Projet Outil Numérique 2018

Estimation du problème Dirichlet par simulation probabiliste

Etudiant : VU Thi Hai Yen

1. Codes

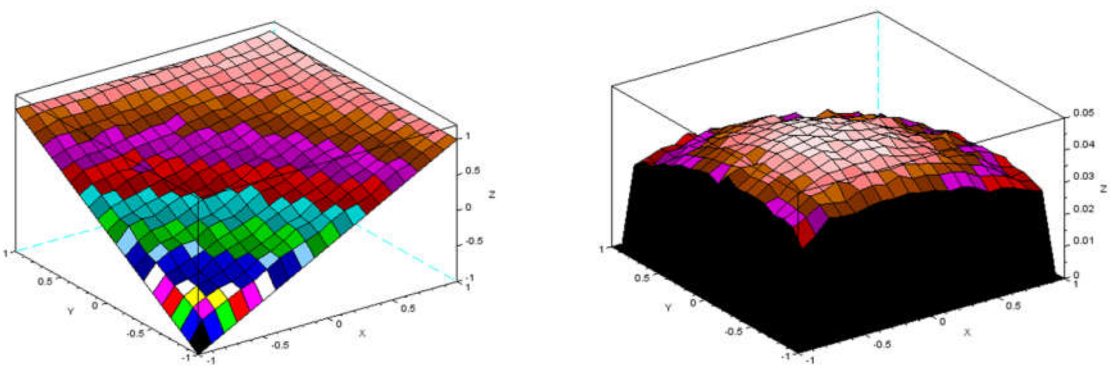
Les codes sont dans les fichiers *.sce*. On note juste que dans le cas du disque, on prend le maillage avec r et θ au lieu de x et y.

2. Quelques résultats graphiques

On va d'abord illustrer quelques résultats graphiques de notre code.

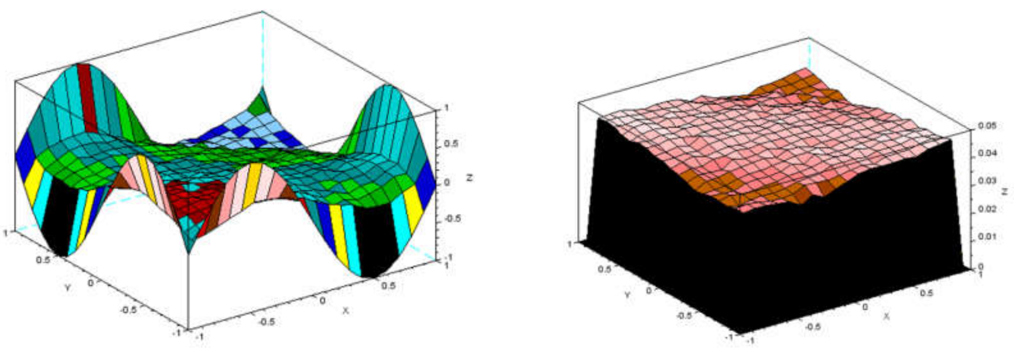
On prend M = 1000 et h = 0.1 dans la suite.

- Cas 1 : $D =]-1,1[^2$
  - g(x,y) = *max(x,y)*



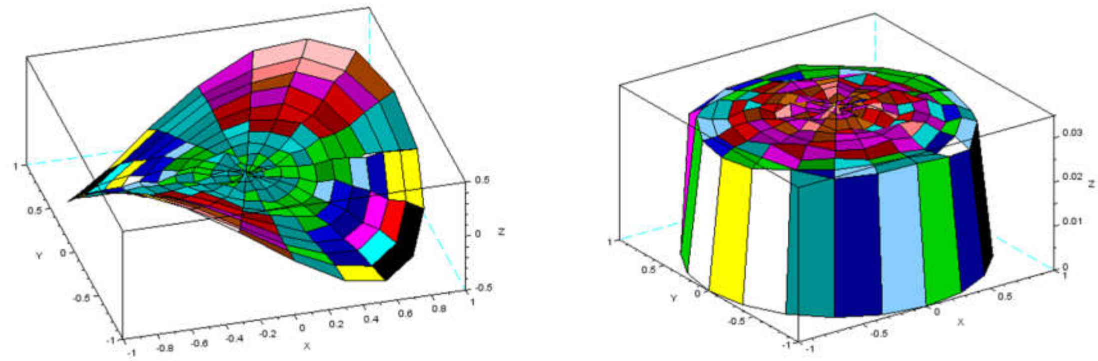
L'approximation de u (gauche) et le rayon de l'intervalle de confiance respectif, g(x,y) = max(x,y) .

