

Le modèle semiLog 0.35

Pour TeXgraph 1.94

Résumé

Description des macros du fichier `semiLog.mac` pour la création de grilles et de tracés logarithmiques ou semi-logarithmiques.

Table des matières

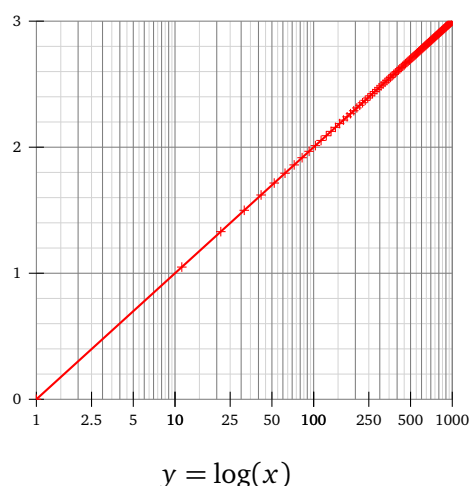
1 Introduction	1
2 La macro <code>viewlog()</code>	1
3 La macro <code>gridlog()</code>	2
4 Les macros <code>dotlog()</code>, <code>linelog()</code> et <code>droitelog()</code>	2
5 Changements	3
5.1 Version 0.3	3
5.2 Version 0.2	3
5.3 Version 0.1	4

1 Introduction

Le modèle *semiLog.mod* charge les macros du fichier *semiLog.mac*, celles-ci permettent la création d'une grille avec l'un des axes (ou les deux) gradué logarithmiquement, ainsi que le dessin d'un nuage de points, d'une ligne polygonale ou d'une droite. L'utilisateur a donc principalement cinq macros à sa disposition :

- **viewlog()** : qui permet de définir le type de représentation ($\log x$ ou $\log y$ ou $\log xy$), ainsi que les intervalles pour x et y ainsi que la taille du graphique. Cette macro est à utiliser en premier.
- **gridlog()** : fait le dessin de la grille en fonction des options.
- **dotlog()** : dessine un nuage de points.
- **linelog()** : dessine une (ou plusieurs) lignes polygonales.
- **droitelog()** : dessine une droite sur la grille.

```
\begin{texgraph}[name=semilogx,file]
Include "semiLog.mod";
Graph objet1=[
viewlog( logx, 1, 1000, 0,3, [size:=7,
Lmargin:=0.5,Bmargin:=1]),
LabelSize:=tiny,
gridlog([labels:=1, nbsubdivx:=2]),
tMin:=1, tMax:=1000,
L:= Get( Cartesienne(ln(x)/ln(10)), 0),
Color:=red, Width:=8, linelog(L),
DotStyle:=cross, dotlog(L),
Color:=black, LabelSize:=normalsize,
Label((Xmax+Xmin)/2+Ymin*i,
"$y=\mathrm{log}(x)$");
\end{texgraph}
```



2 La macro viewlog()

Cette macro prépare le terrain pour le graphique à venir. Sa syntaxe est :

```
viewlog( type, xmin, xmax, ymin, ymax , [options])
```

Le premier paramètre peut prendre comme valeur : `logx` ou `logy` ou `logxy`. La macro initialise la variable globale `typelog` à la valeur de ce paramètre.

Les quatre paramètres suivants déterminent les intervalles sur les deux axes. La macro initialise les variables globales `xmin`, `xmax`, `ymin`, `ymax`, `powerminx`, `powerminy`, `decadex`, `decadey`.

Le dernier paramètre est constitué d'une liste d'options, qui sont :

- `size := < nombre >`, détermine la taille du graphique en cm (marges incluses), cette taille se présente sous la forme : `largeur+i*hauteur`, si la hauteur déclarée est nulle, alors celle-ci est considérée égale à la largeur. La taille vaut 12 par défaut.
- `margin := < nombre >`, marges en cm autour de la grille (nulle par défaut),
- `Tmargin := < nombre >`, marge en cm au-dessus de la grille,
- `Bmargin := < nombre >`, marge en cm en-dessous de la grille
- `Lmargin := < nombre >`, marge en cm à gauche de la grille
- `Rmargin := < nombre >`, marge en cm à droite de la grille

Lorsque `margin` est non nulle, les quatre marges prennent automatiquement cette valeur. Ces marges sont à l'intérieur de la zone de dessin, ce qui permet d'ajouter des éléments graphiques à côté de la grille, comme une légende par exemple.

