Recherche d'une racine simple par la méthode de bissection ou dichotomie

Manuel Luque

17 août 2003

1 Présentation

La méthode de la bissection, exposée ici, est adaptée de celle de R.Dony qu'il développe dans son livre *Graphisme scientifique sur micro-ordinateur* publié chez Masson 1984/1985, aux pages 112,113,114,115,116 et 117. C'est un livre remarquable que je conseille à tous ceux qui souhaitent justement, s'initier aux graphismes 2D et 3D.

Nous nous limitons à la recherche d'une racine simple. La première démarche consiste à déterminer l'intervalle $[\mathrm{x}_1, \mathrm{x}_2]$ où se trouve la racine. Une méthode rudimentaire et simple consiste à tracer la fonction étudiée. L'exemple choisi sera très simple et les racines évidentes, ceci pour permettre une vérification aisée.

J'ai donc écrit trois commandes, qui vont être détaillées par la suite :

- l'une `\pNodeRacine` permet d'afficher graphiquement la racine dans l'intervalle proposé ;
- la deuxième `\pNodeValeurRacine` calcule et affiche la valeur approchée de la racine ;
- la troisième `\pNodeIteration` permet de suivre pas à pas comment l'encadrement autour de la racine se resserre : c'est une méthode qui converge très rapidement, nous en reparlerons ; la précision souhaitée peut être modifiée dans la commande.

2 La méthode de la bissection

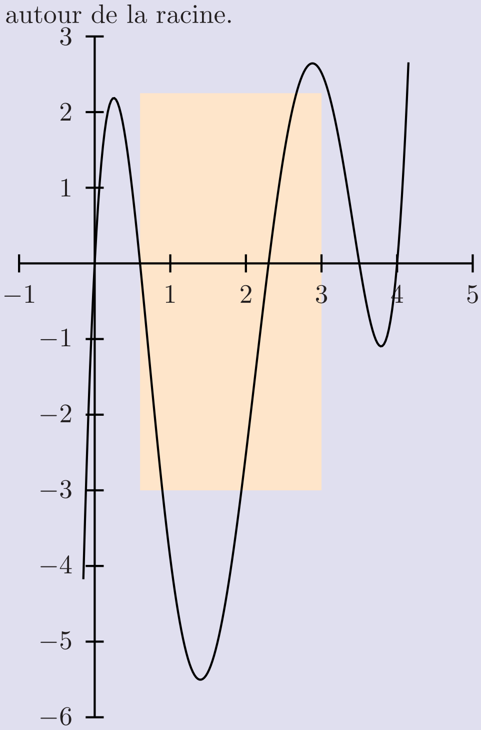
Les explications ci-dessous sont extraites du livre de R.Dony.

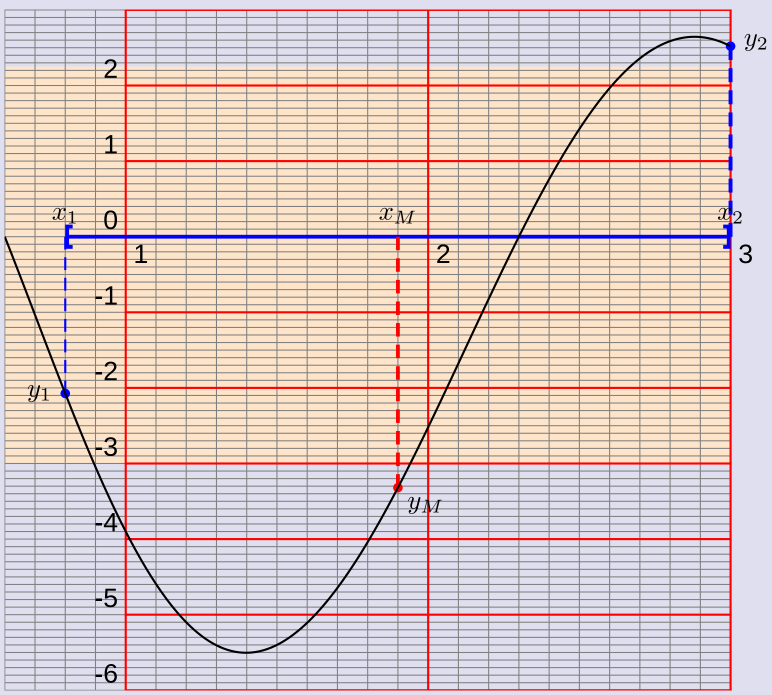
Le choix de l'équation : elle sera définie dans une variable `\Function` par :

```
\newcommand\Function{x x 0.6 sub mul x 2.3 sub mul x 3.5 sub mul x 4 sub mul}
```

$$y = x(x-0.6)(x-2.3)(x-3.5)(x-4)$$

en respectant la syntaxe de PostScript. Les racine sont évidentes $\{0.6, 2.3, 3.5, 4\}$. Nous nous focalisons sur la recherche de $x = 2.3$. Nous allons tout d'abord tracer la fonction dans un large intervalle, puis faire un zoom autour de la racine.





