

plotplots >> plotplots > plotplots

plotplots

Trace par intervalles des courbes ou un repère graphique existant, selon un axe fragmenté ou/et multiéchelle

Syntaxes

```
plotplots(x, y, cutaxis, intervals)
plotplots(axes0, cutaxis, intervals)
plotplots(.., "ticksmode", "alt" | "shift")
plotplots(.., "widths", "isoscaled" | widths)
[Axes, curves] = plotplots(x, y, ..)
Axes = plotplots(axes0, cutaxis, intervals, ..)
```

Sommaire

- [Arguments](#)
- [Description](#)
 - [Introduction](#)
 - [Titres, légendes, Données étiquetées](#)
- [Exemples](#)
 - [Options "ticksmode" et "widths". Adressage et restylage des courbes](#)
 - [Intervalle répliqué et inversé](#)
 - [Axe Y multi-échelle](#)
 - [Exemple avec Matplotlib\(\)](#)
- [Voir aussi](#)
- [Auteur](#)
- [Historique](#)

Arguments

x

vecteur ou matrice d'abscisses réelles.

- Si x et y sont des vecteurs, il faut $\text{length}(x) == \text{length}(y)$
- Si x est un vecteur et y est une matrice, les mêmes abscisses sont partagées par toutes les courbes données en colonnes de y , et nous devons avoir $\text{length}(x) == \text{size}(y, 1)$.
- Si x et y sont des matrices, elles doivent être de mêmes tailles. $x(:, i)$ sont alors les abscisses respectives de $y(:, i)$.

y

nombre réels : ordonnées des courbes à tracer. Si y est une matrice, chacune de ses colonnes représente une courbe à part.

axes0

Repère graphique existant à redessiner par intervalles choisis. `.isoview="off"` est forcé sur tous les intervalles représentés.

cutaxis

"x" ou "y" : indique l'axe fragmenté.

intervals

Matrice de tailles ($N \times 2$) de nombres réels. Chaque ligne représente les bornes `[start, end]` d'un intervalle sur lequel toutes les courbes doivent être tracées. Ces intervalles sont tous situés sur l'axe désigné par `cutaxis`. Le nombre d'intervalles possibles n'est pas limité.

Sur la figure, les intervalles sont représentés de la gauche vers la droite dans l'ordre des lignes de `intervals`. Les différentes portions de l'axe peuvent être ordonnées et présentées à souhait.

Si un intervalle est spécifié tel que `start > end`, l'axe est inversé sur la partie concernée.

"ticksmode"

Par défaut, un espace vide sans graduation ni tracé sépare deux intervalles consécutifs. Les graduations des différentes parties de l'axe fragmenté sont "alignées" les unes avec les autres.

Le mot optionnel "ticksmode" permet d'indiquer d'autres modes d'alternance des graduations de l'axe fragmenté. Il doit être suivi d'une des valeurs suivantes :

"alt" Les intervalles gradués sont dessinés alternativement en bas et en haut pour "x" fragmenté, ou à gauche et à droite pour "y".

"shift" Les intervalles gradués consécutifs sont dessinés légèrement décalés, de manière alternée.

"widths"

Par défaut, si N intervalles sont définis, la pleine largeur du tracé est segmentée en N sections de largeurs égales, quelles que soient les largeurs effectives $|end - start|$ des intervalles de données. Le mot "widths" permet de définir d'autres modes de répartition de l'espace total fragmenté. Il peut être suivi d'un des éléments suivants :

"-" Les espaces graphiques vides séparant les intervalles sont supprimés.

"isoscaled" Dans ce mode, tous les intervalles sont représentés avec la même échelle. La largeur graphique est proportionnelle à $|end - start|$.

"-isoscaled" fera la même chose, en supprimant en outre les espaces graphiques entre intervalles consécutifs.

widths Vecteur de nombres indiquant la largeur relative des différentes sections graphiques selon l'axe fragmenté. Le vecteur doit avoir autant d'éléments qu'il y a d'intervalles définis.