  - g(x,y) = *sin(π(x+y))*



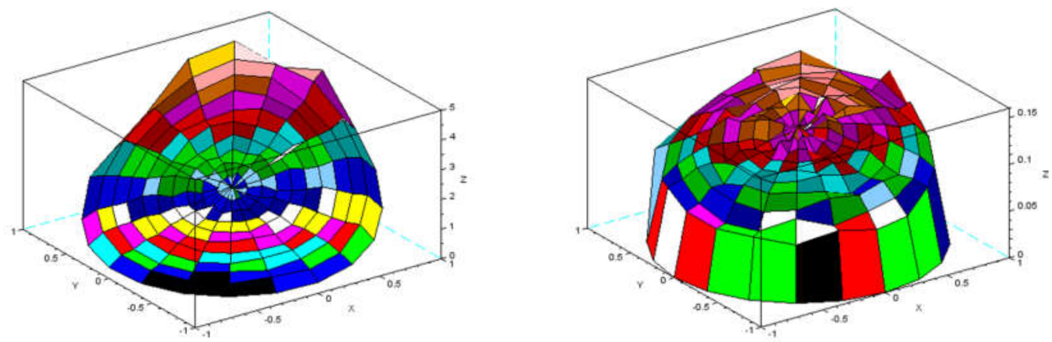
L'approximation de u (gauche) et le rayon de l'intervalle de confiance respectif, g(x,y) = sin(π(x+y)) .

- Cas 2 : $D = \{(x, y), x^2 + y^2 < 1\}$
  - g(x,y) = *xy*



L'approximation de u (gauche) et le rayon de l'intervalle de confiance respectif, g(x,y) = xy.

  - g(x,y) = *exp(x+y)*



L'approximation de u (gauche) et le rayon de l'intervalle de confiance respectif, g(x,y) = exp(x+y).

3. Pistes de réflexion

a. L'influence de h et M sur la précision de l'approximation de $u(x_0, y_0)$ et le temps de calcul

On considère le cas :

$$D = \{(x, y), x^2 + y^2 < 1\}, \ g(x,y) = xy,$$

où la solution exacte est donnée par *u(x,y) = xy*. On obtient les résultats

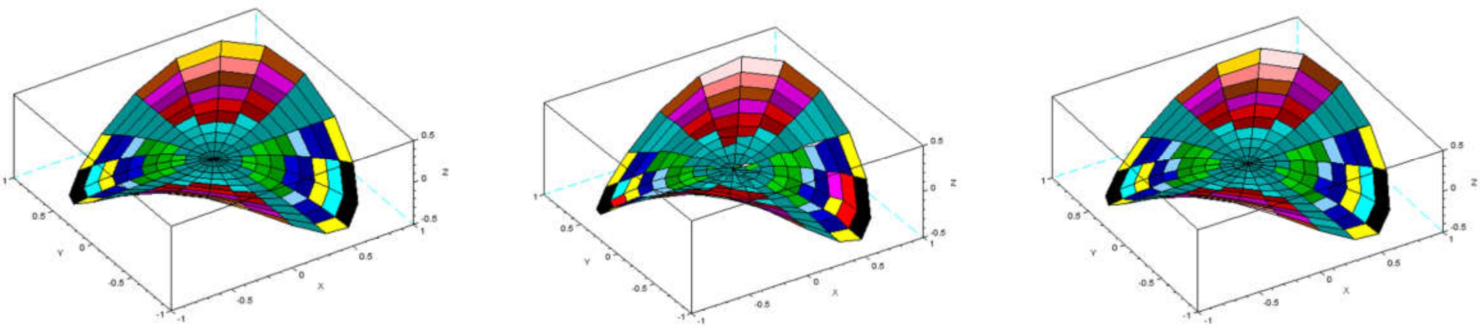
	M = 1000	M = 5000	M = 10000
h = 0.1	0.0112598	0.0054597	0.0037273
h = 0.05	0.0107121	0.0045426	0.0033901
h = 0.01	0.0076269	0.0039515	0.0027265

