le cnam

Mathématiques Appliquées pour le Génie des Procédés et l'Energétique

Paris, automne 2018

Géométrie numérique

Notes du cours 06

Amélie Danlos, Marie Debacq, François Dubois

le cnam

Mathématiques Appliquées pour le Génie des Procédés et l'Énergétique

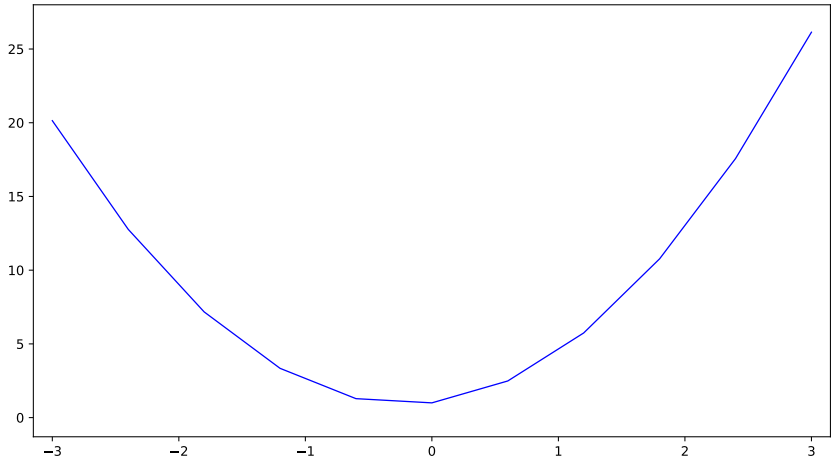
Cours 6 Géométrie numérique

- **Graphe d'une parabole avec le logiciel Python**

Un codage très élémentaire dans le langage Python peut s'écrire

```
###       déclarations préliminaires
import numpy as np
import matplotlib.pyplot as plt
from pylab import plot, axis, savefig, show
from math import log

###       codage des données
a = 2.46 ; b = 1. ; c = 1.
nn = 10 ; xmin = -3. ; xmax = 3.
dx = (xmax-xmin)/nn
xx = np.zeros(nn+1) ; pp = np.zeros(nn+1)
for i in range(0,nn+1) :                    ### début de boucle
    xx[i] = xmin + i*dx
    pp[i] = a*(xx[i]**2) + b*xx[i] + c      ### parabole y = ax^2 + bx + c
print ('xx =', xx) ; print ('pp =', pp)

###       codage du graphique
fig, ax = plt.subplots(figsize=(11,6))
ax.plot (xx, pp, 'b', linewidth=1)
plt.savefig('utc-parabole-01-fig.pdf', transparent=True)
plt.show()
```

On génère ainsi le graphe suivant :



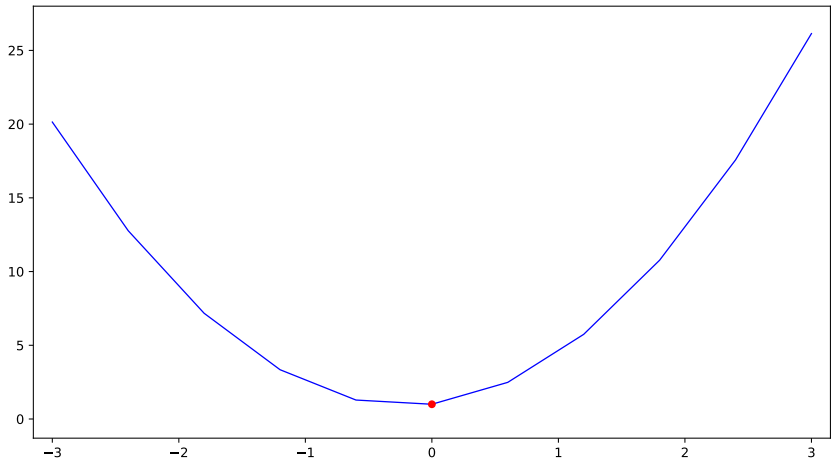
La discrétisation est clairement visible à cause du faible nombre de points utilisés.

