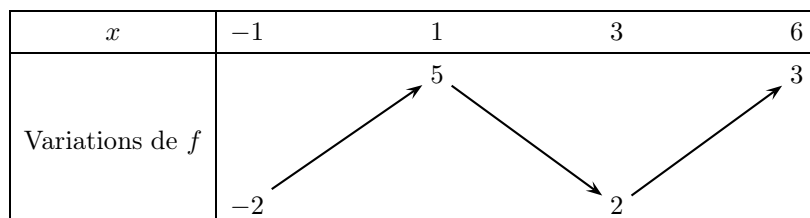


Courbe régulière

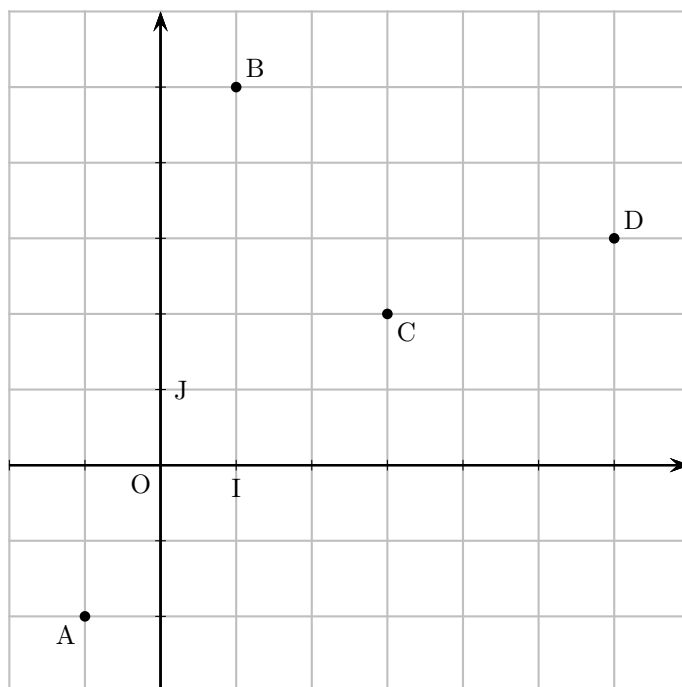
1 Le problème

On donne le tableau des variations d'une fonction f .

x	-1	1	3	6
Variations de f				

Comment tracer une courbe « propre » représentant une fonction admettant ce tableau de variations ?

2 Mise en place



- La zone de travail de `\pspicture` est un rectangle défini par `\xmin`, `\xmax`, `\ymin` et `\ymax`.
- On trace les axes, on nomme l'origine O, puis les points I et J avec `\uput`.
- Les quatre points A, B, C et D sont les points intéressants de la courbe. Je les ai définis au moyen de `\Cnode*`. C'est la variable `radius` qui définit la taille du point ; en mettant `radius` à `0pt`, le point existe mais n'est pas dessiné.

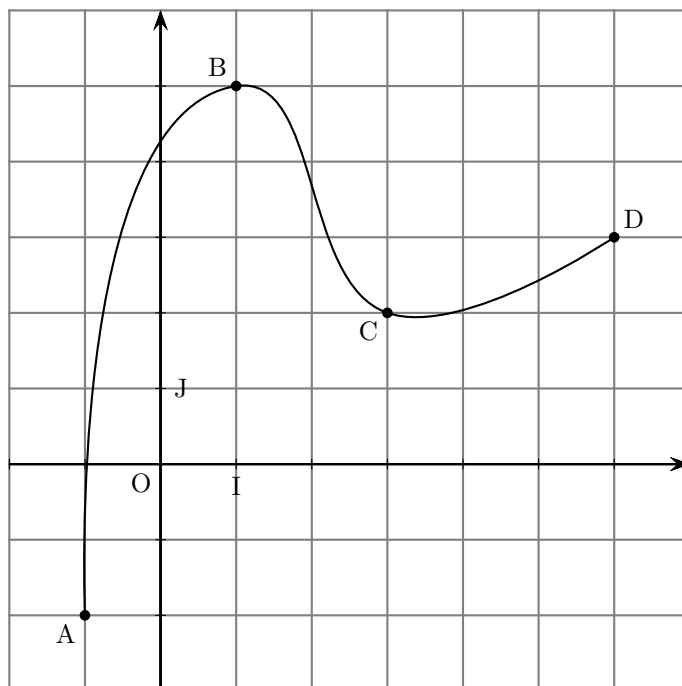
Cette façon de faire permet de parler des points en utilisant leurs noms au lieu de leurs coordonnées ; plus la figure est compliquée, plus ce procédé est efficace.

