Les valeurs extrêmes…un début

© Pierre Lantagne
Enseignant retraité du Collège de Maisonneuve

La première version de ce document est parue en février 2006. Cette feuille `Maple` est une entrée en matière sur les notions de maximum local, de minimum local et de point-selle d'une fonction réelle de deux variables réelles. Cette entrée en matière passera d'abord par l'observation graphique de l'existence de points stationnaires d'une surface bien choisie. La recherche de ces points stationnaires sera faite à l'aide de la macro-commande `fsolve` où il sera nécessaire de lui indiquer des intervalles d'amorces. Finalement, le tracé en trois dimensions des "points extrêmes" sur la surface viendra confirmer la justesse des résultats. La lecture de cette feuille Maple est aussi une occasion pour l'élève d'être introduit aux macro-commandes `contourplot`, `contourplot3d` et `display` de l'extension `plots` et des macro-commandes `line` et `sphere` de l'extension `plottools`.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.2

Initialisation

```
> restart;
> with(plots,contourplot,contourplot3d,display,setoptions,setoptions3d)
  :
  setoptions(axesfont=[TIMES,ROMAN,8],labelfont=[TIMES,ROMAN,8]):
  setoptions3d(labels=[x,y,z],lightmodel=light3,axes=framed,axesfont=
  [TIMES,ROMAN,8],labelfont=[TIMES,ROMAN,8]):
> with(plottools,line,sphere):
```

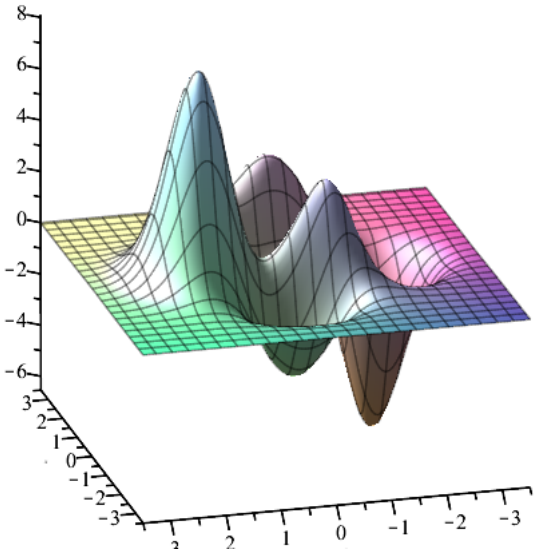
Maximum, minimum et point-selle

Soit la fonction f définie par $f(x,y) = 3\,(x-1)^2\,e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{e^{-(x+1)^2-y^2}}{3}$.

```
> f:=(x,y)->3*(x-1)^2*exp(-x^2-(y+1)^2)-10*(1/5*x-x^3-y^5)*exp(-x^2-
  y^2)-1/3*exp(-(x+1)^2-y^2);
```

$$f := (x,y) \mapsto 3\cdot(x-1)^2\cdot e^{-x^2-(y+1)^2} - 10\cdot\left(\frac{1}{5}\cdot x - x^3 - y^5\right)\cdot e^{-x^2-y^2} - \frac{e^{-(x+1)^2-y^2}}{3} \qquad \textbf{(2.1)}$$

```
> Surface:=plot3d([x,y,f(x,y)],x=-3.5..3.5,y=-3.5..3.5,grid=[90,120],
  thickness=0):
  display(Surface,orientation=[165,70]);
```



Grâce au graphique, nous identifions, trois maximum locaux, trois minimum locaux et trois point-selles. Pour mieux observer ces neuf points stationnaires, effectuer d'abord un agrandissement du graphique avec les carrés de modification. Ensuite, effectuez un zoom avant du tracé. Cliquez sur le graphique et maintenez le bouton gauche de la souris enfoncé pour modifier l'orientation du graphique dans l'espace.