- **On rajoute un point de la courbe**

♯♯♯ peu de nouvelles données
x0 = 0 ; y0 = c

♯♯♯ codage du graphique
fig, ax = plt.subplots(figsize=(11,6))
ax.plot (xx, pp, 'b', linewidth=1)
ax.plot (x0,y0 , 'r',marker='o',markersize=5)
plt.savefig('utc-parabole-01-fig.pdf', transparent=True)
plt.show()

La courbe est enrichie d'un point isolé.

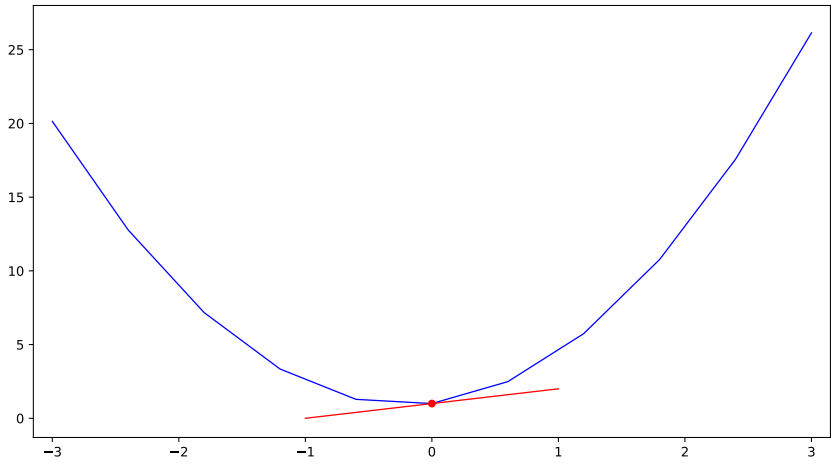


- **On trace la droite tangente qui passe par le point que l'on s'est donné**

On rappelle d'abord que la droite tangente en la courbe d'équation $y = f(x)$ au point de coordonnées $\big(x_0, y_0 = f(x_0)\big)$ est l'unique droite qui passe par le point (x_0, y_0) et de pente $f'(x_0)$. Elle a donc pour équation $y = y_0 + f'(x_0)\,(x - x_0)$. Dans le langage Python, on se contente de tracer un segment de droite.

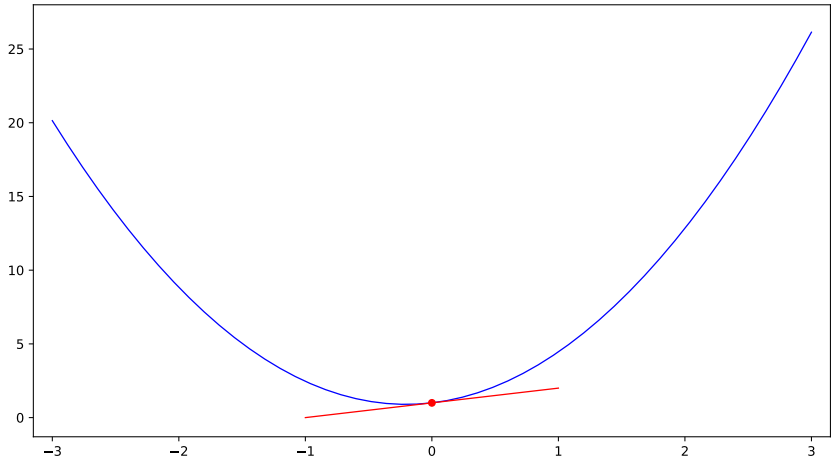
```
###     nouvelles données
x0 = 0 ; y0 = a*x0**2+b*x0+c ; yp0 = 2*a*x0 + b
xtmin = x0-1 ; xtmax = x0+1                    ###     points extrèmes de la tangente
ytmin = y0 + yp0 * (xtmin-x0)
ytmax = y0 + yp0 * (xtmax-x0)                  ###     valeurs des ordonnees

###     codage du graphique
fig, ax = plt.subplots(figsize=(11,6))
ax.plot (xx, pp, 'b', linewidth=1)
ax.plot (x0,y0 , 'r',marker='o',markersize=5)
                                               ###     un segment est décrit par ses deux points extrémaux
ax.plot ([xtmin,xtmax],[ytmin,ytmax] , 'r',linewidth=1)
plt.savefig('utc-parabole-03-fig.pdf', transparent=True)
plt.show()
```