- Les noms des points sont écrits avec `\uput`.
- Une grille complète le tout.

3 Première idée

Il existe une instruction dans `PsTricks` qui permet de tracer une courbe passant par certains points; c'est `\pscurve`.

On essaie donc `\pscurve(A)(B)(C)(D)`.



Plutôt moche!

La courbe part verticalement du point A, semble même revenir en arrière, et le point C n'est pas un minimum.

Impossible de modifier cette courbe; il faut trouver autre chose.

4 Amélioration

Dans le package `pst-bezier`, il existe une instruction `\psbcurve` qui semble prometteuse!

J'ai déjà parlé des courbes de Bézier dans deux chroniques :

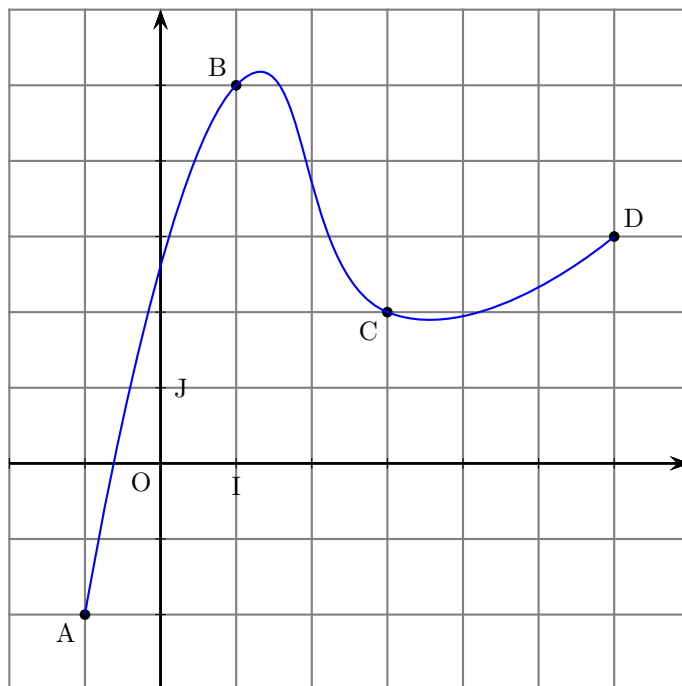
- [chronique de mai 2017](#);
- [chronique de novembre 2018](#).

Inutile de les relire pour utiliser `\psbcurve`; son mode d'emploi est dans le document `pst-bezier-doc`.

Il faut naturellement charger le package : `\usepackage{pst-bezier}`.

En haut de la page suivante de ce document, on voit ce que donne `\psbcurve(A)(B)(C)(D)`.

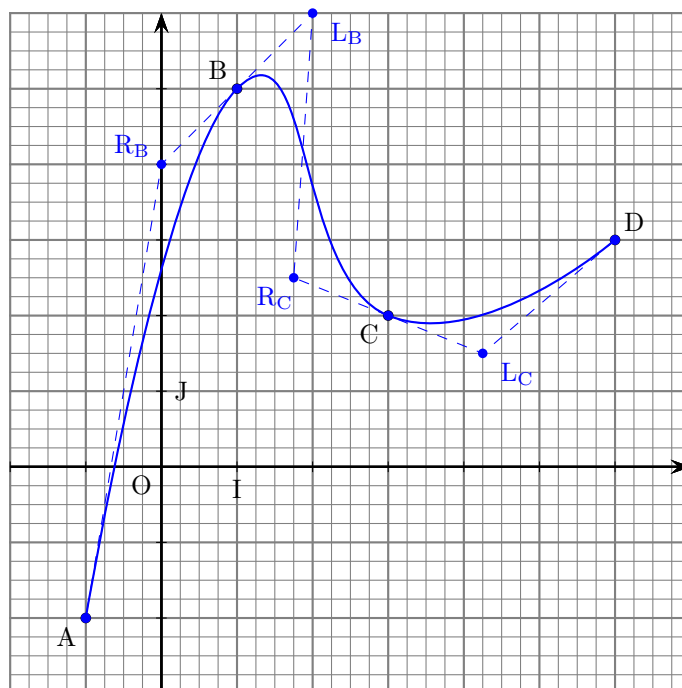
La courbe est nettement plus jolie mais le point B ne correspond plus à un maximum et le point C toujours pas à un minimum.



