Problèmes stationnaires à une dimension du type $f(x) = 0$

Ingénierie numérique et simulations

LYCÉE CARNOT (DIJON), 2013 - 2014

Germain Gondor

Sommaire

1. Introduction
2. Convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
6. Dilemme robustesse/rapidité

Sommaire

1. Introduction
   - Objectifs
   - Support de l'étude
   - Linéarité
   - Non linéarité
2. Convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
6. Dilemme robustesse/rapidité

Objectifs

L'objectif de ce cours est de traiter des problèmes stationnaires (constant au cours du temps), linéaires ou non, conduisant à la résolution approchée d'une équation algébrique ou transcendante.

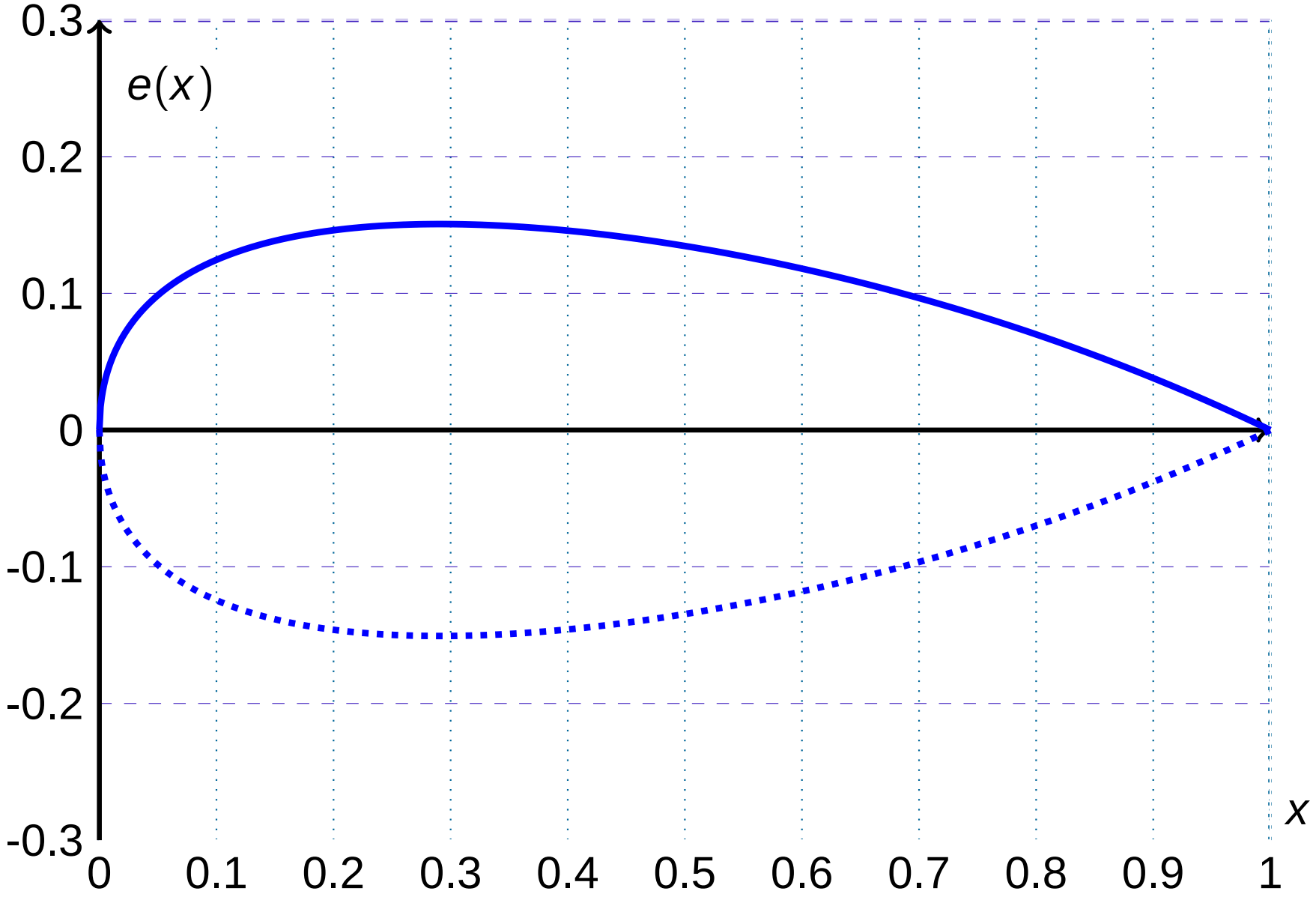
Support de l'étude

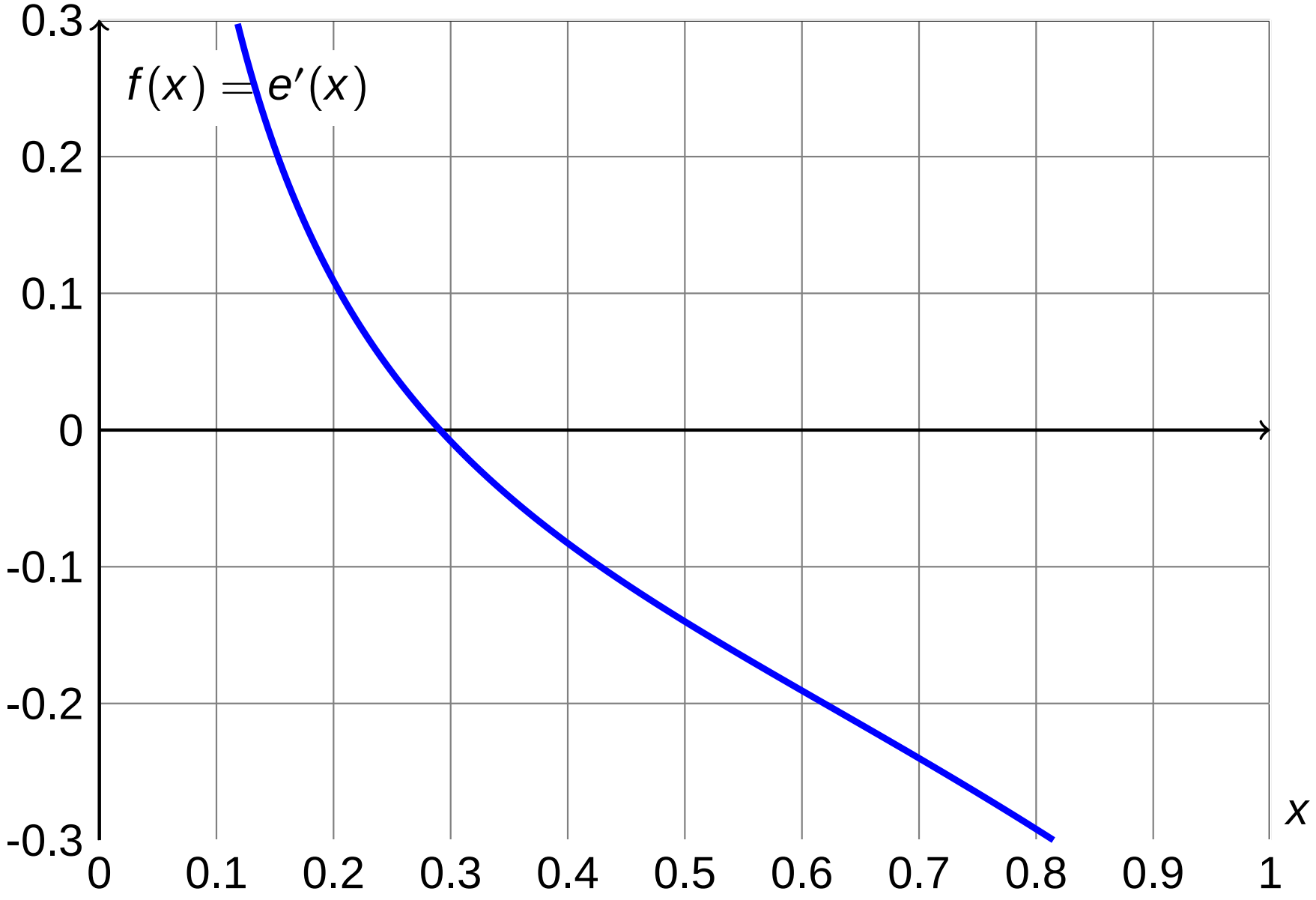
Comme support de cours, nous prenons un profil d'aile d'avion dont la demi-épaisseur $e(x)$ est approchée par l'équation:

$$e(x) = 0,15 \times \left(3,7 \times \sqrt{x} - 3,4 \times x - 0,3 \times x^4\right), \forall x \in [0,1]$$

Rechercher la valeur de x pour laquelle le profil est le plus épais, revient à résoudre l'équation :

$$f(x) = 0,15 \times \left(\frac{3,7}{2.\sqrt{x}} - 3,4 - 1,2 \times x^3\right) = 0$$


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
e(x)
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


$f(x) = e'(x)$
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x

Linéarité

Soit f une fonction définie de E, un espace vectoriel, dans K, un corps commutatif. f est linéaire, si elle vérifie la propriété suivante:

$$\forall (x,y) \in E^2, \forall (\lambda,\mu) \in K^2, f(\lambda.x + \mu.y) = \lambda.f(x) + \mu.f(y)$$

EXEMPLE :

En 1D: Equation de droite

$$f : \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R} \\ x & \mapsto & a.x + b \end{array}$$

avec $(a,b) \in \mathbb{R}^2$

En 2D: Définition d'un plan

$$f(x,y,z) = a.x + b.y + c.z + d$$

La forme linéaire $f(x,y,z) = 0$ définit un plan où le vecteur (a,b,c) est un vecteur normal à ce plan.

