

Le modèle CourbesNiv 0.5

Pour TeXgraph 1.95

Résumé

Description des macros du fichier CourbesNiv.mac pour la création de courbes de niveaux.

Table des matières	6 Définition de plans	2
1 Introduction	1 7 Définition de sections planes	2
2 Définition de la surface	1 8 Définition d'une légende	2
3 Définition de la fenetre	1 9 Variables pour gérer l'affichage	2
4 Définition de la vue	2 10 Affichage de la scène	3
5 Définition des courbes de niveaux	2 11 Exemples	3

1 Introduction

Le modèle *CourbesNiv.mod* charge les macros du fichier *CourbesNiv.mac*, celles-ci permettent le dessin d'une surface cartésienne avec des courbes de niveaux en z . Un certain nombre d'options permettent de gérer les axes, la vue, les traits cachés ...

2 Définition de la surface

Elle se fait par l'intermédiaire de la macro :

`surface("expression", [options])`

Le premier argument est l'expression qui définit la surface, elle peut-être de deux types :

- Expression cartésienne : " $f(x,y)$ ", représentant la surface d'équation cartésienne $z = f(x, y)$. Dans ce cas le paramétrage défini par la macro est : $x(u, v) = u$, $y(u, v) = v$, et $z(u, v) = f(u, v)$.
- Expression paramétrique : " $M(x(u,v), y(u,v), z(u,v))$ ", ou bien " $[x(u,v)+i*y(u,v), z(u,v)]$ ", représentant le paramétrage d'une surface. Les noms des paramètres sont obligatoirement u et v .

Les options sont :

- `mode := < cartésien/paramétrique >`, détermine le type de surface (cartésien par défaut).
- `grille := < [uMin, uMax, vMin, vMax, uNblignes, vNblignes] >`, indique l'intervalle du paramètre u , l'intervalle du paramètre v , le nombre de valeurs prises par le paramètre u et le nombre de valeurs prises par le paramètre v . Par défaut la grille est $[-2, 2, -2, 2, 20, 20]$.

3 Définition de la fenetre

Elle se fait par la macro :

`fenetre([options])`

Les options sont :

- `x := < [xMin, xMax] >`, détermine l'intervalle des abscisses. Par défaut cet intervalle vaut $[-2, 2]$.
- `y := < [yMin, yMax] >`, détermine l'intervalle des ordonnées. Par défaut cet intervalle vaut $[-2, 2]$.
- `z := < [zMin, zMax] >`, détermine l'intervalle des cotes. Par défaut cet intervalle est mis à Nil lors de l'exécution de la macro, si la valeur de z est laissée à Nil alors l'intervalle sera automatiquement calculé pour s'adapter à la surface.
- `pas := < [xPas, yPas, zPas] >`, indique le pas des graduations pour les trois axes (1 par défaut).
- `orthonorme := < 0/1 >`, indique si le repère 3D doit être orthonormé ou non.

4 Définition de la vue

Elle se fait par la macro :

`vue(x0y ou x0z ou y0z)`

Par défaut la vue est une vue dans l'espace, mais cette macro permet d'obtenir une projection sur un des trois plans. Pour revenir à la vue spatiale, on saisit `vue()`.

Remarque : pour la vue dans l'espace, les angles de vue peuvent être modifiés, ils correspondent aux variables θ et ϕ (en radians), par défaut ils valent respectivement $\frac{\pi}{6}$ et $\frac{\pi}{3}$.

5 Définition des courbes de niveaux

Elle se fait par la macro :

`niveaux([liste de valeurs de z] <, [liste de couleurs]>)`

Le deuxième argument est la liste des couleurs, celle-ci est optionnelle et lorsqu'elle est absente, la couleur de chaque niveau est calculée en fonction de z , dans une palette qui est contenue dans la variable *palette*, celle-ci vaut par défaut : [purple, blue, green, yellow, orange, red, crimson].

La liste des valeurs de z est automatiquement rangée dans l'ordre croissant et cette liste est vide par défaut.

6 Définition de plans

Elle se fait par la macro :

`tracerplans("x/y, [valeurs], [attributs], x/y, [valeurs], [attributs],...")`

Cela permet de dessiner sur la surface la trace de plans d'équation $x = a$ ou $y = a$. Les options d'attributs sont des instructions comme par exemple : [Color:=blue, Width:=8, ...].