5 Les points de contrôle

Le `b` de `\psbcurve` signifiant « Bézier », il y a forcément des points de contrôle quelque part !

On fait afficher les points de contrôle en tapant `\psbcurve[showpoints](A)(B)(C)(D)` au lieu de `\psbcurve(A)(B)(C)(D)`, ou en entrant `\psset{showpoints=true}`.



Le quadrillage plus serré permet de lire les coordonnées des points de contrôle ; quatre points sont affichés : $R_B(0 ; 4)$, $L_B(2 ; 6)$, $R_C(1,75 ; 2,5)$ et $L_C(4,25 ; 1,5)$.

J'ai appelé R_B avec un « R » car ce point de coordonnées (0 ; 4) contrôle le point B de la courbe qui est à sa droite (Right). On comprend aisément pourquoi j'ai appelé L_B le point de coordonnées (2 ; 6) quand on sait que « Left » veut dire « Gauche » !

La droite ($R_B L_B$) est tangente en B à la courbe, et la droite ($R_C L_C$) est tangente en C à la courbe ; c'est ce qui assure la « régularité » de la courbe.

Comment ces points interviennent-ils dans l'instruction `\psbcurve` ?

En fait l'instruction `\psbcurve(A)(B)(C)(D)` est la même que

```
\psbcurve(A)r(0,4)(B)l(2,6)r(1.75,2.5)(C)l(4.25,1.5)(D) ;
```

vous pouvez vous en rendre compte en tapant la deuxième expression à la place de la première : c'est la même courbe qui est tracée.

Ce serait bien de pouvoir modifier ces points de contrôle. Mais est-ce possible ?

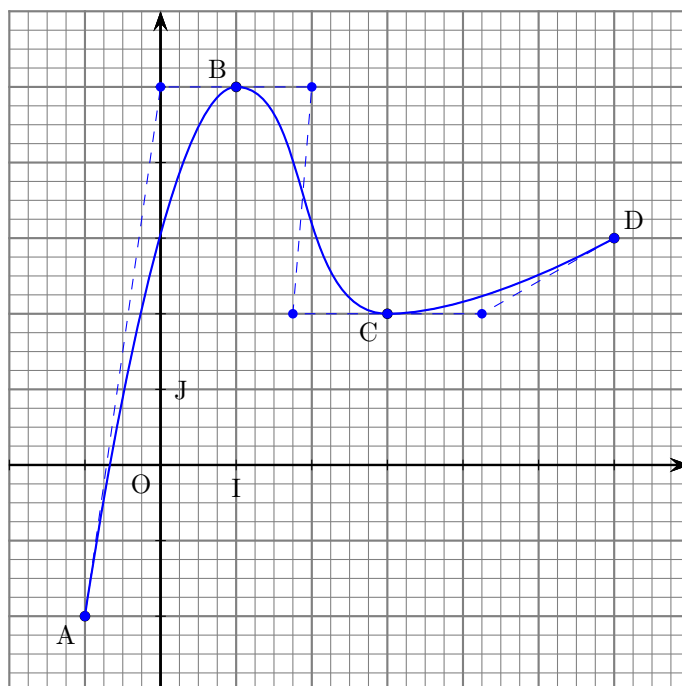
On se doute bien que la réponse est « oui » !

6 La solution

Pour qu'il y ait un maximum au point B, on remplace `r(0,4)(B)l(2,6)` par `r(0,5)(B)l(2,5)`, et pour qu'il y ait un minimum en C, on remplace `r(1.75,2.5)(C)l(4.25,1.5)` par `r(1.75,2)(C)l(4.25,2)`.