L'erreur (absolute) moyenne (sur D) de l'approximation de u avec M, h différents

	M = 1000	M = 5000	M = 10000
h = 0.1	Steps moyen : 5.4220736 (35s)	Steps moyen : 5.401903 (176s)	Steps moyens : 5.3975429 (366s)
h = 0.05	Steps moyen : 9.339684 (54s)	Steps moyen : 9.3470017 (290s)	Steps moyens : 9.3370719 (584s)
h = 0.01	Steps moyen : 38.290082 (228s)	Steps moyen : 38.319757 (1083s)	Steps moyens : 38.317207 (2026s)

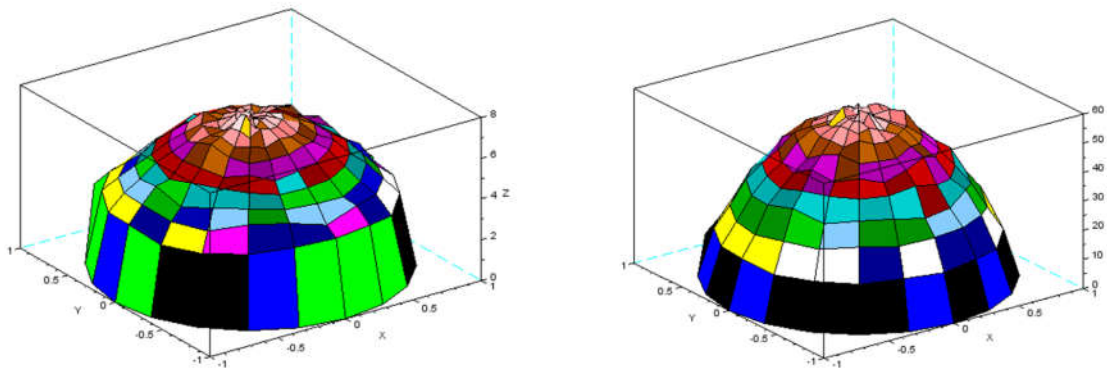
Nombre de steps moyen (sur D) pour sortie de D et les temps de calcul nécessaires

On voit que en diminuant *h*, l'erreur diminue mais pas rapide que quand on augmente *M*. On déduit que l'augmentation de M accélère la précision. Le temps de calcul augmente linéairement d'après *h* et *M*, mais le nombre de steps moyen pour sortie de D ne dépend que de *h*.



De gauche à droite :
La solution exacte, l'approximation avec M = 1000 et h = 0.1, l'approximation avec M = 10000 et h = 0.1

On voit que la fonction est plus lisse si on augmente M.



Le nombre de steps moyen (en chaque point de D) pour sortie de D

Gauche : avec M = 1000, h = 0.1 – Droite : avec M = 1000, h = 0.01

On voit dans la figure précédente que le nombre moyen de steps dépend de la distance de ces points aux bords de D et dépend (presque) linéairement de *h*.

b. L'influence de la position de point de départ (x_0, y_0) sur la précision

On considère le cas :

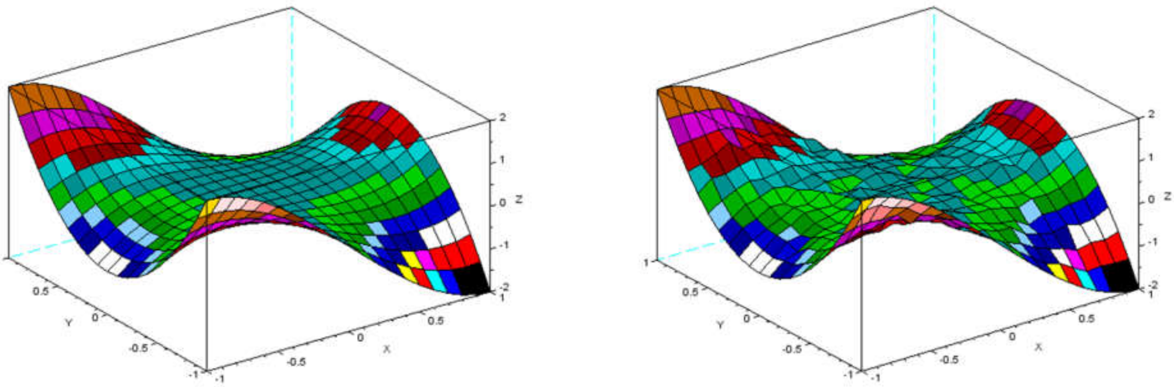
$$D =]-1,1[^2,\ g(x,y) = x^3 - 3xy^2,$$

où la solution exacte est donnée par $u(x,y) = x^3 - 3xy^2$.