Le point isolé est muni de sa tangente.



- **On rajoute des points afin d'avoir une parabole plus réaliste**

On remplace l'instruction $nn = 10$ par $nn = 50$ au début du programme lors du codage des données. Le graphique est alors le suivant :



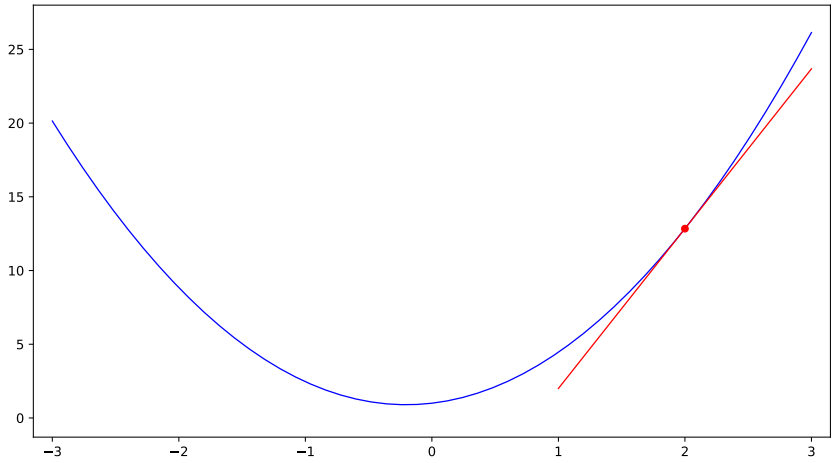
ce qui est beaucoup plus réaliste

- **On déplace le point isolé**

Le codage est très simple : on ajoute simplement les lignes suivantes au programme précédent :

```
### quelques données
x0 = 2 ; y0 = a*x0**2+b*x0+c ; yp0 = 2*a*x0 +b
xtmin = x0-1 ; xtmax = x0+1
ytmin = y0 + yp0*(xtmin-x0) ; ytmax = y0 + yp0*(xtmax-x0)
```

et le codage du graphe est inchangé.

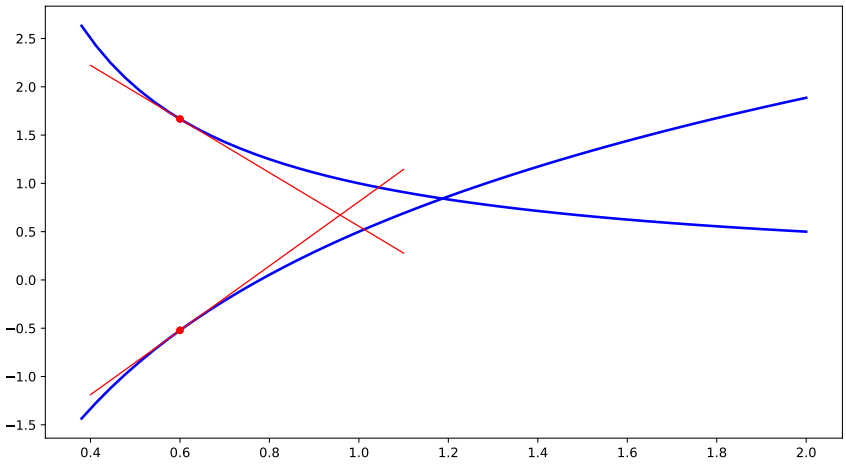


- **On dessine deux courbes**

En particulier une courbe logarithmique représentée par la fonction $f_1(x) = a\log(x) + b$ et une hyperbole associée à la fonction $f_2(x) = \frac{1}{x}$. On rajoute aussi un point de même abscisse sur chacune des courbes et les tangentes associées.