3 La macro gridlog()

Cette macro fait le dessin de la grille. Sa syntaxe est :

```
gridlog( [options] )
```

Les options sont :

- `width := < nombre >`, épaisseur du trait de la grille principale (4 par défaut),
 - `subgridwidth := < nombre >`, épaisseur du trait de la grille secondaire (2 par défaut),
 - `color := < nombre >`, couleur de la grille principale (gray par défaut),
 - `subgridcolor := < nombre >`, couleur de la grille secondaire (lightgray par défaut),
 - `dots := < 0/1 >`, indique si la grille principale est en pointillés ou non (0 par défaut),
 - `subgriddots := < 0/1 >`, indique si la grille secondaire est en pointillés ou non (0 par défaut),
 - `y := < [ymin, ymax] >`, indique l'étendue sur Oy lorsque l'axe Ox seulement est logarithmique. Par défaut c'est l'étendue déclarée dans `viewlog`, mais on peut diminuer cette étendue pour avoir une marge en haut et/ou en bas du graphique pour d'éventuels labels par exemple,
 - `x := < [xmin, xmax] >`, indique l'étendue sur Ox lorsque l'axe Oy seulement est logarithmique. Par défaut c'est l'étendue déclarée dans `viewlog`, mais on peut diminuer cette étendue pour avoir une marge à gauche et/ou à droite du graphique pour d'éventuels labels par exemple,
 - `nbdivy := < entier positif >`, nombre d'intervalles sur l'axe Oy lorsque celui-ci n'est pas logarithmique, (1 division par unité par défaut),
 - `nbsubdivy := < entier positif >`, nombre de sous-intervalles par division sur l'axe Oy même si celui-ci est logarithmique (1 par défaut s'il est logarithmique, 5 sinon),
 - `nbsubdivx := < entier positif >`, nombre de sous-intervalles par division sur l'axe Ox même si celui-ci est logarithmique (1 par défaut s'il est logarithmique, 5 sinon),
 - `labels := < 0/1 >`, indique si les labels doivent apparaître avec la grille sur les deux axes (0 par défaut),
 - `labelx := < 0/1 >`, indique si les labels doivent apparaître avec la grille sur l'axe Ox (0 par défaut),
 - `labeledy := < 0/1 >`, indique si les labels doivent apparaître avec la grille sur l'axe Oy (0 par défaut),
- Si `labels=1` alors `labelx` et `labeledy` sont ignorés.

- **loglabelx** := < liste de nombres >, indique les labels qui doivent apparaître sur l'axe Ox lorsque celui-ci est logarithmique. Si *loglabelx*=Nil (valeur par défaut), TeXgraph prend des labels par défaut.
- **loglabely** := < liste de nombres >, indique les labels qui doivent apparaître sur l'axe Oy lorsque celui-ci est logarithmique. Si *loglabely*=Nil (valeur par défaut), TeXgraph prend des labels par défaut.
- **labelcolor** := < nombre >, couleur des labels (black par défaut).

4 Les macros dotlog(), linelog() et droitelog()

`dotlog(liste de points, [otions])`

Cette macro permet le dessin d'un nuage de points sur la grille. Ces points sont exprimés sous forme d'affixe. Les options sont :

- **clip** := < 0/1 >, indique si la grille doit clipper le dessin (1 par défaut),
- **epsilon** := < nombre positif >, si le clipping a lieu, celui-ci est fait à *epsilon* près (ce paramètre est nul par défaut).

Dans les options il est possible de modifier les attributs (Color, DotStyle, ...), ces changements seront pris en compte mais ils sont locaux.

`linelog(liste de points, [otions])`

Cette macro permet le dessin d'une ligne polygonale sur la grille. Ces points sont exprimés sous forme d'affixe. Les options sont :

- **close** := < 0/1 >, indique si la ligne doit être refermée (0 par défaut),
- **clip** := < 0/1 >, indique si la grille doit clipper le dessin (1 par défaut),
- **epsilon** := < nombre positif >, si le clipping a lieu, celui-ci est fait à *epsilon* près (ce paramètre est nul par défaut).

Dans les options il est possible de modifier les attributs (Color, DotStyle, ...), ces changements seront pris en compte mais ils sont locaux.

`droitelog(A, B, [otions])`

Cette macro permet le dessin de la droite (AB) sur la grille. Ces points sont exprimés sous forme d'affixe. Les options sont :

- **clip** := < 0/1 >, indique si la grille doit clipper le dessin (1 par défaut),
- **epsilon** := < nombre positif >, si le clipping a lieu, celui-ci est fait à *epsilon* près (ce paramètre est nul par défaut).

Dans les options il est possible de modifier les attributs (Color, DotStyle, ...), ces changements seront pris en compte mais ils sont locaux.

5 Changements

5.1 Version 0.3

- Correction d'un bug dans la macro *gridlogxy*.

5.2 Version 0.2

- options pour la macro *viewlog* :
 - **size** := < nombre >
 - **margin** := < nombre >
 - **Tmargin** := < nombre >

- `Bmargin := < nombre >`
- `Lmargin := < nombre >`
- `Rmargin := < nombre >`
- nouvelles options pour la macro *gridlog* :
 - `labelx := < 0/1 >`
 - `labeledy := < 0/1 >`
 - `loglabelx := < liste de nombres >`
 - `loglabeledy := < liste de nombres >`
- options pour la macro *dotlog* :
 - `clip := < 0/1 >`
 - `epsilon := < nombre positif >`
- options pour la macro *linelog* :
 - `close := < 0/1 >`
 - `clip := < 0/1 >`
 - `epsilon := < nombre positif >`
- nouvelle macro *droitelog*(*A*, *b*, [*options*]) qui trace la droite (AB) sur la grille.

5.3 Version 0.1

Première version