Sachant que si la fonction f passe par une valeur extrême, nous devons avoir, si elles existent, les dérivées partielles nulles. Déterminons alors les couples (a, b) du domaine de la fonction f qui annuleront simultanément les dérivées partielles f_x et f_y.

```
> fx:=D[1](f)(x,y);
  fy:=D[2](f)(x,y);
```

$$fx := 6\,(x-1)\,e^{-x^2-(y+1)^2} - 6\,(x-1)^2\,x\,e^{-x^2-(y+1)^2} - 10\left(\frac{1}{5} - 3x^2\right)e^{-x^2-y^2} + 20\left(\frac{1}{5}x - x^3 - y^5\right)x\,e^{-x^2-y^2} - \frac{(-2-2x)\,e^{-(x+1)^2-y^2}}{3}$$

$$fy := 3\,(x-1)^2\,(-2-2y)\,e^{-x^2-(y+1)^2} + 50\,y^4\,e^{-x^2-y^2} + 20\left(\frac{1}{5}x - x^3 - y^5\right)y\,e^{-x^2-y^2} + \frac{2y\,e^{-(x+1)^2-y^2}}{3} \qquad \textbf{(2.2)}$$

```
> solve({fx=0,fy=0},{x,y});
Warning, solutions may have been lost
```

Maple n'a pu résoudre analytiquement ce système… nous le comprenons et nous lui pardonnons ☺. Résolvons alors numériquement avec la macro-commande `fsolve`.

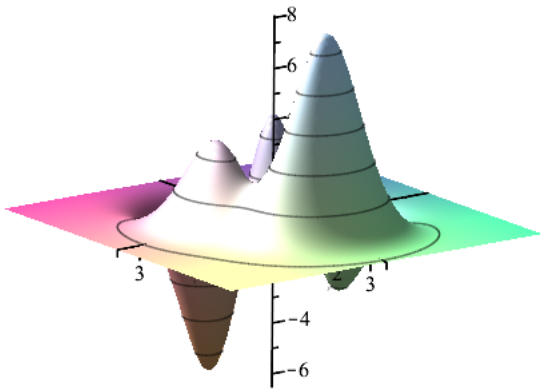
```
> fsolve({fx=0,fy=0},{x,y});
```

$$fsolve\left(\left\{3\,(x-1)^2\,(-2-2y)\,e^{-x^2-(y+1)^2} + 50\,y^4\,e^{-x^2-y^2} + 20\left(\frac{1}{5}x - x^3 - y^5\right)y\,e^{-x^2-y^2} + \frac{2y\,e^{-(x+1)^2-y^2}}{3} = 0, 6\,(x-1)\,e^{-x^2-(y+1)^2} - 6\,(x-1)^2\,x\,e^{-x^2-(y+1)^2} - 10\left(\frac{1}{5} - 3x^2\right)e^{-x^2-y^2} + 20\left(\frac{1}{5}x - x^3 - y^5\right)x\,e^{-x^2-y^2} - \frac{(-2-2x)\,e^{-(x+1)^2-y^2}}{3} = 0\right\}, \{x, y\}\right) \qquad \textbf{(2.3)}$$

Maple a encore été incapable de résouder ce système d'équations. Sachez que la macro-commande `fsolve` utilise une méthode d'approximations successives pour donner une approximation des racines d'une équation ou d'un système d'équations. L'effacité d'une telle méthode repose sur des valeurs amorçant le processus. Pour augmenter ses "chances" de trouver des solutions à notre système, il faut donc trouver des valeurs d'amorces. Un bon choix de valeurs d'amorce, serait de prendre des valeurs "proches" des racines (un tel choix ne garantit pas tout de même le succès de la méthode).

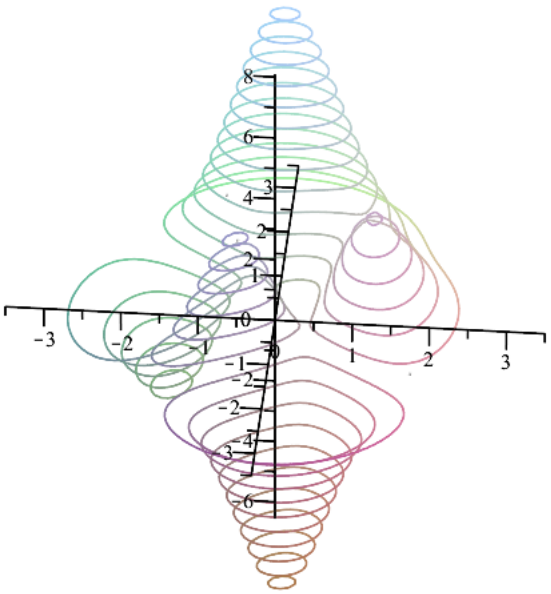
Avec le graphique précédent, il est plutôt difficile de lire des valeurs d'abscisse et d'ordonnée de ces points stationnaires. Tâchons de le faire avec les courbes de niveau.

```
> display(Surface,style=patchcontour,axes=normal);
```