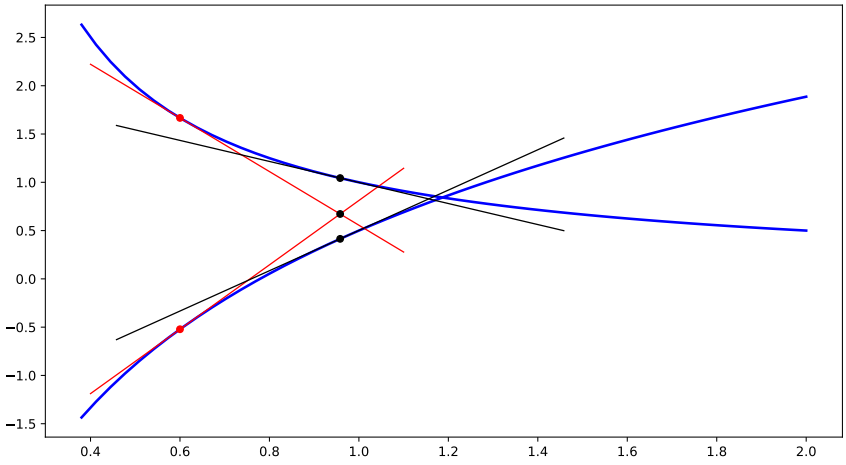
```
###       quelques données
a = 2 ; b = .5
nn = 50 ; xmin = .5 ; xmax = 2 ; dx = (xmax-xmin)/nn
xx = np.zeros(nn+1) ; f1 = np.zeros(nn+1) ; f2 = np.zeros(nn+1)
for i in range(0,nn+1) :
    xx[i] = xmin + i*dx
    f1[i] = a*log(xx[i]) + b -(1./xx[i])
    f2[i] = (1./xx[i])
x0 = .6 ; y1 = a*log(x0) + b ; y2 = 1/x0 ; yp1 = a/x0 ; yp2 = -1/(x0*x0)
xtmin = x0-.2 ; xtmax = x0+.5
y1tmin = y1 + yp1 * (xtmin-x0) ; y1tmax = y1 + yp1 * (xtmax-x0)
y2tmin = y2 + yp2 * (xtmin-x0) ; y2tmax = y2 + yp2 * (xtmax-x0)
```

```
###       graphique des deux courbes
fig, ax = plt.subplots(figsize=(11,6))
ax.plot (xx, hy, 'b',linewidth=2)
ax.plot (xx, f1, 'b',linewidth=2)
ax.plot (x0,y0 , 'r',marker='o',markersize=5)
ax.plot (x0,f2, 'r',marker='o',markersize=5)
ax.plot ([xtmin,xtmax],[y1tmin,y1tmax] , 'r',linewidth=1)
ax.plot ([xtmin,xtmax],[y2tmin,y2tmax] , 'r',linewidth=1)
plt.savefig('utc-intersection-courbes-fig.pdf', transparent=True)
plt.show()
```



