

ANIMATION AVEC wxMAXIMA : WITH_SLIDER_DRAW

La commande `with_slider_draw` permet de créer des animations au sein d'une session wxMaxima.

Le contenu ci-dessous, développé par David A. Stegenga, fait partie d'un ensemble de ressources disponibles à l'adresse <http://www.math.hawaii.edu/home/wxmaxima.shtml>.

Des exemples supplémentaires sur wxMaxima et les graphiques en deux dimensions sont disponibles à l'adresse http://www.cs.us.es/~jalonso/cursos/ilm/temas/tema-CS7/tema_CS7.html.

L'animation présentée ci-dessous est activée en utilisant :

- soit le bouton de lecture (triangle),
- soit le curseur situé à sa droite dans la barre d'outils.

De plus amples instructions sont données ci-dessous. Les animations se déroulent au sein de la session wxMaxima. Pour des informations sur l'animation avec `draw`, consultez

<http://www.telefonica.net/web2/biomates/maxima/gpdraw/animation>.

NDT : depuis l'écriture du document par l'auteur, les contenus des liens ont été modifiés :

- le premier lien est obsolète et ne donne que le lien vers le site de wxMaxima;
- le second lien est très riche mais en espagnol;
- le troisième lien n'est plus disponible;
- le logiciel KompoZer n'est plus disponible, son développement ayant été arrêté en 2011.

Ces liens ont été conservés pour être fidèle au document d'origine.

Cette session wxMaxima contient deux exemples.

- Le premier utilise `wxdraw2d` et son équivalent pour la génération d'animations `with_slider_draw`.
- Le second exemple utilise `wxdraw3d` et son équivalent `with_slider_draw3d`.

Le contenu développé ici est exporté vers un fichier HTML, légèrement modifié avec l'éditeur open-source KompoZer (disponible sur <http://kompozer.net/>). Cette version n'est disponible que sur la version originale en anglais de ce document.

Les illustrations ont été développées avec la version 0.8.7 de wxMaxima et Maxima version 5.23.2 (date de compilation : 11 :22 02/02/2011) dans un environnement Windows XP, et testées avec la version 0.8.5 de wxMaxima et Maxima version 5.22.1 (date de compilation : 11/08/2010) dans un environnement Ubuntu.

NDT : le fichier `wxmx` a été mis à jour lors de cette traduction et le notebook wxMaxima fonctionne parfaitement avec wxMaxima version 26.01 et Maxima version 5.49.

Journal des modifications :

- (14.03.2019, eg, erklue@gmail.com) : Modification du nommage des graphiques car celui-ci produisait une erreur sous wxMaxima 19.03.1 (avec Maxima 5.42.2). Suppression de l'ancien `label`, remplacé par `key`. La valeur de l'aire du rectangle est maintenant mise à jour pendant l'animation.
- (15.03.2019, wxMaxima@physikbuch.de) : Suppression de nombreux retours à la ligne forcés dans les commentaires, wxMaxima prenant maintenant en charge le retour à la ligne automatique.

1 Utilisation de `with_slider_draw` dans `wxdraw2d`

Exemple. Le problème consiste à trouver le plus grand rectangle inscrit dans un demi-cercle de rayon 1.

Avant d'utiliser `with_slider_draw` pour animer le graphique, nous générons une représentation graphique statique, en utilisant le package graphique `draw` sans animation, montrant un tel rectangle typique inscrit sous la courbe $y = f(x)$ où $f(x) = (1 - x^2)^{1/2}$.

Le rectangle est déterminé par son sommet supérieur droit situé au point $(t, f(t))$, où $0 < t < 1$, et son aire est donnée par $(2t) \cdot f(t)$. Nous prenons d'abord $t = 0,5$ pour observer la figure.

La fonction et les calculs correspondants apparaissent dans le tableau ci-dessous.

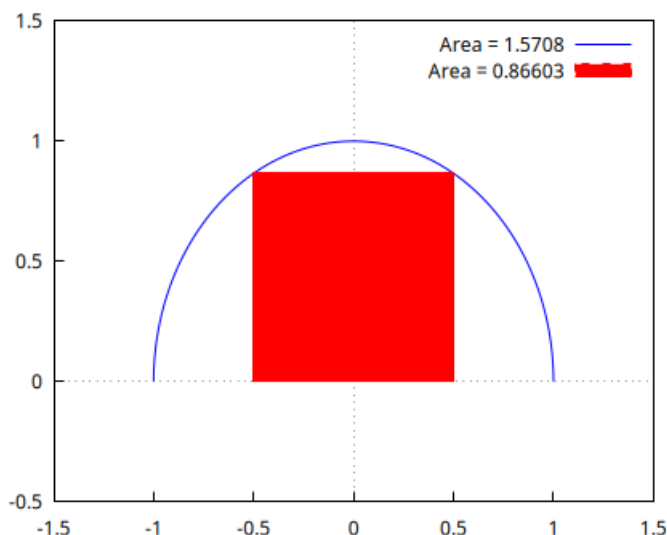
```
-> kill(all)$ fpprintprec :5$ load(draw)$
set_draw_defaults(font="Tahoma", font_size=9);
f(x) := (1-x^2)^(1/2)$ a : 0.5$ f(a)$ A : 2*a*f(a)$
matrix(["f(x)", "a", "f(a)", "A"], [f(x), a, f(a), A]);
```

```
(% o3) [font = "Tahoma", font_size = 9]
```

```
(% o8) [ "f(x)" "a" "f(a)" "A"
        [sqrt(1-x^2) 0.5 0.86603 0.86603 ]
```

