Université Partis-Sud — **Année universitaire 2002-2003**
DEUG SMPE — S3 — **Stage Mathematica**

Séance n°4

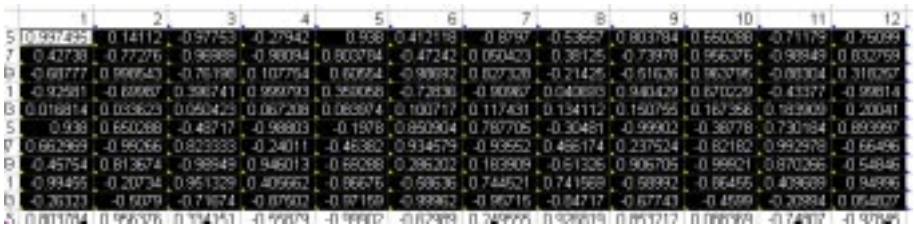
Jeudi 19 septembre 2002

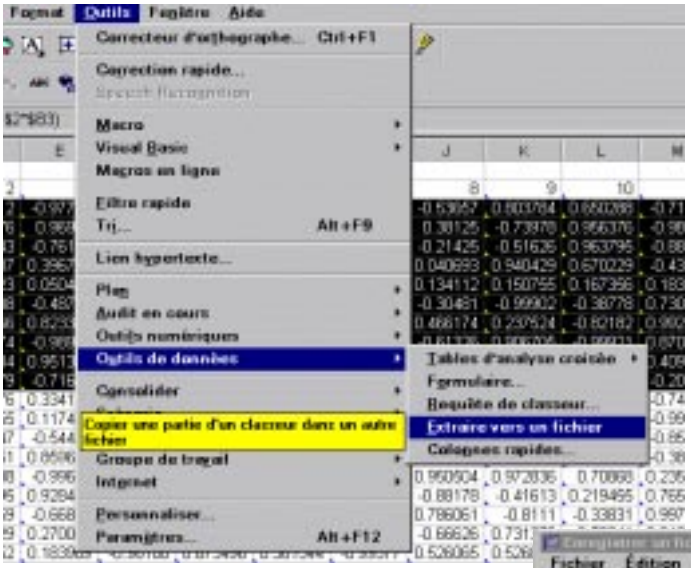
Importation et exportation de données, listes, graphiques, animations,

Résumé du cours

Importation et exportation de données

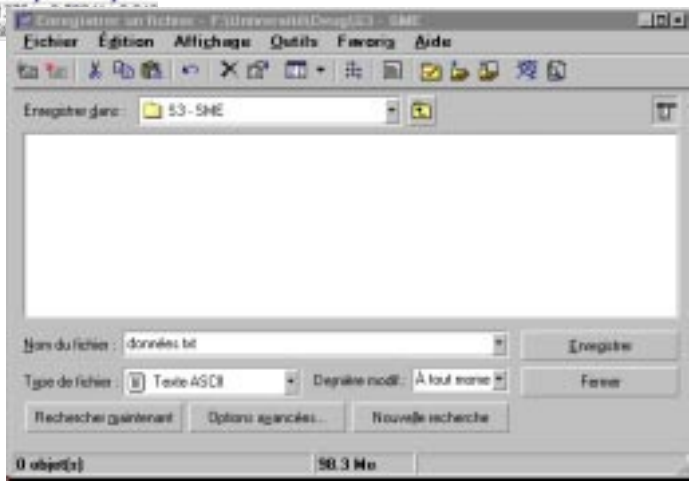
Une des utilisations de Mathematica est le traitement de données obtenues de sources diverses, et l'exportation des résultats, en particulier des graphiques. Typiquement, en SMPE, Mathematica pourra servir à traiter des données d'expérimentation qui arriveront sous des formats divers dépendant des appareils ayant servi à leur capture. Nous allons donc détailler ici un exemple d'importation à partir d'un tableur. Tout en restant semblable dans les grandes lignes, la procédure devra être adaptée pour chaque source de données. En TP Mathematica, on procédera à une simulation en Mathematica : après avoir produit une liste qui servira de test, on exportera cette liste qu'on réimportera ensuite pour la structurer différemment.





Voici la procédure à suivre pour exporter à partir d'un tableur un champ de données.

Dans un premier temps, sélectionner le champ que l'on veut exporter.



Cliquer ensuite sur le menu d'extraction vers un fichier, et bien spécifier que l'on exporte les valeurs et non les formules. Choisir Texte ASCII comme type de fichier, donner un nom au fichier et l'enregistrer.

Passons à l'importation en Mathematica. La première chose à faire et d'indiquer un répertoire de travail correct, dans lequel seront stockés les fichiers. On indique le répertoire par l'instruction SetDirectory. Noter les deux \\ après le nom du répertoire, et les guillemets entourant le nom du répertoire. Vérifier que l'instruction a bien été prise en compte en demandant "Directory[]". Noter que lorsqu'on travail sur un serveur distant, on n'a souvent pas le choix du répertoire de travail. Dans ce cas, cette étape peut être omise.

```
SetDirectory["F:\\"]

F:\

Directory[]

F:\
```

Trucs d'utilisateur

☞ Toujours importer en donnant un nom aux données importées ; cela facilite leur manipulation.

☞ Si le fichier source des données que l'on a importées risque de disparaître, ou si l'on désire travailler sur une machine n'ayant pas nécessairement accès au répertoire de stockage des données, rentrer physiquement les données dans le fichier Mathematica. Pour cela, importer les données sous un nom, par exemple a. Demander l'affichage de a. Copier le résultat de cette demande dans une cellule commençant par aa := . Travailler ensuite avec aa.

L'importation des données dans Mathematica va se faire par l'instruction "Import" sous Windows, "Get" sous Unix. Les illustrations du présent document viennent d'une version Windows de Mathematica. Pour une version Unix, il convient d'adapter *mutatis mutandis*. L'importation est réussie lorsque Mathematica identifie à l'intérieur d'une liste le contenu de chacune des cellules du fichier de données. Pour un fichier texte issu d'un tableur, il convient de préciser à Mathematica que l'on importe des données provenant d'un tableau. Pour comprendre ce qui se passe lorsqu'on ne précise pas, effectuons les deux importations.

```
a := Import["données.txt"]

a

0.997494986604055    0.141120008059867    -0.977530117665097    -
   0.279415498198926    0.937999976774739    0.412118485241757
   0.87969575997167    -0.536572918000435    0.803784426551621
   0.650287840157117    -0.711785342369123    -0.750987246771676
0.42737988023383    -0.772764487555987    0.969889810845086    -
   0.980936230066492    0.803784426551621    -0.472421986398469
   0.0504226878068148    0.38125049165494    -0.739778585077894
   0.956375928404503    -0.989487083254536    0.832759485307784
-0.687766159183974    0.998543345374605    -0.761983583919032
   0.107753652299442    0.605539869719601    -0.986915558120649
```

L'importation ci-contre n'est pas facilement utilisable, car le fichier n'a pas une structure de liste permettant la manipulation des données.

```
b := Import["données.txt", "Table"]

b

{{0.997495, 0.14112, -0.97753, -0.279415, 0.938, 0.412118,
   -0.879696, -0.536573, 0.803784, 0.650288, -0.711785, -0.750987},
  {0.42738, -0.772764, 0.96989, -0.980936, 0.803784, -0.472422,
   0.0504227, 0.38125, -0.739779, 0.956376, -0.989487, 0.832759},
  {-0.687766, 0.998543, -0.761984, 0.107754, 0.60554, -0.986916,
   0.827328, -0.214253, -0.516262, 0.963795, -0.883038, 0.318257},
 {-0.925815, 0.699875, 0.396741, 0.999793, 0.359058, 0.728361
```

Comme on peut le voir sur cette image partielle, le fichier b est structuré en liste.

Sous Unix, l'importation par l'"instruction Get d'un fichier de données peut conduire à un contenu parasite avec des chaînes de caractères excédentaires au début et à la fin du fichier. Un moyen de contourner cette difficulté est de demander l'affichage du contenu du fichier grâce à l'instruction "!!", qui ici donnerait par exemple !!données.txt. On peut ensuite tout simplement sélectionner la partie intéressante du fichier et la copier dans une cellule de la feuille de travail Mathematica.

Indexation et restructuration de listes

```
Length[b]
```

```
10
```

```
Length[b[[1]]]
```

```
12
```

Après avoir importé des données, il faut en déterminer ou en vérifier les dimensions. L'instruction Length donne la longueur d'une liste. Ici, on a importé au total 120 nombres. Supposons que la structuration du tableau en 10 lignes de 12 termes ne convienne pas, et qu'on veuille interpréter les données comme 6 listes de 10 termes formés de paires de nombres. Voici une situation où une telle structure pourrait se rencontrer : données provenant de 6 réalisations successives d'une expérience au cours de laquelle on a enregistré 10 mesures de la position d'un objet dans un plan (2 coordonnées pour un point). On pourrait créer directement une liste à trois niveaux correspondant à cette structure, en partitionnant d'abord la liste complète en 6 morceaux, puis en partitionnant chacun des 6 morceaux en 10. Il est plus simple de définir une fonction d'indexation. Une telle fonction est parfaitement adaptée aux calculs. Pour comprendre ce que fait une telle fonction, d, la valeur $d(n, p, q)$ sera la q-ième composante du p-ième point mesuré au cours de la n-ième expérience. On va donc définir d sur $\{1, \cdots, 6\} \times \{1, \cdots, 10\} \times \{1, 2\}$.

L'instruction Flatten[b] a enlevé la sous-structure de la liste b pour donner une liste "plate" de 120 éléments.

```
d := Flatten[b][[20 (#1 - 1) + 2 (#2 - 1) + #3 ]] &
```

Une fois qu'on dispose d'une fonction d'indexation d'une liste, il est généralement inutile de la restructurer. C'est bien sûr possible, et très simple, justement grâce à la fonction d'indexation.

```
bbb := Table[d[n, p, q], {n, 1, 6}, {p, 1, 10}, {q, 1, 2}]
```

```
bbb
```

```
{{{0.997495, 0.14112}, {-0.97753, -0.279415},
  {0.938, 0.412118}, {-0.879696, -0.536573}, {0.803784, 0.650288},
  {-0.711785, -0.750987}, {0.42738, -0.772764},
  {0.96989, -0.980936}, {0.803784, -0.472422}, {0.0504227, 0.38125}},
 {{-0.739779, 0.956376}, {-0.989487, 0.832759}, {-0.687766, 0.998543},
  {-0.761984, 0.107754}, {0.60554, -0.986916}, {0.827328, -0.214253},
  {-0.516262, 0.963795}, {-0.883038, 0.318257},
```

Sur cette vue partielle du début de l'affichage de la liste bbb, on en voit bien la structure.

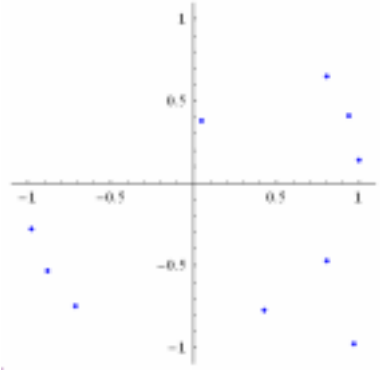
Graphiques, Animations

Grâce à notre fonction d'indexation, nous allons créer 6 graphiques représentant chacun les dix points du plan.

```
Table[
 ListPlot[Table[{d[k, q, 1], d[k, q, 1]}, {q, 1, 10}]],
 {k, 1, 6}]
```

Cette instruction nous donne bien les graphiques, mais pour pouvoir les comparer dans une animation, il faut les améliorer : Les points sont très petits, difficiles à discerner. Le partie du plan représentée change à chaque figure. L'échelle aussi. Les polices utilisées ne sont pas très belles. L'ajout d'options graphiques dans l'instruction va nous permettre de remédier à ces problèmes.

Ci-contre, une copie du premier des six graphiques.



```
Table[ListPlot[Table[{d[k, q, 1], d[k, q, 2]}, {q, 1, 10}],
  PlotStyle → {RGBColor[0, 0, 1], Thickness[0.01], PointSize[0.015]},
  TextStyle → {FontFamily → "Times", FontSize → 12},
  PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}}, AspectRatio → 1],
 {k, 1, 6}]
```

Pour une animation, il est important de contrôler le domaine de visualisation des graphiques. C'est l'option PlotRange qui effectue ce contrôle. L'option AspectRatio donne le rapport entre la hauteur et la largeur d'un graphique. Combinée à PlotRange, elle permet d'assurer l'orthonormalité du repère de visualisation, qui permet en particulier d'obtenir des cercles qui soient bien des cercles et non seulement des ellipses.

Une fois la liste de graphiques créée, on peut l'animer en sélectionnant la poignée de la liste des graphiques, puis en tapant Ctrl Y. Sur certains claviers Unix, c'est la combinaison ◊Y.

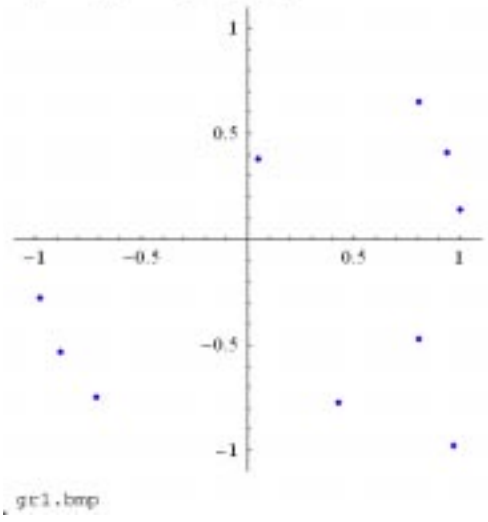
Exportation de données et de graphiques

```
gr[k_] := ListPlot[Table[{d[1, q, k], d[2, q, k]}, {q, 1, 10}],
  PlotStyle → {RGBColor[0, 0, 1], Thickness[0.01`], PointSize[0.015`]},
  TextStyle → {FontFamily → "Times", FontSize → 12},
  PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}}, AspectRatio → 1]
```

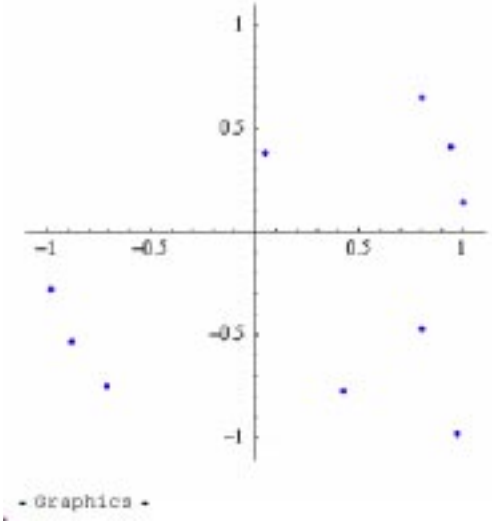
Mathematica exporte les données grâce à l'instruction "Export" sous Windows, "Put" sous Unix..

Commençons par exporter les six graphiques que nous venons de créer. Pour cela, on commence par définir une fonction donnant chacun d'eux. Ensuite, l'instruction Export permet le créer le fichier voulu. On a un grand choix de formats graphiques possibles pour une exportation.

```
Export["gr1.bmp", gr[1]]
```




```
ggr1 := Import["gr1.bmp"]
Show[ggr1]
```



Une fois le graphique exporté, penser à vérifier le résultat en le ré-important. (ci-contre, à gauche)

Si l'on désire créer un film, on va avoir besoin de créer un grand nombre de fichiers graphiques (plusieurs centaines). Pour que le logiciel de création de films puisse incorporer facilement les fichiers, il faut que Mathematica les numérote dans l'ordre.

Truc d'utilisateur

☞ Pour choisir les images dans un film, elle apparaissent classées par ordre alphabétique. Si l'on a plus d'une centaine d'image à récupérer, les numéroter en partant de 100. En effet, si l'on part de 1, l'ordre des fichiers est 1, 10, 11, ...19, 2, 20, etc.

Pour cette exportation "en chaîne", on utilise Do, qui a une structure de la forme Do[xx, {itérateur}]. Pour donner le nom de fichier, on regroupe dans un même mot trois morceaux, "gr", "numéro" qui donne le numéro du graphique (en partant de 100), et ".bmp", l'extension. C'est StringJoin qui effectue ce regroupement à partir des morceaux. Les morceaux littéraux sont créés avec ToString.

```
Do[
 Export[StringJoin[ToString["gr"], ToString[k+100], ToString[".bmp"]],
  gr[k]], {k, 1, 6}
]
```

Pour exporter des données, c'est encore l'instruction Export (Put sous Unix) que l'on utilise. Cette fois-ci, on choisit un nom de fichier adapté au contenu des données. L'extension .dat convient pour stocker une matrice. Par exemple, un triangle de Pascal.

```
pascal[n_] := Table[Binomial[p, q], {p, 0, n}, {q, 0, n}]

Export["pascal5.dat", pascal[5]]

pascal5.dat

Import["pascal5.dat"]

{{1, 0, 0, 0, 0, 0}, {1, 1, 0, 0, 0, 0}, {1, 2, 1, 0, 0, 0},
 {1, 3, 3, 1, 0, 0}, {1, 4, 6, 4, 1, 0}, {1, 5, 10, 10, 5, 1}}
```

Exercices

1. CRÉATION D'UNE ANIMATION REPRÉSENTANT UNE ONDE SE PROPAGEANT — Soit f la fonction définie par $f(x, y, k) := \sin(x - k)\cos(y)$. Représenter $(x, y) \mapsto f(x, y, 0)$ sur le carré $[-4\pi, 4\pi]^2$. Etudier l'effet de l'option "PlotPoints -> xx" sur le résultat.

Créer une fonction $k \mapsto \mathrm{gr}_1(k)$ créant une représentation de $(x, y) \mapsto f(x, y, k)$ sur le carré précédent. Créer une animation à l'aide d'une liste de $\mathrm{gr}_1(k)$. Comment choisir le pas d'incrémentation pour un mouvement fluide ? Comment choisir le domaine de k pour que l'on ne voit pas le "raccord" lorsque l'animation repart au début ?

2. ETUDE D'UNE SUITE DE FONCTIONS — Etudier graphiquement la convergence de la suite de fonctions $u_n : [-\pi, \pi] \to \mathbb{R}$, $x \mapsto \sin nx \exp(-nx^2)$. On réalisera une animation montrant à chaque image une fonction u_n, et une série de figures montrant toutes les u_k, pour k variant de 0 à n. Laquelle de ces deux méthodes permet de mieux comprendre le comportement de la suite de fonctions ?

Ind. On pensera à garder une échelle et un domaine constant.

3. Représenter la suite des premières sommes partielles des séries de termes généraux $a_n = (1/n) + (-1)^n/n^{1/2}$, $a_n = (1/n) + \ln((n-1)/n)$ et $a_n = (\sin n)/n^2$. Ces dessins permettent-ils de conjecturer la convergence ou la divergence des séries ?

4. Créer une matrice M_n de taille n+1 de terme général $\sin\left(\binom{k}{p}\right)$, $0 \le k \le n$, $0 \le p \le n$. Exporter M_{19} dans un fichier. Réimporter ce fichier, puis en restructurer les données de différentes manières indiquées ci-dessous. Pour chacune, créer aussi une fonction de réindexation.