- **Algorithme de Newton**

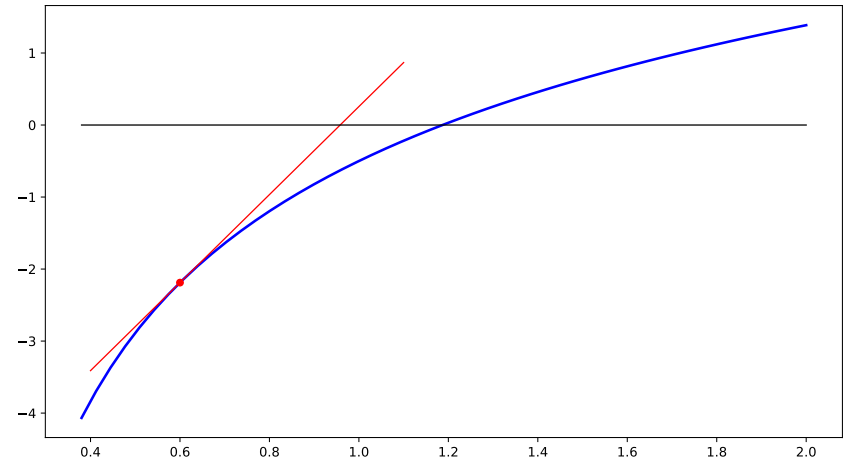
Si on veut calculer numériquement le point d'intersection de ces deux courbes, l'algorithme de Newton remplace la courbe par sa tangente. Après calcul du point d'intersection, on recommence le processus décrit plus haut.



Après la seconde itération de l'algorithme de Newton, le point obtenu, intersection des deux tangentes en noir, est très proche visuellement du point d'intersection des deux courbes bleues.

- **Une autre façon de poser l'algorithme de Newton**

Calculer le point d'intersection des courbes représentatives des deux fonctions f_1 et f_2 revient à résoudre l'équation $a\log(x)+b-\frac{1}{x}=0$, c'est à dire à chercher la valeur du réel x qui annule la fonction f définie par $f(x)=a\log(x)+b-\frac{1}{x}$. Le graphe de cette fonction est donné ci-dessous.



La première itération de l'algorithme de Newton revient à calculer le point d'intersection de la tangente (en rouge) avec l'axe des abscisses (en noir). Le calcul est laissé au lecteur.

le cnam

Conservatoire National des Arts et Métiers

Mathématiques Appliquéees pour le Génie des Procédés et l'Énergétique

Devoir numéro 1, à rendre le mercredi 11 décembre 2018

On se donne trois nombres strictement positifs a, b et R. Dans les applications numériques, on prendra $a = 2,46\,;\quad b = 0,29\,;\quad R = 100$.

On cherche à résoudre l'équation suivante $\dfrac{1}{\sqrt{f/2}} = a\,\ln\Big(R\,\sqrt{f/2}\Big) + b.$

1) On suppose connue la solution x de l'équation suivante

(1) $\dfrac{1}{x} = a\,\ln\big(R\,x\big) + b\,.$

Comment calculer $f/2$ en fonction de x ?

2) Pour $x > 0$, on pose $F(x) = a\,\ln(x) + b + a\,\ln(R) - \dfrac{1}{x}$.

a) Que vaut la dérivée $F'(x)$?

b) Montrer que la fonction F est strictement croissante lorsque $0 < x < +\infty$.

c) Montrer que $\lim_{x\longrightarrow 0} = -\infty$ et que $\lim_{x\longrightarrow +\infty} = +\infty$.

d) En déduire que l'équation (1) a une solution unique.

3) On introduit $c = b + a\,\ln(R)$.

a) Avec les donnés numériques proposées, montrer que $c > 1$.

b) Montrer que $F\big(\frac{1}{c}\big) < 0$ et $F(1) > 0$.

c) En déduire que la solution x de l'équation (1) vérifie $\dfrac{1}{c} < x < 1$.

4) Effectuer quelques itérations de l'algorithme de dichotomie afin d'améliorer l'encadrement de la solution x de l'équation (1).

5) Calculer une bien meilleure approximation de la solution x de l'équation (1) en initialisant l'algorithme de Newton $x_{n+1} = x_n - \dfrac{F(x_n)}{F'(x_n)}$ avec l'approximation trouvée à l'issue de la question 4).

6) En déduire la valeur $f/2$ de l'équation initiale donnée dans le préambule.

7) Joindre la feuille de calcul Excel (ou équivalent) ou le programme Python.

8) Question complémentaire. Reprendre les questions 3) à 6) avec une valeur de R de votre choix supérieure à $3/2$.