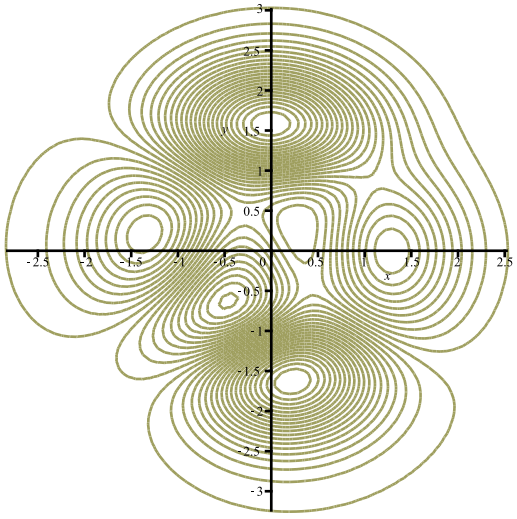
On pourrait y arriver mais, procédons comme suit. Utilisons la macro-commande `contourplot3d` pour obtenir un nombre voulu de courbes de niveau.

```
> Contours:=contourplot3d(f(x,y),x=-3.5..3.5,y=-3.5..3.5,contours=24,
  grid=[80,80],thickness=0):
  display(Contours,axes=normal,orientation=[-85,55]);
```



Traçons plutôt les courbes de niveau en 2D pour mieux localiser les valeurs des abscisses et des ordonnées des points extrêmes.

```
> Courbes_niveau:=contourplot(f(x,y),x=-3.5..3.5,y=-3.5..3.5,
  xtickmarks=12,ytickmarks=12,contours=42,grid=[90,90],thickness=0,
  color=khaki):
  Courbes_niveau;
```



En s'aidant des deux graphiques précédents (zoom avant utile), nous pouvons obtenir une première approximation des valeurs d'abscisse et d'ordonnée de chaque point stationnaire.

Point_maximum: $(1.3, 0)$, $(0, 1.6)$ et $(-0.3, -0.4)$;
Point_minimun: $(-1.4, 0.2)$, $(0.3, 0.3)$ et $(0.2, -1.6)$;
Point_selle: $(1.1, 0.8)$, $(-0.3, 0.4)$ et $(0.4, -0.4)$;

Obtenons ensuite de meilleurs approximations pour les points maximum avec la macro-commande `fsolve`.

]Par exemple, pour le point $(1.3, 0)$, précisons, dans la macro-commande fsolve, l'intervalle d'amorce $[-1.2, 1.4]$ pour x et l'intervalle $[-0.1, 0.1]$ pour y.

```
> fsolve({fx=0,fy=0},{x,y},{x=1.2..1.4,y=-0.1..0.1});
```

$$\{x = 1.2856846970, y = -0.0048475591\} \quad \textbf{(2.4)}$$

Bingo! Vérifions cette approximation en calculant f_x et f_y.

```
> assign((2.4));
  D[1](f)(x,y);
  D[1](f)(x,y);
  unassign('x,y');
```

$$4.2300000000\,10^{-10}$$
$$4.2300000000\,10^{-10} \quad \textbf{(2.5)}$$

C'est une bonne approximation. N'oublions pas que les calculs tiennent comptent de la variable d'environnement `Digits` et qu'elle est initialisé à 10 par défaut. Augmentons sa valeur à 40 chiffres et voyons si cela améliore la précision des approximations.

```
> Digits:=40;
  fsolve({fx=0,fy=0},{x,y},{x=1.2..1.4,y=-0.1..0.1});
```

$$Digits := 40$$

$$\{x = 1.2856846972, y = -0.0048475591\} \quad \textbf{(2.6)}$$

```
> assign((2.6));
  D[1](f)(x,y);
  D[1](f)(x,y);
  unassign('x,y');
```

$$-2.8600000000\,10^{-39}$$

$$-2.8600000000\,10^{-39} \quad \textbf{(2.7)}$$

Il est évident de constater que les valeurs de f_x et f_y sont davantages près de 0. Mais, pour nos besoin, 10 chiffres est satisfaisant.

```
> Digits:=10;
```

$$Digits := 10 \quad \textbf{(2.8)}$$

Obtenons de meilleurs approximation pour les deux autres points maximum.

```
> fsolve({fx=0,fy=0},{x,y},{x=-0.1..0.1,y=1.5..1.7});
  fsolve({fx=0,fy=0},{x,y},{x=-0.5..-0.3,y=-0.7..-0.5});
```

$$\{x = -0.0093175820, y = 1.5813679630\}$$

$$\{x = -0.4600245180, y = -0.6291965087\} \quad \textbf{(2.9)}$$

Bingo encore!