La commande `draw` crée deux étiquettes. La première affiche le texte "Aire =" et la seconde montre la valeur de l'aire. La commande `explicit` crée la courbe et la commande `rectangle` crée le rectangle inscrit.

```
-> wxdraw2d(xrange = [-1.5,1.5], yrange = [-.5,1.5],
xaxis = true, yaxis = true,
key=concat("Area = ", float(%pi/2)),
color = blue, explicit(f(x),x,-1,1),
key=concat("Area = ", float(2*a*f(a))),
color = red, rectangle([-a,0],[a,f(a)]) )$
```



```
(% t9)
```

Avec cet ensemble de valeurs, nous obtenons une aire d'environ 0,866.

Nous créons ensuite une animation en utilisant la commande `with_slider_draw()` avec plusieurs valeurs du compteur t (d'autres noms peuvent être attribués au compteur). La commande `with_slider_draw` remplace la commande `wxdraw2d()`. Les deux premières entrées entre parenthèses doivent être un compteur et une liste de valeurs. Dans la plupart des cas, le moyen le plus simple de créer la liste est d'utiliser la commande `makelist`. Dans le cas ci-dessous, les valeurs de t allant de 0 à 1 sont créées, avec des incréments de 0,05.

Après avoir exécuté l'ensemble des commandes dans la cellule ci-dessous, cliquez sur le graphique. Le graphique sera mis en surbrillance. À ce stade, un clic sur le bouton de lecture dans la barre d'outils amène wxMaxima à faire défiler les 21 valeurs de t et à projeter les résultats à l'écran. L'animation continue jusqu'à ce que vous cliquiez n'importe où sur l'écran. Alternativement, cliquer sur le curseur après avoir mis le graphique en surbrillance permet une animation plus contrôlée.

```

-> with_slider_draw(      t, makelist(.05*i,i,0,20),
    xrange = [-1.5,1.5], yrange = [-.5,1.5],
    xaxis = true, yaxis = true,
    key=concat("Area=", float(%pi/2)),
    color = blue, explicit((1-x^ 2)^ .5,x,-1,1),
    key=concat("Area=", float(2*t*f(t))),
    color = red, rectangle([-t,0],[t,f(t)])      )$

```

(% t10) [animated graphics - not shown in TeX export]

Les animations ne peuvent pas être incluses dans un fichier PDF. Elles sont visibles sur le fichier wxMaxima disponible en téléchargement (fichier source à télécharger sur la documentation française de wxMaxima :

<https://maxima-french-doc.fr/documentation/>

Quelques expérimentations suggèrent que l'aire maximale est d'environ 1 et que la valeur de x qui génère cette aire est d'environ 0,7. Le plus grand rectangle n'est clairement pas un carré.

Bien sûr, Maxima peut être utilisé pour déterminer la valeur de x qui donne l'aire maximale. La série de commandes ci-dessous montre que la valeur critique de x est $x = 1/\sqrt{2}$ (environ 0,707), et que l'aire résultante est $A(1/\sqrt{2}) = 1,0$.

```

-> A(x) := 2*x*f(x);
    diff(A(x), x);
    solve(%, x);
    rhs(%[2]),numer;
    A(%);

```

(% o12) $A(x) := 2x f(x)$

(% o13) $2\sqrt{1-x^2} - \frac{2x^2}{\sqrt{1-x^2}}$

(% o14) $\left[x = -\left(\frac{1}{\sqrt{2}}\right), x = \frac{1}{\sqrt{2}} \right]$

(% o15) 0.70711

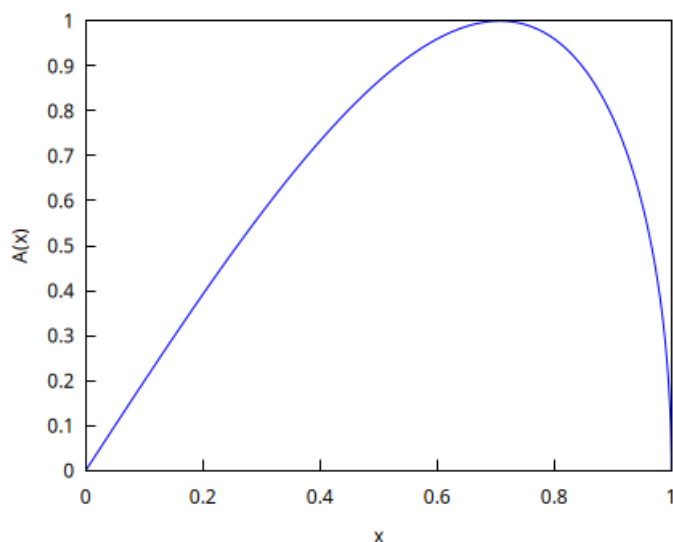
(% o16) 1.0

Le graphique ci-dessous confirme qu'un maximum est déterminé (comme l'animation l'a déjà démontré) et qu'un seul maximum local existe sur l'intervalle considéré.

```

-> wxdraw2d(xlabel = "x", ylabel = "A(x)", explicit(A(x),x,0,1))$