7 Définition de sections planes

Elle se fait par la macro :

`sections(plan1, plan2, ...)`

Chaque plan est défini par un point et un vecteur normal, un plan est donc une liste de la forme : [point3D, vecteur3D], c'est à dire : [M(x,y,z), M(x,y,z)] (le vecteur normal est en deuxième position. La surface sera sectionnée par le plan et seule la partie qui est dans le demi-espace contenant le vecteur normal sera affichée.

8 Définition d'une légende

Elle se fait par la macro :

`legende("titre")`

Elle sera affichée sous le graphique, et centrée.

9 Variables pour gérer l'affichage

- `dollar := < 0/1 >`, permet d'ajouter un dollar autour des labels du repère, et les légendes des niveaux (1 par défaut).
- `lissage := < 0/1 >`, permet de lisser ou non la surface (0 par défaut, réservé aux sorties eps et pstricks).
- `voiraxes := < 0/1 >`, permet d'afficher ou non le repère (1 par défaut).
- `voirboite := < 0/1 >`, permet d'afficher ou non les arêtes de la boîte contenant la scène (0 par défaut).
- `voirlegendeNiv := < 0/1 >`, permet d'afficher ou non la légende des lignes de niveau (z =valeur) (1 par défaut).
- `voirtraces := < 0/1 >`, permet d'afficher ou non la trace des plans sur la boîte contenant la scène (0 par défaut).
- `styleNiv := < styleColor/styleLine/styleColor/styleCont >`, permet de choisir le style d'affichage des lignes de niveaux : bandes de couleurs, simples traits de couleurs, type chaleur (les facettes sont peintes en fonction de l'altitude z , des couleurs les plus froides vers les couleurs les plus chaudes), ou type continu (les facettes sont peintes en fonction de l'altitude z , la couleur variant de manière continue dans la palette) par défaut la valeur est `styleColor`.
- `traitscaches := < noline/solid/dashed/dotted >`, permet de choisir le style des lignes cachées (*dotted* par défaut).

10 Affichage de la scène

Elle se fait par la macro :

```
afficher([options axes], [actions], [options surfaces])
```

La liste [options axes] contient les options pour la macro *BoxAxes3D* (consulter l'aide sur cette macro), par défaut on a les options :

```
[grid:=1, Color:=black, LabelSize:=scriptsize, FillStyle:=full, FillColor:=beige,
xstep:=Xpas, ystep:=Ypas, zstep:=Zpas,
xlabelsep:=0.25, ylabelsep:=0.25, zlabelsep:=0.25]
```

La liste [actions] est une liste d'instructions qui sera exécutée après le dessin des axes, et avant le dessin de la surface, cette liste est vide par défaut.

La liste [options surfaces] contient des options d'attributs pour le dessin de la surface. Par défaut ces options sont :

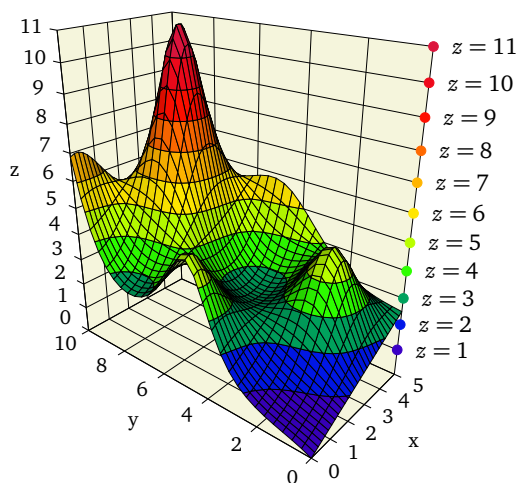
```
[Width:=8, Color:=black, FillStyle:=full, FillColor:=slategray,
LineCap:=round, LineStyle:=solid]
```