Seuls les rapports $|widths| / \text{sum}(|widths|)$ sont considérés, de sorte que $[1, 2, 2]$ et $[0.5, 1, 1]$ ou $[0.2, 0.4, 0.4]$ sont par exemple équivalents.

Une valeur négative indique que la séparation graphique précédant l'intervalle doit être supprimée. La valeur absolue est alors prise pour largeur relative effective.

Axes

vecteur des identifiants des repères graphiques correspondant aux différents intervalles. Ses éléments permettent de modifier les attributs des repères définissant l'axe fragmenté, tels que le mode logarithmique, etc. Voir les exemples illustrés.

Axes(\$) adresse le repère graphique définissant l'axe commun, la grille, et la légende associées. C'est le repère actif au retour de `plotplots()`.

curves

liste de vecteurs d'identifiants graphiques des morceaux des courbes tracées.

- `curves(i)` est le vecteur des identifiants graphiques de toutes les sections tracées pour la courbe n° i définie dans `y(:,i)`.
- `curves(i)(j)` adresse la courbe n° i tracée sur l'intervalle n° j défini par `intervals(j,:)`.

Ces identifiants permettent de modifier les attributs des courbes après leur tracé. Par exemple, `curves(2).foreground = color("red")` mettra en rouge tous les intervalles de la courbe n° 2. `curves(2)(3).line_style = 3` mettra uniquement sa 3ème section en tirets. Un exemple est fourni et illustré plus loin.

💡 Le vecteur des identifiants des courbes n° $c1$ à $c2 > c1$ pour l'unique intervalle n° i est donné par `Axes(i).children($).children($-c1+1:-1:$-c2+1)`.

Description

`plotplots()` permet de tracer des courbes selon un axe non régulier, par exemple comportant des intervalles cachés. Les courbes sont ainsi tracées sur les intervalles consécutifs visibles, sans faire apparaître en vide les intervalles à ignorer.

Lorsqu'un repère graphique `axes0` est fourni au lieu de courbes, il est répliqué par intervalle. Tout le contenu du repère est répliqué. Les titres de ses axes X Y et Z et son titre général sont ignorés.

Dans les deux cas (courbes ou repère entier), `plotplots()` dessine dans la zone graphique de `gca()`. Si `axes0` est le repère graphique courant `gca()`, `plotplots()` le supprime à l'issue de la réplification par intervalle. Sinon, `axes0` est conservé dans sa propre zone graphique.

Les intervalles représentés sur l'axe discontinu peuvent avoir des échelles différentes, différents modes logarithmique/linéaire, être directs ou inversés.

Un axe peut être gradué symétriquement en répétant un intervalle et en inversant sa copie.

💡 `plotplots()` peut être utilisé notamment en conjonction avec les fonctionnalités suivantes :

- `subplot(..)` OU `xsetech(..)` peuvent être utilisées pour définir la zone de dessin avant d'appeler `plotplots()`.
- Un rectangle à zoomer portant sur plusieurs intervalles peut être défini de manière interactive.

Titres, légendes, et étiquettes de courbes

axes0 initial répliqué :

Dans ce cas, le titre général et les légendes des axes X et Y de axes0 sont transférés dans le nouveau repère graphique composite. Si axes0 comporte des étiquettes de données (datatips), celles-ci sont également transférées. Chaque étiquette peut être interactivement bougée mais est astreinte à rester sur son intervalle.

plotplots(x,y,..) de courbes :

Titre général, titre de l'axe continu : title(), xtitle(), xlabel() lorsque cutaxis est "y", ou ylabel() lorsque cutaxis est "x" peuvent être utilisés comme d'habitude après avoir appelé plotplots().

Titre de l'axe fragmenté : il est fortement recommandé d'utiliser le titre de l'intervalle le plus central, plutôt que le titre du repère graphique général (lequel introduit un décalage parasite). L'identifiant de l'intervalle ciblé sera utilisé en appelant xlabel() ou ylabel(). Voir les exemples.

Étiquettes de courbes (datatips) : elles peuvent être définies comme d'habitude sur les courbes tracées par plotplots(). Cependant, aucune étiquette ne peut alors être glissée sur la courbe d'un intervalle à l'autre.