L'intervalle choisi est volontairement large $[x_1 = 0.8, x_2 = 3]$.

«À chaque étape, l'intervalle étant divisé par deux, au bout de 10 itérations, l'intervalle initial sera réduit dans un facteur de 2^{10} !

`ligne 2 :` On calcule l'abscisse x_M et l'ordonnée du milieu de $[x_1 = 0.8, x_2 = 3]$.

`ligne 7 :` Si $y_M = 0$ alors y_M est la racine.

`lignes 8, 9, 10 :` Si $y_1 \times y_M > 0$ alors la racine appartient à l'intervalle $[x_M, x_2]$ et on pose $x_1 = x_M$, sinon elle appartient à l'intervalle $[x_1, x_M]$ et on pose $x_2 = x_M$.

`ligne 11 :` Si la condition d'arrêt n'est pas satisfaite, on recommence le processus avec un nouvel intervalle réduit de moitié. »

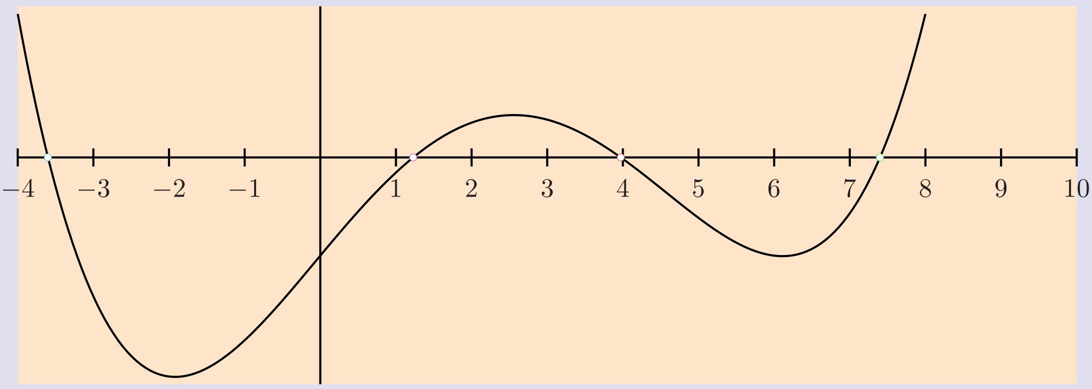
```
1   {
2      /XM X1 X2 add 2 div def % abscisse du milieu
3      /x X1 def
4      /Y1 Function def % valeur de la fonction Y1=y(x) pour x=X1
5      /x XM def
6      /YM Function def % valeur de la fonction YM=y(XM) pour x=XM
7       YM 0 eq {exit} if % si y(XM)=0 la racine est XM
8       Y1 YM mul 0 ge {/X1 XM def}
9                      {/X2 XM def}
10      ifelse
11    X1 X2 sub abs 1e-6 le {exit} if
12  } loop
```

3 Les commandes PSTricks permettant d'expérimenter

3.1 La recherche graphique d'une racine simple : \pNodeRacine

Cette commande s'écrit `\pNodeRacine(x1,x2){I}{\Function}`.

- Le couple `(x1,x2)` correspond aux deux bornes de l'intervalle initial encadrant la racine.
- `{I}` est le nom du point dont l'abscisse correspond à la racine cherchée.
- `{\Function}` contient la définition de la fonction étudiée. `{\Function}` aura été définie ou redéfinie au préalable. Si elle a déjà été définie, comme dans la première partie, on utilisera la commande suivante, pour en définir une nouvelle : $y = x^4 - 9x^3 - 2x^2 + 120x - 130$.
  `\renewcommand\Function{x 4 exp 9 x 3 exp mul sub 2 x 2 exp mul sub 120 x mul add 130 sub}`
  Exemple que j'emprunte au livre de R.Dony, page 112.




```
\begin{pspicture}(-4,-4)(10,2)
\psaxes[Dy=10](0,0)(-4,-4)(10,2)
\psset{xunit=1,yunit=0.01}
\psplot[plotpoints=200]{-4}{8}{\Function}
\pNodeRacine(-5,-2){I}{\Function}
\psdots[dotstyle=o,dotsize=1mm,linecolor=cyan](I)
\pNodeRacine(0,2){I}{\Function}
\psdots[dotstyle=o,dotsize=1mm,linecolor=magenta](I)
\pNodeRacine(3,5){I}{\Function}
\psdots[dotstyle=o,dotsize=1mm,linecolor=red](I)
\pNodeRacine(6,8){I}{\Function}
```

```
\psdots[dotstyle=o,dotsize=1mm,linecolor=green](I)
\end{pspicture}
```

Le cadre du dessin `\begin{pspicture}(-4,-4)(10,2)` doit être déterminé auparavant, on jouera aussi sur les échelles `\psset{xunit=1,yunit=0.01}`.