Résoudre une équation linéaire 1D dans $\mathbb{R}$ ou $\mathbb{C}$ est immédiat:

$$\forall a \neq 0,\ a.x + b = 0 \quad \Rightarrow \quad x = -\frac{b}{a}$$

EXEMPLE : $3.i.x - 12.i + 6 = 0 \quad \Rightarrow \quad x = 4 + 2.i$

m

Résoudre un système linéaire de n équations à n inconnues suppose la résolution d'un système matriciel du type:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} . \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad \Leftrightarrow \quad [A].[X] = [B] \overset{\det(A) \neq 0}{\Rightarrow} [X] = [A$$

Nous verrons dans un chapitre suivant comment le résoudre par un pivot de Gauss.

Non linéarité

Les cas non-linéaires sont plus difficiles à résoudre. Parfois une résolution analytique est possible comme dans la résolution d'une équation du second degré :

EXEMPLE :

$$x^2 - x - 6 = 0 \quad \Rightarrow \quad \Delta = (-1)^2 - 4.1.(-6) = 25 = 5^2$$

$$\Rightarrow \quad \begin{cases} x_+ & = & \dfrac{-(-1)+5}{2} = 2 \\ x_- & = & \dfrac{-(-1)-5}{2} = -3 \end{cases}$$

Dans d'autre cas, la solution analytique n'est pas connue (exemple : profil d'aile d'avion). On utilise alors une approche numérique pour la calculer. C'est l'objet de ce cours.

Sommaire

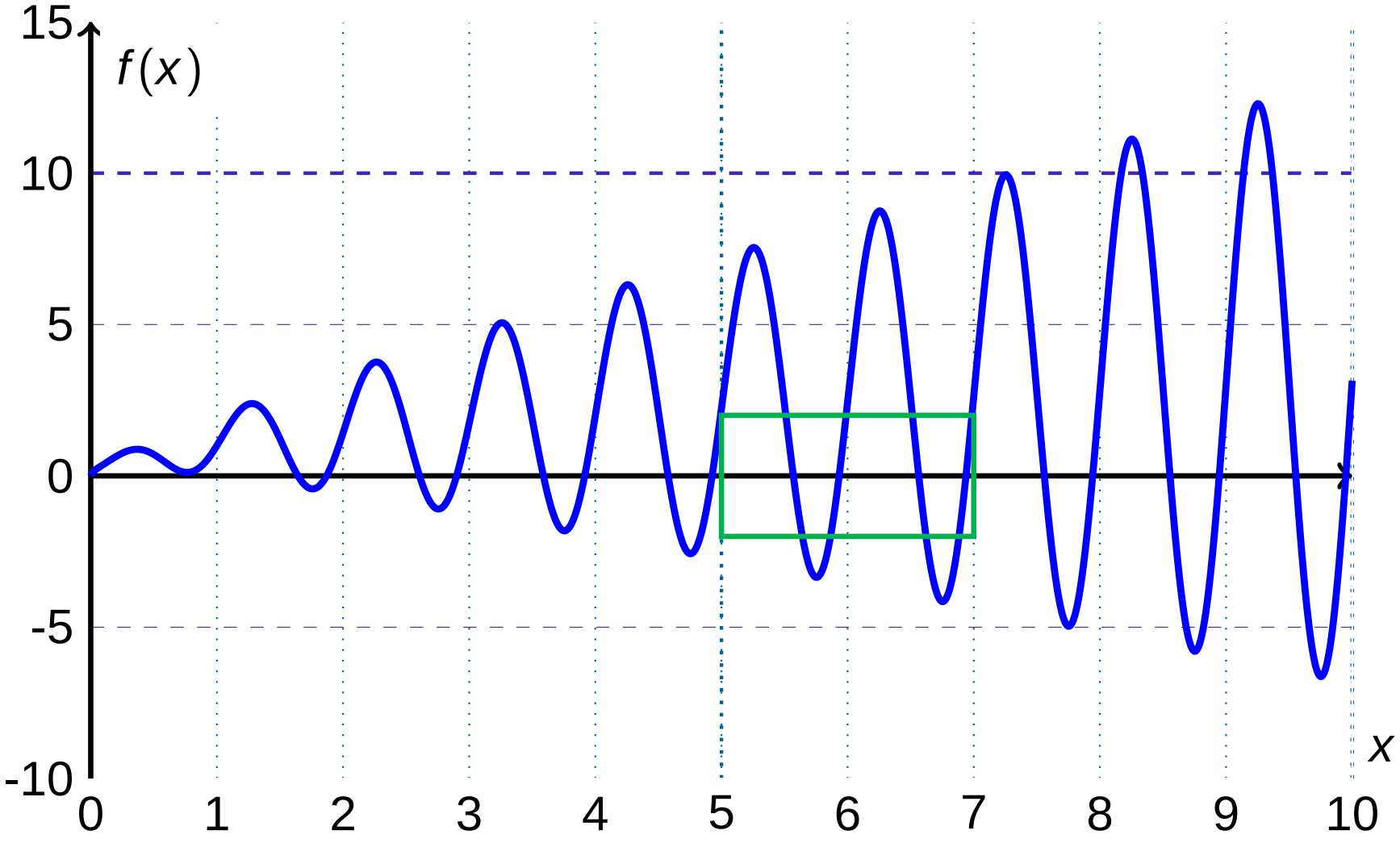
1. Introduction
2. Convergence
   - Séparation des racines
   - Représentations graphiques
   - Critères de convergence
   - Ordre de convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
6. Dilemme robustesse/rapidité

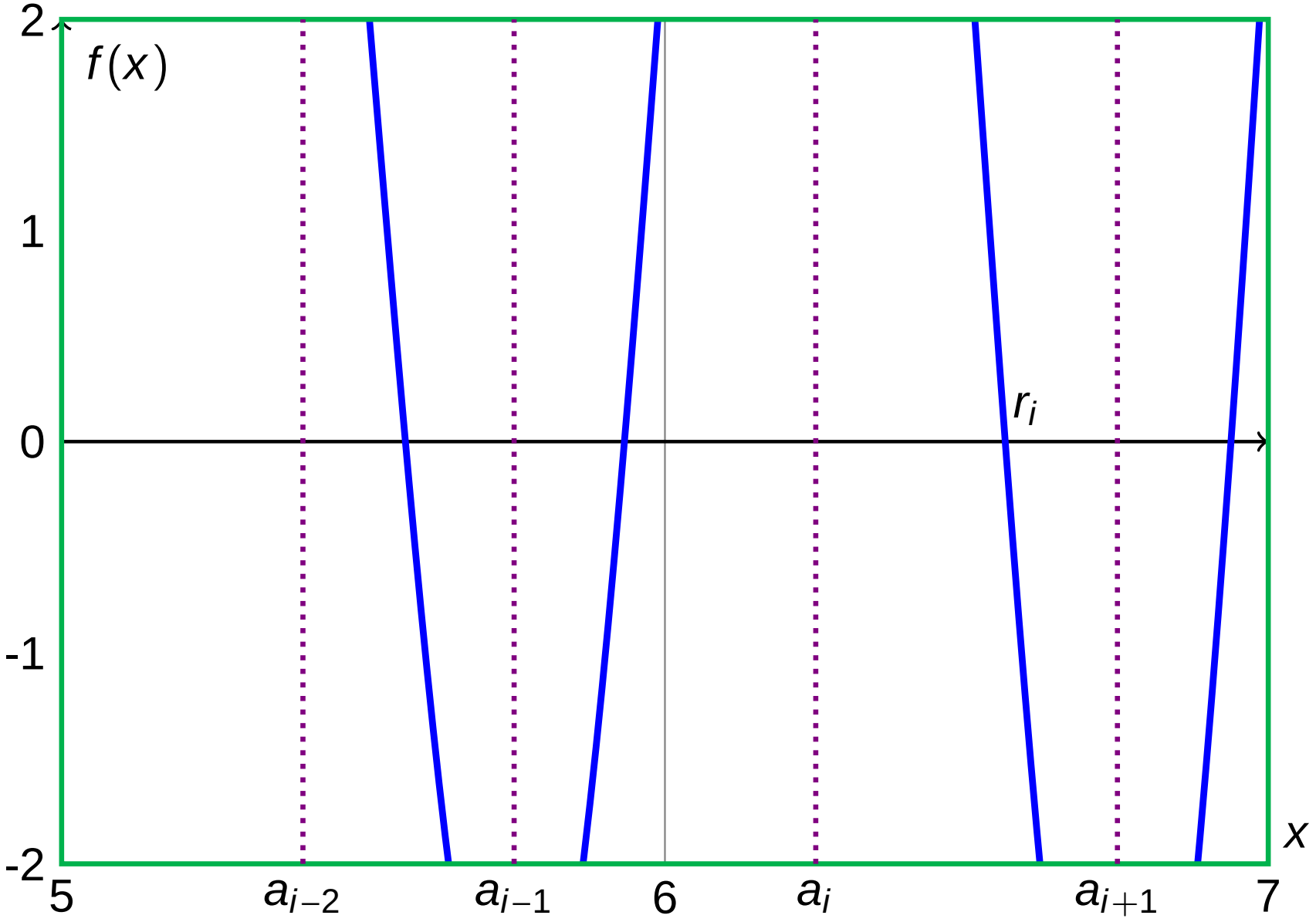
Séparation des racines

La plupart des méthodes de recherche de racines d'une fonction, se comportent bien si une seule racine r (valeur pour laquelle la fonction s'annule) est présente dans l'intervalle d'étude.