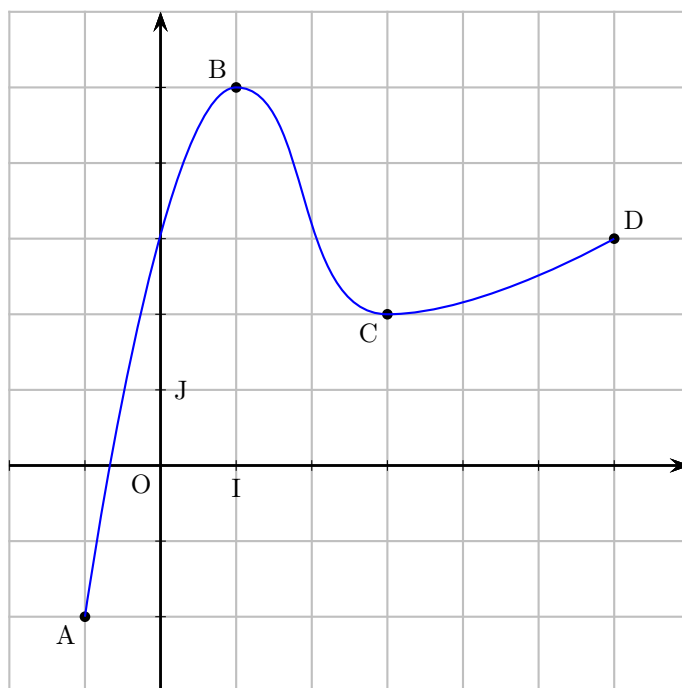
Autrement dit, on trace la courbe en entrant

```
\psbcurve(A)r(0,5)(B)l(2,5)r(1.75,2)(C)l(4.25,2)(D).
```



Le code de ce graphique se trouve en page 7.

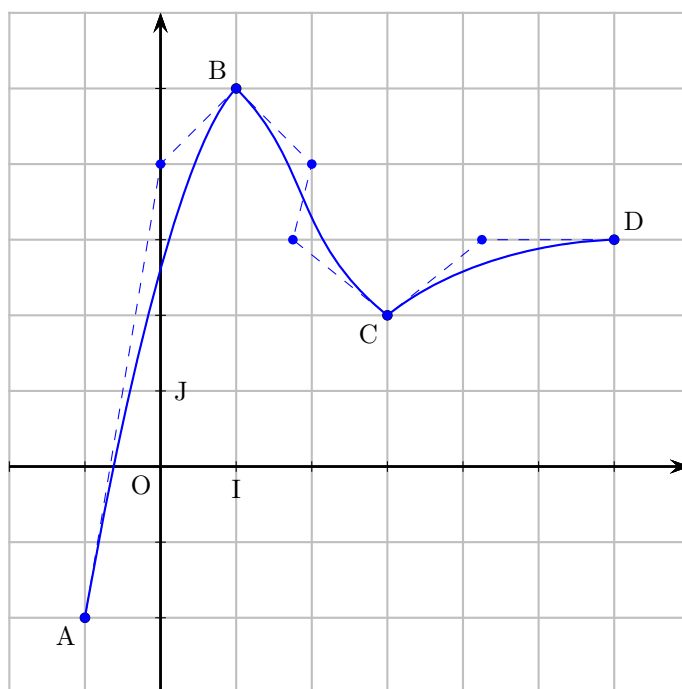
Ce qui donne, en nettoyant un peu la figure :



7 Points anguleux

Rien n'empêche de triturer un peu les points de contrôle pour dessiner des points anguleux en B et/ou en C. Voici ce qu'on obtient en entrant

```
\psbcurve(A)r(0,4)(B)l(2,4)r(1.75,3)(C)l(4.25,3)(D) :
```