Créons immédiatement de petites sphères de couleur "navy" qui illustrrons à la fin ces points maximum.

```
> Pmax_1:= sphere([1.285684697,-.4847559076e-2,f(1.285684697,
  -.4847559076e-2)], 0.10,style=patchnogrid,color=navy):
  Pmax_2:= sphere([-.9317581960e-2,1.581367963,f(-.9317581960e-2,
  1.581367963)], 0.10,style=patchnogrid,color=navy):
  Pmax_3:= sphere([-.4600245180,-.6291965087,f(-.4600245180,
  -.6291965087)], 0.10,style=patchnogrid,color=navy):
```

Faisons un travail similaire pour les mimimum locaux: $(-1.4, 0.2)$, $(0.3, 0.3)$ et $(0.2, -1.6)$

```
> fsolve({fx=0,fy=0},{x,y},{x=-1.5..-1.3,y=0.1..0.3});
  fsolve({fx=0,fy=0},{x,y},{x=0.2..0.4,y=0.2..0.4});
  fsolve({fx=0,fy=0},{x,y},{x=0.1..0.3,y=-1.7..-1.5});
```

$$\{x = -1.3473962440, y = 0.2045188661\}$$

$$\{x = 0.2964455538, y = 0.3201962477\}$$

$$\{x = 0.2282789206, y = -1.6255349570\} \quad \textbf{(2.10)}$$

```
> Pmin_1:= sphere([-1.347396244,.2045188661,f(-1.347396244,.2045188661)
  ], 0.10,style=patchnogrid,color=magenta):
  Pmin_2:= sphere([.2964455538,.3201962477,f(.2964455538,.3201962477)],
  0.10,style=patchnogrid,color=magenta):
  Pmin_3:= sphere([.2282789206,-1.625534957,f(.2282789206,-1.625534957)
  ], 0.10,style=patchnogrid,color=magenta):
```

Faisons un travail similaire pour les points-selle: $(1.1, 0.8)$, $(-0.3, 0.4)$ et $(0.4, -0.4)$

```
> fsolve({fx=0,fy=0},{x,y},{x=1..1.2,y=0.7..0.9});
  fsolve({fx=0,fy=0},{x,y},{x=-0.4..-0.2,y=0.3..0.5});
  fsolve({fx=0,fy=0},{x,y},{x=0.3..0.5,y=-0.5..-0.3});
```

$$\{x = 1.0982728340, y = 0.8544609795\}$$

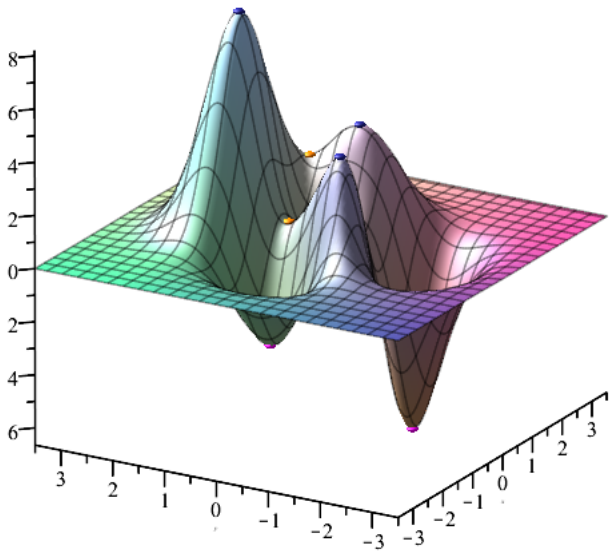
$$\{x = -0.2659375568, y = 0.4667399937\}$$

$$\{x = 0.4163233217, y = -0.3941450984\} \quad \textbf{(2.11)}$$

```
> Pselle_1:= sphere([1.098272834,.8544609795,f(1.098272834,.8544609795)
  ], 0.10,style=patchnogrid,color=coral):
  Pselle_2:= sphere([-.2659375568,.4667399937,f(-.2659375568,
  .4667399937)], 0.10,style=patchnogrid,color=coral):
  Pselle_3:= sphere([.4163233217,-.3941450984,f(.4163233217,
  -.3941450984)], 0.10,style=patchnogrid,color=coral):
> Points_stationaires:=Pmax_1,Pmax_2,Pmax_3,Pmin_1,Pmin_2,Pmin_3,
  Pselle_1,Pselle_2,Pselle_3:
```