Séparer la racines r_i (ième annulation de la fonction sur l'intervalle de départ) revient à trouver un intervalle $]a_i, a_{i+1}[$ où cette racine est unique.

EXEMPLE : $f(x) = \sqrt{x} + x.\sin(2.\pi.x)$




2
1
0
-1
-2
$f(x)$
r_i
x
5
a_{i-2}
a_{i-1}
6
a_i
a_{i+1}
7

Il convient alors de faire un test aux bornes de l'intervalle d'étude pour éliminer les cas d'inadéquation de la comparaison à zéro, i.e, les cas sans racines sur l'intervalle. Il est en effet, possible d'élaborer une procédure permettant de déterminer la parité du nombre de racines sur l'intervalle :

- Posons $p_i = f(a_i).f(a_{i+1})$
- En tenant compte de l'ordre de multiplicité des racines, nous avons:
  - si $p_i < 0$, il existe $2.n + 1$ racines dans $]a_i, a_{i+1}[,\ n \in \mathbb{N}$
  - si $p_i > 0$, il existe $2.n$ racines dans $]a_i, a_{i+1}[,\ n \in \mathbb{N}$
  - si $p_i = 0$, alors a_i ou a_{i+1} est racine (voire les deux).

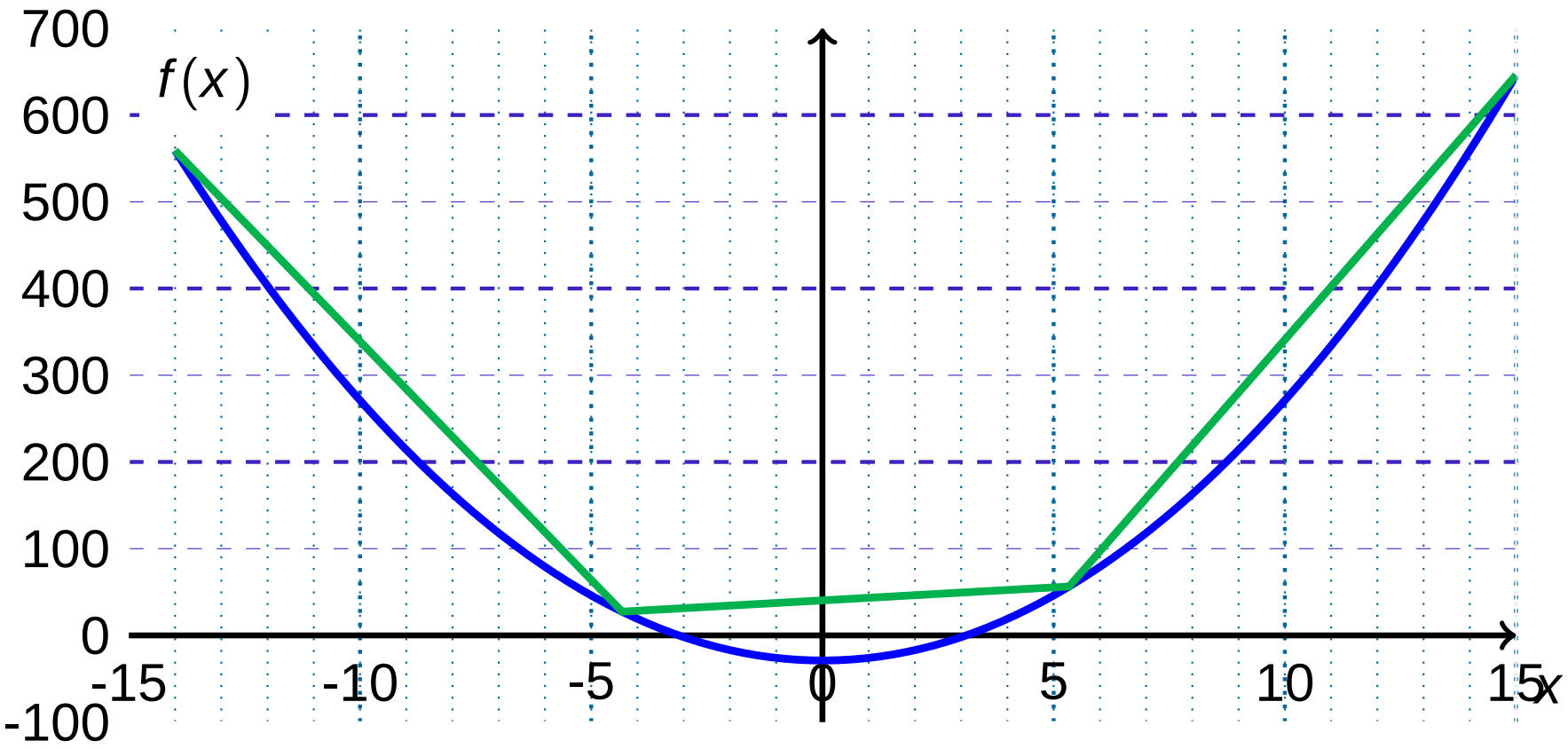
Evidemment, cet algorithme ne donne pas la valeur de n. Nous verrons plus loin l'utilité de ce test, notamment dans la recherche de la racine par dichotomie.

Représentations graphiques

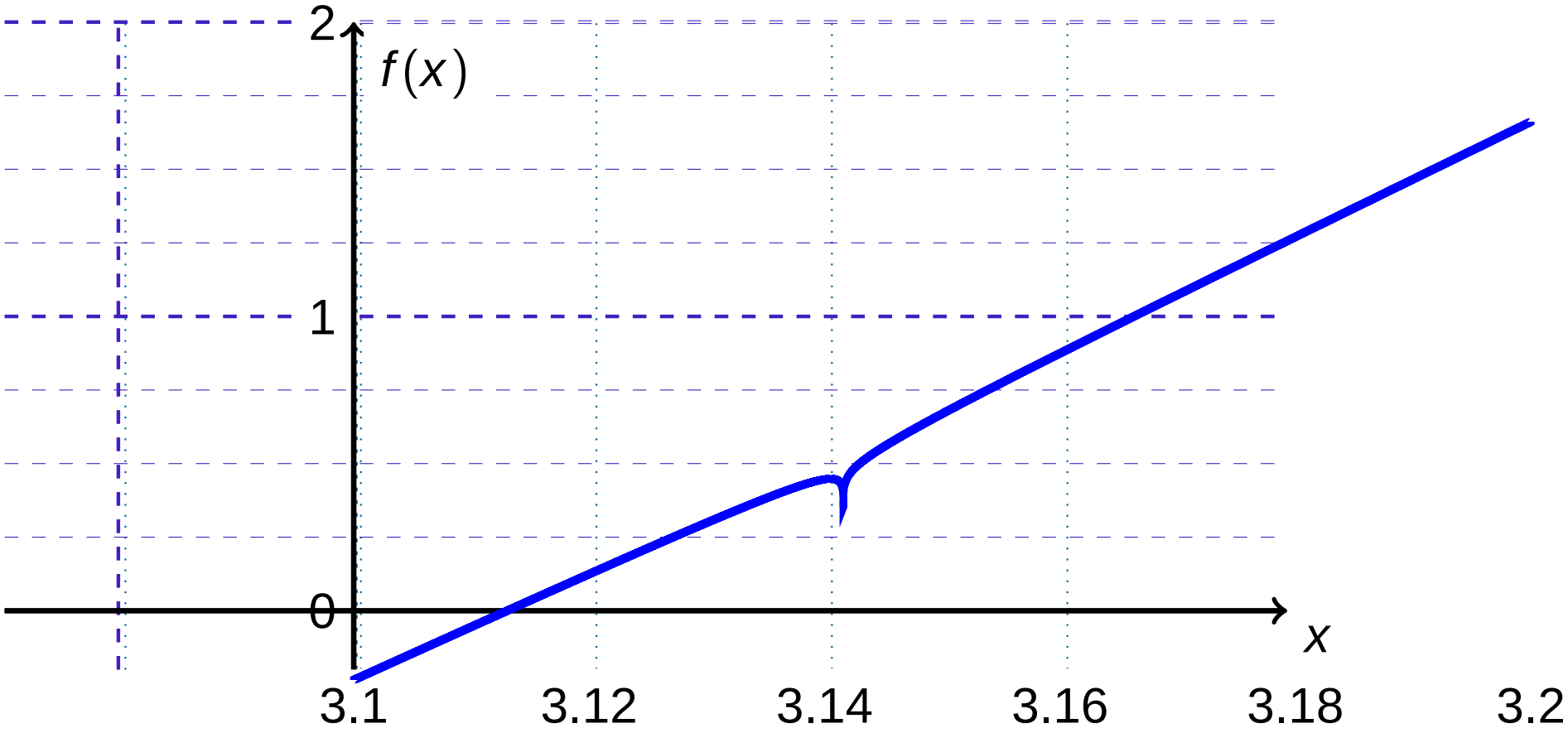
Les représentations graphiques sont utiles parfois pour avoir une idée des lieux de convergence. Cela permet d'obtenir à vu d'œil un premier intervalle ou un premier candidat pour démarrer une méthode numérique.