1. Matrice 40 × 10, matrice 10 × 40. **2.** Liste de 5 sous-listes de 8 sous-listes de 10 éléments.

3. Liste de 4 sous-listes de 5 sous-sous-listes de 10 sous-sous-sous-listes de paires d'éléments.

Université Paris-Sud

Deug SMPE - S3 - Année 2002-2003

Stage Mathematica

Quatrième séance - Importation et exportation de données, paramétrage des graphiques, animations

Cours

Importation d'un fichier de données provenant d'un tableur

```
SetDirectory["F:\\"]
```

```
F:\
```

```
Directory[]
```

```
F:\
```

```
a := Import["données.txt" ]
```

a

```
0.997494986604055    0.141120008059867    -0.977530117665097    -
  0.279415498198926    0.937999976774739    0.412118485241757    -
  0.87969575997167    -0.536572918000435    0.803784426551621
  0.650287840157117    -0.711785342369123    -0.750987246771676
0.42737988023383    -0.772764487555987    0.969889810845086    -
  0.980936230066492    0.803784426551621    -0.472421986398469
  0.0504226878068148    0.38125049165494    -0.739778585077894
  0.956375928404503    -0.989487083254536    0.832759485307784
-0.687766159183974    0.998543345374605    -0.761983583919032
  0.107753652299442    0.605539869719601    -0.986915558120649
  0.827327900595377    -0.214252540295884    -0.516262220079927
  0.963795386284088    -0.883038191005416    0.318256511102482
-0.925814682327732    -0.699874687593544    0.396740573130612
  0.999792900142669    0.359058354022172    -0.728360767831594    -
  0.909666671833528    0.0406932573498578    0.940428962947004
  0.67022917584338    -0.433765540975678    -0.998136615932704
0.0168139004843506    0.0336230472211385    0.0504226878068148
  0.0672080725254785    0.0839744556917504    0.100717096992508
  0.117431262827103    0.134112227645664    0.150755275285169
  0.167355700302814    0.183908809306357    0.200409922281082
0.937999976774739    0.650287840157116    -0.487174512460513    -
  0.988031624092861    -0.197798799636455    0.850903524534122
  0.787705226984114    -0.304810621102224    -0.999021480034636    -
  0.387781635409417    0.730183859153177    0.893996663600552
0.662969230082182    -0.992659380470633    0.823333000738082    -
  0.240111597953781    -0.463815515983821    0.934579458388415    -
  0.935524037149618    0.466174406187185    0.237524226238826    -
  0.821817836630815    0.992978107132719    -0.664961621482308
-0.457535893775321    0.813673737507105    -0.989487083254536
  0.946012582626908    -0.692884954233696    0.286201759556747
  0.183908809306343    -0.613261931806889    0.906705358702901    -
  0.999206834186354    0.870266394499704    -0.54845952264626
-0.994552588203989    -0.207336420606759    0.951328738786783
  0.405661876555336    -0.866759574260759    -0.586357025092822
  0.744520521019825    0.741568747154996    -0.589924161317414    -
  0.864551448610608    0.409689345269107    0.949960227810725
-0.263231791365803    -0.507896590390625    -0.716737023160658    -
  0.87502257898945    -0.971588623516111    -0.999623893557639    -
  0.957150927951623    -0.847165547313093    -0.677425544533781    -
  0.459903490689579    -0.209942252562513    0.0548272128502745
```

```
b := Import["données.txt", "Table" ]
```

```
b
```

```
{{0.997495, 0.14112, -0.97753, -0.279415, 0.938, 0.412118,
  -0.879696, -0.536573, 0.803784, 0.650288, -0.711785, -0.750987},
 {0.42738, -0.772764, 0.96989, -0.980936, 0.803784, -0.472422,
  0.0504227, 0.38125, -0.739779, 0.956376, -0.989487, 0.832759},
 {-0.687766, 0.998543, -0.761984, 0.107754, 0.60554, -0.986916,
  0.827328, -0.214253, -0.516262, 0.963795, -0.883038, 0.318257},
 {-0.925815, -0.699875, 0.396741, 0.999793, 0.359058, -0.728361,
  -0.909667, 0.0406933, 0.940429, 0.670229, -0.433766, -0.998137},
 {0.0168139, 0.033623, 0.0504227, 0.0672081, 0.0839745, 0.100717,
  0.117431, 0.134112, 0.150755, 0.167356, 0.183909, 0.20041},
 {0.938, 0.650288, -0.487175, -0.988032, -0.197799, 0.850904,
  0.787705, -0.304811, -0.999021, -0.387782, 0.730184, 0.893997},
 {0.662969, -0.992659, 0.823333, -0.240112, -0.463816, 0.934579,
  -0.935524, 0.466174, 0.237524, -0.821818, 0.992978, -0.664962},
 {-0.457536, 0.813674, -0.989487, 0.946013, -0.692885, 0.286202,
  0.183909, -0.613262, 0.906705, -0.999207, 0.870266, -0.54846},
 {-0.994553, -0.207336, 0.951329, 0.405662, -0.86676, -0.586357,
  0.744521, 0.741569, -0.589924, -0.864551, 0.409689, 0.94996},
 {-0.263232, -0.507897, -0.716737, -0.875023, -0.971589, -0.999624,
  -0.957151, -0.847166, -0.677426, -0.459903, -0.209942, 0.0548272}}
```

Analyse et structuration des données importées

```
Length[b]
```

```
10
```

```
Length[b[[1]]]
```

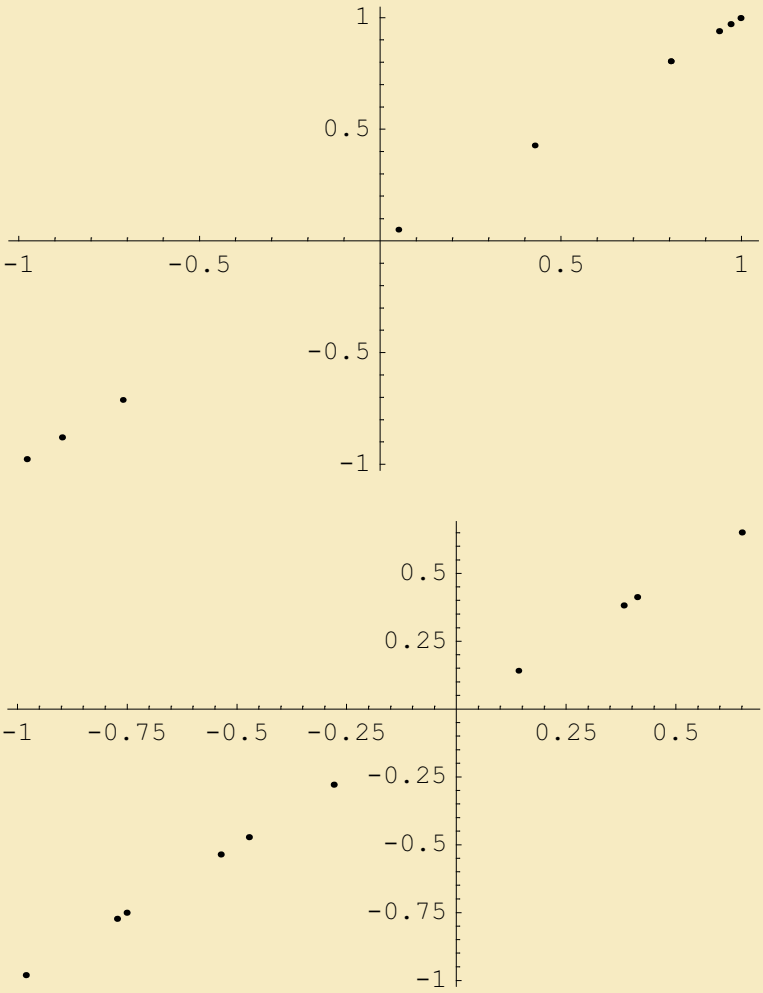
```
12
```

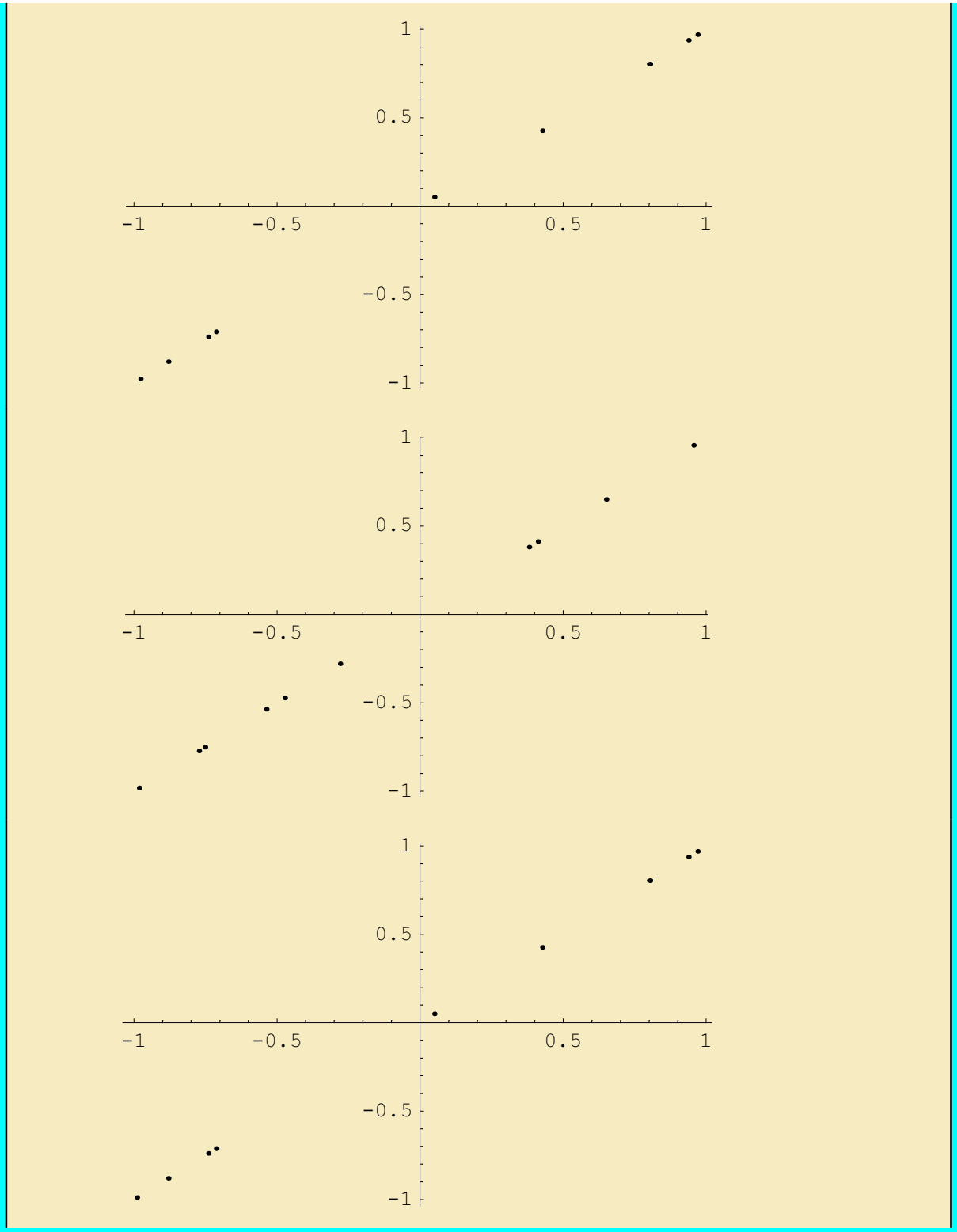
Indexation des 120 termes en 6 sous-listes de longueur 20 (1ère variable de 1 à 6), contenant chacune 10 sous-sous-listes (deuxième variable de 1 à 10) de deux éléments (troisième variable de 1 à 2).

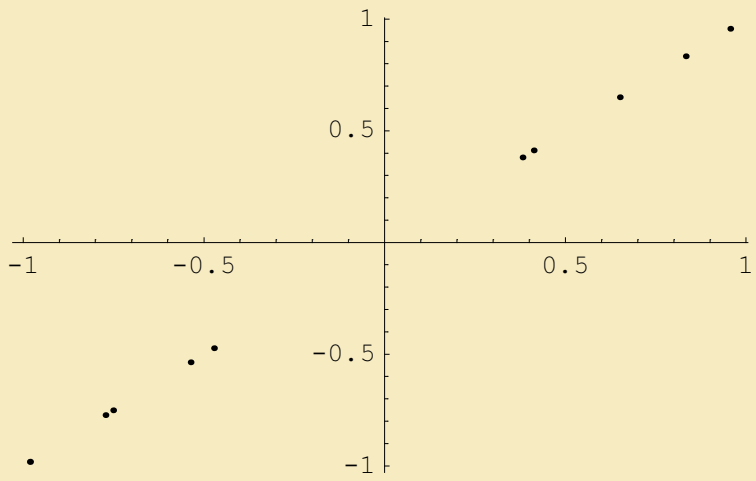
```
d := Flatten[b][[20 (#1 - 1) + 2 (#2 - 1) + #3 ]] &
```

Création d'une animation

```
Table[ListPlot[Table[{d[k, q, 1], d[k, q, 1]},
   {q, 1, 10}]], {k, 1, 6}]
```

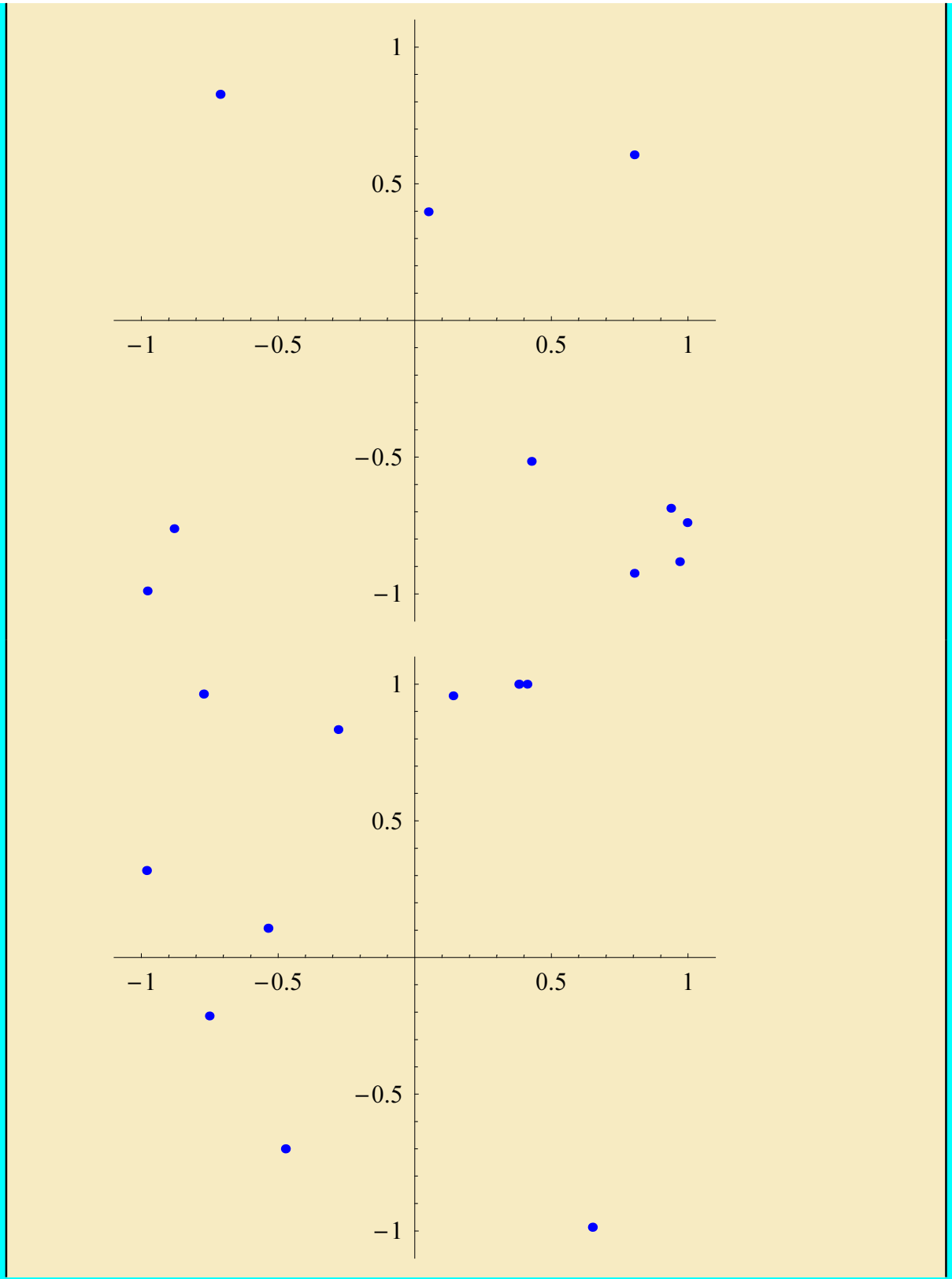


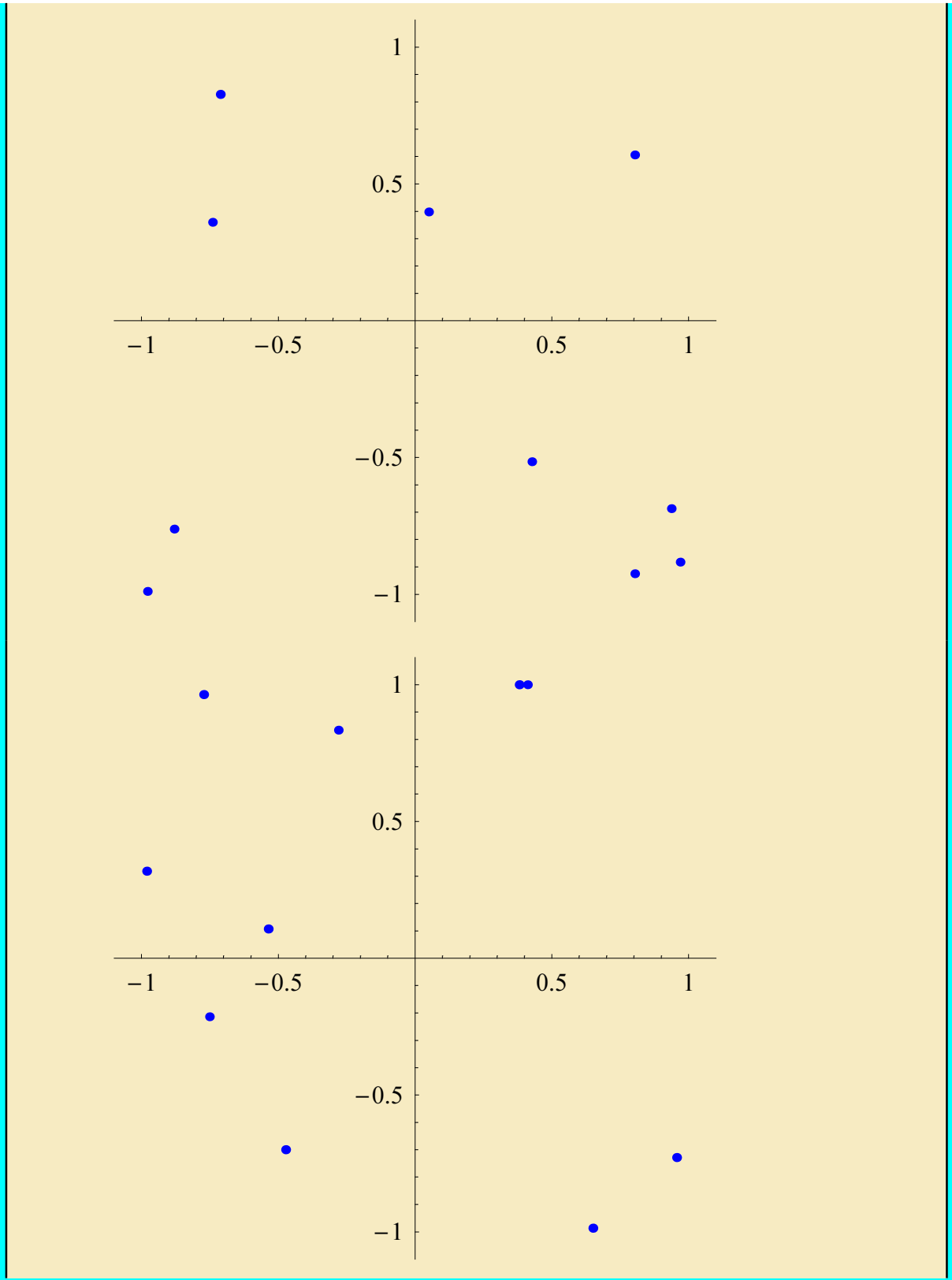

1
0.5
-1
-0.5
0.5
1
-0.5
-1
1
0.5
-1
-0.5
0.5
1
-0.5
-1
1
0.5
-1
-0.5
0.5
1
-0.5
-1