11 Exemples



Style bandes de couleurs

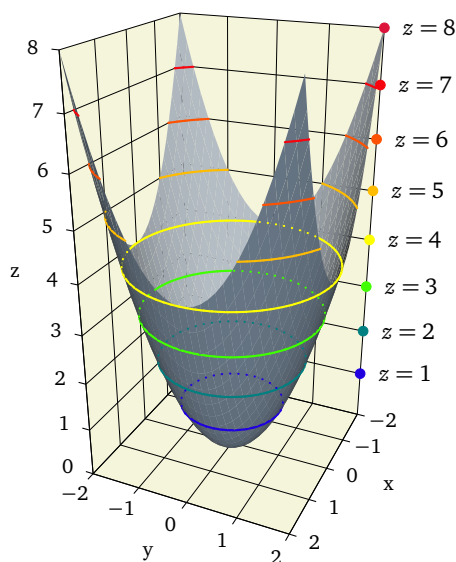
```
\begin{texgraph}[name=boite1, file]
Include "CourbesNiv.mod";
Graph image = [
Marges(0, 0, 0, 0), view(-10, 10, -11, 8), size(9), theta := 220*deg,
surface( "(x+y)/(2+sin(y)*cos(x))", [ grille := [0, 5, 0, 10, 30, 30] ]),
fenetre( [ x := [0, 5], y := [0, 10], z := [0, 11], pas := [1, 2, 1], orthonorme := 1 ],
niveaux( for k from 1 to 11 do k od ), legende( "$\frac{x+y}{2+\sin(y)\cos(x)}$",
afficher()
];
\end{texgraph}
```



$$\frac{x+y}{2+\sin(y)\cos(x)}$$

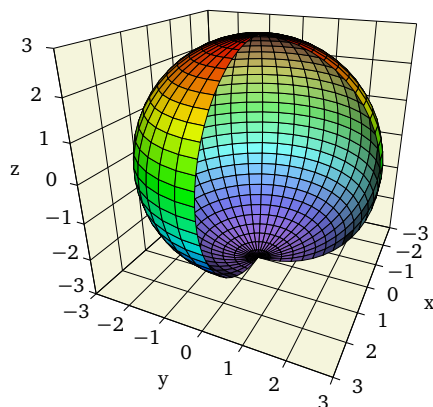
Style traits

```
\begin{texgraph}[name=boite2, file]
Include "CourbesNiv.mod";
Graph image = [
view(-8, 7.5, -8.5, 7), Marges(0, 0, 0, 0), size(9),
surface( "-u+i*v,u^2+v^2", [ mode :=parametrique, grille :=[-2, 2, -2, 2, 30, 30] ]),
fenetre( [x :=[-2, 2], y :=[-2, 2], z :=[0, 8], pas :=[1, 1, 1], orthonorme :=1] ),
niveaux( for k from 1 to 8 do k od ), styleNiv := styleLine,
afficher([], [], [LineStyle :=noline])
];
\end{texgraph}
```



Style chaleur

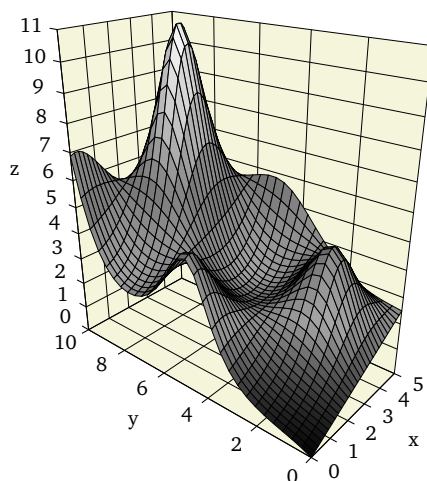
```
\begin{texgraph}[name=boite3, file]
Include "CourbesNiv.mod";
Graph image = [
Marges(0, 0, 0, 0), size(8),
surface( "3*[exp(-i*u)*sin(v),cos(v)]", [ mode :=parametrique, grille :=[0, 3*pi/2, 0, pi, 30, 30] ]),
fenetre( [x :=[-3, 3], y :=[-3, 3], z :=[-3, 3], pas :=[1, 1, 1], orthonorme :=1] ),
styleNiv := styleCalor, afficher()
];
\end{texgraph}
```