Cependant, il convient aussi de se méfier des représentations graphiques. Pour cela, il est préférable d'observer attentivement l'expression de la fonction et son domaine de définition.

EXEMPLE : $f(x) = 3.x^3 + \frac{1}{\pi^4}.\ln\left[(\pi - x)^2\right] - 29$



Courbes tracées entre -14 et 15. La courbe verte n'ayant que 4 points, on ne constate pas d'intersection avec l'axe des abscisses.



Avec le zoom sur l'intervalle $[3,1;3,2]$, il semble encore qu'il n'y ait qu'une racine sur l'intervalle. Or $\lim_{x \mapsto \pi} f(x) = -\infty$.

Par continuité sur les intervalle $[3,14;\pi[$ et $[\pi;3,15]$, la fonction f admet donc 3 racines dans l'intervalle $[3,1;3,2]$.

Critères de convergence

Les méthodes numériques ne permettent pas d'obtenir une réponse formelle à un problème.

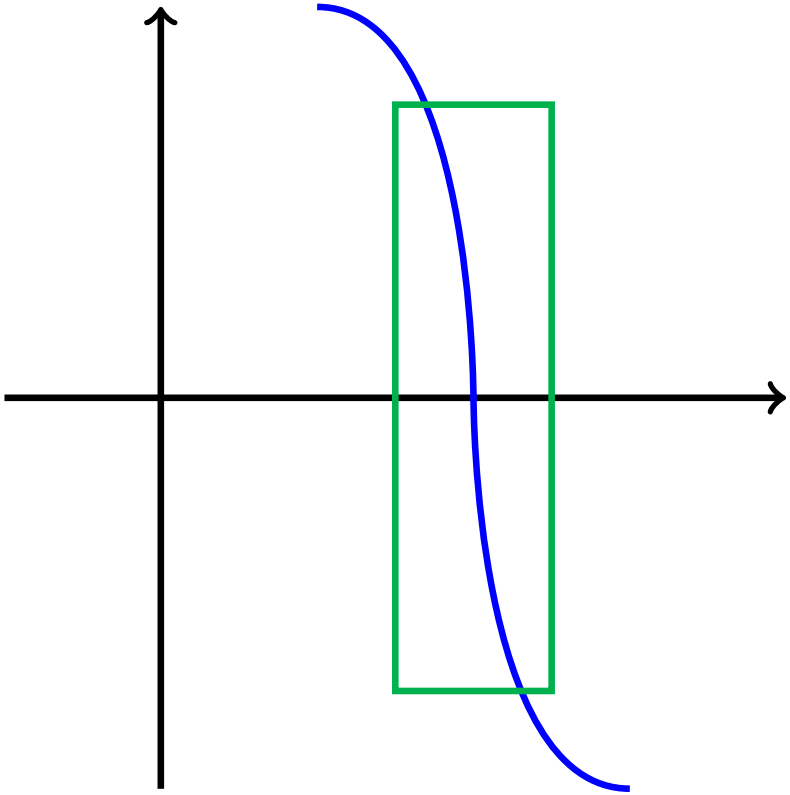
En revanche, elles permettent d'approcher la solution d'un problème avec une précision fonction du type de stockage des nombres (cf premier semestre) et du calcul numérique nécessaire pour obtenir cette solution (cf la sensibilité de certains calculs aux erreurs d'arrondis).

Pour une erreur normée en abscisse (fonction de l'"ordre de grandeur" de x), une tolérance de 10^{-8} est raisonnable. En revanche, chercher à obtenir 10^{-16} est a priori non nécessaire et impossible.

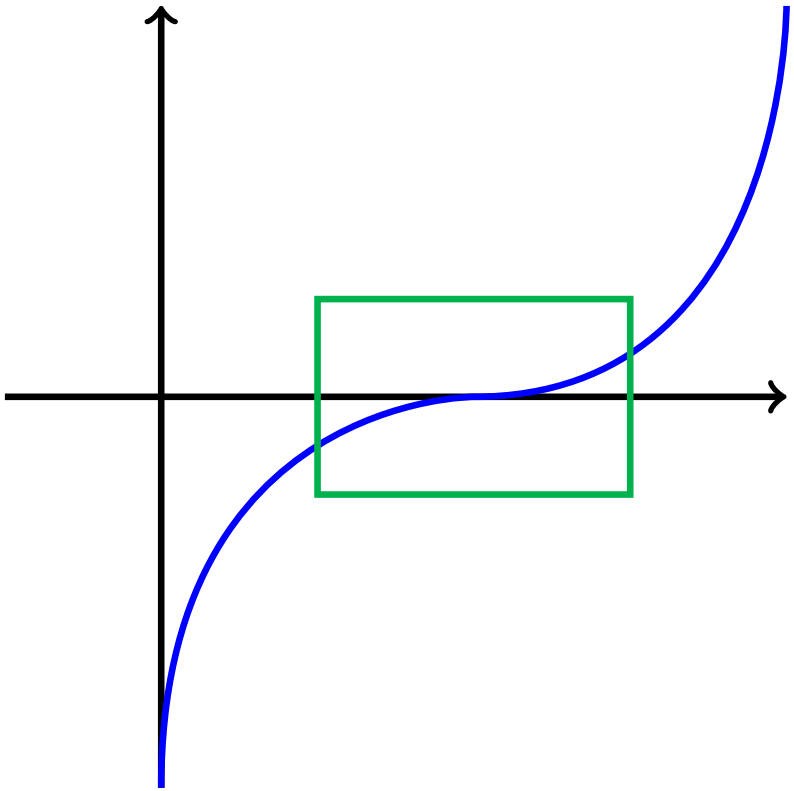
Dans le cas de la méthode par dichotomie, il est possible de faire un test de convergence sur la valeur de x et/ou sur la valeur de la fonction:

$$\left|\frac{b_n - a_n}{a_n + b_n}\right| < \varepsilon_x \quad \text{et/ou} \quad |f(c_n)| < \varepsilon_f$$

Suivant le gradient (pente) de la fonction, un des deux tests sera privilégiés. Pour un fort (faible) gradient, il convient de suivre la valeur de f (de x).



Fort gradient



Faible gradient

En absence d'information, on peut suivre les deux (si la méthode donne une information sur l'erreur en x).

Ordre de convergence

Soit x_n la énième valeur calculée pour résoudre l'équation $f(x) = 0$ sur l'intervalle donné. Soit r la solution exacte de cette équation. On peut alors définir ε_n comme l'erreur en abscisse à l'itération n.

La méthode numérique est qualifié de méthode à l'ordre p si: $\exists\ a \in \mathbb{N} / \forall n \in \mathbb{N}, n > a \quad \Rightarrow \quad \varepsilon_{n+1} \approx \varepsilon_n^p$
Si $p = 1$, la méthode est dite linéaire. Si $p > 1$, la méthode est qualifié de super-linéaire. Enfin, la méthode est quadratique si $p = 2$.

Sommaire

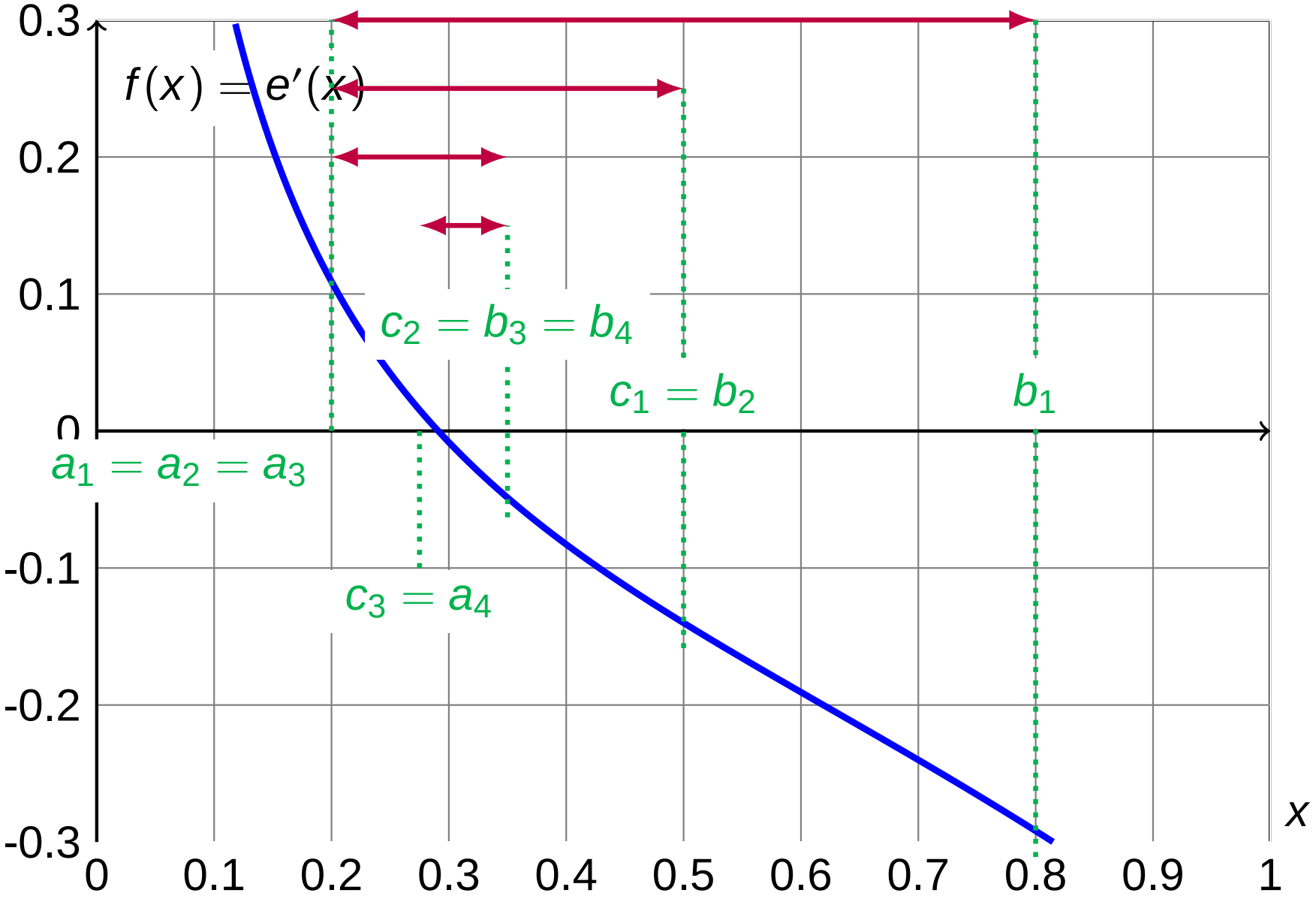
1. Introduction
2. Convergence
3. Rappel sur la méthode par dichotomie
   - Principe
   - Algorithme
   - Convergence de la méthode
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
6. Dilemme robustesse/rapidité

