a_1 + s\,(a_2 + s\,(a_3 + \cdots + s\,(a_{d-1} + s a_d)\,\cdots)))$$

2.1 Proposition

En notant $q_{i-1} = a_i + s q_i \ \forall i \in \{1, d\}$ et $q_d = 0$, on a

$$P(x) = (x - s)Q(x) + r_0$$

où $Q(x) = q_0 + q_1 x + \cdots + q_{d-1} x^{d-1}$ et $r_0 = a_0 + s q_0 = P(s)$

Preuve : au (vrai) tableau ! $\blacksquare$

II. a) Algorithme de Horner

Algorithme d'évaluation de Horner :

$q_d \leftarrow 0$

Pour i=d **à** 1

  $q_{i-1} \leftarrow a_i + s q_i$

$r_0 \leftarrow a_0 + s q_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(a_{d-1} + s a_d\right) \cdots \right)\right)\right)$$

II. a) Algorithme de Horner

Algorithme d'évaluation de Horner :

$q_d \leftarrow 0$

Pour i=d **à** 1

$\quad q_{i-1} \leftarrow a_i + s q_i$

$r_0 \leftarrow a_0 + s q_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(\underbrace{a_{d-1} + s a_d}_{i\,=\,d-1}\right)\cdots\right)\right)\right)$$

II. a) Algorithme de Horner

Algorithme d'évaluation de Horner :

$q_d \leftarrow 0$

Pour i=d **à** 1

$\quad q_{i-1} \leftarrow a_i + s q_i$

$r_0 \leftarrow a_0 + s q_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + \underbrace{s\left(\underbrace{a_{d-1} + s a_d}_{i = d-1}\right)}_{i = d-2} \cdots\right)\right)\right)$$

II. a) Algorithme de Horner

Algorithme d'évaluation de Horner :

$\mathrm{q_d} \leftarrow 0$

Pour i=d **à** 1

  $\mathrm{q}_{i-1} \leftarrow \mathrm{a}_i + \mathrm{sq}_i$

$\mathrm{r}_0 \leftarrow \mathrm{a}_0 + \mathrm{sq}_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(\underbrace{a_{d-1} + s a_d}_{i=d-1}\right) \cdots \right)\right)\right)$$

$i = d-1$

$i = d-2$

$\vdots$

$i = 3$

II. a) Algorithme de Horner

Algorithme d'évaluation de Horner :

$q_d \leftarrow 0$

Pour i=d **à** 1

$\quad q_{i-1} \leftarrow a_i + sq_i$

$r_0 \leftarrow a_0 + sq_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\,(a_1 + s\,(a_2 + s\,(a_3 + \cdots + s\,(a_{d-1} + sa_d)\,\cdots)\,)\,)$$

Remarque :

L'algorithme de Horner permet donc deux choses :

1. Evaluer efficacement P en s
2. Déterminer Q(x) et R t.q. P(x) = (x-s) Q(x) + R

II. a) Algorithme de Horner

On appelle division Euclidienne d'un polynôme $P(x)$ par $D(x)$ l'opération consistant à déterminer $Q(x)$ et $R(x)$ t.q. :

$$P(x) = Q(x)D(x) + R(x)$$

où $\deg(R(x)) < \deg(D(x))$.

Exemple :

Calculer la division Euclidienne de :

$P(x) = 3x^3 - 8x^2 + 9x - 9$ par $Q(x) = x - 2$

II. a) Algorithme de Horner

La division euclidienne de P(x) par D(x) = (x-s) donne :

$$P(x) = (x - s)Q(x) + R$$

où :

- $P(x) = a_0 + a_1 x + \cdots + a_d x^d$
- $Q(x) = q_0 + q_1 x + \cdots + q_{d-1} x^{d-1}$

  où $\boxed{q_{i-1} = a_i + s q_i \ \ \forall i \in \{1, d\} \ \text{ et } \ q_d = 0}$
- $\boxed{R = r_0 = a_0 + s q_0}$

Pour déterminer une racine de P(x), on cherche s t.q. R(s) = 0.

II. a) Algorithme de Horner

Méthode de Newton-Horner :

La division euclidienne de P(x) par D(x) = (x-s) donne :

$$P(x) = (x - s)Q(x) + R$$

où $R = r_0 = a_0 + sq_0$ et on cherche s t.q. R(s) = 0.