Bloc de légendes de courbes multiples : un tel bloc peut être défini dans le repère graphique d'un des intervalles, en ciblant préalablement l'identifiant graphique correspondant :

```
sca(axes(i));
legend(["Curve 1" "Curve 2"], "in_upper_left");
```

La position du bloc dans le repère intervalle se réfère aux bornes de celui-ci. Il n'est pas possible de cibler le repère graphique général axes(\$) pour définir et héberger ce bloc.

Exemples

Options "ticksmode" et "widths". Identification et modifications des courbes

```
x = [-7.5:0.5:200]';
y = exp(x/30).*(1+0.5*sin(x));

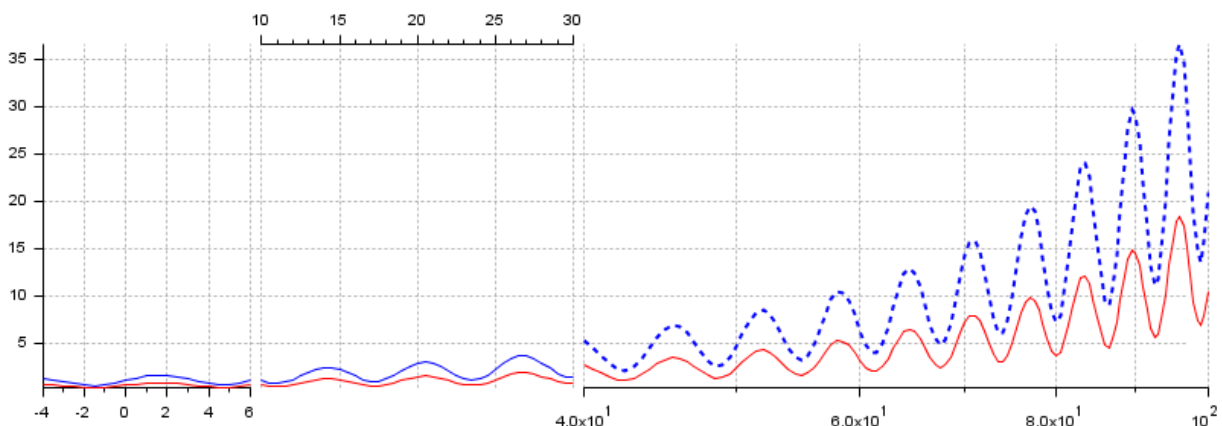
f = gcf(); f.axes_size = [1000 300];
clf

intervals = [-4 6; 10 30; 40 100];
[a, c] = plotplots(x, [y y/2], "x", intervals, "ticksmode", "alt", "widths", [1 1.5 3]);

// Adressage des courbes pour modifications
c(2).foreground = color("red"); // 2nde courbe en rouge
c(1)(3).line_style = 8; // 3ème partie de la 1ère courbe en tirets courts
c(1)(3).thickness = 2; // .. et trait plus épais

// Inversion haut-bas de L'axe normal Y
//a.axes_reverse = ["off" "on"];

// Mise en mode Logarithmique de La 3ème partie de L'axe fragmenté X
a(3).log_flags = "lnn";
```



Intervalle répliqué et inversé

plotplots() permet d'utiliser des intervalles se recouvrant partiellement, se répétant avec ou sans inversion ou changement d'échelle, etc. Cette souplesse peut être mise à profit de diverses façons. Par exemple, pour une abscisse radiale positive, il est possible de graduer en valeurs positives les distances à 0 des deux côtés, autant vers la gauche que vers la droite. Un côté peut être gradué en échelle radiale linéaire, et l'autre en échelle radiale logarithmique mettant en évidence un comportement asymptotique :