Principe

Le principe est de diviser l'intervalle en deux parts égales, de conserver la partie contenant la racine et d'y reproduire l'opération jusqu'à ce que le critère de convergence soit satisfait.

A partir d'un intervalle donné $[a, b]$, encadrant une racine de la fonction f étudiée :

- Calculer le point c milieu de l'intervalle : $c = \dfrac{a+b}{2}$
- Evaluer $p = f(a).f(c)$ puis test :
  - si $p > 0$, il n'y a pas de racine dans l'intervalle $[a, c]$. La racine est donc dans l'intervalle $[c, b]$. On donne alors à a la valeur de c
  - si $p < 0$, la racine est dans $[a, c]$. On donne alors à b la valeur de c
  - si $p = 0$, alors la racine est c. Ô Miracle,...
- test d'arrêt : Evaluer le critère de convergence
- Recommencer si le critère de convergence n'est pas satisfait


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
$f(x) = e'(x)$
$c_2 = b_3 = b_4$
$c_1 = b_2$
b_1
$a_1 = a_2 = a_3$
$c_3 = a_4$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x

Algorithme

Méthode par dichotomie pour la fonction f sur $[a, b]$ avec une tolérance ε :

Entrées : f, a, b, ε
$ai, bi \leftarrow a, b$
$fa, fb \leftarrow f(ai), f(bi)$
tant que $bi - ai > 2.\varepsilon$ **faire**
  $ci \leftarrow (ai + bi)/2$
  $fc \leftarrow f(ci)$
  si $fa.fc \geq 0$ **alors**
    $bi \leftarrow ci$
    $fd \leftarrow fc$
  sinon
    $ai \leftarrow ci$
    $fa \leftarrow fc$
Sorties : $\dfrac{ai + bi}{2}$

```
def Dichotomie(f,a,b,eps):
assert f(a)*f(b)<=0 and eps >0
ai,bi=a,b
fa,fb=f(ai),f(bi)
while bi-ai>2*eps:
c=(ai+bi)/2.
fc=f(c)
if fa*fc<=0:
bi,fb=c,fc
else:
ai,fa=c,fc
return (ai+bi)/2.
```

Convergence de la méthode

A l'itération n, l'erreur ϵ_n entre la solution exacte et la solution numérique calculée peut être majorée par $|\epsilon_n| < \frac{|b_n - a_n|}{2}$. Ainsi, la racine r de la fonction peut être encadrée :

$$c_n - \frac{b_n - a_n}{2} \leq r \leq c_n + \frac{b_n - a_n}{2}$$

Soit $[a_n, b_n]$ l'intervalle à l'itération n. Définissons ε_n par $\varepsilon_n = b_n - a_n$. Comme l'intervalle est divisé par deux à chaque itération, $\left|\epsilon_{n+1}\right| \leq \varepsilon_{n+1} = \frac{\varepsilon_n}{2} = \frac{\varepsilon_1}{2^n}$. La convergence est donc linéaire.

Pour test d'arrêt, il est alors possible de prendre :

- un encadrement de la solution exacte : $b_n - a_n \leq \varepsilon$
- la valeur de la fonction au point calculé : $\left|f(c_n)\right| < \varepsilon$

Le nombre d'itération n pour obtenir un erreur inférieur à ε est tel que :

$$\varepsilon \leq \frac{b-a}{2^n} \quad \Rightarrow \quad 2^n \leq \frac{b-a}{\varepsilon}$$

$$\Rightarrow \quad n.\ln(2) \leq \ln\left(\frac{b-a}{\varepsilon}\right) \quad \Rightarrow \quad n \geq \frac{1}{\ln 2}.\ln\left(\frac{b-a}{\varepsilon}\right)$$

Avec $\varepsilon = 10^{-p}$ (resp. $\varepsilon = 2^{-p}$) pour une approche décimal (resp. binaire) :

$$n \geq \frac{\ln(b-a) + p.\ln(10)}{\ln 2} \qquad \left(\text{resp. } n \geq p + \frac{\ln(b-a)}{\ln 2}\right)$$

La vitesse de convergence est lente mais la méthode est robuste. Si:

- la fonction f est définie et continue sur l'intervalle $[a, b]$
- la fonction n'admet qu'une seule racine sur l'intervalle $[a, b]$

alors la méthode converge.

Sommaire

1. Introduction
2. Convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
   - Principe
   - Convergence de la méthode
   - Algorithme
5. Méthode de Newton
6. Dilemme robustesse/rapidité

Principe

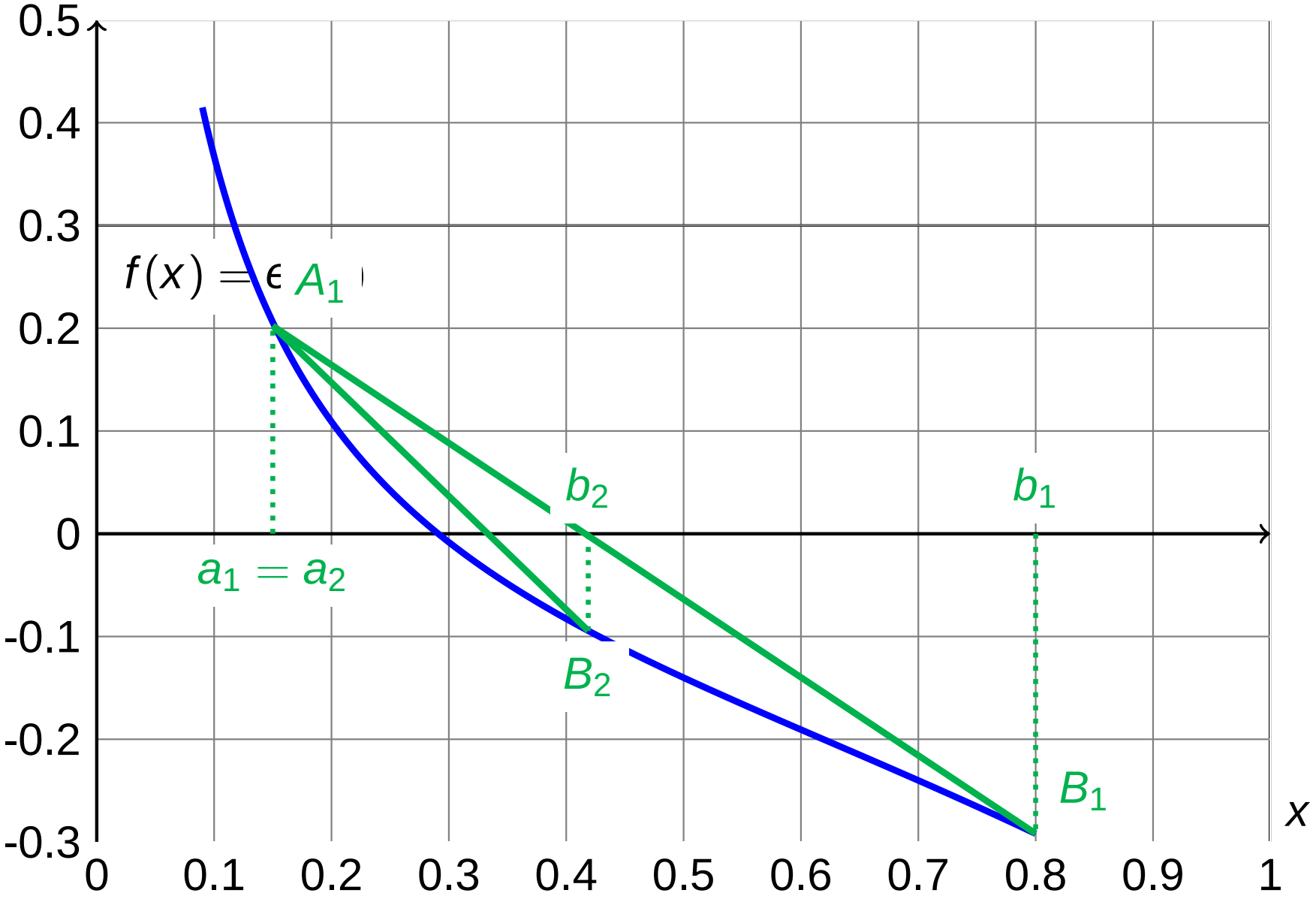
Le principe de la méthode de la corde est de diviser l'intervalle en deux, de conserver la partie contenant la racine et d'y reproduire l'opération jusqu'à ce que le critère de convergence soit satisfait. Au lieu de diviser en deux parts égales, on cherche le point c (diviseur de l'intervalle intersection de l'axe des abscisses et de la droite passant par les points $A = (a, f(a))$ et $B = (b, f(b))$.