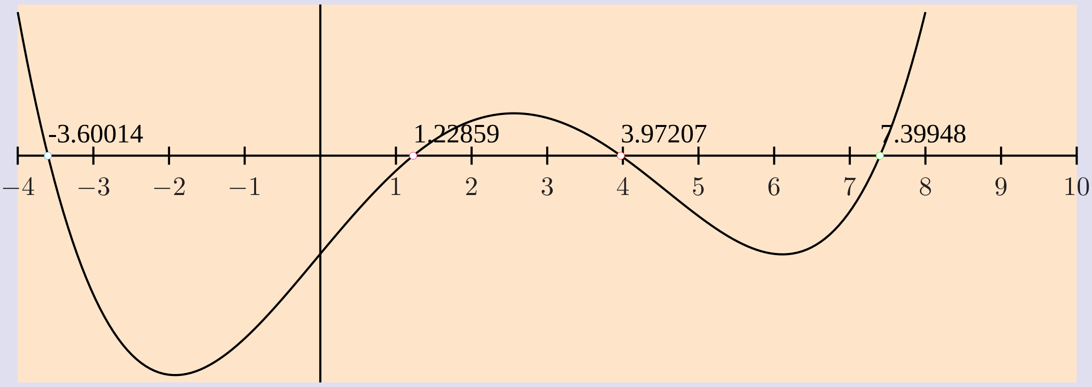
3.2 La recherche de la valeur approchée d'une racine simple : `\pNodeValeurRacine`

Elle s'utilise comme précédemment.

```
La racine comprise dans l'intervalle [-5,-2] vaut $x=\pNodeValeurRacine(-5,-2){I}{\Function}$
```

La racine comprise dans l'intervalle [-5,-2] vaut $x =$ -3.60014
La racine comprise dans l'intervalle [0,2] vaut $x =$ 1.22859
La racine comprise dans l'intervalle [3,5] vaut $x =$ 3.97207
La racine comprise dans l'intervalle [6,8] vaut $x =$ 7.39948
Si l'on souhaite afficher la valeur de la racine au-dessus du point d'intersection, il suffit d'opérer ainsi :

```
\pNodeRacine(-5,-2){I}{\Function}
\psdots[dotstyle=o,dotsize=1mm,linecolor=cyan](I)
\uput[90](I){\pNodeValeurRacine(-5,-2){I}{\Function}}
```

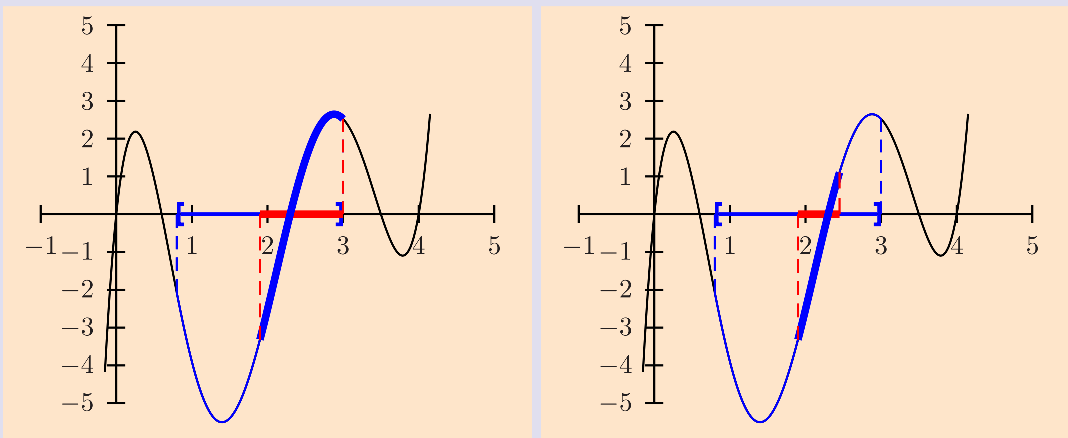


4 Expérimenter pas à pas la méthode de la bissection

Il existe une commande dédiée à cela : `\pNodeIteration`. Elle s'utilise de la manière suivante :

```
\pNodeIteration[NbreIterations=0](0.8,3){I}{\Function}
```

`[NbreIterations=0]` permet de fixer le nombre d'itérations. L'intervalle s'affiche en rouge. L'intervalle initial est en bleu.



Une animation utilisant cette commande est visible sur la page `html` correspondante. On s'apercevra de la convergence très rapide de la méthode.