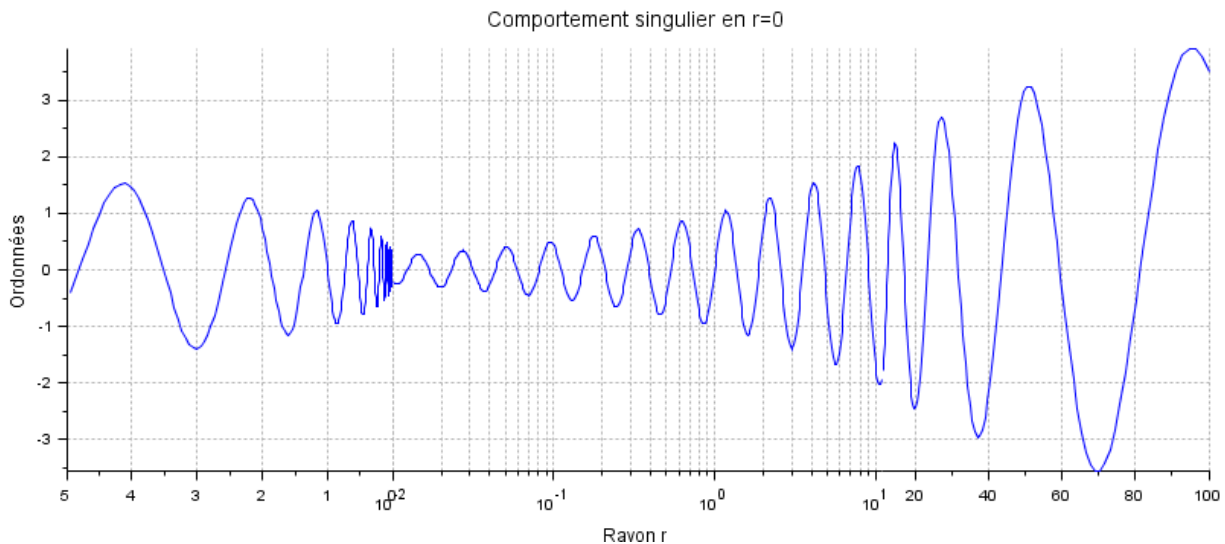
```

x = logspace(-2,2,500)';
y = sin(10*log(x)).*(x.^0.3);

f = gcf(); f.axes_size = [1000 370];
clf

intervals = [5 0 ; 0.01 11 ; 11 max(x)];
[a, c] = plotplots(x, y, "x", intervals, "widths", -[2 3 2]);
a(2).log_flags = "lnn";
a(2).sub_ticks(1) = 8;
title("Comportement singulier en r=0", "fontsize", 3)
ylabel("Ordonnées", "fontsize", 2)
xlabel(a(2), "Rayon r", "fontsize", 2);

```



Axe Y multiéchelle :

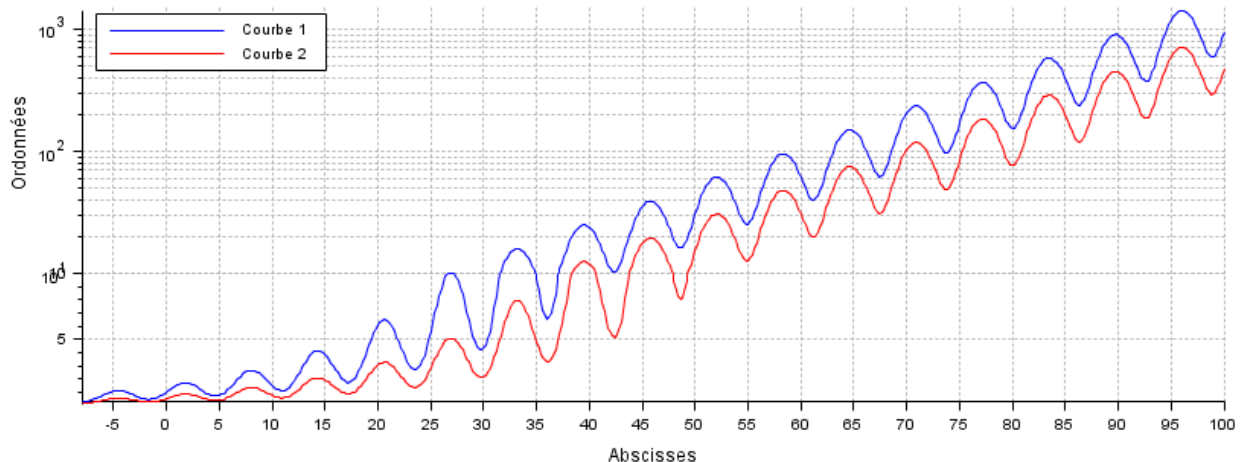
Exemple n° 1, avec légendes des courbes

```

x = [-8;0.2:100]';
y = exp(x/14).*(1+0.5*sin(x));
intervals = [0.01 10 ; 10 max(y)];

f = gcf(); f.axes_size = [1000 400];
clf
[a, c] = plotplots(x, [y y/2], "y", intervals, "widths", -[1 2]);
c(2).foreground = color("red");
a(2).log_flags = "nln";
a(2).sub_ticks(2) = 8;
xlabel("Abscisses", "fontsize", 2)
ylabel(a(2),"Ordonnées", "fontsize", 2) // a(2) est maintenant Le repère graphique actif
legend(["Courbe 1" "Courbe 2"], "in_upper_left");
// Grille
// a($).grid(1) = -1; // supprime la grille verticale
// for i=1:length(a)-1, a(i).grid(2)=-1; end // supprime Les grilles horizontales