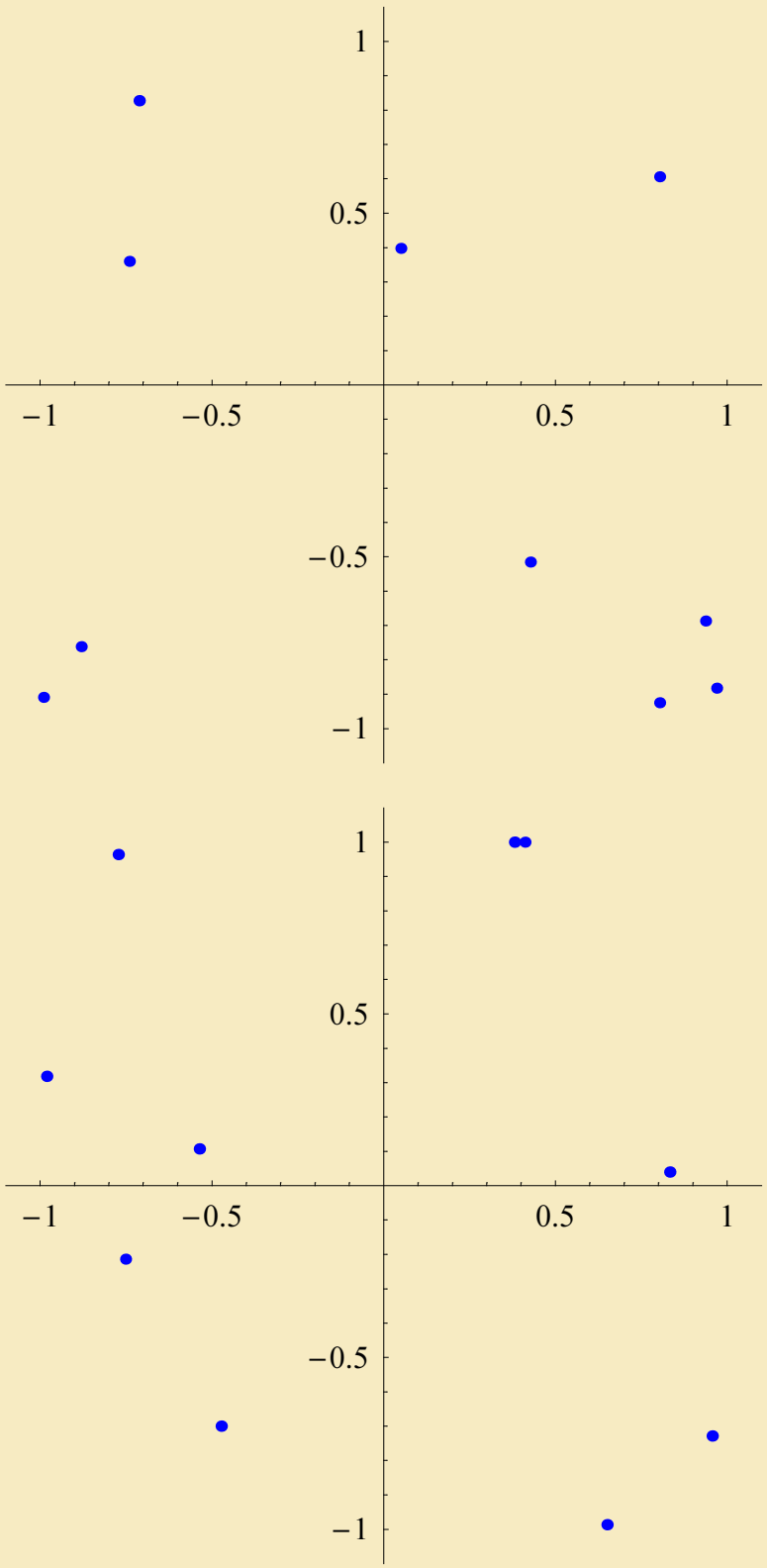



```
{- Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -}
```

```
Table[ListPlot[
  Table[{d[k, q, 1], d[k, q, 2]}, {q, 1, 10}],
  PlotStyle → {RGBColor[0, 0, 1],
    Thickness[0.01], PointSize[0.015]},
  TextStyle → {FontFamily → "Times", FontSize → 12},
  PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}},
  AspectRatio → 1], {k, 1, 6}]
```


1
0.5
−1
−0.5
0.5
1
−0.5
−1
1
0.5
−1
−0.5
0.5
1
−0.5
−1


1
0.5
−1
−0.5
0.5
1
−0.5
−1
1
0.5
−1
−0.5
0.5
1
−0.5
−1

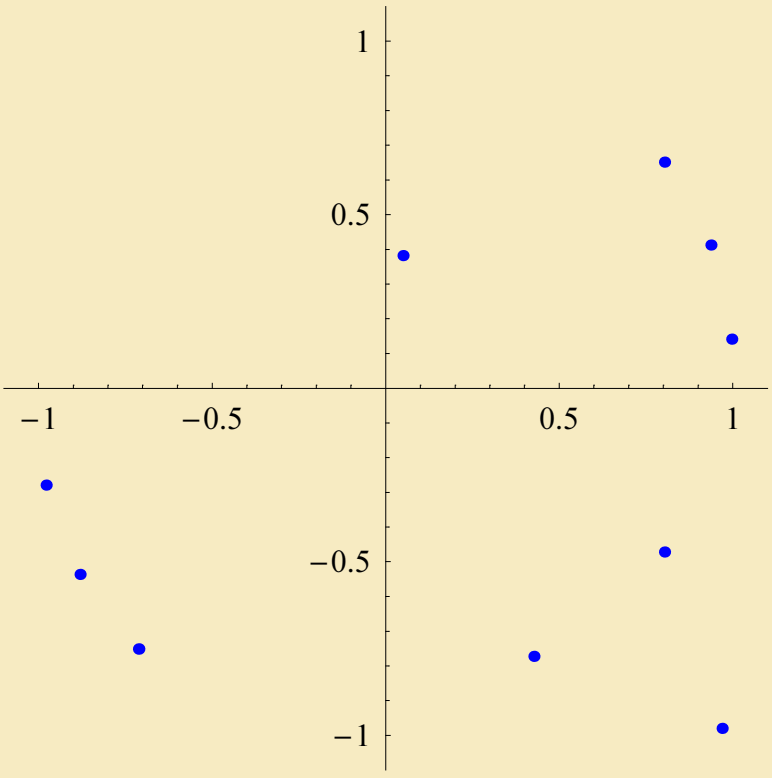



```
{- Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -}
```

Exportation de données

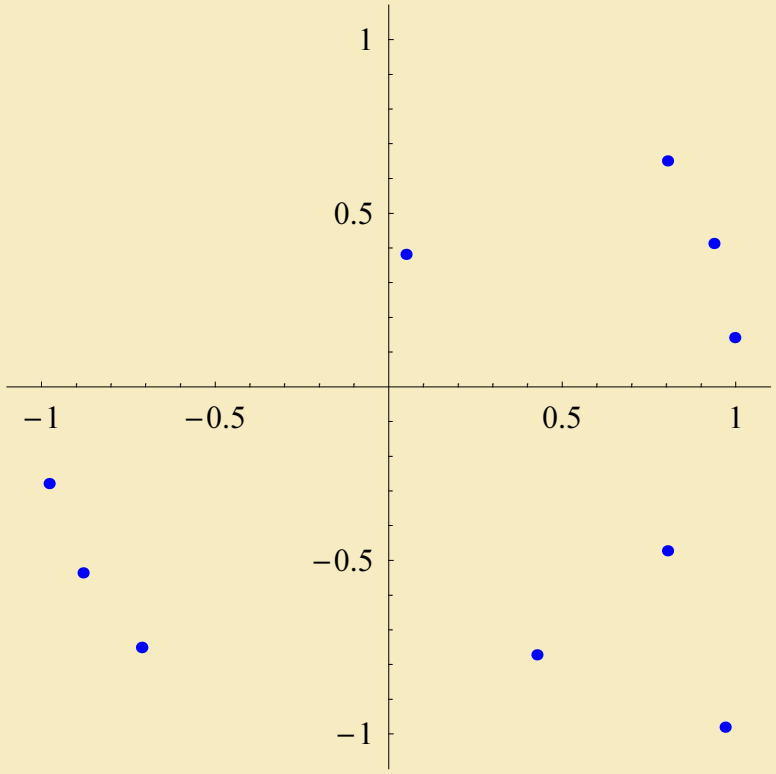
```
gr[k_] := ListPlot[
  Table[{d[1, q, k], d[2, q, k]}, {q, 1, 10}],
  PlotStyle → {RGBColor[0, 0, 1],
    Thickness[0.01`], PointSize[0.015`]},
  TextStyle → {FontFamily → "Times", FontSize → 12},
  PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}},
  AspectRatio → 1]
```

```
gr[1]
```



- Graphics -

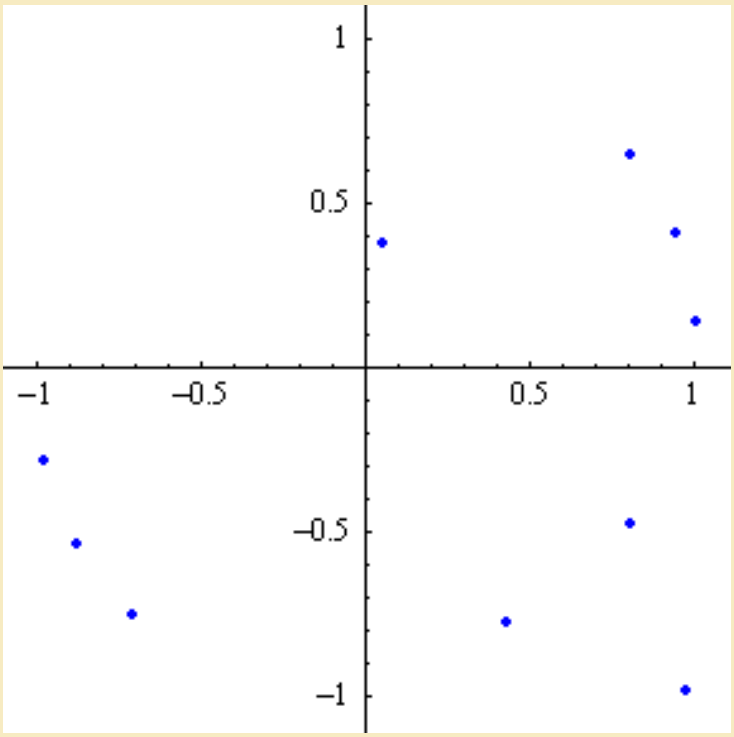
```
Export["gr1.bmp", gr[1]]
```



gr1.bmp

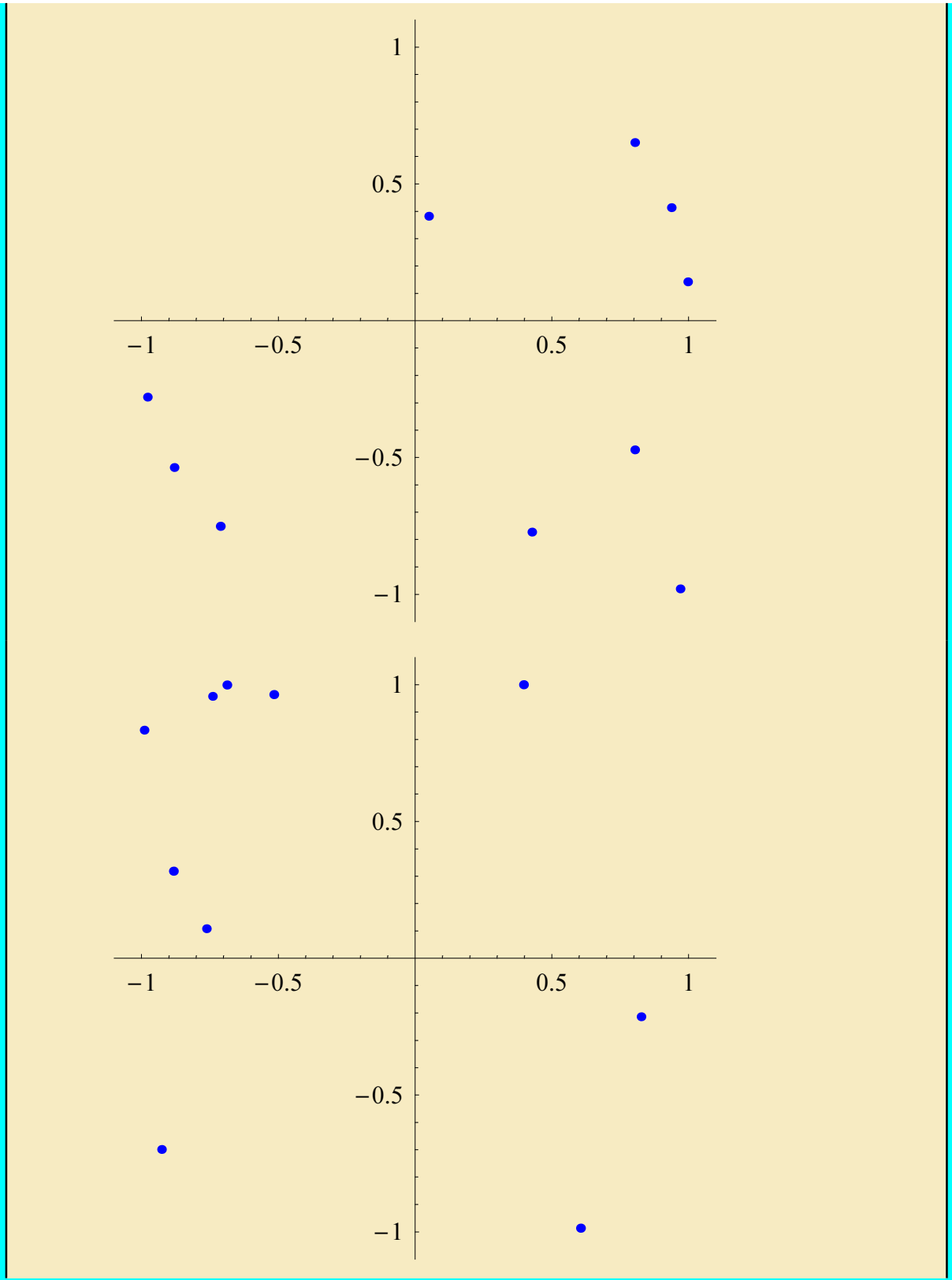
```
ggr1 := Import["gr1.bmp"]
```

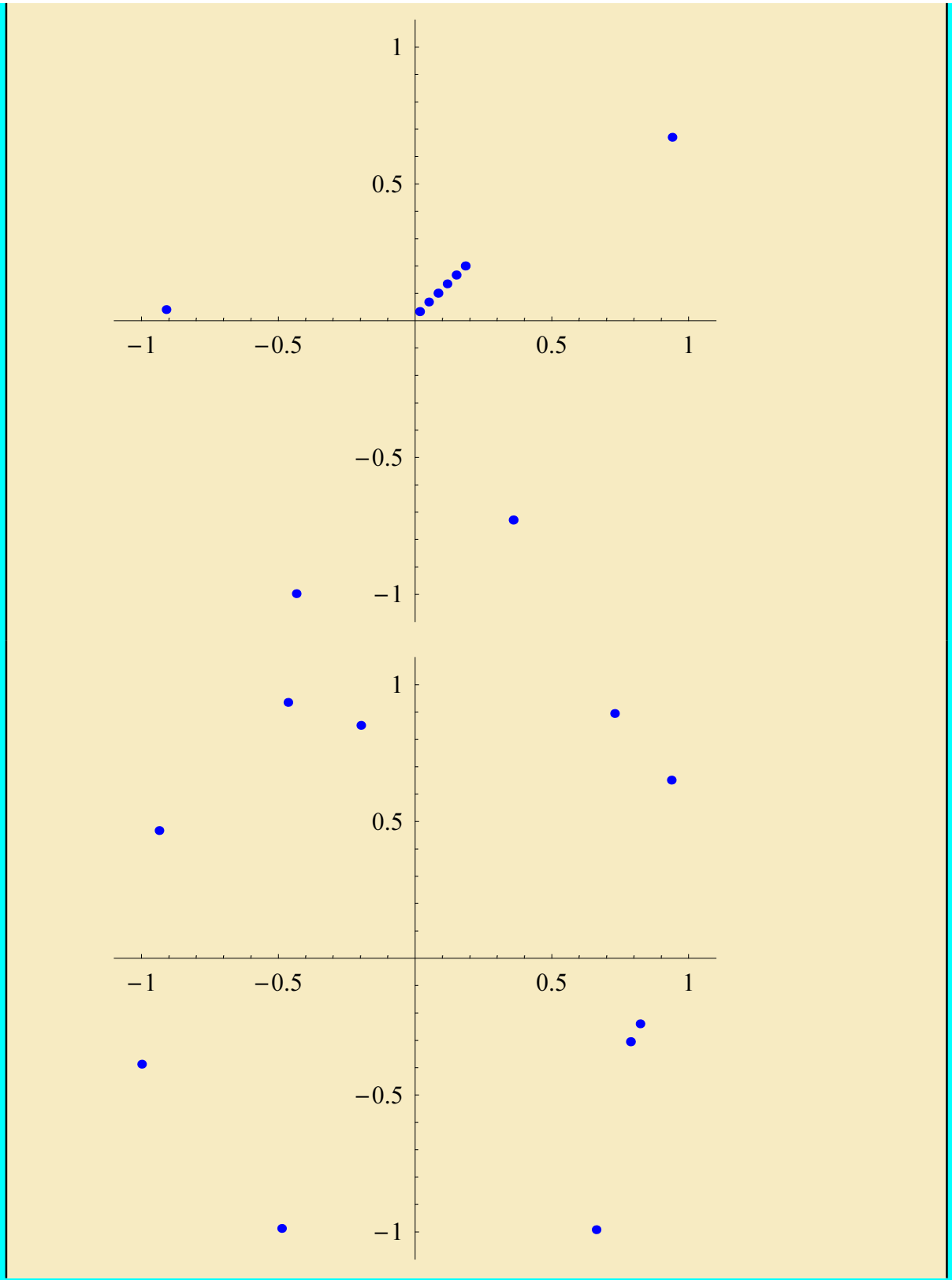
```
Show[ggr1]
```

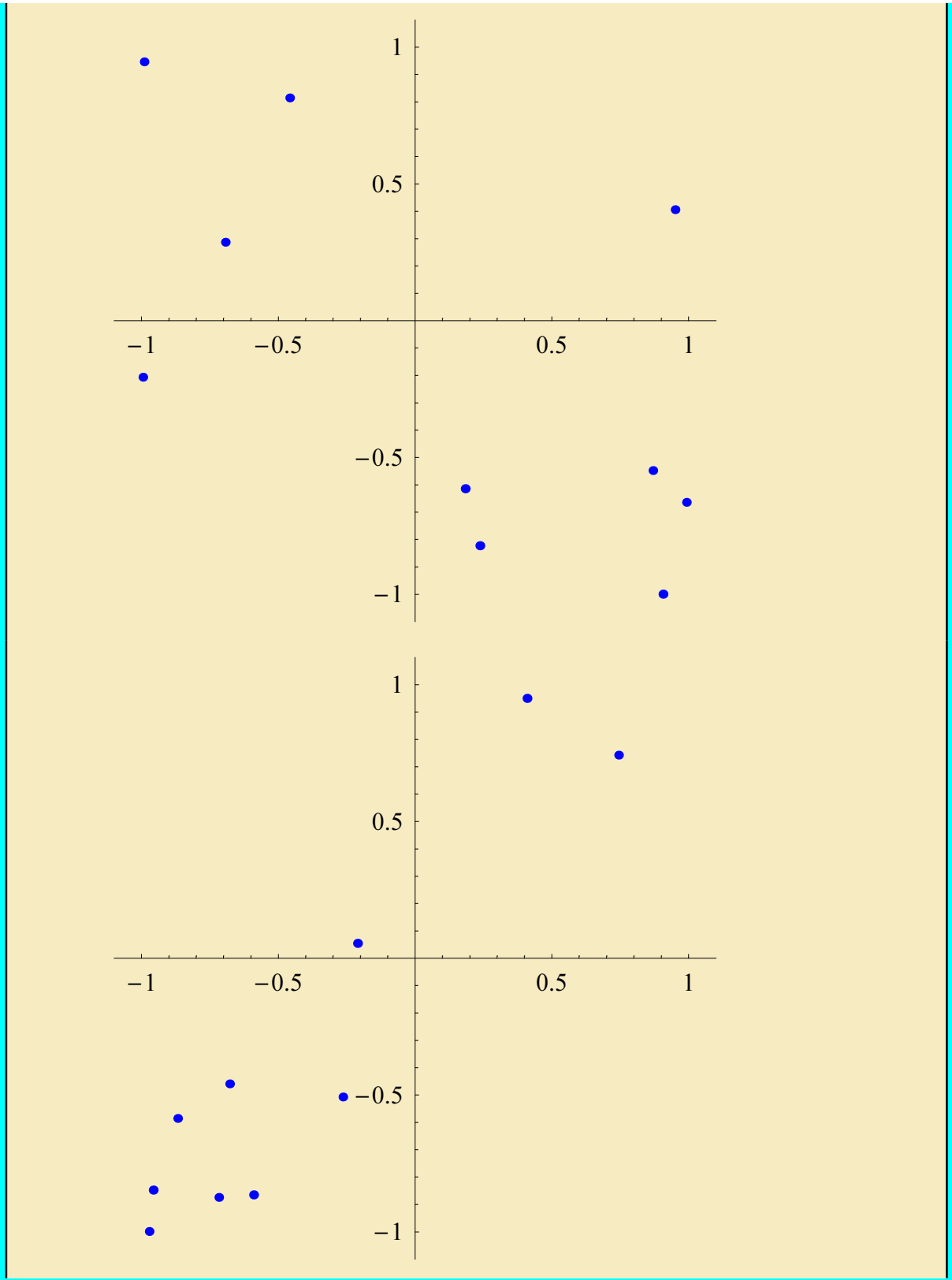


- Graphics -

```
Do[
 Export[StringJoin[ ToString["gr"],
   ToString[ k + 100], ToString[".bmp"]],
  gr[k]], {k, 1, 6}
]
```