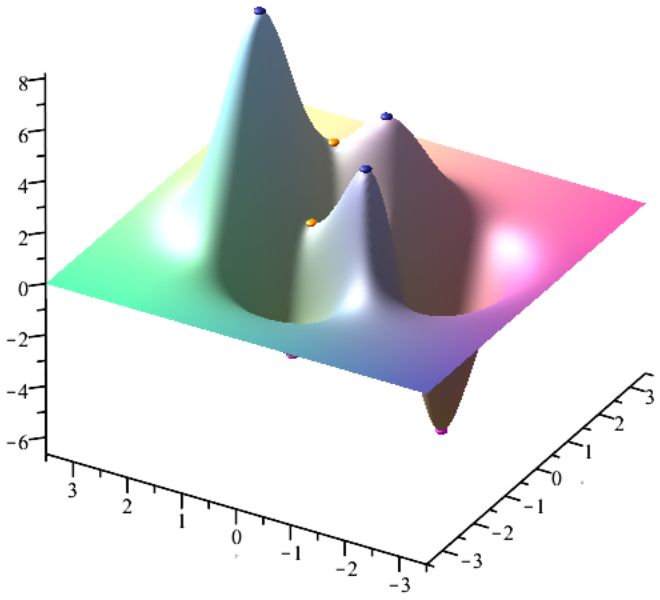
Superposons, dans un même graphique, la surface ainsi que les neuf points qui ont été créés. N'ayant pas spécifié l'option `scaling=constrained`, ces petites sphères ont plutôt la forme d'une pastille. Agrandissez le graphique sinon, on ne voit pas grand chose.

```
> display({Surface,Points_stationaires},axes=framed,orientation=[-150,
  70]);
```



Nous pouvons aussi spécifier un affichage avec le style patchnogrid.

```
> display({Surface,Points_stationaires},style=patchnogrid,orientation=
  [-150,60]);
```



Voici une belle construction graphique résumant toute cette situation. Les étapes de cette construction ne seront pas commentées…

```
> Segment_max1:=line([1.285684697,-.4847559076e-2,-8],
                     [1.285684697,-.4847559076e-2,f(1.285684697,
  -.4847559076e-2)],
                      linestyle=3,thickness=2,color=navy):
  Segment_max2:=line([-.9317581960e-2,1.581367963,-8],
                     [-.9317581960e-2,1.581367963,f(-.9317581960e-2,
  1.581367963)],
                      linestyle=3,thickness=2,color=navy):
  Segment_max3:=line([-.4600245180,-.6291965087,-8],
                     [-.4600245180,-.6291965087,f(-.4600245180,-.6291965087)
  ],
                      linestyle=3,thickness=2,color=navy):
> Segment_min1:=line([-1.347396244,.2045188661,-8],
                     [-1.347396244,.2045188661,f(-1.347396244,.2045188661)],
                     linestyle=3,thickness=2,color=magenta):
  Segment_min2:=line([.2964455538,.3201962477,-8],
                     [.2964455538,.3201962477,f(.2964455538,.3201962477)],
                     linestyle=3,thickness=2,color=magenta):
  Segment_min3:=line([.2282789206,-1.625534957,-8],
                     [.2282789206,-1.625534957,f(.2282789206,-1.625534957)],
                     linestyle=3,thickness=2,color=magenta):
> Segment_point_selle1:=line([1.098272834,.8544609795,-8],
                     [1.098272834,.8544609795,f(1.098272834,.8544609795)],
                     linestyle=3,thickness=2,color=coral):
  Segment_point_selle2:=line([-.2659375568,.4667399937,-8],
                     [-.2659375568,.4667399937,f(-.2659375568,.4667399937)],
                     linestyle=3,thickness=2,color=coral):
  Segment_point_selle3:=line([.4163233217,-.3941450984,-8],
```

```
                    [.4163233217,-.3941450984,f(.4163233217,-.3941450984)],
                    linestyle=3,thickness=2,color=coral):
> t := plottools[transform]((x,y) -> [x,y,-8]):

> display([Surface,Points_stationaires,t(Courbes_niveau),
           Segment_max||(1..3),
           Segment_min||(1..3),
           Segment_point_selle||(1..3)],
           orientation=[-170,65],lightmodel=light1,style=wireframe,
           axes=framed);
```