```



(% t17)

2 Utilisation de with_slider_draw dans wxdraw3d

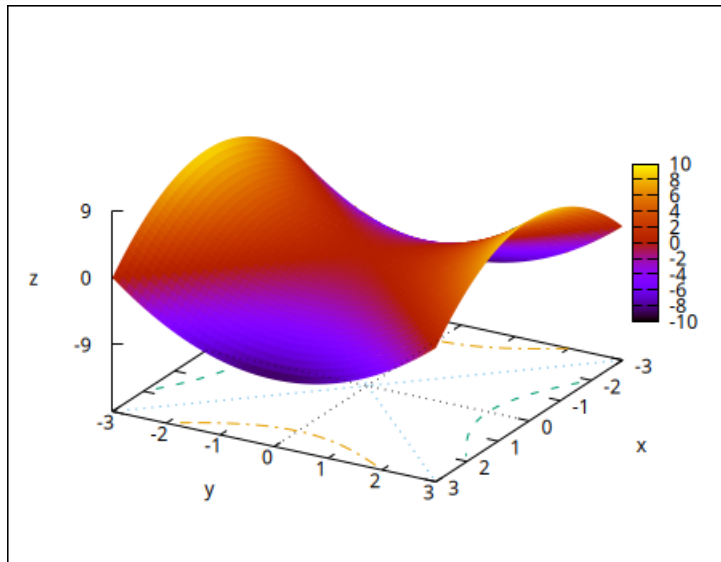
Cette section démontre que `with_slider_draw` en trois dimensions fonctionne de la même manière qu'en deux dimensions. Nous utilisons une surface en forme de selle et montrons comment l'animation peut être appliquée. La fonction est $z(x, y) = y^2 - x^2$.

Là encore, nous commençons par un graphique statique. Le graphique ci-dessous montre la surface et trois courbes de niveau : celles pour $z = -5$, $z = 0$ et $z = 5$. Les courbes de niveau apparaissent à la base du graphique et sur la surface, ce qui est le réglage par défaut.

Ci-dessous, nous limitons le graphique à une paire de courbes de niveau à la fois, mais les niveaux changent.

```
-> kill(all)$ z(x, y) := y^2 - x^2;  
wxdraw3d(view=[55,120], xaxis=true, yaxis=true,  
enhanced3d=true, contour = both,  
ztics=-9, 0, 9, xlabel="x", ylabel="y", zlabel="z",  
explicit(z(x, y), x, -3, 3, y, -3, 3)  
)$
```

```
(% o1)  $z(x, y) := y^2 - x^2$ 
```



```
(% t2)
```

La commande `with_slider_draw3d` remplace `wxdraw3d` et nous ajoutons un compteur, à nouveau nommé t . Les niveaux de contour sont fixés à deux valeurs données, $t - 1$ et $t + 1$.

```
-> with_slider_draw3d(t, makelist(0.2*i, i, -10, 10),  
xaxis=true, yaxis=true,  
enhanced3d=true, contour = both,  
ztics=-9, 0, 9, xlabel="x", ylabel="y", zlabel="z",  
contour_levels = t-1, t+1,  
explicit(z(x, y), x, -3, 3, y, -3, 3) )$
```

```
(% t3) [animated graphics - not shown in TeX export]
```

Les animations ne peuvent pas être incluses dans un fichier PDF. Elles sont visibles sur le fichier wxMaxima disponible en téléchargement (fichier source à télécharger sur la documentation française de wxMaxima :

<https://maxima-french-doc.fr/documentation/>

Le compteur peut être utilisé pour déplacer la surface elle-même, comme ci-dessous.

```
-> with_slider_draw3d(t, makelist(0.2*i,i,-10,10),  
    xaxis=true, yaxis=true,  
    enhanced3d=true, contour = both,  
    ztics=-9, 0, 9, xlabel="x", ylabel="y", zlabel="z",  
    contour_levels = t-1, t+1,  
    explicit(z(t*x, y), x, -3, 3, y, -3, 3) )$
```

(% t5) [animated graphics - not shown in TeX export]

Les animations ne peuvent pas être incluses dans un fichier PDF. Elles sont visibles sur le fichier wxMaxima disponible en téléchargement (fichier source à télécharger sur la documentation française de wxMaxima :

<https://maxima-french-doc.fr/documentation/>