II. a) Algorithme de Horner

Méthode de Newton-Horner :

La division euclidienne de $P(x)$ par $D(x) = (x-s)$ donne :

$$P(x) = (x-s)Q(x) + R$$

où $R = r_0 = a_0 + sq_0$ et on cherche s t.q. $R(s) = 0$.

Pour résoudre l'équation $R(s) = 0$, on va appliquer la méthode de Newton :

$$s_{n+1} = s_n - \frac{R(s_n)}{R'(s_n)}$$

II. a) Algorithme de Horner

Méthode de Newton-Horner :

La division euclidienne de $P(x)$ par $D(x) = (x-s)$ donne :

$$P(x) = (x-s)Q(x) + R$$

où $R = r_0 = a_0 + sq_0$ et on cherche s t.q. $R(s) = 0$.

Pour résoudre l'équation $R(s) = 0$, on va appliquer la méthode de Newton :

$$s_{n+1} = s_n - \frac{R(s_n)}{R'(s_n)} = s_n - \frac{P(s_n)}{Q(s_n)} \qquad \left(= s_n - \frac{P(s_n)}{P'(s_n)} \right)$$

On exploitera l'algorithme de Horner pour évaluer les polynômes P et Q en s_n.

II. a) Algorithme de Horner

Méthode de Newton-Horner :

La division euclidienne de $P(x)$ par $D(x) = (x-s)$ donne :

$$P(x) = (x-s)Q(x) + R$$

où $R = r_0 = a_0 + sq_0$ et on cherche s t.q. $R(s) = 0$.

Pour résoudre l'équation $R(s) = 0$, on va appliquer la méthode de Newton :

$$s_{n+1} = s_n - \frac{R(s_n)}{R'(s_n)} = s_n - \frac{P(s_n)}{Q(s_n)} \qquad \left(= s_n - \frac{P(s_n)}{P'(s_n)} \right)$$

On exploitera l'algorithme de Horner pour évaluer les polynômes P et Q en s_n.

Ensuite, pour déterminer les autres racines, on procédera de même sur le polynôme $Q(x)$ (déflation).

II. a) Algorithme de Horner

Algorithme de Newton-Horner :

Pour k=d **à** 1

  Tant que $\|s_0 - s_1\| \geq \varepsilon$

    $[P(s_0), Q] \leftarrow \text{Horner}(P, s_0)$

    $[Q(s_0), \sim] \leftarrow \text{Horner}(Q, s_0)$

    $s_1 \leftarrow s_0 - P(s_0)/Q(s_0)$

  $\text{racine}(k) = s_1$

  $P \leftarrow Q$

} $P(x) = (x - s_0)Q(x) + R$
Cal. $P(s_0)$ et $Q(s_0)$

} Itérations Newton

} Déflation

II. a) Algorithme de Horner

Algorithme de Newton-Horner :

Pour k=d **à** 1

  Tant que $\|s_0 - s_1\| \geq \varepsilon$

    $[P(s_0), Q] \leftarrow \text{Horner}(P, s_0)$

    $[Q(s_0), \sim] \leftarrow \text{Horner}(Q, s_0)$

    $s_1 \leftarrow s_0 - P(s_0)/Q(s_0)$

  $\text{racine}(k) = s_1$

  $P \leftarrow Q$

$P(x) = (x - s_0)Q(x) + R$
Cal. $P(s_0)$ et $Q(s_0)$

Itérations Newton

Déflation

Remarque :

Lors du calcul de $P(s)$ avec l'algorithme de Horner, on détermine le polynôme $Q(x)$.

II. a) Algorithme de Horner

Algorithme de Newton-Horner :

Pour k=d **à** 1

  Tant que $\|s_0 - s_1\| \geq \varepsilon$

    $[P(s_0), Q] \leftarrow \text{Horner}(P, s_0)$

    $[Q(s_0), \sim] \leftarrow \text{Horner}(Q, s_0)$

    $s_1 \leftarrow s_0 - P(s_0)/Q(s_0)$

  $\text{racine}(k) = s_1$

  $P \leftarrow Q$

} $P(x) = (x - s_0)Q(x) + R$
Cal. $P(s_0)$ et $Q(s_0)$

} Itérations Newton

} Déflation

Remarque 2 :

Pour déterminer les racines complexes, si le polynôme est à coefficients réels, il faut initialiser s complexe.