1
0.5
−1
−0.5
0.5
1
−0.5
−1
1
0.5
−1
−0.5
0.5
1
−0.5
−1


1
0.5
−1
−0.5
0.5
1
−0.5
−1
1
0.5
−1
−0.5
0.5
1
−0.5
−1


1
0.5
−1
−0.5
0.5
1
−0.5
−1
1
0.5
−1
−0.5
0.5
1
−0.5
−1

Exportation d'un fichier de données

```
pascal[n_] :=
 Table[Binomial[p, q], {p, 0, n}, {q, 0, n}]
```

```
Export["pascal5.dat", pascal[5]]
```

```
pascal5.dat
```

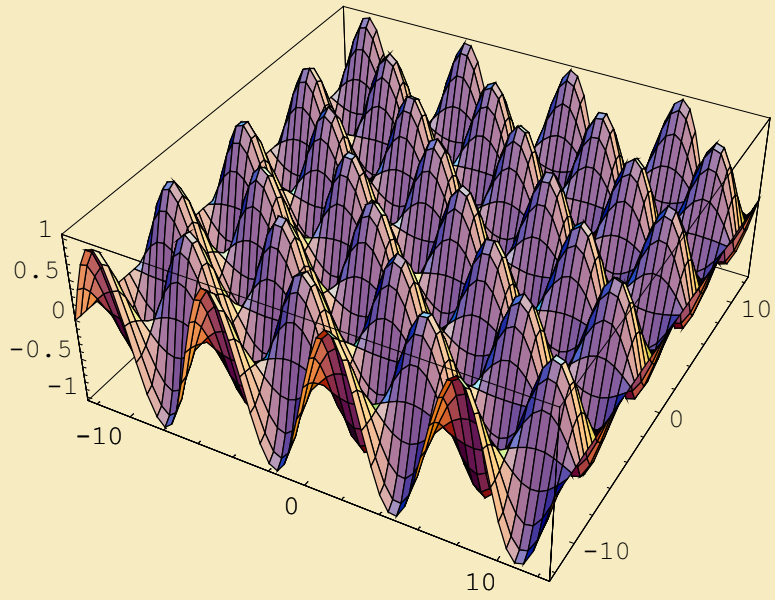
```
Import["pascal5.dat"]
```

```
{{1, 0, 0, 0, 0, 0}, {1, 1, 0, 0, 0, 0}, {1, 2, 1, 0, 0, 0},
 {1, 3, 3, 1, 0, 0}, {1, 4, 6, 4, 1, 0}, {1, 5, 10, 10, 5, 1}}
```

Exercices

Exercice 1

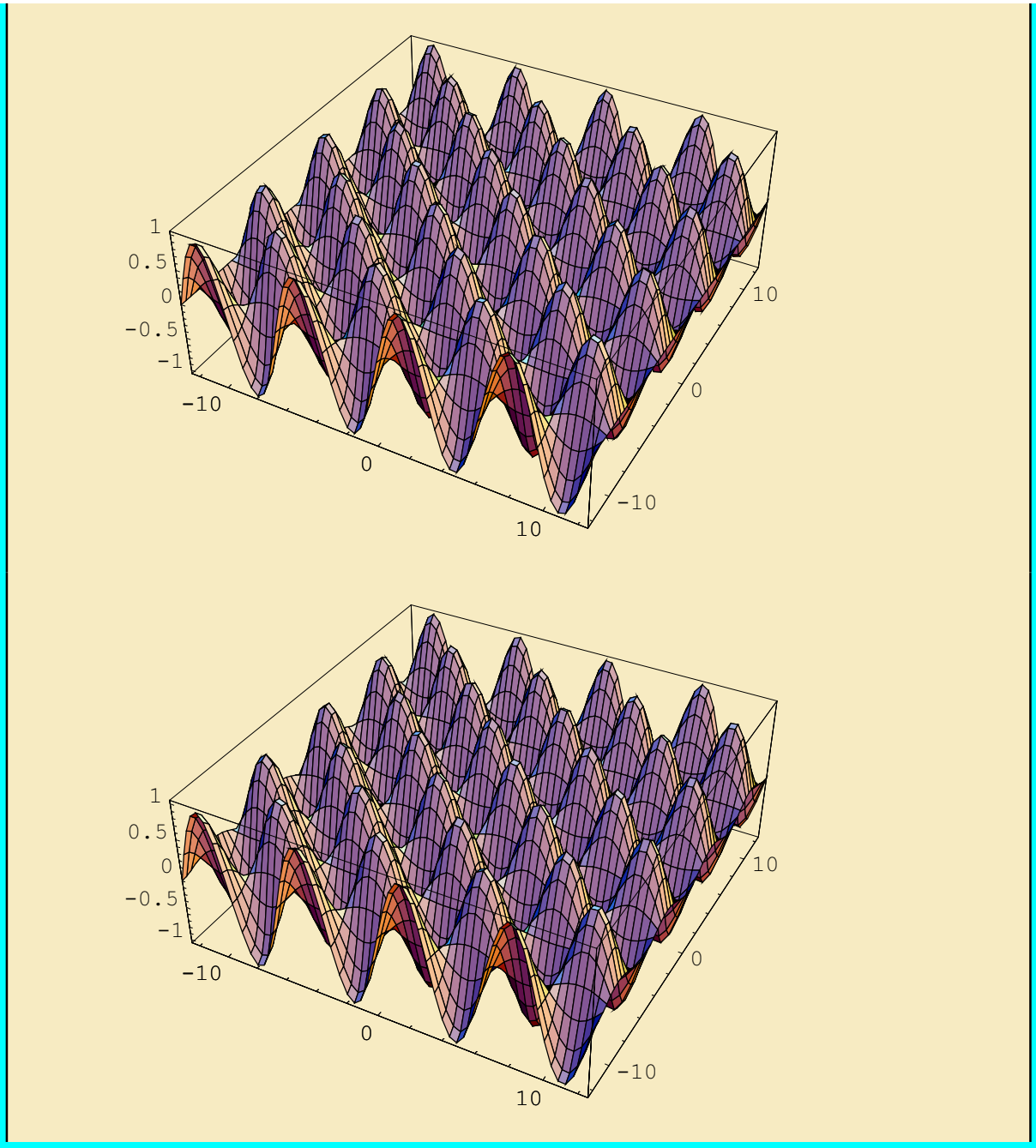
```
Plot3D[Sin[x] Cos[y], {x, -4 Pi, 4 Pi},
 {y, -4 Pi, 4 Pi}, PlotPoints → 60]
```

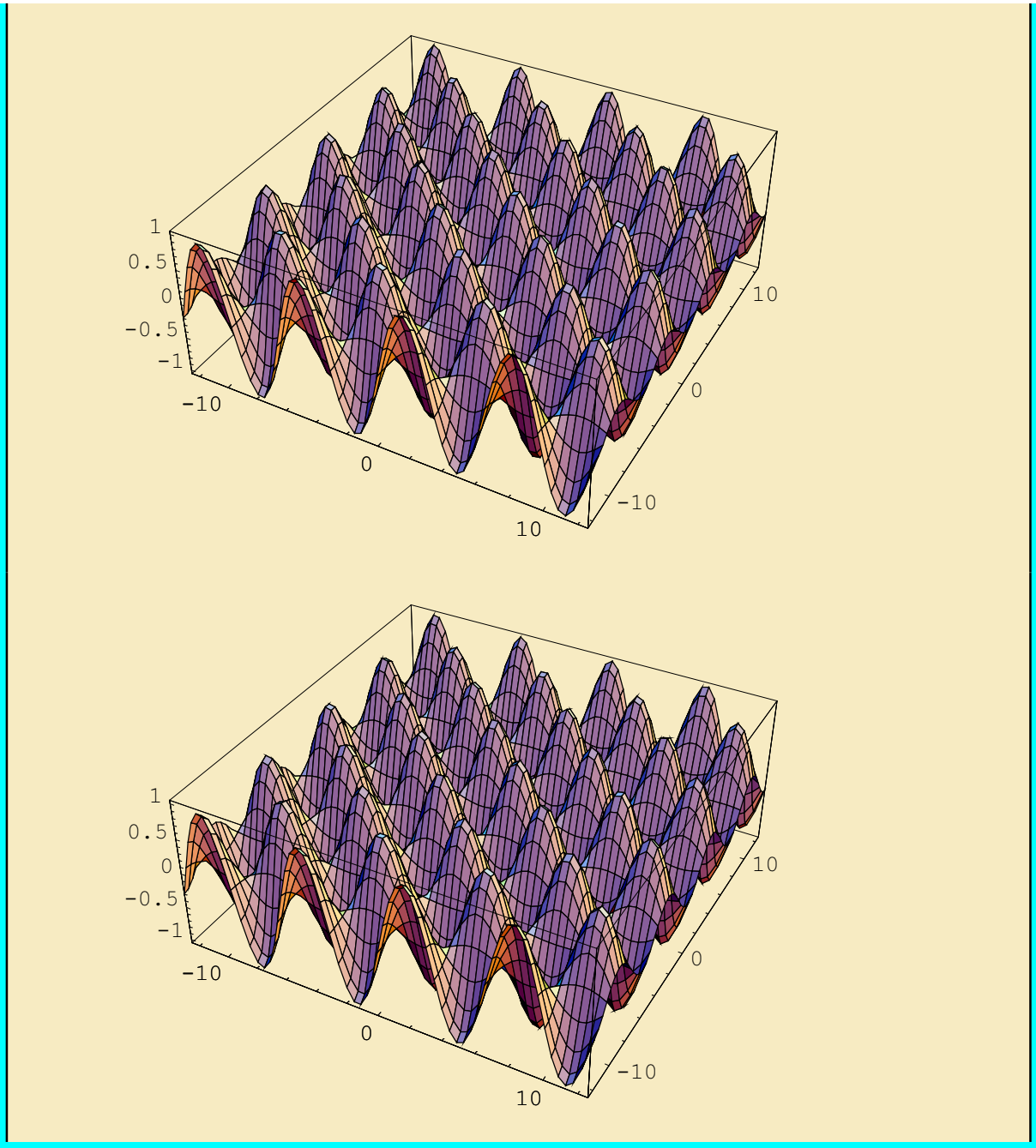


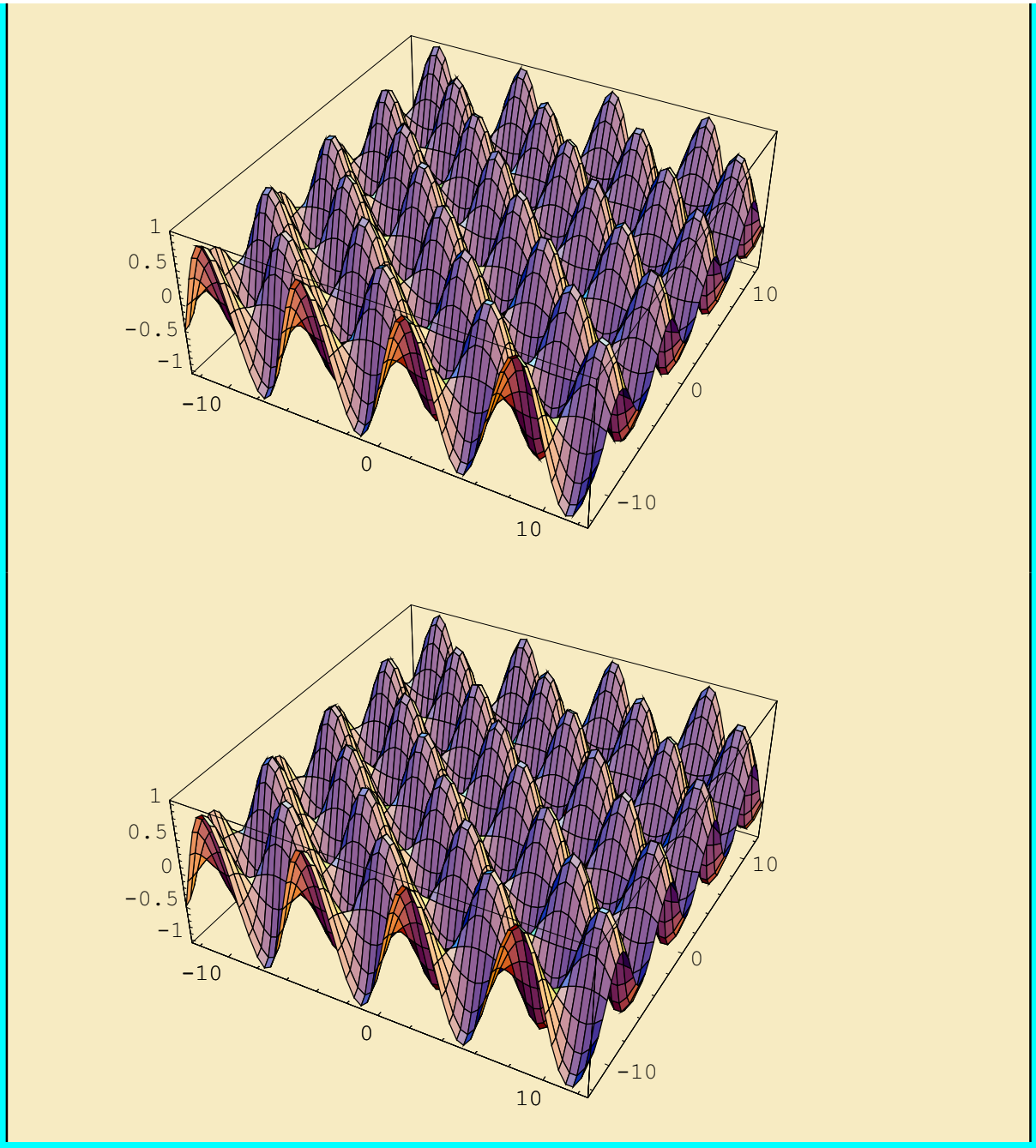
- SurfaceGraphics -

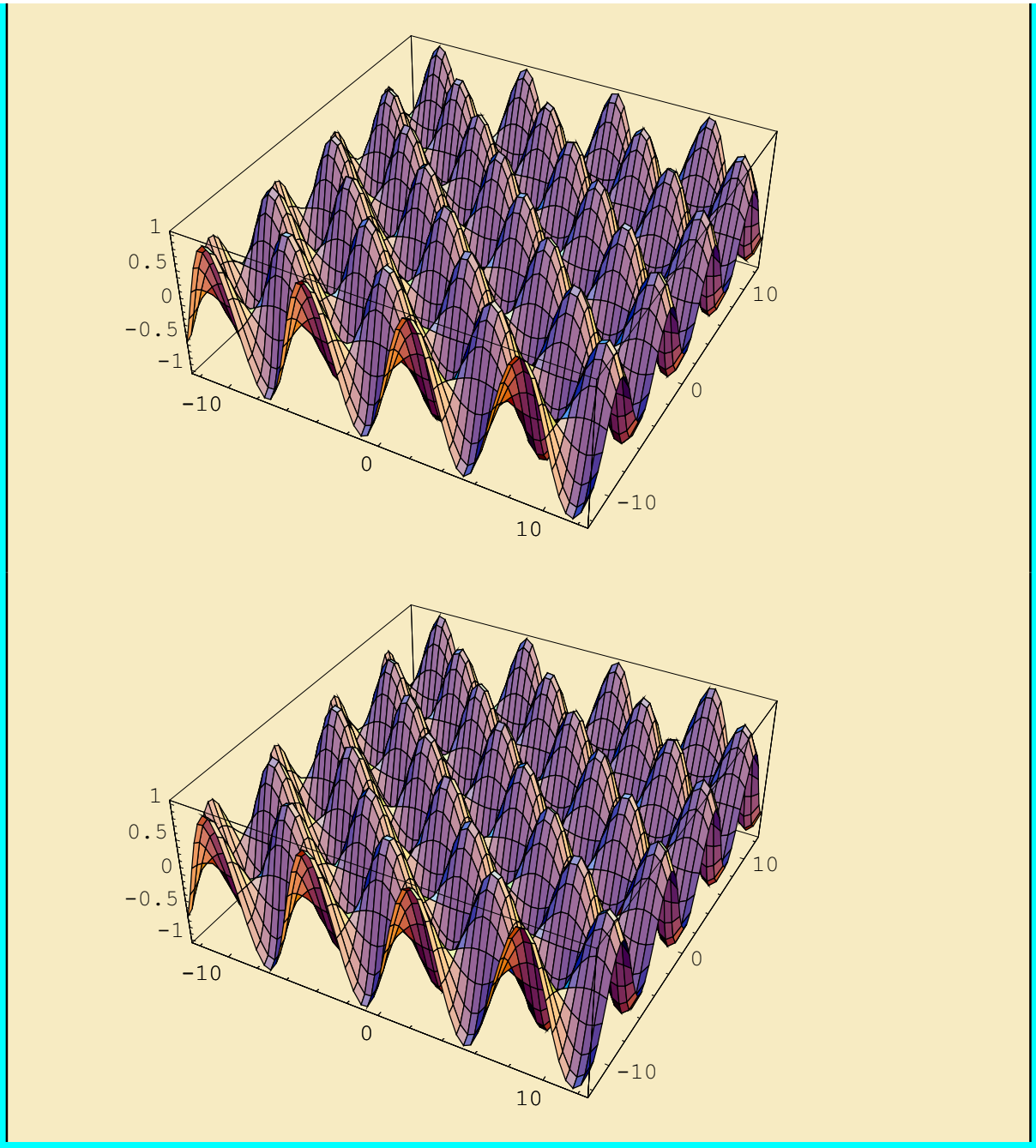
```
g[k_] :=
 Plot3D[Sin[x - k / 10] Cos[y], {x, -4 Pi, 4 Pi},
  {y, -4 Pi, 4 Pi}, PlotPoints → 60]
```