Style continu

```
\begin{texgraph}[name=boite4, file]
Include "CourbesNiv.mod";
Graph image = [
Marges(0, 0, 0, 0), view(-10, 10, -11, 8), size(9), theta :=220*deg,
surface( "(x+y)/(2+sin(y)*cos(x))", [ grille :=[0, 5, 0, 10, 30, 30] ]),
fenetre( [x :=[0, 5], y :=[0, 10], z :=[0, 11], pas :=[1, 2, 1], orthonorme :=1] ),
legende("$\frac{x+y}{2+\sin(y)\cos(x)}$"), palette :=[black, darkgray, gray, lightgray, white],
styleNiv :=styleCont, afficher()
];
\end{texgraph}
```

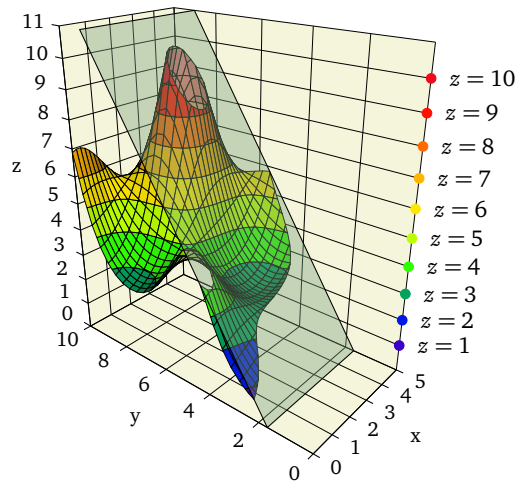


$$\frac{x+y}{2+\sin(y)\cos(x)}$$



Section plane

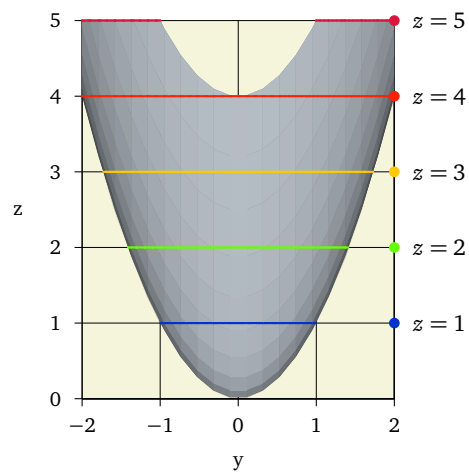
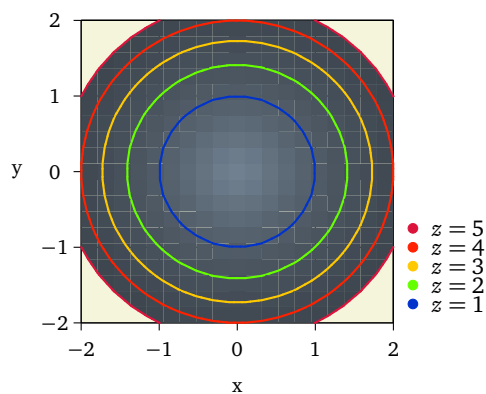
```
\begin{texgraph}[name=boite5, file]
Include "CourbesNiv.mod";
Graph image = [
Marges(0, 0, 0, 0), view(-10, 10, -11, 8), size(9), theta :=220*deg,
surface( "(x+y)/(2+sin(y)*cos(x))", [ grille :=[0, 5, 0, 10, 30, 30] ]),
fenetre( [x :=[0, 5], y :=[0, 10], z :=[0, 11], pas :=[1, 2, 1], orthonorme :=1] ),
niveaux( for k from 1 to 10 do k od ), legende("$\frac{x+y}{2+\sin(y)\cos(x)}$"),
sections( [ M(2, 5, 5), -M(0, -1.5, 1)] ), afficher(),
DrawFacet(getplan( [ M(2, 5, 5), -M(0, -1.5, 1)] ), [color :=darkseagreen, FillOpacity :=0.4])
];
\end{texgraph}
```