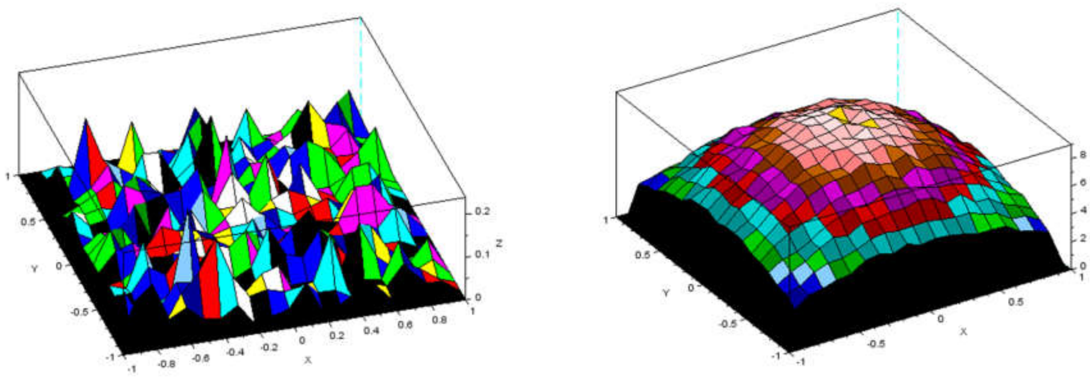
On prendra $M = 1000$, $h = 0.1$.

On obtient l'erreur (absolute) moyenne de l'approximation : 0.0365424

Comparer le résultat avec la solution exacte :



La solution exacte (gauche) et l'approximation de u



L'erreur (absolute) moyenne de l'approximation (gauche) et le nombre de steps moyen pour sortie de D

Comme on a déjà vu, le nombre de steps moyen pour sortie de D est plus haut dans le centre, on voit que l'erreur est un peu base dans le centre, mais celui-ci n'est pas si clair.

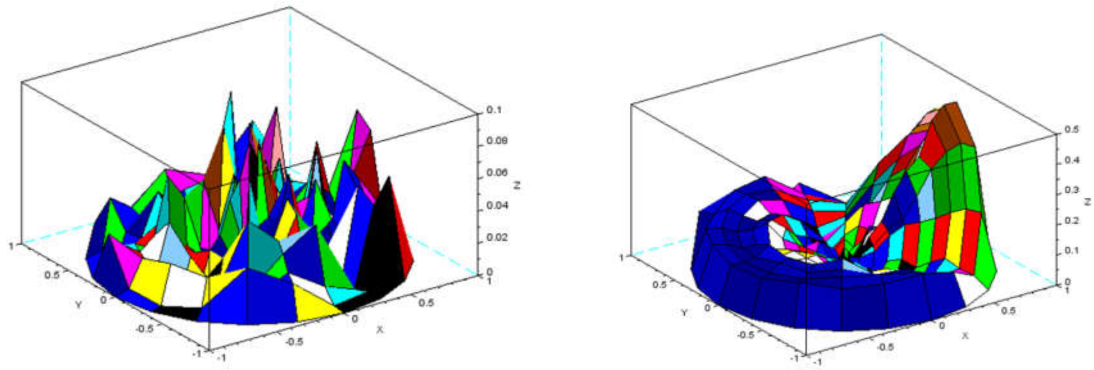
c. Le choix de point final

On verra la différence entre choisi *(X,Y)* comme le premier qui sort ou son prédécesseur. Dans cette partie, on considère le cas :