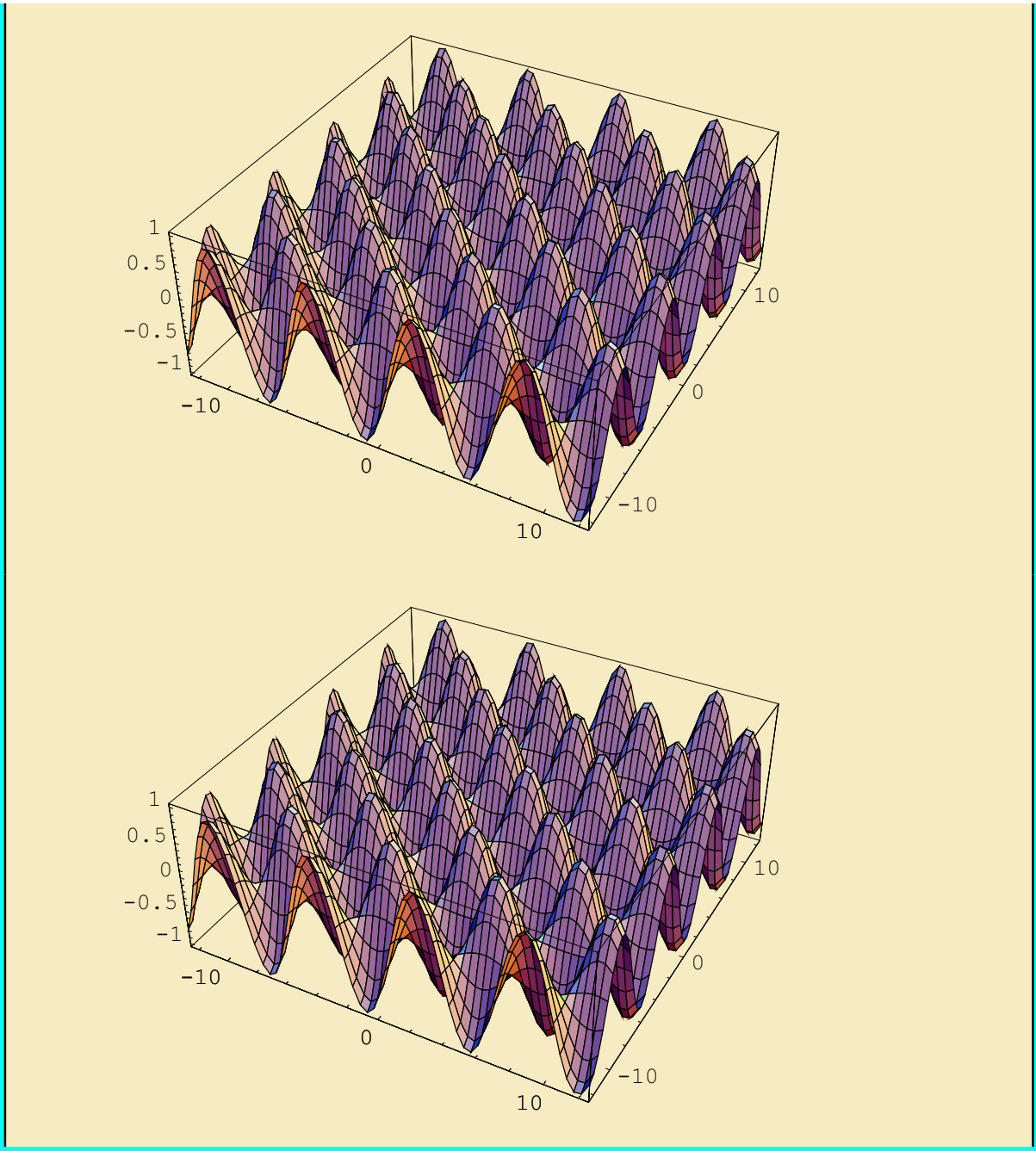
```
Table[g[k], {k, 0, 61}]
```

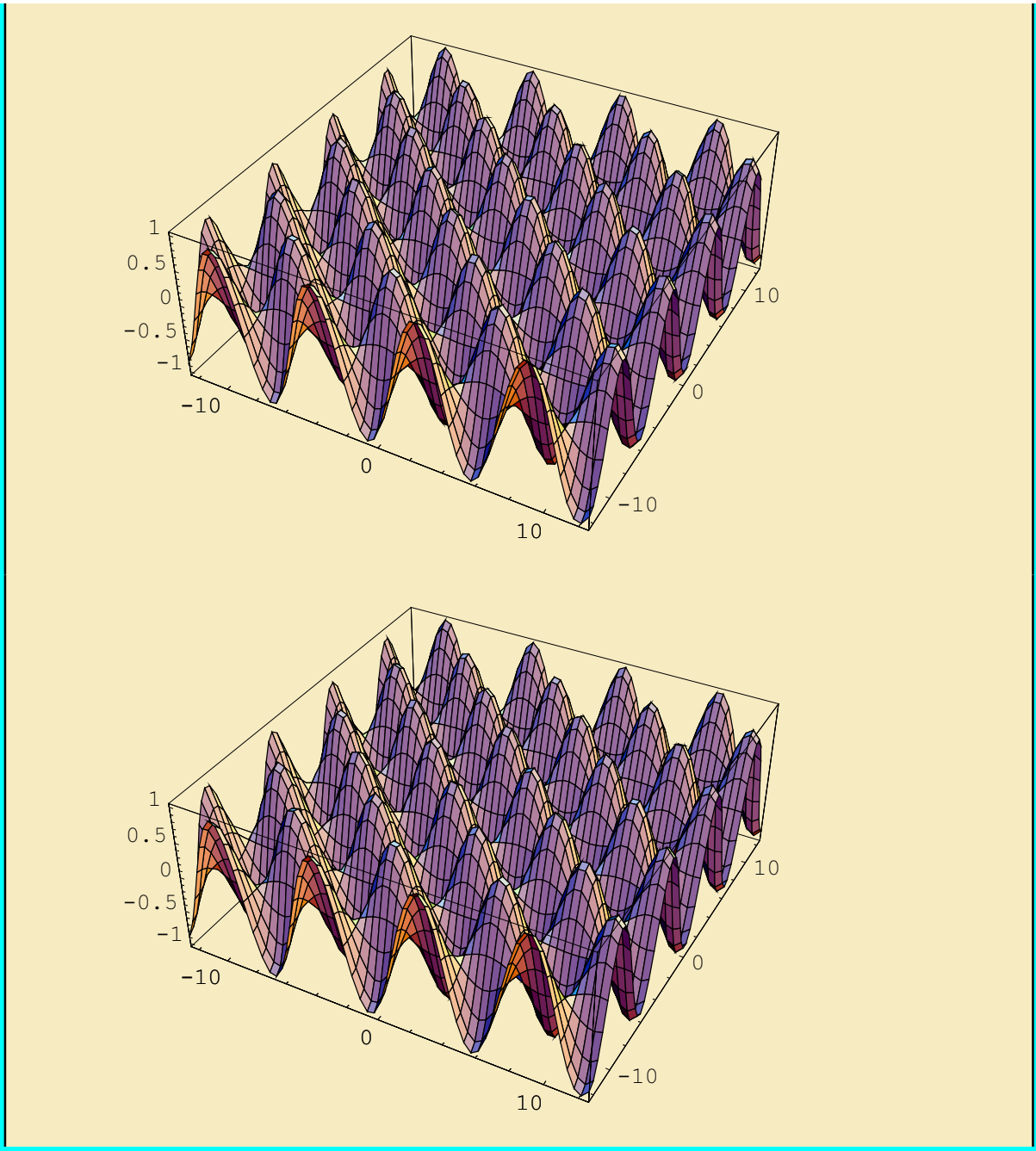

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

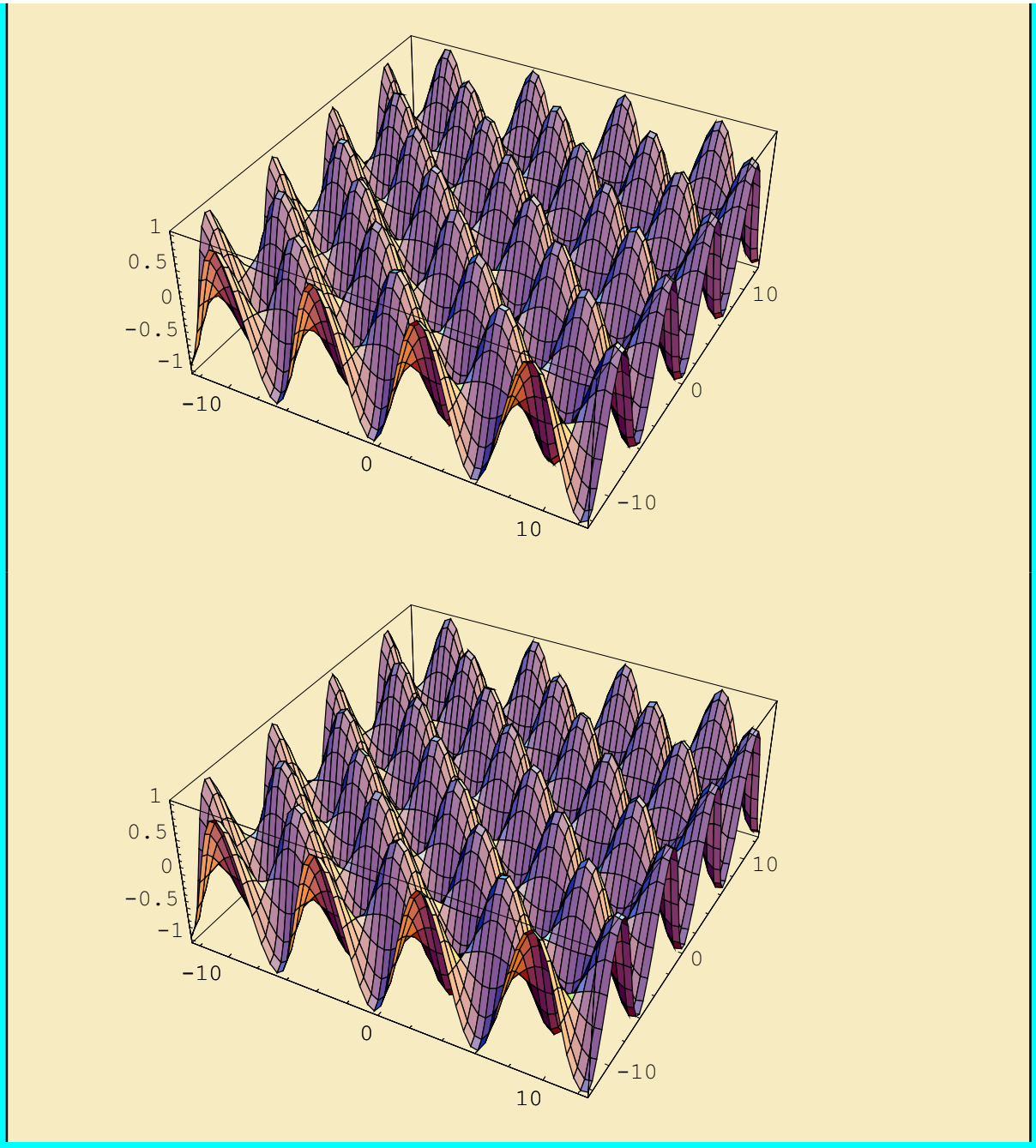

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

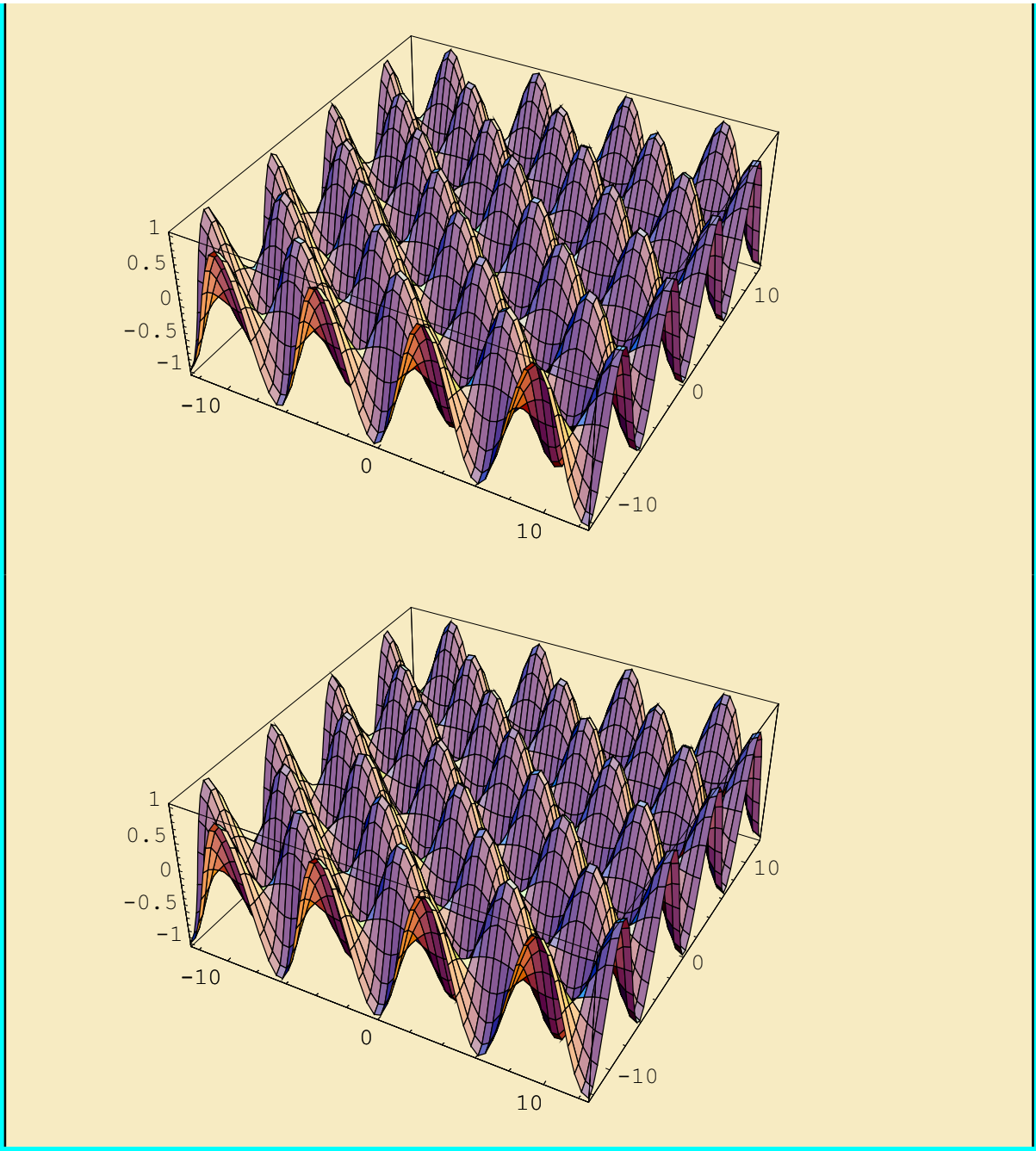

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

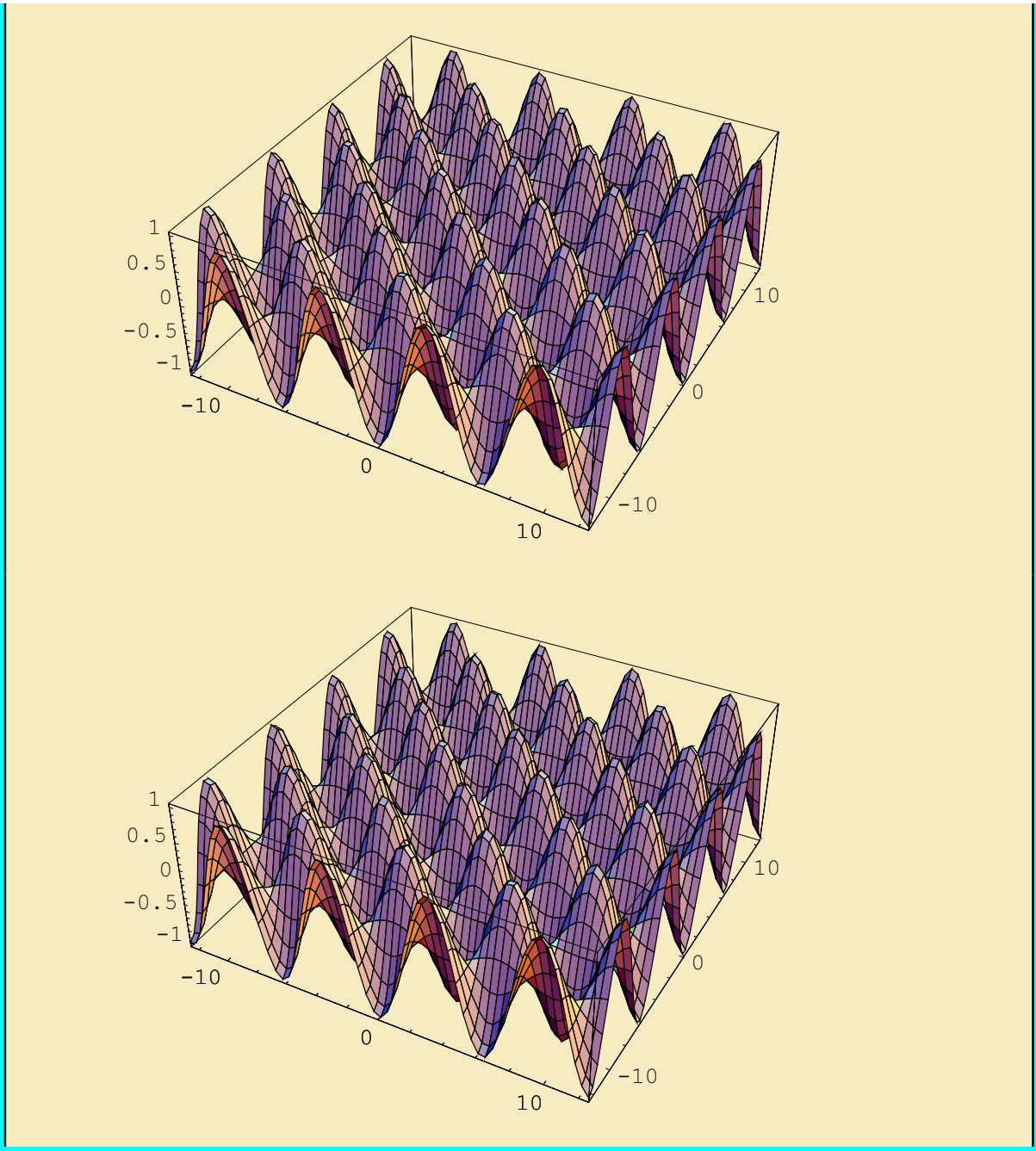

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

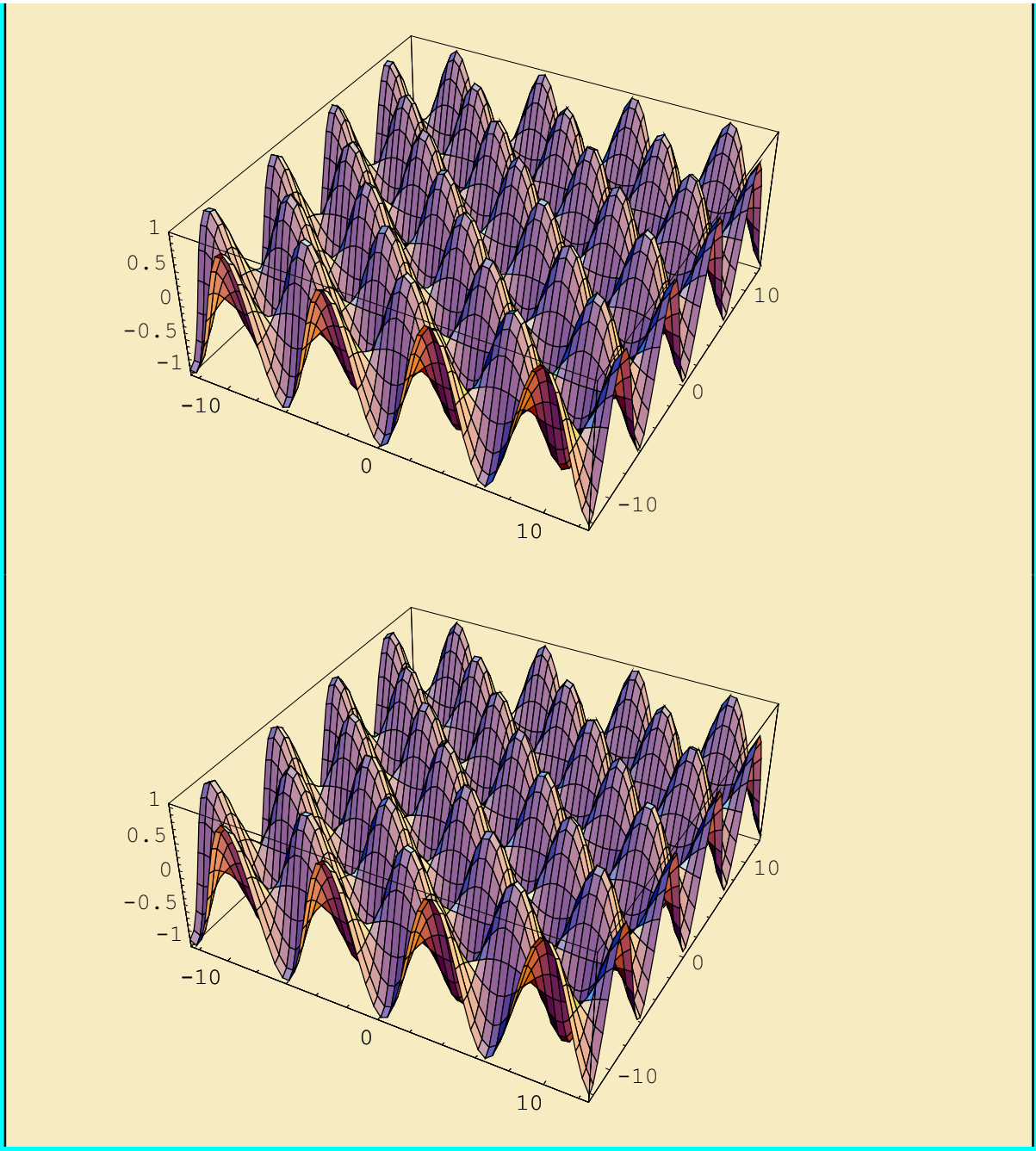