L'équation de la corde est donnée par:

$$y = \frac{f(b) - f(a)}{b - a}.(x - a) + f(a) \quad \Rightarrow \quad 0 = \frac{f(b) - f(a)}{b - a}.(c - a) + f(a)$$

A partir d'un intervalle donné $[a, b]$, encadrant une racine de la fonction f étudiée :

- Calculer le point c sur la corde et sur l'axe des abscisses : $c = \dfrac{a.f(b) - b.f(a)}{f(b) - f(a)}$
- Evaluer $p = f(a).f(c)$ puis test :
  - si $p > 0$, il n'y a pas de racine dans l'intervalle $[a, c]$. La racine est donc dans l'intervalle $[c, b]$. On donne alors à a la valeur de c
  - si $p < 0$, la racine est dans $[a, c]$. On donne alors à b la valeur de c
  - si $p = 0$, alors la racine est c. Ô Miracle,...
- test d'arrêt : Evaluer le critère de convergence
- Recommencer si le critère de convergence n'est pas satisfait


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
$f(x) =$
A_1
b_2
b_1
$a_1 = a_2$
B_2
B_1

Convergence de la méthode

Comme nous pouvons le voir sur la figure précédente, dans le cas d'une fonction convexe sur $[a, b]$ telle que $f(a) > 0$ et $f(b) < 0$, $\forall n \in \mathbb{N}$, $a_n = a_1$.

Il se sera donc pas possible d'encadrer la solution exacte par intervalle dont la longueur tendrait vers 0 avec n. En effet :

A l'itération n, r étant la solution exacte du problème $f(x) = 0$ sur l'intervalle d'étude :

$$c_n = \frac{a_n.f(b_n) - b_n.f(a_n)}{f(b_n) - f(a_n)} \quad \Rightarrow \quad \left\{ \begin{array}{lcl} a_n & = & c_n + \dfrac{b_n - a_n}{f(b_n) - f(a_n)}.f(a_n) \\ b_n & = & c_n + \dfrac{b_n - a_n}{f(b_n) - f(a_n)}.f(b_n) \end{array} \right.$$

or $a_n < r < b_n$

$$\Rightarrow \; c_n + \frac{b_n - a_n}{f(b_n) - f(a_n)}.f(a_n) < r < c_n + \frac{b_n - a_n}{f(b_n) - f(a_n)}.f(b_n)$$

d'où $\quad |c_n - r| < \dfrac{\max\left(\left|f(a_n)\right|, \left|f(b_n)\right|\right)}{\left|f(b_n) - f(a_n)\right|}.|b_n - a_n|$

Il convient alors de prendre pour test d'arrêt:

- la valeur de la fonction au point calculé : $\left|f(c_n)\right| < \varepsilon$
- l'évolution de l'algorithme : $|c_n - c_{n-1}| < \varepsilon$

Algorithme

Méthode de la corde de Lagrange pour la fonction f sur $[a, b]$ avec une tolérance ε :

Entrées : f, a, b, ε
$ai, bi \leftarrow a, b$
$fa, fb \leftarrow f(ai), f(bi)$
$fc \leftarrow fa$
tant que $|fc| > \varepsilon$ **faire**
  $ci \leftarrow (ai.fb - bi.fa)/(fb - fa)$
  $fc \leftarrow f(ci)$
  si $fa.fc \geq 0$ **alors**
    $bi \leftarrow ci$
    $fd \leftarrow fc$
  sinon
    $ai \leftarrow ci$
    $fa \leftarrow fc$
Sorties : $\dfrac{ai.fb - bi.fa}{fb - fa}$

```
def Corde(f,a,b,eps):
assert f(a)*f(b)<=0 and eps >0
ai,bi=a,b
fa,fb=f(ai),f(bi)
fc=fa
while abs(fc)>eps:
c=(ai*fb-bi*fa)/(fb-fa)
fc=f(c)
if fa*fc<=0:
bi=c
fb=fc
else:
ai,fa=c,fc
return (ai*fb-bi*fa)/(fb-fa)
```

Sommaire

1. Introduction
2. Convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
   - Principe
   - Convergence de la méthode
   - Algorithme
   - Limites et précautions
   - Fausse position
   - Fausse position
   - Méthode de Newton-Raphson

Principe

Les méthodes de Schröder sont basés sur les n premières dérivées d'une fonction f pour une méthode d'ordre $n+1$. Dans le cas de la méthode de Newton, on n'utilise que la dérivée première.

RAPPEL Si la fonction est de classe C^1 sur l'intervalle $[a,b]$ alors, le développement de Taylor à l'ordre 1 donne:

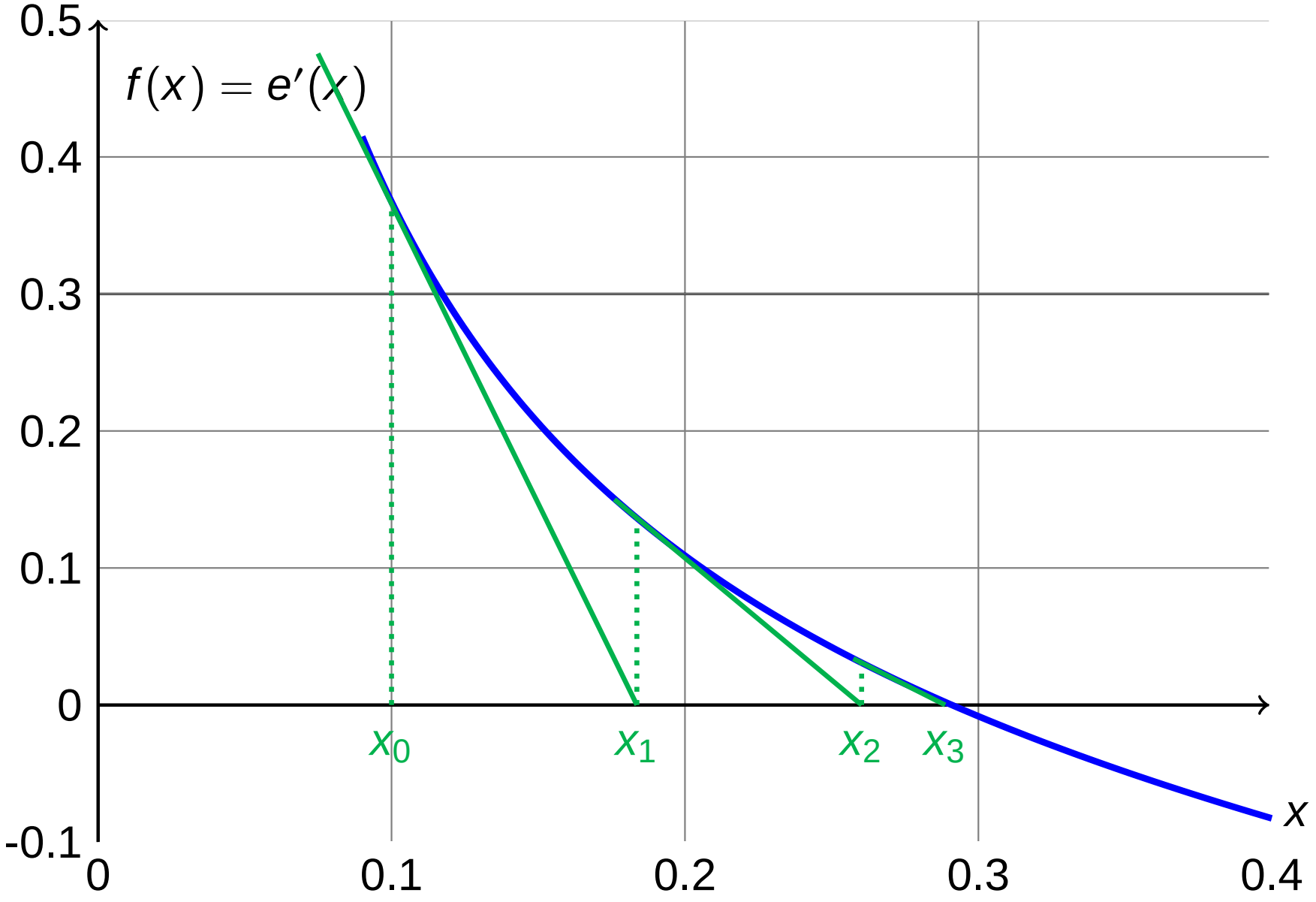
$$f(b) = f(a) + f'(a).(b-a) + o(b-a)$$

A partir d'un point x_0, la méthode consiste à rechercher le point suivant x_1 en le supposant racine de la fonction et en négligeant le terme $o(b-a)$ dans le développement de Taylor à l'ordre 1. Ainsi:

$$0 = f(x_1) = f(x_0) + f'(x_0).(x_1 - x_0) \quad \Rightarrow \quad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Soit f de classe C^1 sur $I = [a, b]$:

- Choisir un premier candidat x_0
- Test d'arrêt : Tester le critère de convergence $|f(x_0)| < \epsilon$ et $x_0 \notin I$
- Si le critère n'est pas atteint, calculer le nouveau candidat $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$
- Reprendre depuis le test d'arrêt


0.5
0.4
0.3
0.2
0.1
0
-0.1
0
0.1
0.2
0.3
0.4
$f(x) = e'(x)$
x_0
x_1
x_2
x_3
x

Convergence de la méthode

Soit r la racine de la fonction f sur l'intervalle d'étude et x_n, la énième valeur calculée par itération de Newton. Notons alors $\epsilon_n = x_n - r$. Le développement de Taylor à l'ordre 2 au voisinage de r donne:

$$\begin{aligned} f(x_n) &= \cancel{f(r)} + f'(r).\varepsilon_n + f''(r).\frac{\varepsilon_n^2}{2} + o(\varepsilon_n^2) \\ f'(x_n) &= f'(r) + f''(r).\varepsilon_n + o(\varepsilon_n) \end{aligned}$$

$$\begin{aligned} \text{alors} \quad \varepsilon_{n+1} &= x_{n+1} - r = x_n - r - \frac{f(x_n)}{f'(x_n)} \\ &= \varepsilon_n - \frac{f'(r).\varepsilon_n + f''(r).\frac{\varepsilon_n^2}{2} + o(\varepsilon_n^2)}{f'(r) + f''(r).\varepsilon_n + o(\varepsilon_n)} = \frac{\varepsilon_n^2}{2}.\frac{f''(r)}{f'(r)} + o(\varepsilon_n^2) \end{aligned}$$

La convergence est donc quadratique.

Algorithme

Méthode de Newton pour la fonction f sur $[a, b]$ avec une tolérance ε :

Entrées : f, f', x_0, ε
$xi \leftarrow x_0$
$fx \leftarrow f(xi)$
tant que $|fx| > \varepsilon$ **faire**
  $fpx \leftarrow f'(xi)$
  si $fpx=0$ **alors**
    break
  sinon
    $xi \leftarrow xi - fx/fpx$
    $fx \leftarrow f(xi)$
Sorties : xi

```
%# f = fonction et fp = sa dérivée
def Newton(f,fp,x,eps):
  xi=x
  fx=f(xi)
  while abs(fx)>eps:
    fpx=fp(xi)
    if fpx==0.:
      print('Tangeante nulle\n')
      quit
    else:
      res=xi-fx/fpx
      xi,fx=res,f(res)
  return xi
```

Limites et précautions

La méthode est bien adapté si la valeur de x_0 est proche de la racine r de la fonction f et si la dérivée n'est pas trop faible (pente trop horizontale).

Limites et précautions

La méthode est bien adapté si la valeur de x_0 est proche de la racine r de la fonction f et si la dérivée n'est pas trop faible (pente trop horizontale).

Soit la fonction $f(x) = x - 2.\sin(x)$. La dérivée $f'(x) = 1 - 2.\cos(x)$ s'annule pour $x = \frac{\pi}{3} \approx 1,0472$.

Limites et précautions

La méthode est bien adapté si la valeur de x_0 est proche de la racine r de la fonction f et si la dérivée n'est pas trop faible (pente trop horizontale).

Soit la fonction $f(x) = x - 2.\sin(x)$. La dérivée $f'(x) = 1 - 2.\cos(x)$ s'annule pour $x = \frac{\pi}{3} \approx 1,0472$.

Prendre comme valeur $x_0 = \frac{\pi}{3}$ conduit tout de suite au crash. La dérivée étant nulle, x_1 ne pourra pas être calculé. Prendre $0 < x_0 < \frac{\pi}{3}$ conduit à la racine $r_0 = 0$. Pour trouver la deuxième racine $r_1 \approx 1,9$, il convient de prendre une valeur de départ $x_0 > \frac{\pi}{3}$.

n	x_n	$f(x_n)$
0	$1,1$	$-0,6824147$
1	$8,452992$	$6,801205$
2	$5,256414$	$6,967679$
3	$203,384184$	$201,922795$
$\vdots$	$\vdots$	$\vdots$
57	$-0,380713$	$0,362452$
58	$0,042317$	$-0,042292$
59	$-0,000051$	$0,000051$
60	$0,000000$	$-0,000000$

Cependant, prendre pour valeur $x_0 = 1,1$ ou $x_0 = 1,2$ conduit dans le premier cas à r_0 et dans le deuxième r_1, même si la fonction est monotone sur $\left[\frac{\pi}{3}, \pi\right]$ et que $\frac{\pi}{3} < 1,1 < 1,2 < \pi$.

Notons dans les deux cas, la convergence quadratique à proximité de r.

n	x_n	$f(x_n)$
0	$1,2$	$-0,6640782$
1	$3,612334$	$4,519429$
2	$1,988080$	$0,159694$
3	$1,899879$	$0,007200$
4	$1,895505$	$0,000018$
5	$1,895494$	$0,000000$

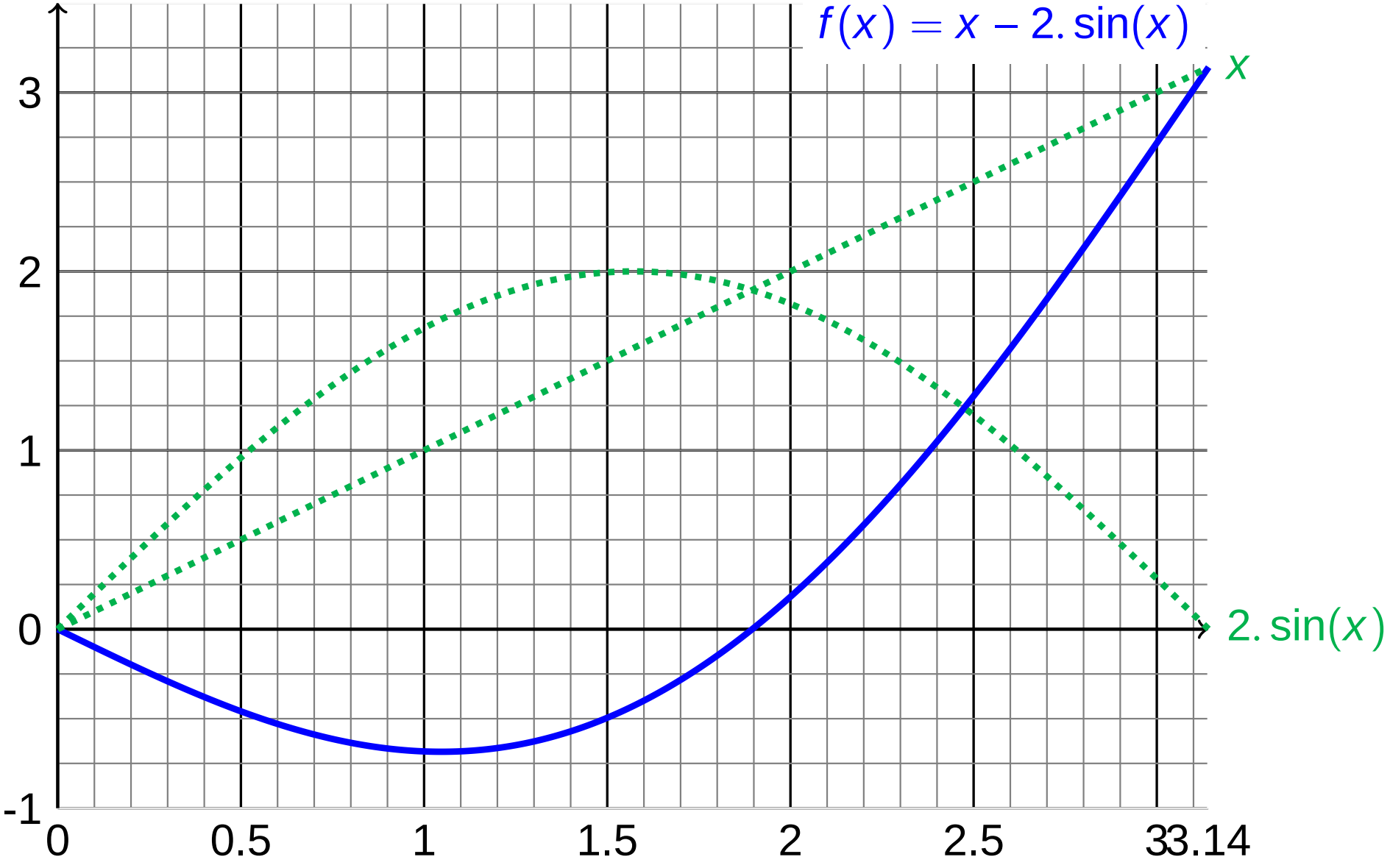
n	x_n	$f(x_n)$
0	$1,2$	$-0,6640782$
1	$3,612334$	$4,519429$
2	$1,988080$	$0,159694$
3	$1,899879$	$0,007200$
4	$1,895505$	$0,000018$
5	$1,895494$	$0,000000$

Il est donc préférable que la fonction soit monotone sur l'intervalle d'étude et que la première valeur x_0 soit proche de r.

n	x_n	$f(x_n)$
0	$1,2$	$-0,6640782$
1	$3,612334$	$4,519429$
2	$1,988080$	$0,159694$
3	$1,899879$	$0,007200$
4	$1,895505$	$0,000018$
5	$1,895494$	$0,000000$

Il est donc préférable que la fonction soit monotone sur l'intervalle d'étude et que la première valeur x_0 soit proche de r.