$$\frac{x+y}{2+\sin(y)\cos(x)}$$

Vues

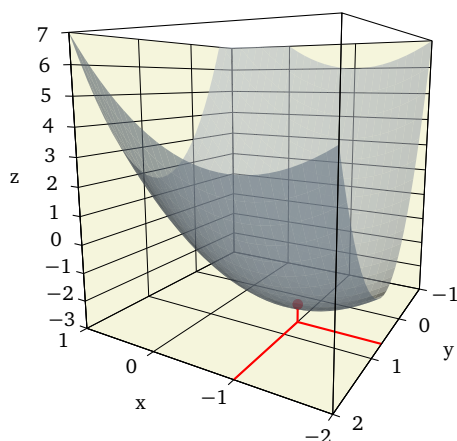
```
\begin{texgraph}[name=boite6, file]
Include "CourbesNiv.mod";
Graph image = [
view(-16, 16, -8.5, 7), Marges(0, 0, 0, 0), size(16+9*i, 1),
surface( "x^2+y^2", [grille :=[-2, 2, -2, 2, 20, 20] ]),
fenetre( [x :=[-2, 2], y :=[-2, 2], z :=[0, 5], pas :=[1, 1, 1], orthonorme :=1 ] ),
niveaux( for k from 1 to 5 do k od ),
SaveWin(), view(-16, 0, -8.5, 7), ChangeWinTo([-8-8.5*i, 7.5+7*i]),
vue(xOy), afficher([], [], [LineStyle :=noline]), RestoreWin(),
SaveWin(), view(0, 16, -8.5, 7), ChangeWinTo([-8-8.5*i, 7.5+7*i]),
vue(yOz), styleNiv :=styleLine, afficher([], [], [LineStyle :=noline]),
RestoreWin()
];
\end{texgraph}
```



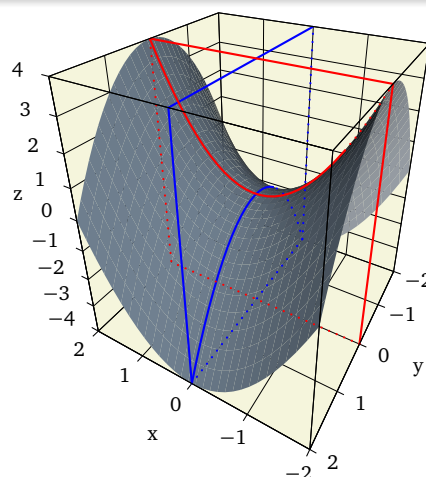


Plans en x et y

```
\begin{texgraph}[name=boite7, file]
Include "CourbesNiv.mod";
Graph image = [
view(-20, 20, -13, 7), Marges(0, 0, 0, 0), size(16+9*i, 1), theta :=120*deg,
SaveWin(), view(-20, 0, -12, 7), ChangeWinTo([-12-14*i, 12+7*i]), phi :=80*deg,
surface("x^2+3*y^2+2*x-4*y", [ grille :=[-1, 2, -1, 2, 30, 30] ]),
parametrage("[-u+i*v, f(-u,v)]"),
fenetre( [x :=[-2, 1], y :=[-1, 2], z :=[-3, 7], pas :=[1, 1, 1], orthonorme :=0] ),
tracerplans("", legende("Minimum en $M(-1,\frac{2}{3},-\frac{7}{3})$"),
afficher([],
[Color :=red, Width :=8, Point3D(M(-1, 2/3, f(-1, 2/3))),
Ligne3D([M(-1, 2, -3), M(-1, 2/3, -3), M(-2, 2/3, -3), jump,
M(-1, 2/3, -3), M(-1, 2/3, f(-1, 2/3))], 0)
],
[LineStyle :=noline, FillOpacity :=0.6]),
RestoreWin(),
SaveWin(), view(0, 20, -12, 7), ChangeWinTo([-12-14*i, 13+7*i]), phi :=60*deg,
surface("x^2-y^2", [ grille :=[-2, 2, -2, 2, 30, 30] ]),
fenetre( [x :=[-2, 2], y :=[-2, 2], z :=[-4, 4], pas :=[1, 1, 1], orthonorme :=0] ),
tracerplans("x,0,[Color :=blue,Width :=8], y,0,[Color :=red,Width :=8]"),
legende("Point col en $M(0,0,0)$", voirtraces :=1, voirboite :=1,
afficher([zlabelstyle :=right], [], [LineStyle :=noline]), RestoreWin()
]);
\end{texgraph}
```



Minimum en $M(-1, \frac{2}{3}, -\frac{7}{3})$



Point col en $M(0, 0, 0)$.