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

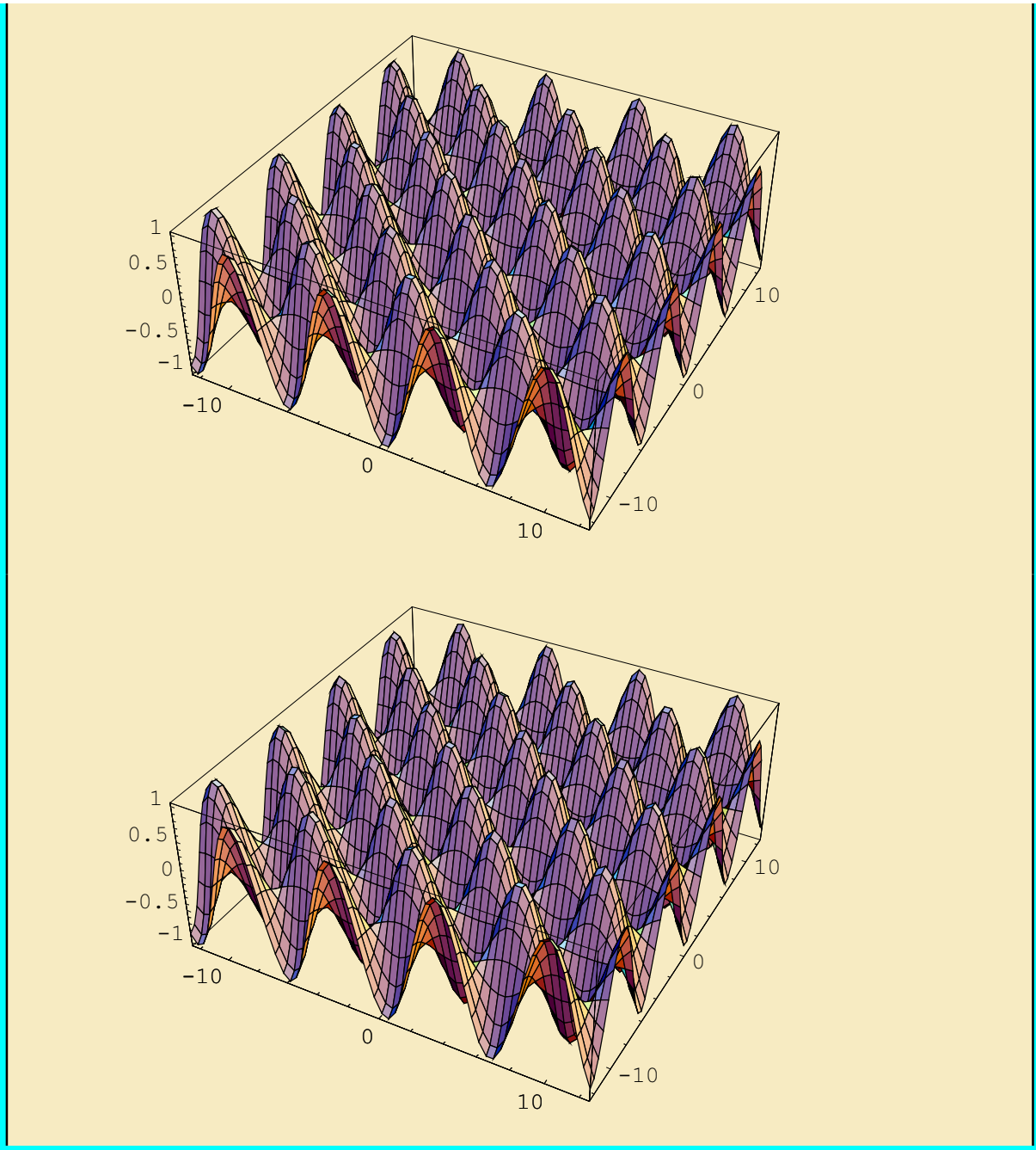

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

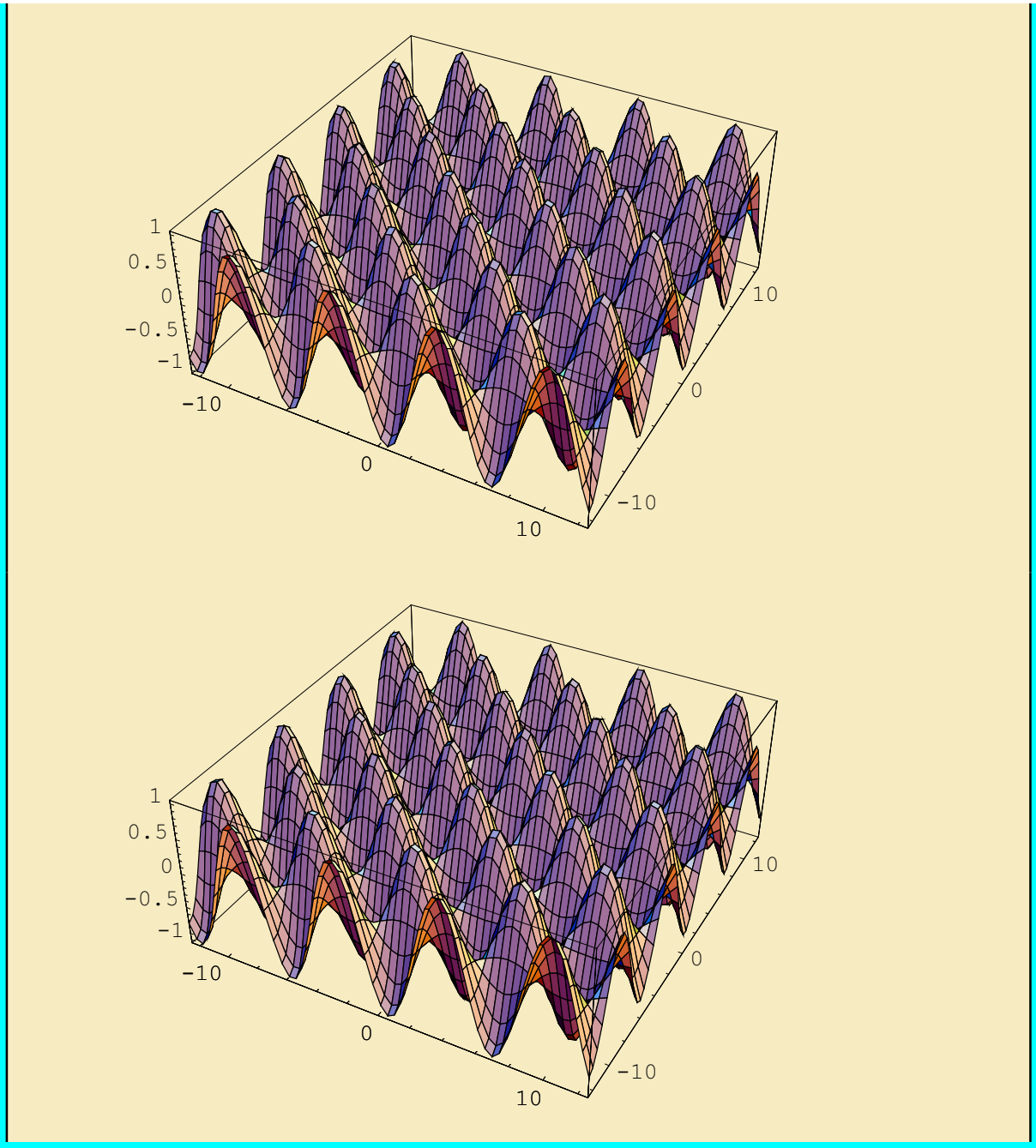

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

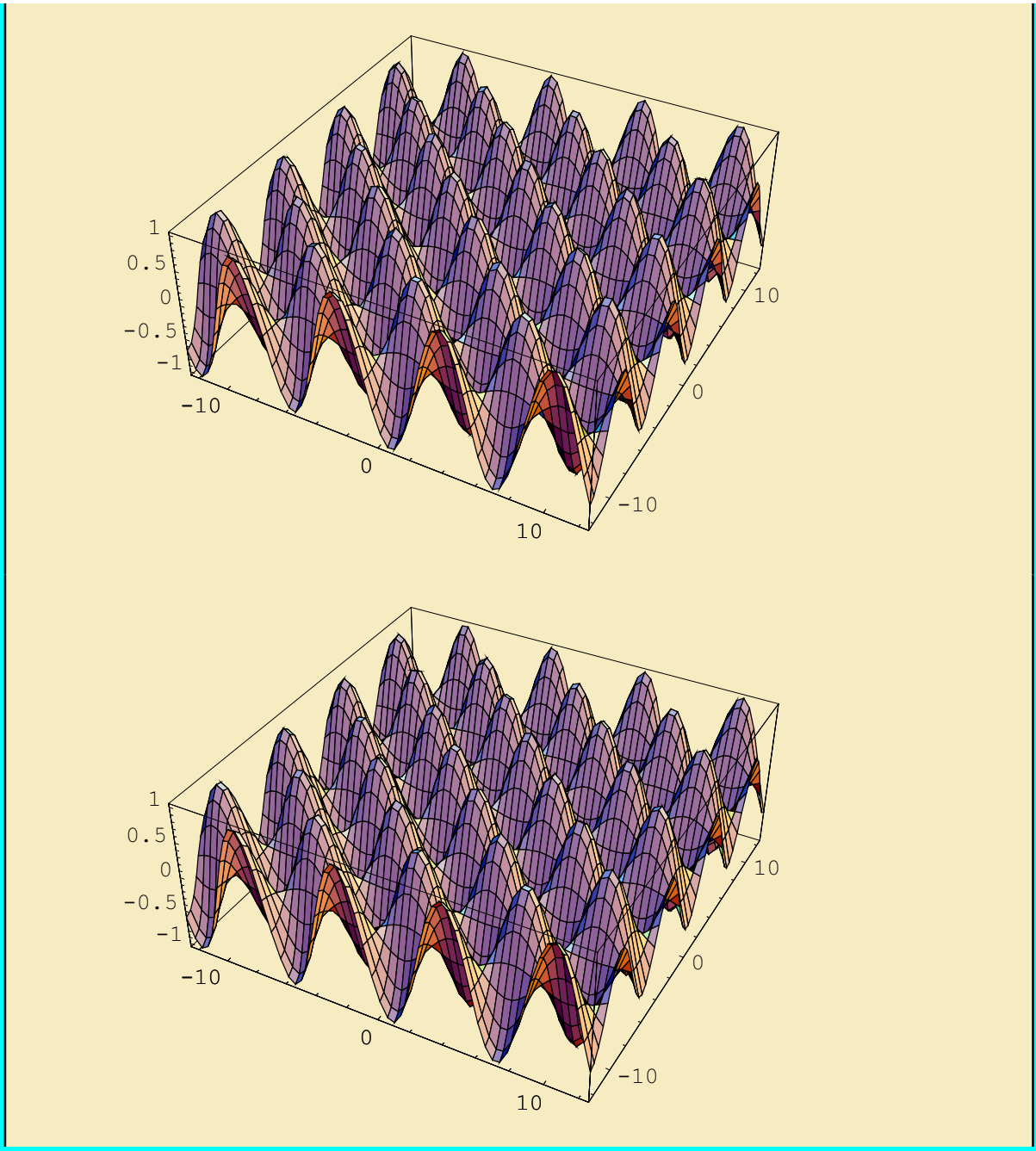

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

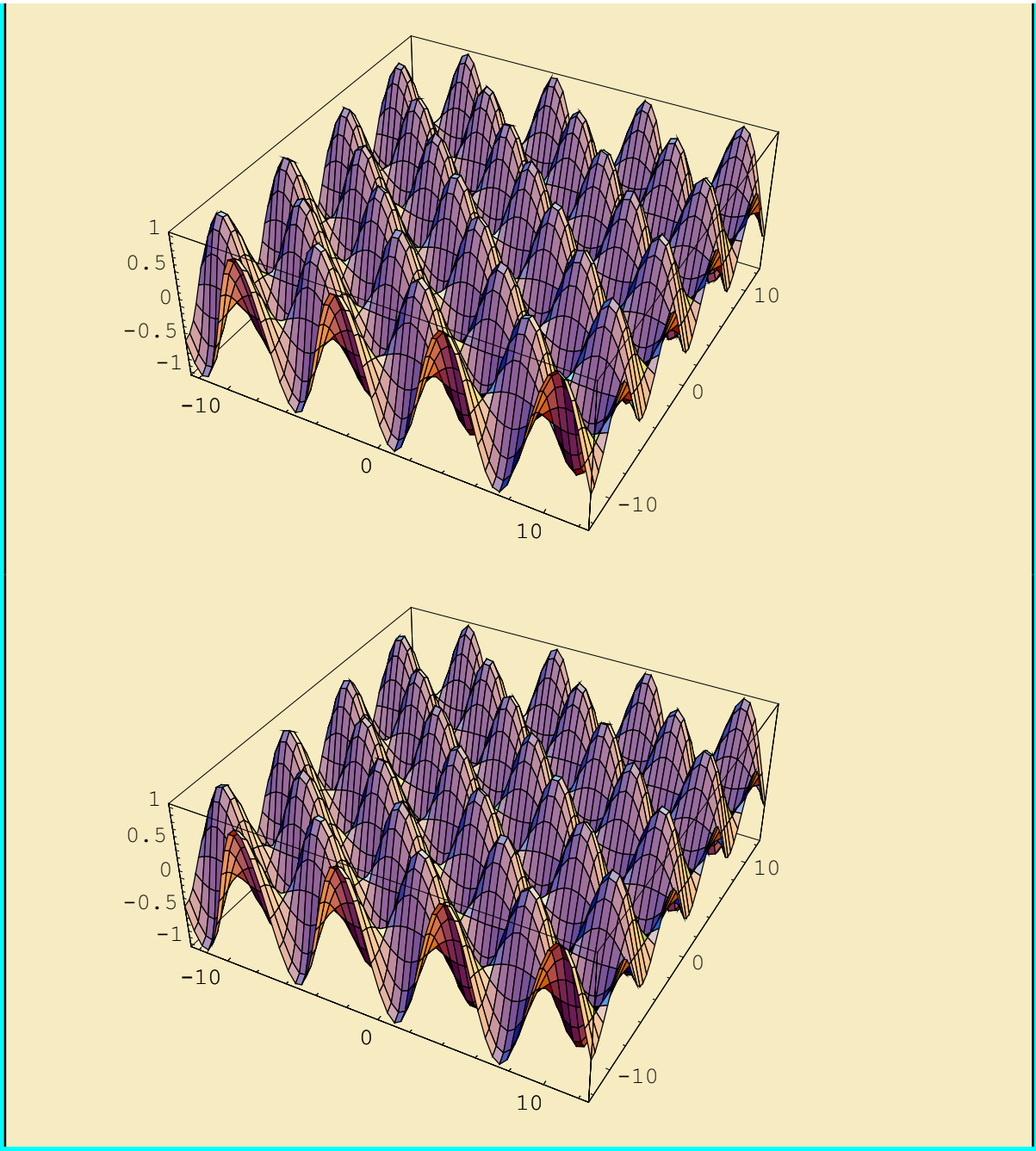

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

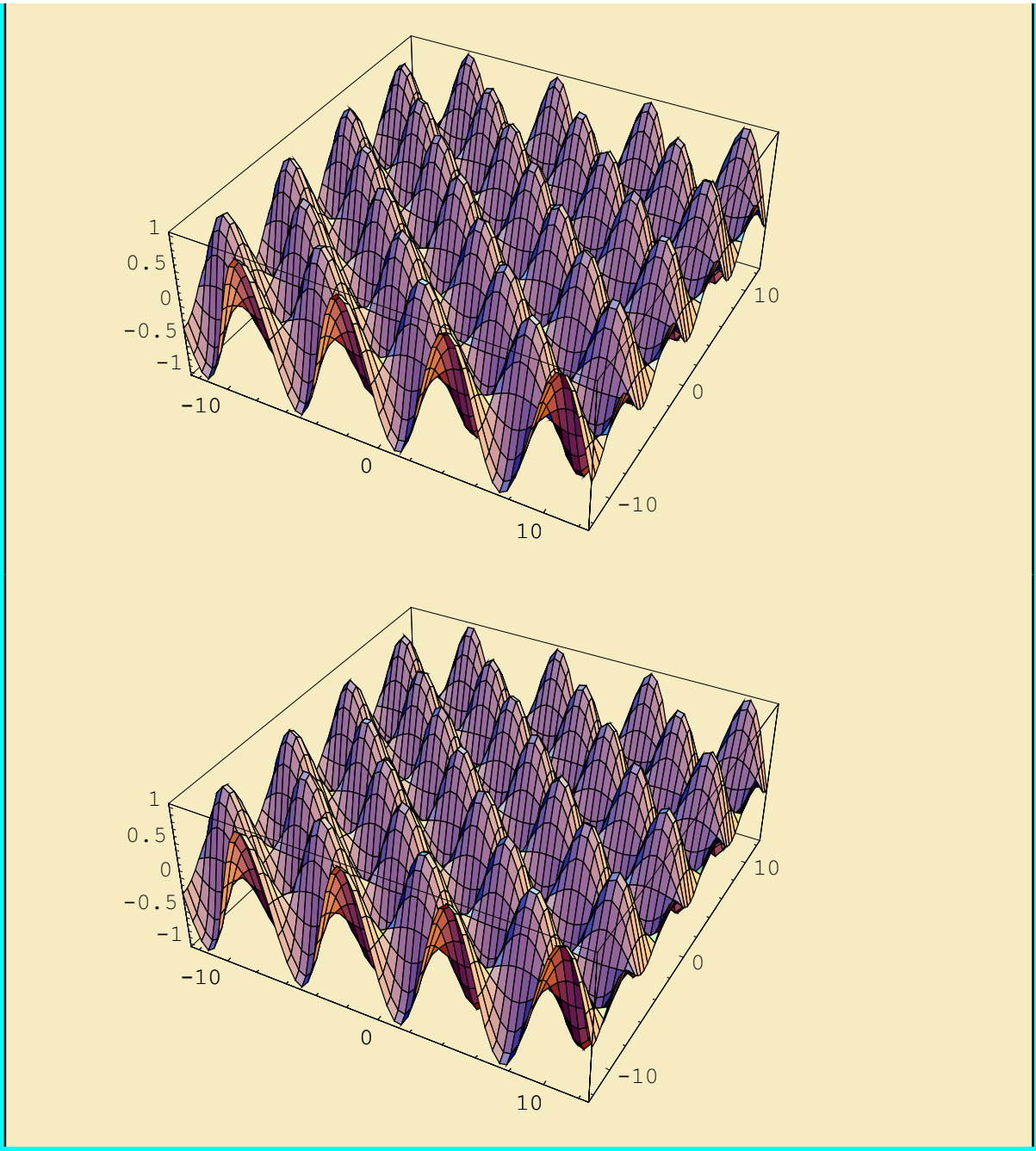