Lorsque l'algorithme patine ou que l'itération conduit à une valeur hors du domaine de définition, il est possible d'utiliser une méthode par dichotomie et reprendre le schéma de Newton près de la racine.


$f(x) = x - 2.\sin(x)$
x
$2.\sin(x)$
3
2
1
0
-1
0
0.5
1
1.5
2
2.5
3
3.14

Fausse position

Principe

La méthode de Newton étant basée sur un développement de Taylor à l'ordre 1, il est nécessaire de calculer la dérivée première.

La méthode de la fausse position est basée sur une estimation de la dérivée à partir des deux points précédents. Ce n'est pas la valeur exacte comme dans le schéma de Newton. Nous avons alors :

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

$$\Rightarrow \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - f(x_n).\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$
$$= \frac{x_{n-1}.f(x_n) - x_n.f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Il est donc préférable que la fonction soit monotone sur l'intervalle d'étude et que la première valeur x_0 soit proche de r.

Lorsque l'algorithme patine ou que l'itération conduit à une valeur hors du domaine de définition, il est possible d'utiliser une méthode par dichotomie et reprendre le schéma de Newton près de la racine.

Soit f définie sur $I = [a, b]$:

- Choisir un premier couple candidat $(x_0, x_1) \in I^2$
- Test d'arrêt : Tester le critère de convergence $|f(x_1)| < \varepsilon$ et $x_1 \notin I$
- Si le critère n'est pas atteint :
  - calculer $f(x_1)$
  - si $f(x_1) = f(x_0)$ soit arrêt, soit choisir une autre valeur de x_0
  - calculer le nouveau candidat $x_2 = x_1 - f(x_1).\dfrac{x_1 - x_0}{f(x_1) - f(x_0)}$
- Reprendre depuis le test d'arrêt avec : si $f(x_n) = f(x_{n-1})$ soit arrêt, soit prendre x_{n_2} à la place de x_{n_1}

Fausse position

Convergence de la méthode

Soit r la racine de la fonction f sur l'intervalle d'étude et x_n, la énième valeur calculée par la méthode de la fausse position. Notons alors $\varepsilon_n = x_n - r$. Le développement de Taylor à l'ordre 2 au voisinage de r donne:

$$\begin{aligned} f(x_n) &= \cancel{f(r)} + f'(r).\varepsilon_n + f''(r).\frac{\varepsilon_n^2}{2} + o(\varepsilon_n^2) \\ f(x_{n-1}) &= \cancel{f(r)} + f'(r).\varepsilon_{n-1} + f''(r).\frac{\varepsilon_{n-1}^2}{2} + o(\varepsilon_{n-1}^2) \end{aligned}$$

$$
\begin{aligned}
\text{alors } \varepsilon_{n+1} &= x_{n+1} - r \\
&= \frac{x_{n-1}.f(x_n) - x_n.f(x_{n-1})}{f(x_n) - f(x_{n-1})} - r \\
&= \frac{\varepsilon_{n-1}.f(x_n) - \varepsilon_n.f(x_{n-1})}{f(x_n) - f(x_{n-1})} \\
&= \frac{f''(r).\left(\frac{\varepsilon_n^2}{2}.\varepsilon_{n-1} - \frac{\varepsilon_{n-1}^2}{2}.\varepsilon_n\right) + o(\varepsilon_n^2) - o(\varepsilon_{n-1}^2)}{f'(r).(\varepsilon_n - \varepsilon_{n-1}) + o(\varepsilon_n) - o(\varepsilon_{n-1})} \approx \frac{f''(r).\varepsilon}{f'(}
\end{aligned}
$$

Nous cherchons à connaître l'ordre p de convergence, i.e, p tel que $\varepsilon_{n+1} = \varepsilon_n^p$. Or:

$$\varepsilon_{n+1} \equiv \varepsilon_n.\varepsilon_{n-1} \quad \Rightarrow \quad \varepsilon_n^p \equiv \varepsilon_n.\varepsilon_n^{\frac{1}{p}} \quad \Rightarrow \quad p = 1 + \frac{1}{p}$$

$$\Rightarrow \quad p^2 - p - 1 = 0 \quad \Rightarrow \quad p = \frac{1+\sqrt{5}}{2} \approx 1,62 < 2$$

La convergence est donc super-linéaire ($p > 1$) mais inférieure à la convergence quadratique ($p = 2$) de la méthode de Newton. En revanche, elle ne nécessite pas le calcul de la dérivée première qui peut parfois être très lourd.

Méthode de Newton-Raphson

La méthode de Newton-Raphson est une généralisation de la méthode de Newton pour les fonctions de plusieurs variables. Il s'agit de résoudre un système de n équations à n inconnues :

$$\vec{F}(\vec{X}) = \vec{0} \quad \Leftrightarrow \quad \begin{bmatrix} F_1(x_1, \ldots, x_i, \ldots, x_n) \\ \vdots \\ F_i(x_1, \ldots, x_i, \ldots, x_n) \\ \vdots \\ F_n(x_1, \ldots, x_i, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

La formule de Taylor à l'ordre 1 donne pour chaque F_i :

$$F_i(\vec{x}+\vec{\delta x}) = F_i(\vec{x}) + \sum_{j=1}^{n} \frac{\partial F_i(\vec{x})}{\partial x_j}.\delta x_j + o(\vec{\delta x}),\ \forall i \in [\![1,n]\!]$$

En introduisant alors la matrice jacobienne $\mathbb{J}$ définie par

$$[\mathbb{J}]_{(i,j)} = J_{i,j} = \frac{\partial F_i(\vec{x})}{\partial x_j} \quad \Leftrightarrow \quad \mathbb{J} = \begin{bmatrix} \frac{\partial F_1(\vec{x})}{\partial x_1} & \cdots & \frac{\partial F_1(\vec{x})}{\partial x_j} & \cdots & \frac{\partial F_1(\vec{x})}{\partial x_n} \\ \vdots & \ddots & \vdots & & \vdots \\ \frac{\partial F_i(\vec{x})}{\partial x_1} & \cdots & \frac{\partial F_i(\vec{x})}{\partial x_j} & \cdots & \frac{\partial F_i(\vec{x})}{\partial x_n} \\ \vdots & & \vdots & \ddots & \vdots \\ \frac{\partial F_n(\vec{x})}{\partial x_1} & \cdots & \frac{\partial F_n(\vec{x})}{\partial x_j} & \cdots & \frac{\partial F_n(\vec{x})}{\partial x_n} \end{bmatrix}$$

la formule générale du développement limité à l'ordre 1 de $\vec{F}$ au voisinage de $\vec{x}$ devient:

$$\vec{F}(\vec{x}+\vec{\delta x}) = \vec{F}(\vec{x}) + \mathbb{J}(\vec{x}).\vec{\delta x} + o(\vec{\delta x})$$

L'algorithme de Newton - Raphson s'obtient en cherchant $\overrightarrow{X_{n+1}}$ solution du problème $\vec{F}(\vec{x}) = \vec{0}$. Le développement de Taylor, conduit à :

$$\cancel{\vec{F}(\overrightarrow{X_{n+1}})} = \vec{F}(\overrightarrow{X_n}) + \mathbb{J}(\overrightarrow{X_n}).\left(\overrightarrow{X_{n+1}} - \overrightarrow{X_n}\right) + o\left(\overrightarrow{X_{n+1}} - \overrightarrow{X_n}\right)$$

d'où $\quad \overrightarrow{X_{n+1}} = \overrightarrow{X_n} - \mathbb{J}^{-1}(\overrightarrow{X_n}).\vec{F}(\overrightarrow{X_n})$

Chaque itération demande alors le calcul de l'inverse de la matrice Jacobienne...

Sommaire

1. Introduction
2. Convergence
3. Rappel sur la méthode par dichotomie
4. Méthode de la corde (de Lagrange)
5. Méthode de Newton
6. Dilemme robustesse/rapidité

Dilemme robustesse/rapidité

Nous avons vu différentes méthodes de résolution de l'équation

$$f(x) = 0$$

Que choisir ?

Dilemme robustesse/rapidité

Robustesse

La méthode la plus naïve est la dichotomie.

A partir d'une fonction continue sur un intervalle $[a, b]$, avec $f(a).f(b) \leq 0$, la recherche de x tel que $f(x) = 0$ est simple et robuste mais lente (convergence linéaire).

Si l'on souhaite un programme simple et robuste, la dichotomie est une solution.

Dilemme robustesse/rapidité

Rapidité

Si on cherche une valeur avec précision, une convergence quadratique sera préférée (méthode de Newton).

A ce moment là, il convient d'avoir une idée approximative de la solution pour proposer un premier candidat au voisinage cette solution de sorte que la fonction y présente de bonnes propriétés.

Si on ne souhaite pas calculer la dérivée de la fonction, on peut alors utiliser la méthode de la fausse position.

Dilemme robustesse/rapidité

Efficacité - compromis

Dans le cas où l'on cherche rapidité et stabilité, on peut utiliser la méthode de par dichotomie dans une premier temps pour localiser le zéro de la fonction, puis appliquer un algorithme de Newton.

En cas d'instabilité de l'algorithme de Newton, il est toujours de réutiliser une méthode par dichotomie.

Enfin, mentionnons qu'il existe aussi d'autres types d'algorithme, comme les algorithmes génétiques pour trouver les différentes solutions du problème.