II. b) Algorithme de Bairstow

L'idée de la méthode de Bairstow est de chercher un polynôme de degré 2 $D(x) = x^2 - sx + p$ t.q. :

$$P(x) = Q(x)D(x) + R(x)$$

où le reste $R(x) = 0$.

II. b) Algorithme de Bairstow

L'idée de la méthode de Bairstow est de chercher un polynôme de degré 2 $D(x) = x^2 - sx + p$ t.q. :

$$P(x) = Q(x)D(x) + R(x)$$

2.2 Proposition

En notant :

- $Q(x) = q_0 + q_1 x + \cdots + q_{d-2} x^{d-2}$
- $R(x) = r_0 + r_1 x$

on a les relations suivantes :

$$q_{i-2} = a_i + s q_{i-1} - p q_i \quad \forall i \in \{2, \cdots, d\} \quad \text{et} \quad \begin{aligned} r_1 &= a_1 + s q_0 - p q_1 \\ r_0 &= a_0 - p q_0 \end{aligned}$$

où $q_d = q_{d-1} = 0$

Preuve : au (vrai) tableau ! $\blacksquare$

II. b) Algorithme de Bairstow

La division euclidienne de $P(x)$ par $D(x)$ donne :

$$P(x) = Q(x)D(x) + R(x)$$

où $D(x) = x^2 - sx + p$ et $R(x) = r_0 + r_1 x$ avec :

$$r_1(s,p) = a_1 + sq_0 - pq_1$$

$$r_0(s,p) = a_0 - pq_1$$

On cherche les valeurs de s et p annulant ces deux coefficients :

$$\begin{bmatrix} r_1(s,p) \\ r_0(s,p) \end{bmatrix} = \underline{0}$$

II. b) Algorithme de Bairstow

La division euclidienne de P(x) par D(x) donne :

$$P(x) = Q(x)D(x) + R(x)$$

où $D(x) = x^2 - sx + p$ et $R(x) = r_0 + r_1 x$ avec :

$$r_1(s,p) = a_1 + sq_0 - pq_1$$

$$r_0(s,p) = a_0 - pq_1$$

On cherche les valeurs de s et p annulant ces deux coefficients :

$$\begin{bmatrix} r_1(s,p) \\ r_0(s,p) \end{bmatrix} = \underline{0}$$

Comme précédemment, on va résoudre ce problème par la méthode de Newton.

Il est donc nécessaire de calculer $\partial_{s,p} r_1$ et $\partial_{s,p} r_0$

II. b) Algorithme de Bairstow

On a vu que :

$$q_{i-2} = a_i + sq_{i-1} - pq_i \ \forall i \in \{2, \cdots, d\} \quad \text{et} \quad \begin{array}{l} r_1 = a_1 + sq_0 - pq_1 \\ r_0 = a_0 - pq_0 \end{array}$$

dont on déduit :

$$\partial_s q_{i-2} = q_{i-1} + s\partial_s q_{i-1} - p\partial_s q_i \quad \text{et} \quad \partial_p q_{i-2} = s\partial_p q_{i-1} - q_i - p\partial_p q_i$$

II. b) Algorithme de Bairstow

On a vu que :

$$q_{i-2} = a_i + sq_{i-1} - pq_i \ \forall i \in \{2, \cdots, d\} \quad \text{et} \quad \begin{aligned} r_1 &= a_1 + sq_0 - pq_1 \\ r_0 &= a_0 - pq_0 \end{aligned}$$

dont on déduit :

$$\partial_s q_{i-2} = q_{i-1} + s\partial_s q_{i-1} - p\partial_s q_i \quad \text{et} \quad \partial_p q_{i-2} = s\partial_p q_{i-1} - q_i - p\partial_p q_i$$

puis :

$$\partial_s r_1 = q_0 + s\partial_s q_0 - p\partial_s q_1 \quad \text{et} \quad \partial_p r_1 = s\partial_p q_0 - q_1 - p\partial_p q_1$$

$$\partial_s r_0 = -p\partial_s q_0 \quad \text{et} \quad \partial_p r_0 = q_0 - p\partial_p q_0$$

II. b) Algorithme de Bairstow

Algorithme de Bairstow :

I. Calculer les coefficients des polynômes Q et R

$$q_{i-2} = a_i + s q_{i-1} - p q_i \quad \forall i \in \{2, \cdots, d\} \quad \text{et} \quad \begin{array}{l} r_1 = a_1 + s q_0 - p q_1 \\ r_0 = a_0 - p q_0 \end{array}$$