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

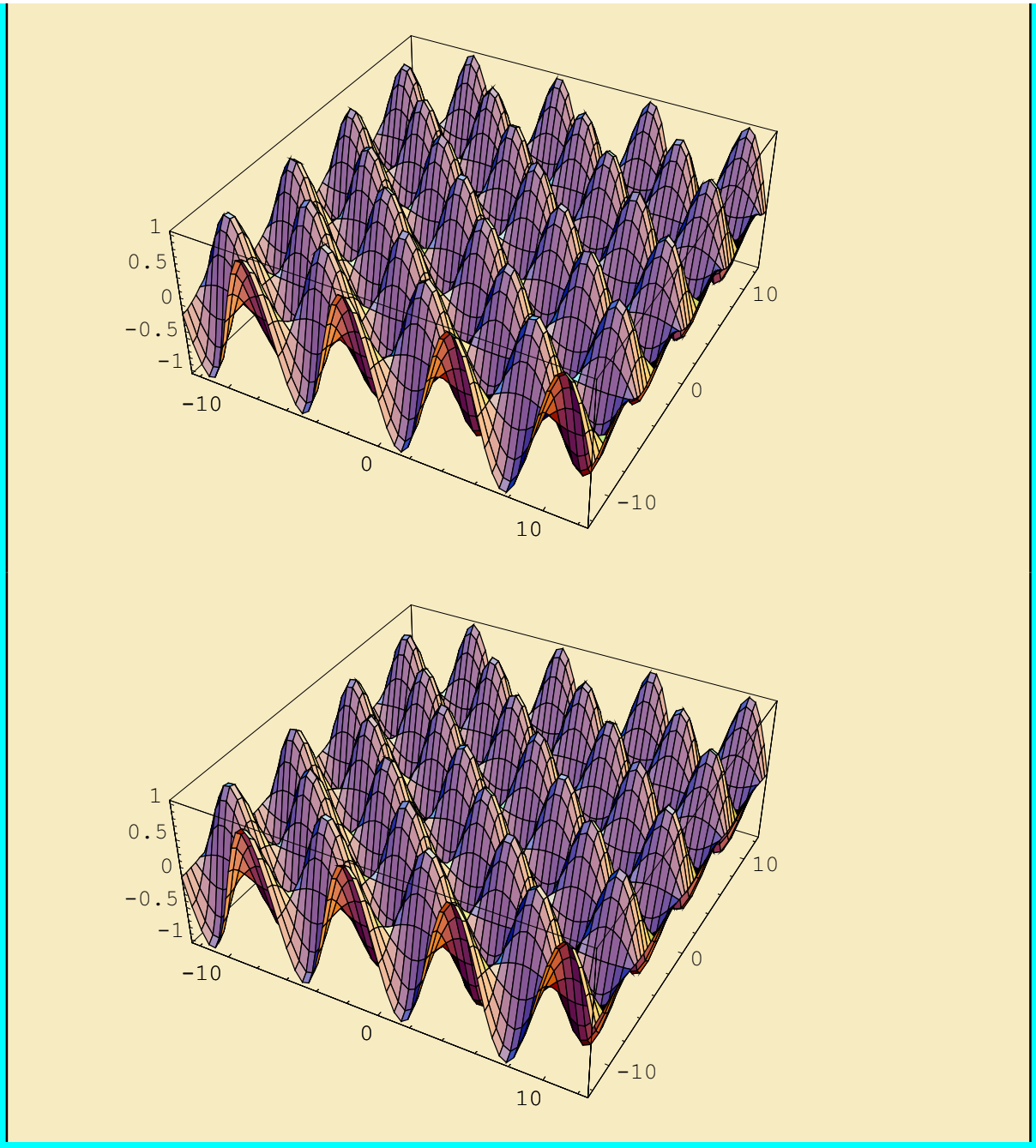

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

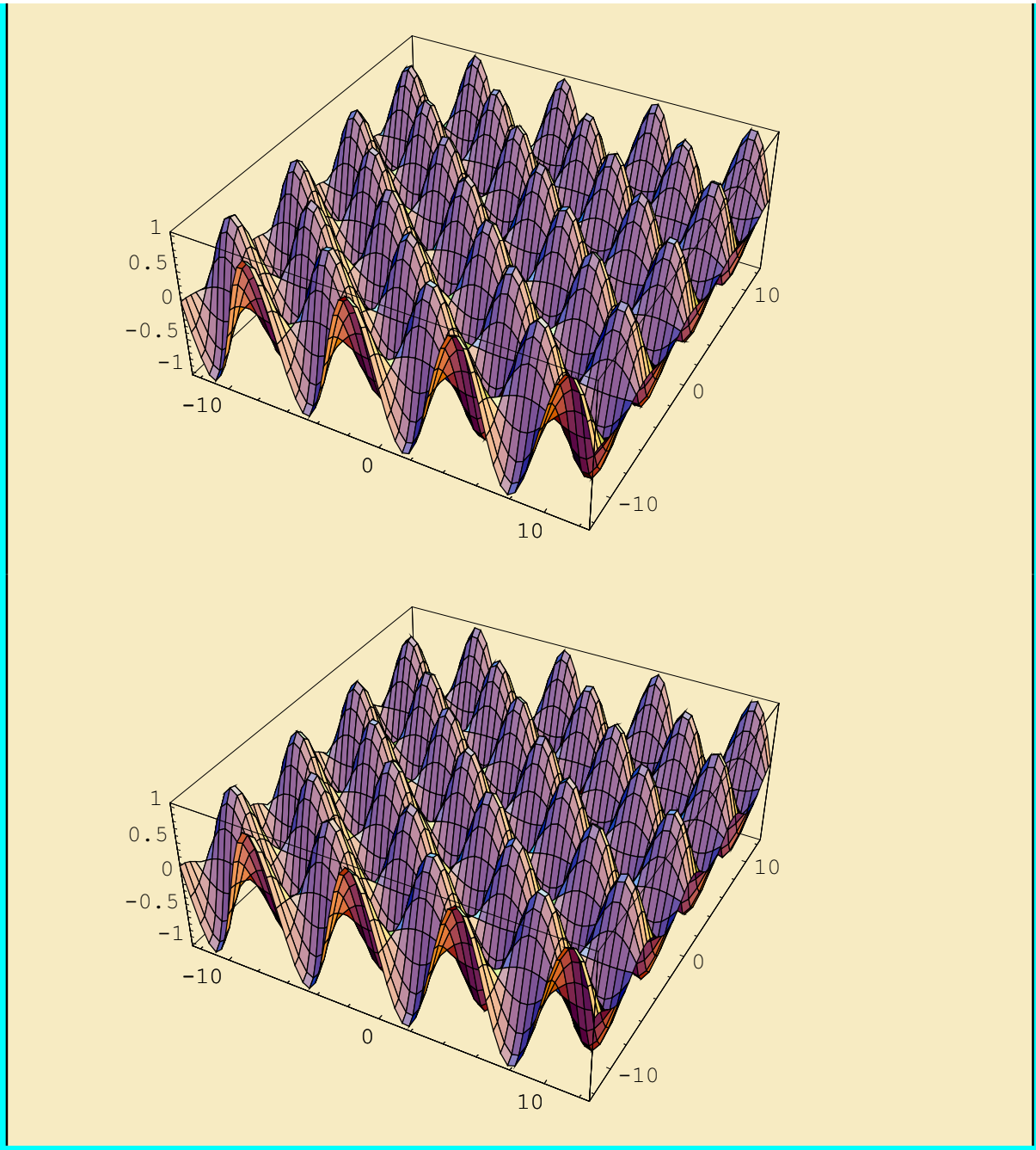

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

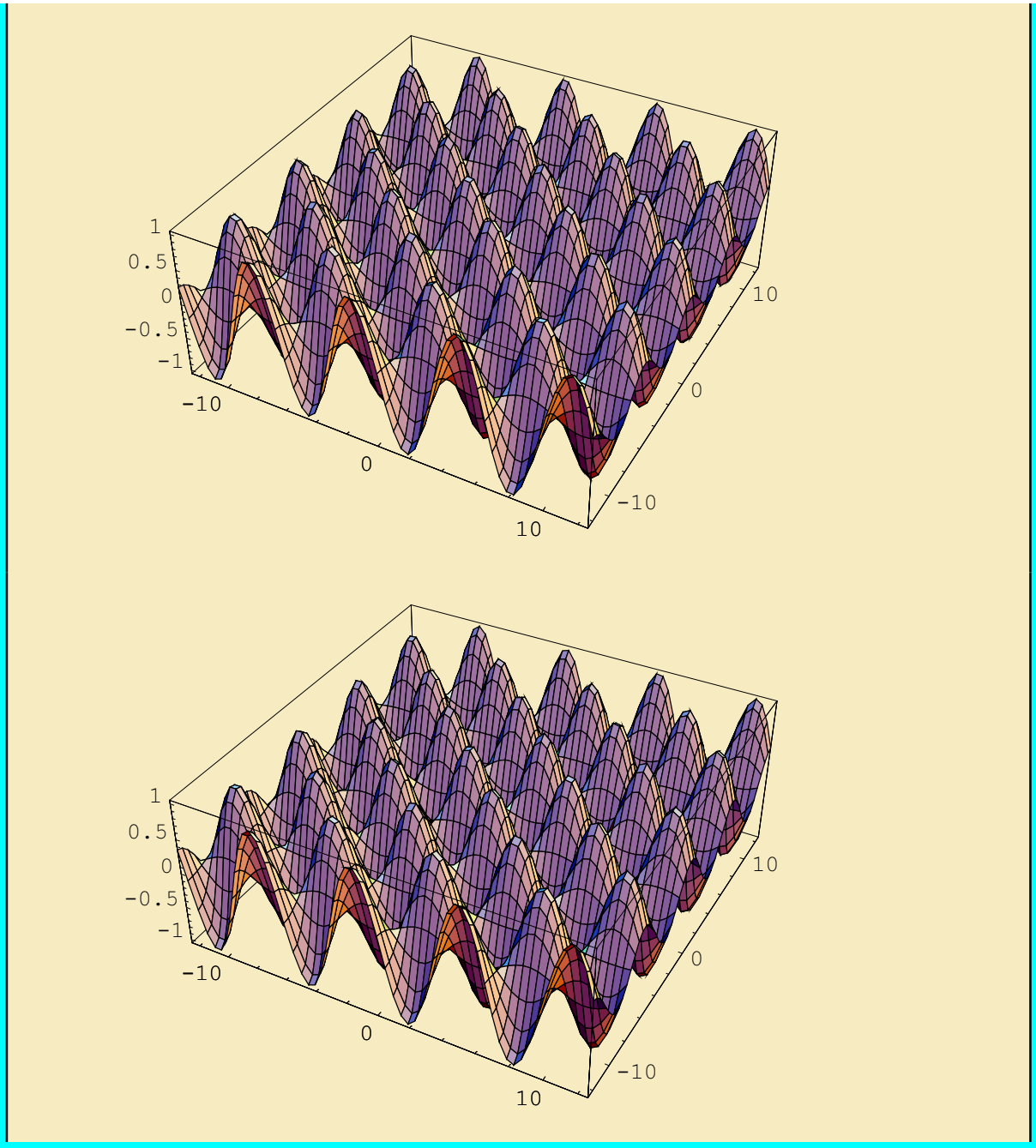

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

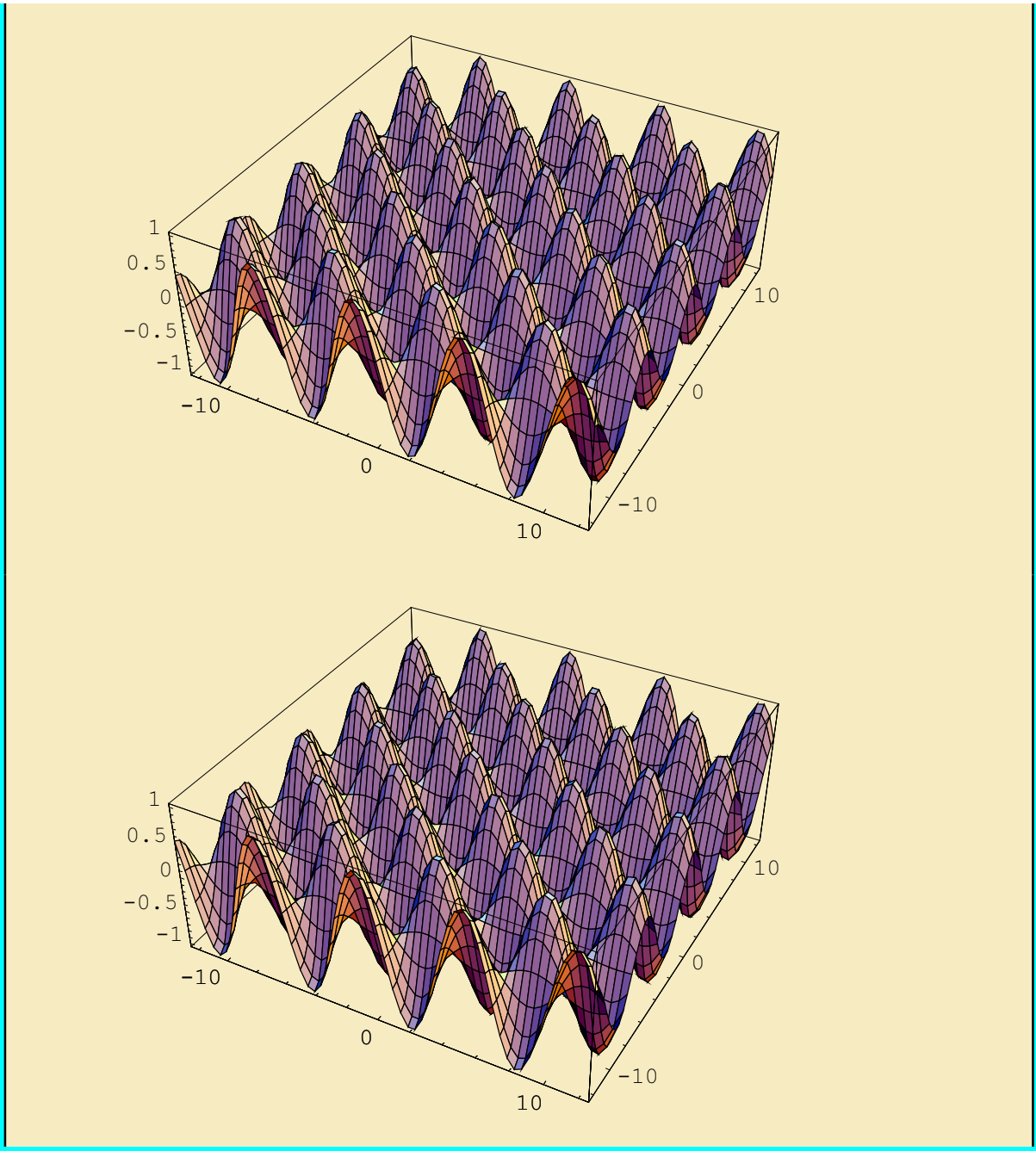

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

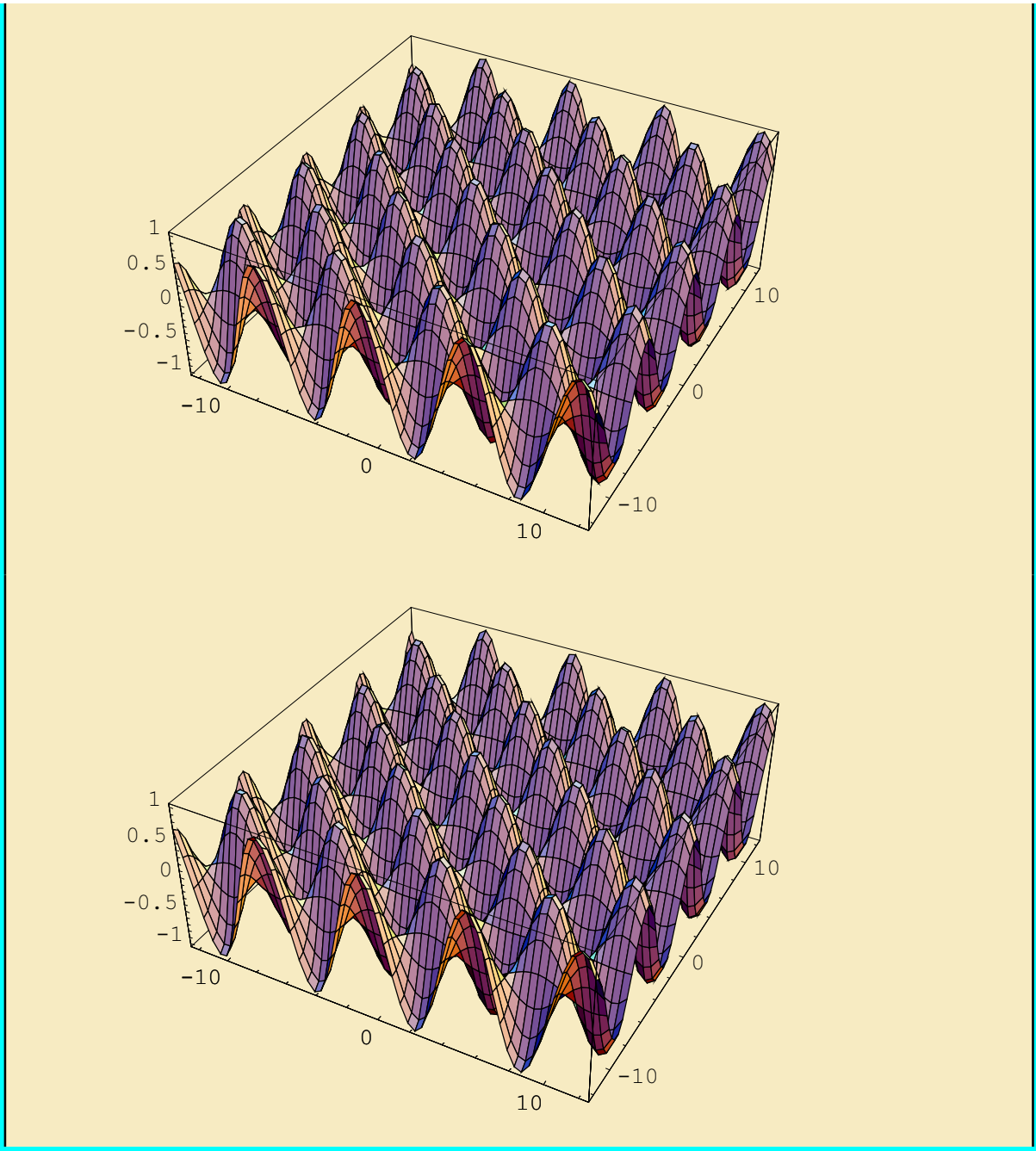

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