```



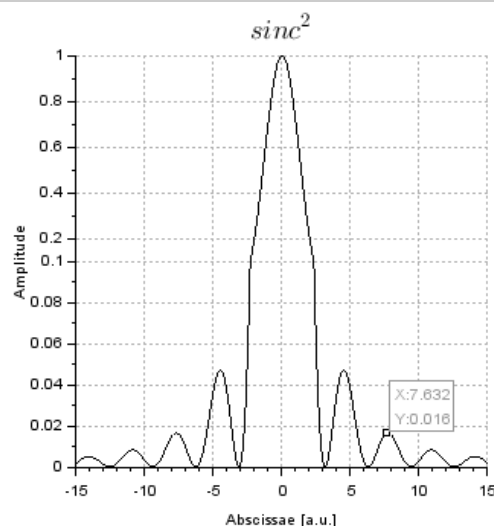
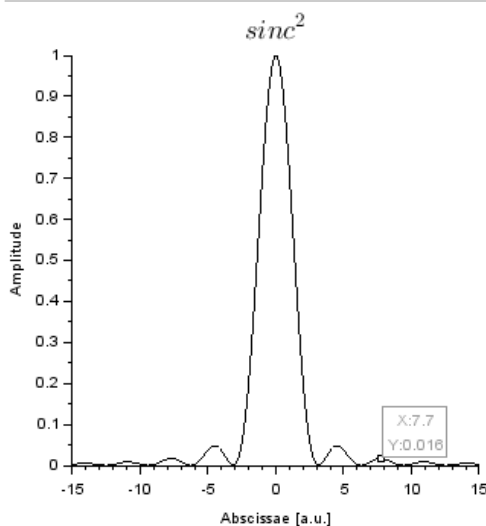
Exemple n° 2: faibles rebonds des ailes de sinc²()

Ci-dessous, nous traçons la courbe pour créer son repère graphique initial dans la zone subplot(1,2,1). Nous déplaçons ensuite gca() en subplot(1,2,2), avant d'appeler plotplots(..) :

```
x = linspace(-15,15,400);
clf
subplot(1,2,1)
gcf().axes_size = [720 360];
plot2d(x, sinc(x).^2)
ax0 = gca();

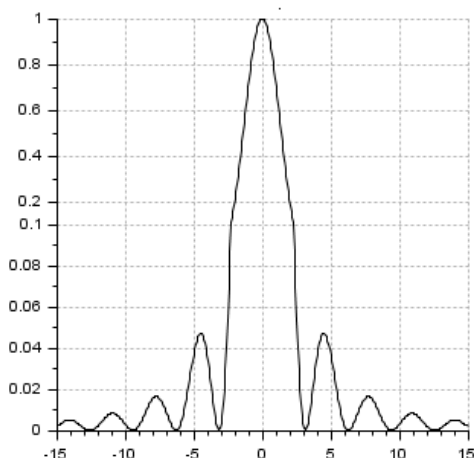
d = datatipCreate(gcf().children, [7.7 0.02]);
xtitle("\$\\LARGE sinc^2\$", "Abscissae [a.u.]", "Amplitude")
grey = color("grey60");
set(d, "mark_size",4, "foreground",grey, "font_foreground",grey, "orientation",8);

subplot(1,2,2)
Axes = plotplots(ax0, "y",[0 0.1 ; 0.1 1], "widths","-");
is_handle_valid(ax0) // %T: Le repère graphique initial subsiste
```