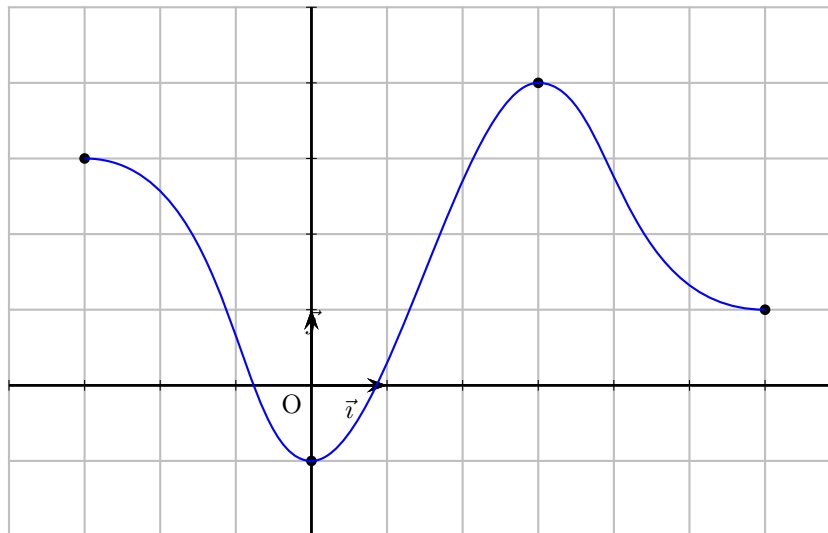


8 Autre exemple

On part de ce tableau de variations :

x	-3		0		3		6			
Signe de f'	\emptyset	-	\emptyset	+	\emptyset	-	\emptyset			
Variations de f	3	↘		-1	↗		4	↘		1

Et on trace cette courbe :



On remarque les tangentes horizontales aux deux extrémités de la courbe ; il a fallu rajouter un point de contrôle de type 1 après le point A, et un point de contrôle de type r avant le point D :

`\psbcurve(A)l(-1,3)r(-1,-1)(B)l(1,-1)r(2,4)(C)l(4,4)r(4,1)(D)`

Le code de ce graphique se trouve en page 7.

9 Résumé

- On place les points A, B, C, etc. par lesquels la courbe doit passer.
- On trace la courbe passant par les points en affichant les points de contrôle :
`\psbcurve[showpoints](A)(B)(C)...`
- On identifie les points de contrôle sur le graphique.
- On modifie les points de contrôle pour que la courbe ressemble à ce que l'on souhaite.
- On trace la courbe sans afficher les points de contrôle.

10 Quelques codes

Code de la courbe de le page 4 paragraphe 6.

```
\psset{unit=1cm,radius=2pt,arrowsize=3pt 3}  
\def\xmin {-2} \def\xmax {7}  
\def\ymin {-3} \def\ymax {6}  
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)  
\psgrid[subgriddiv=1,gridlabels=0,gridcolor=gray]  
\psaxes[ticks=-2pt 2pt, labels=none]{->}(0,0)(\xmin,\ymin)(\xmax,\ymax)  
\uput[d1](0,0){0} \uput[d](1,0){I} \uput[r](0,1){J}  
\Cnode*(-1,-2){A} \Cnode*(1,5){B}  
\Cnode*(3,2){C} \Cnode*(6,3){D}  
\uput[d1](A){A} \uput[d1](C){C}  
\uput[u1](B){B} \uput[ur](D){D}  
\psbcurve[linecolor=blue](A)(B)(C)(D)  
\end{pspicture}
```

Code de la courbe de le page 6 paragraphe 8.

```
\psset{unit=1cm,radius=2pt,arrowsize=3pt 3}  
\def\xmin {-4} \def\xmax {7}  
\def\ymin {-2} \def\ymax {5}  
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)  
\psgrid[subgriddiv=1,gridlabels=0,gridcolor=lightgray]  
\psaxes[ticks=-2pt 2pt, labels=none](0,0)(\xmin,\ymin)(\xmax,\ymax)  
\psaxes[ticks=-2pt 2pt, labels=none]{->}(0,0)(1,1)  
\uput[d1](0,0){0}  
\uput[d](0.5,0){$\vec{\imath}$}  
\uput[l](0,0.5){$\vec{\jmath}$}  
\Cnode*(-3,3){A} \Cnode*(0,-1){B}  
\Cnode*(3,4){C} \Cnode*(6,1){D}  
\psbcurve[linecolor=blue](A)l(-1,3)r(-1,-1)(B)l(1,-1)r(2,4)(C)l(4,4)r(4,1)(D)  
\end{pspicture}
```