$$D = \{(x, y), x^2 + y^2 < 1\},\ g(x,y) = e^x\cos(y),$$

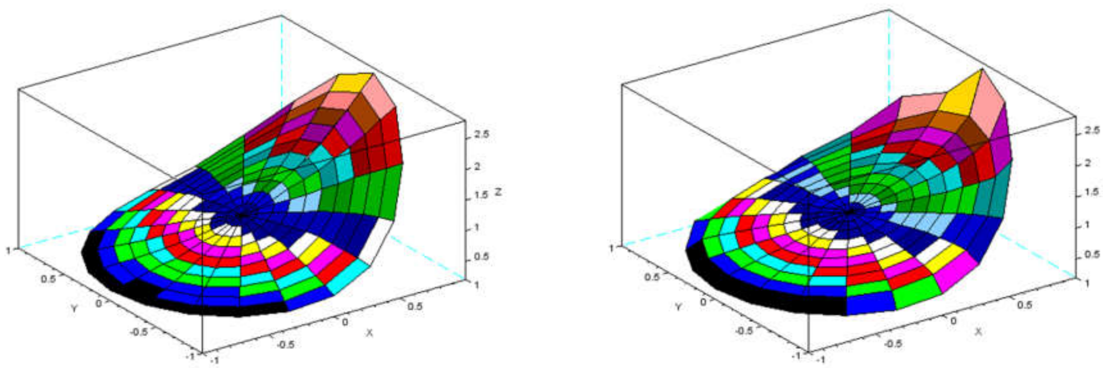
où la solution exacte est donnée par $u(x,y) = e^x\cos(y)$.

On prend $M = 1000$, $h = 0.1$



L'erreur par rapport à la solution exacte
Gauche : si on prend (X,Y) – Droite : si on prend (X_pred, Y_pred)

On voit bien que dans ce cas, le choix de *(X,Y)* est bien meilleur que *(X_pred, Y_pred)*. Plus précisément, l'erreur moyen pour *(X,Y)* est de 0.0216941, et l'autre est de 0.099683, beaucoup plus élevé.



Résultats si on prend le premier (X,Y) qui sort (gauche) ou son prédécesseur.

Si on opte pour le choix probabiliste :

1. *Calculer d = dist((X_pred, Y_pred), D^C)*
2. *Simuler U ~ Unif(0, 1)*
3. *Comparer U avec p = P(chi^2(2) > d^2/h) : si U > p, on choisit (X, Y) sinon (X_pred, Y_pred)*

On obtient dans ce cas l'erreur moyenne : 0.073227, qui est plus bas que le deuxième choix, mais plus haut que le premier choix.

d. Cas spéciaux

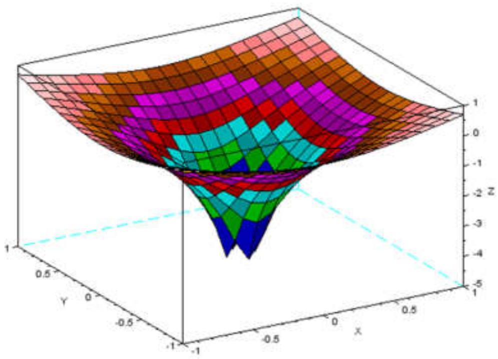
Dans cette partie, on va essayer d'exécuter notre code dans quelques cas extrêmes pour voir comment ça marchera.

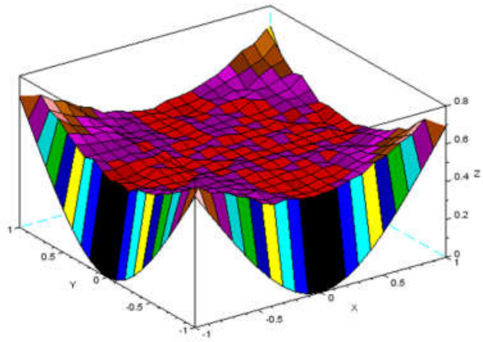
- *D = {(x, y), $x^2 + y^2 > 1$}* : le cas où D n'est pas borné

Dans ce cas, le code n'arrête jamais, car il y a une forte possibilité que les chemins n'atteignent pas le bord.

- g(x,y) = log($x^2 + y^2$) : le cas où la solution exact n'est pas borné

Dans ce cas, on obtient le mauvais résultat.





La solution exacte (gauche) et son approximation. Dans ce cas on voit que l'approximation n'est pas bonne.