II. Calculer les dérivées partielles en s et p des q_i et r_i (cf slide précédent)

III. Mettre à jour les valeurs de s et p par la méthode de Newton.

IV. Répéter le I. tant que les coefficients du reste R sont supérieurs à un seuil

II. b) Algorithme de Bairstow

Quelques remarques :

- Dans la méthode de Bairstow, une fois les coefficients s et p du diviseur D(x) obtenus, on détermine 2 racines de P(x). On peut ensuite continuer l'algorithme en l'appliquant à Q(x) qui est de degré p-2

II. b) Algorithme de Bairstow

Quelques remarques :

- Dans la méthode de Bairstow, une fois les coefficients s et p du diviseur $D(x)$ obtenus, on détermine 2 racines de $P(x)$. On peut ensuite continuer l'algorithme en l'appliquant à $Q(x)$ qui est de degré $p-2$

- La méthode de Bairstow est comme la méthode de Newton, pour une racine simple, au moins d'ordre 2. Néanmoins, elle converge en général plus vite.

II. b) Algorithme de Bairstow

Quelques remarques :

- Dans la méthode de Bairstow, une fois les coefficients s et p du diviseur D(x) obtenus, on détermine 2 racines de P(x). On peut ensuite continuer l'algorithme en l'appliquant à Q(x) qui est de degré p-2
- La méthode de Bairstow est comme la méthode de Newton, pour une racine simple, au moins d'ordre 2. Néanmoins, elle converge en général plus vite.
- En cas de racine double, la méthode de Bairstow converge à l'ordre 2 alors que celle de Newton seulement à l'ordre 1.

II. c) Polynômes orthogonaux

2.3 Définition

On dit qu'une famille de polynômes $(P_k)_k$ est une famille de polynômes orthogonaux ssi elle vérifie une relation de récurrence à 3 termes :

$$P_k(x) = (A_k x + B_k)P_{k-1}(x) - C_k P_{k-2}(x) \quad \text{où} \quad \deg(P_k) = k$$

et (A_k), (B_k), (C_k) sont trois suites réelles, $P_0(x)$ et $P_1(x)$ étant deux polynômes de degrés 0 et 1 non nuls donnés.

II. c) Polynômes orthogonaux

2.3 Définition

On dit qu'une famille de polynômes $(P_k)_k$ est une famille de polynômes orthogonaux ssi elle vérifie une relation de récurrence à 3 termes :

$$P_k(x) = (A_k x + B_k)P_{k-1}(x) - C_k P_{k-2}(x) \quad \text{où} \quad \deg(P_k) = k$$

et (A_k), (B_k), (C_k) sont trois suites réelles, $P_0(x)$ et $P_1(x)$ étant deux polynômes de degrés 0 et 1 non nuls donnés.

Remarque :

Pour être exact, on définit les polynômes orthogonaux comme une famille de polynômes vérifiant une relation d'orthogonalité :

$$\int_a^b P_k(x)P_j(x)W(x)dx = \delta_{kj}$$

II. c) Polynômes orthogonaux

2.4 Proposition

Soit $(P_k)_k$ une suite de polynôme orthogonaux t.q. $C_k > 0$

Le polynôme $P_n(x)$ admet alors n racines distincts. De plus, ces racines sont séparées par les racines de $P_{n-1}(x)$

II. c) Polynômes orthogonaux

Exemple :

On chercher à déterminer les valeurs propres d'une matrice tridiagonale A via le développement de son déterminant :

$$\det(A - \lambda Id) = \det \begin{bmatrix} a_1 - \lambda & b_1 & 0 & \cdots & \cdots & 0 \\ c_2 & a_2 - \lambda & b_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & c_{k-1} & a_{k-1} - \lambda & b_{k-1} \\ 0 & \cdots & \cdots & 0 & c_k & a_k - \lambda \end{bmatrix}$$

(cf détails au (vrai) tableau)

Plan détaillé du chapitre 4

Plan :

I. Les équations non linéaires

a) Les équations de point fixe

b) La méthode de Newton

c) La méthode de la fausse position

II. Le cas des polynômes

a) L'algorithme de Horner

b) La méthode de Bairstow

c) Les polynômes orthogonaux

Quelques images (fractales Newton)