Ci-après, gca() reste sur la zone où nous traçons la courbe dans son repère graphique initial, avant d'appeler plotplots(). En conséquence, plotplots dessine dans la même zone et supprime le repère graphique initial après sa réplication par intervalles :

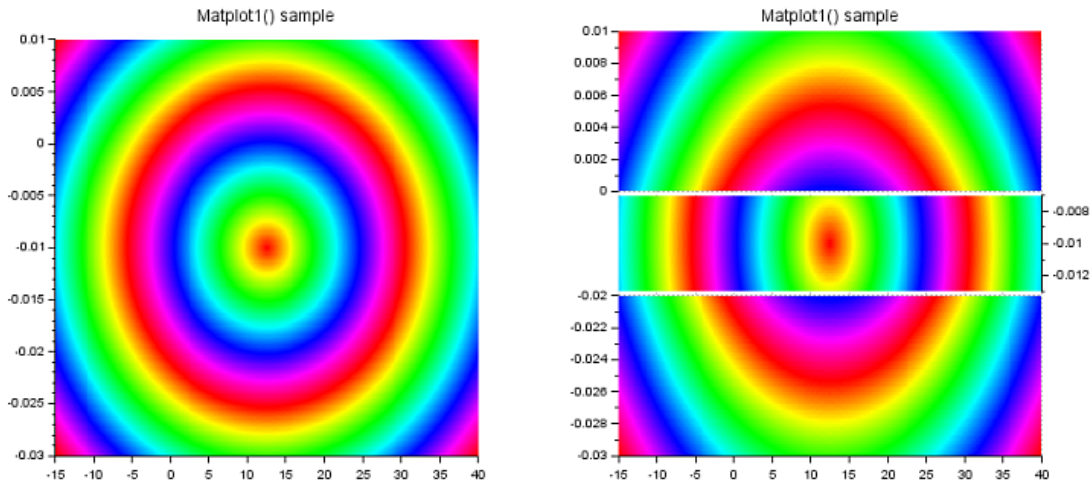
```
x = linspace(-15,15,400);
clf
subplot(1,2,1)
gcf().axes_size = [720 360];
plot2d(x, sinc(x).^2)
ax0 = gca();
Axes = plotplots(ax0, "y",[0 0.1 ; 0.1 1], "widths","-");
is_handle_valid(ax0) // %F: Le repère graphique initial a été supprimé
```



Exemple avec Matplot1()

```
clf
gcf().axes_size = [925 455];
subplot(1,2,1)
Matplot1();
xtitle("", "", "");
title("Matplot1() sample", "fontsize", 3);
x0 = gca();

subplot(1,2,2)
intervals = [-0.03 -0.02; -0.013 -0.007 ; 0 0.01];
axes = plotplots(x0, "y", intervals, "ticksmode", "alt", "widths", "isoscaled");
```



Voir aussi

- <https://atoms.scilab.org/toolboxes/plotplots>
- [plot2d](#)
- [plot](#)

Auteur

(C) Samuel GOUGEON

Historique

Version	Description
1.0 - 2018-01	Publication initiale.
2.0 - 2018-06	<ul style="list-style-type: none">• <code>plotplots()</code> peut désormais traiter un repère graphique existant avec tous ses composants.• <code>plotplots(...,intervals,cutaxis,...)</code> devient <code>plotplots(...,cutaxis,intervals,...)</code>• <code>[hcurves, Axes] = plotplots(x,y,...)</code> devient <code>[Axes, hcurves] = plotplots(x,y,...)</code>• Les messages d'erreur sont désormais traduits en français, allemand, japonais, brésilien, russe, et espagnol.

[Report an issue](#)

[<< plotplots](#)

[plotplots](#)