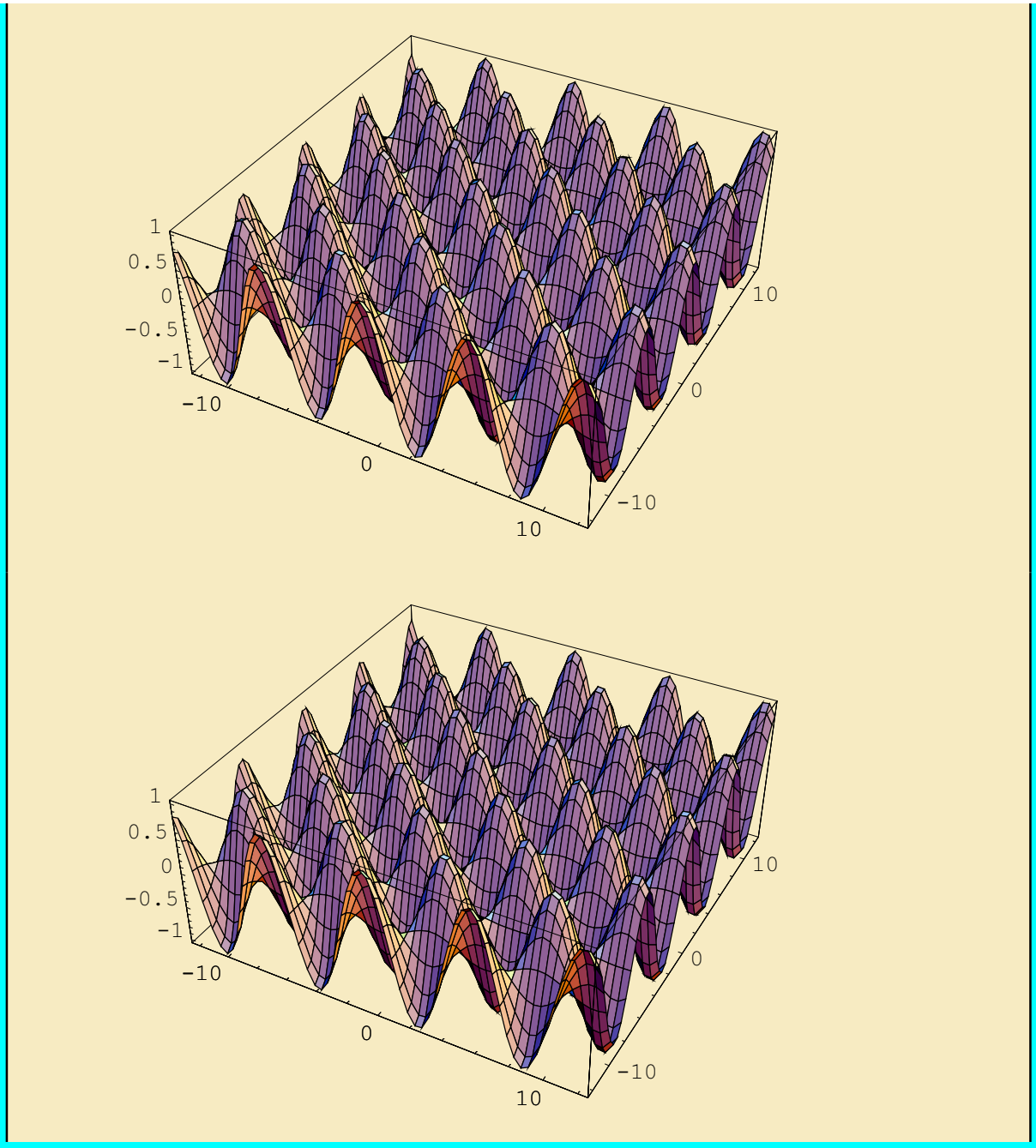

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

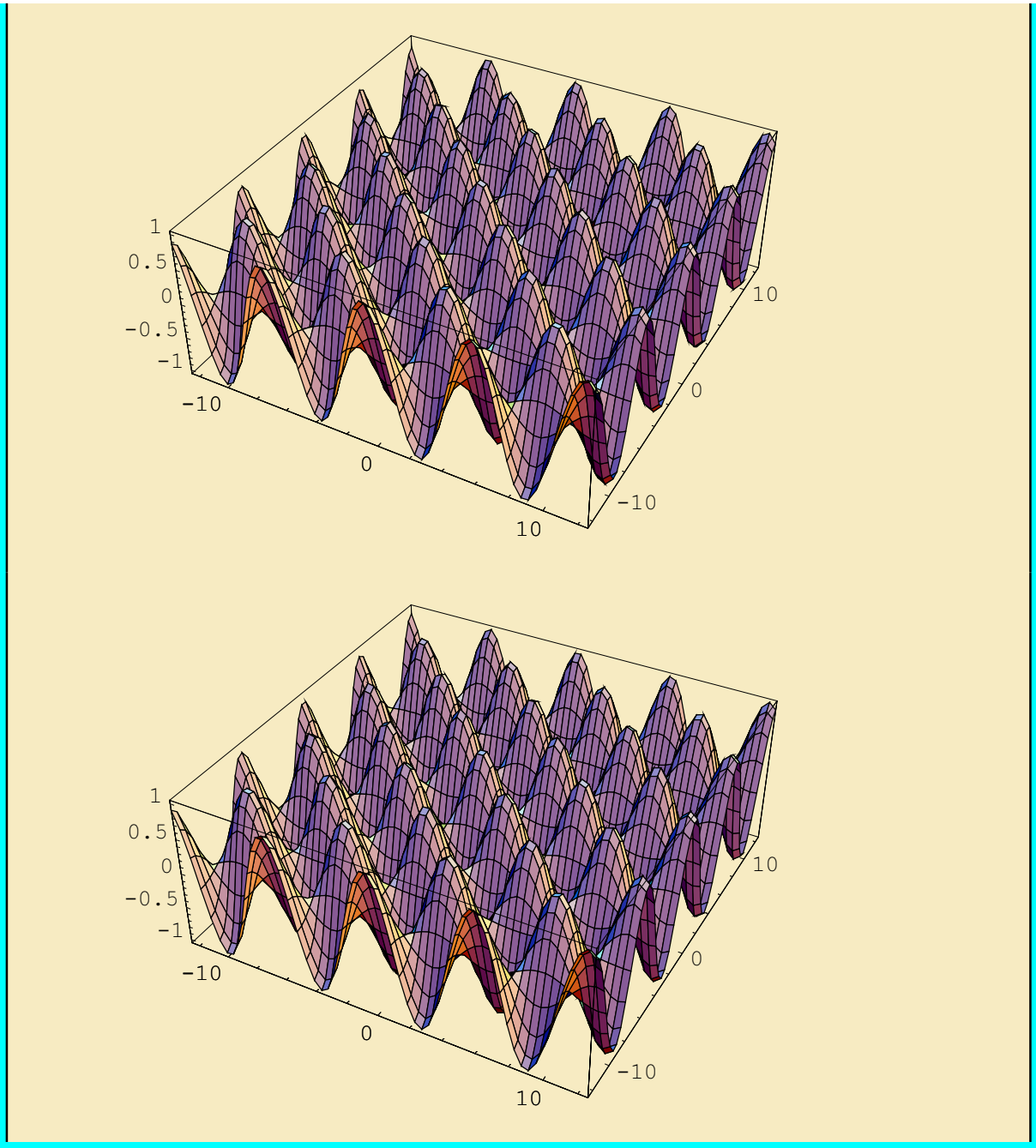

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

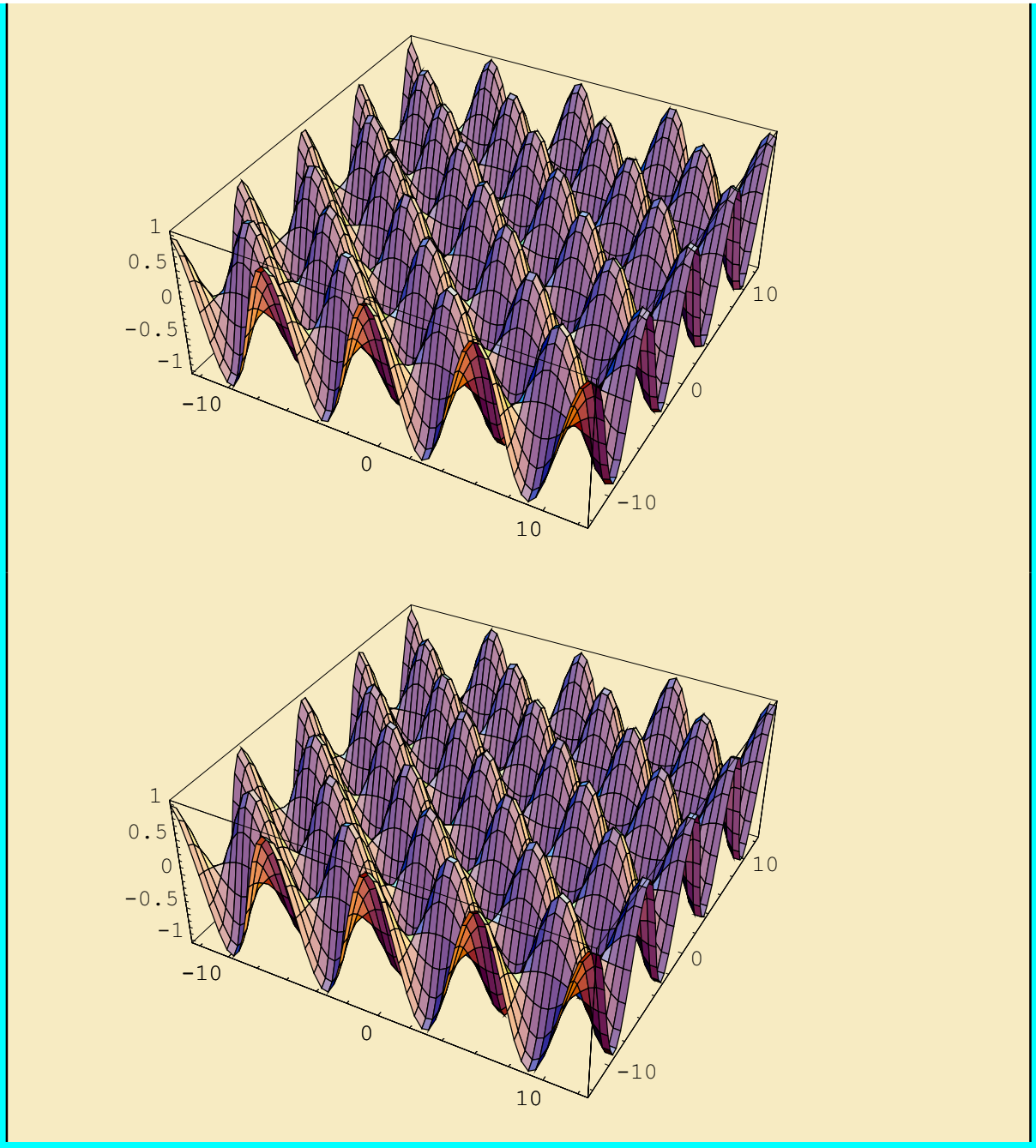

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

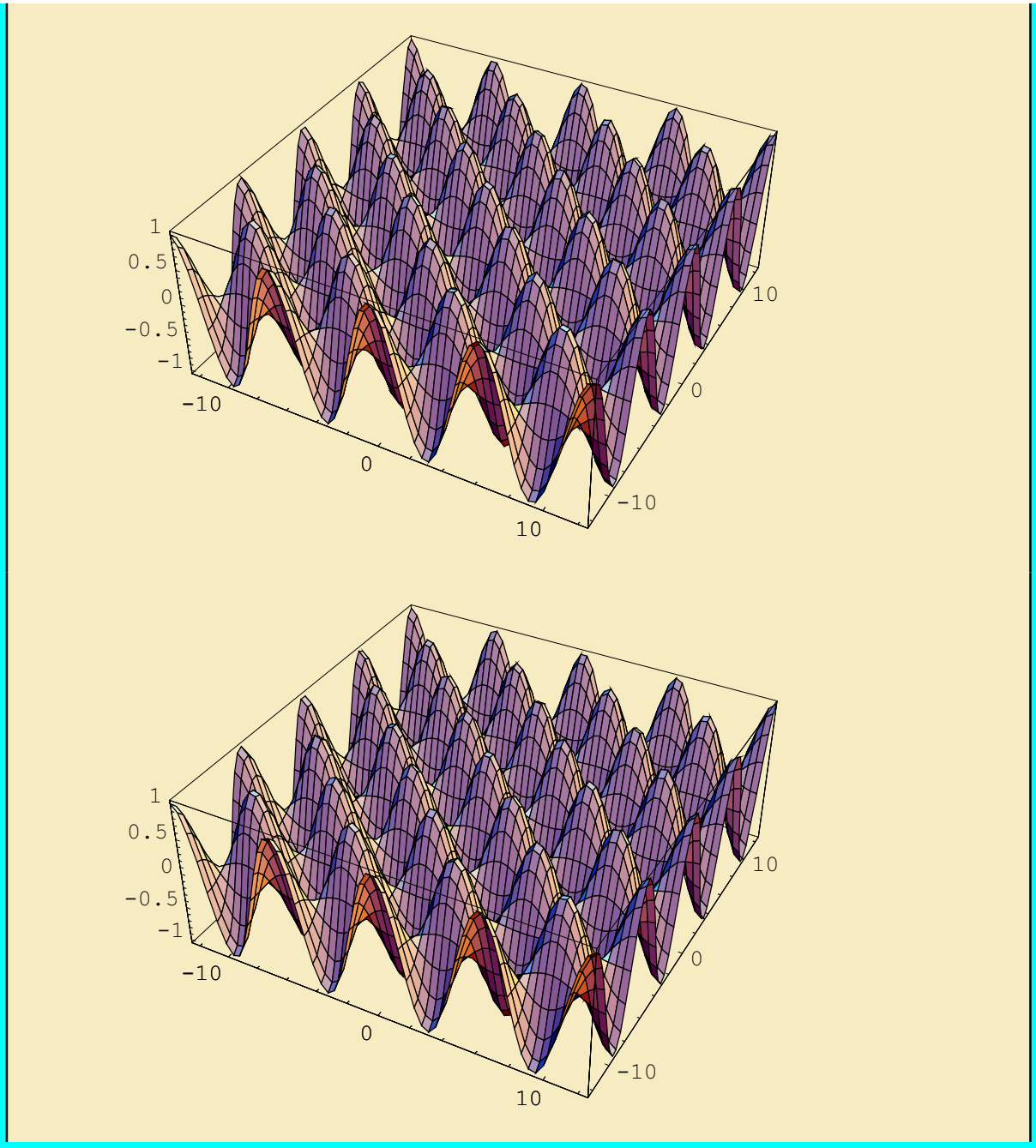

1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

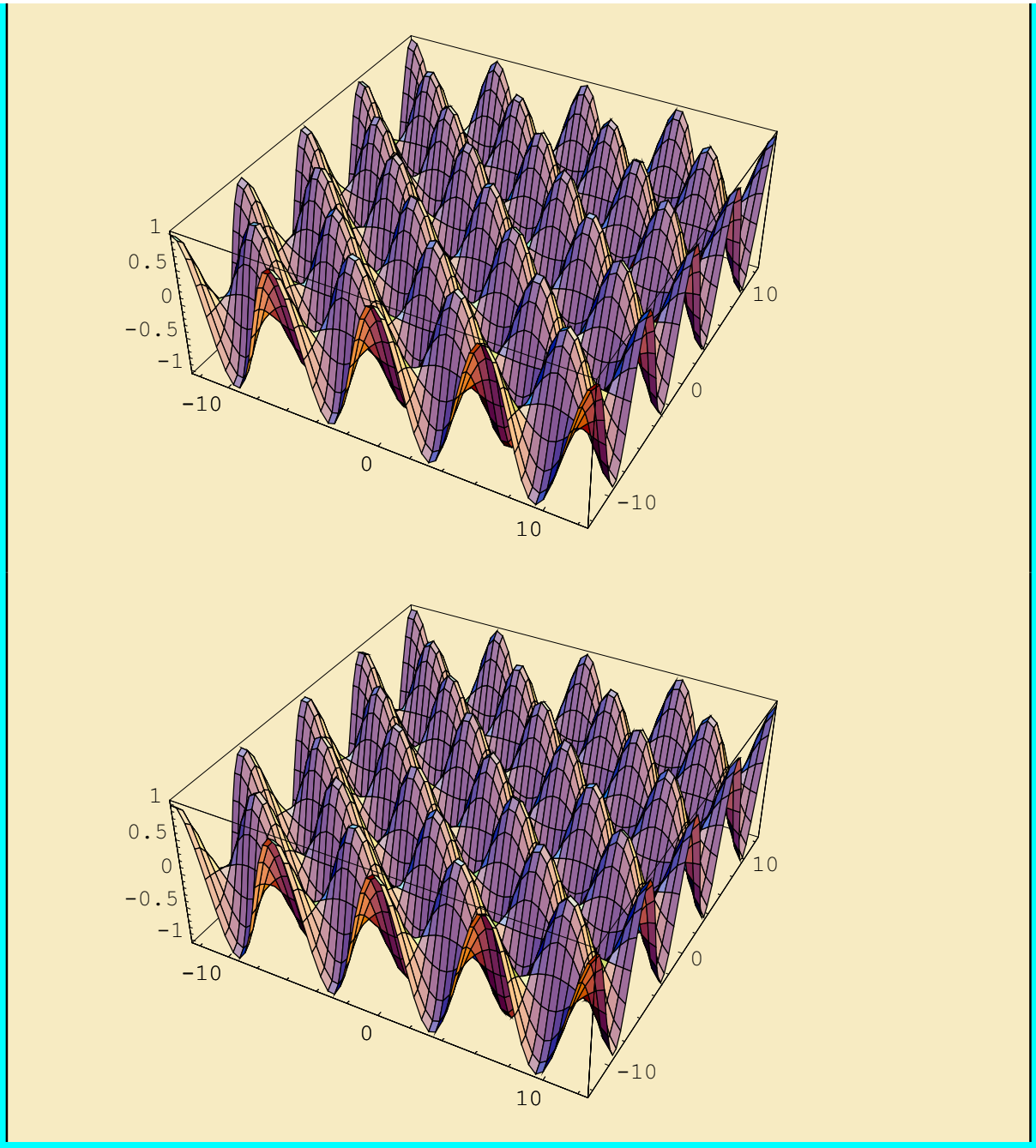

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

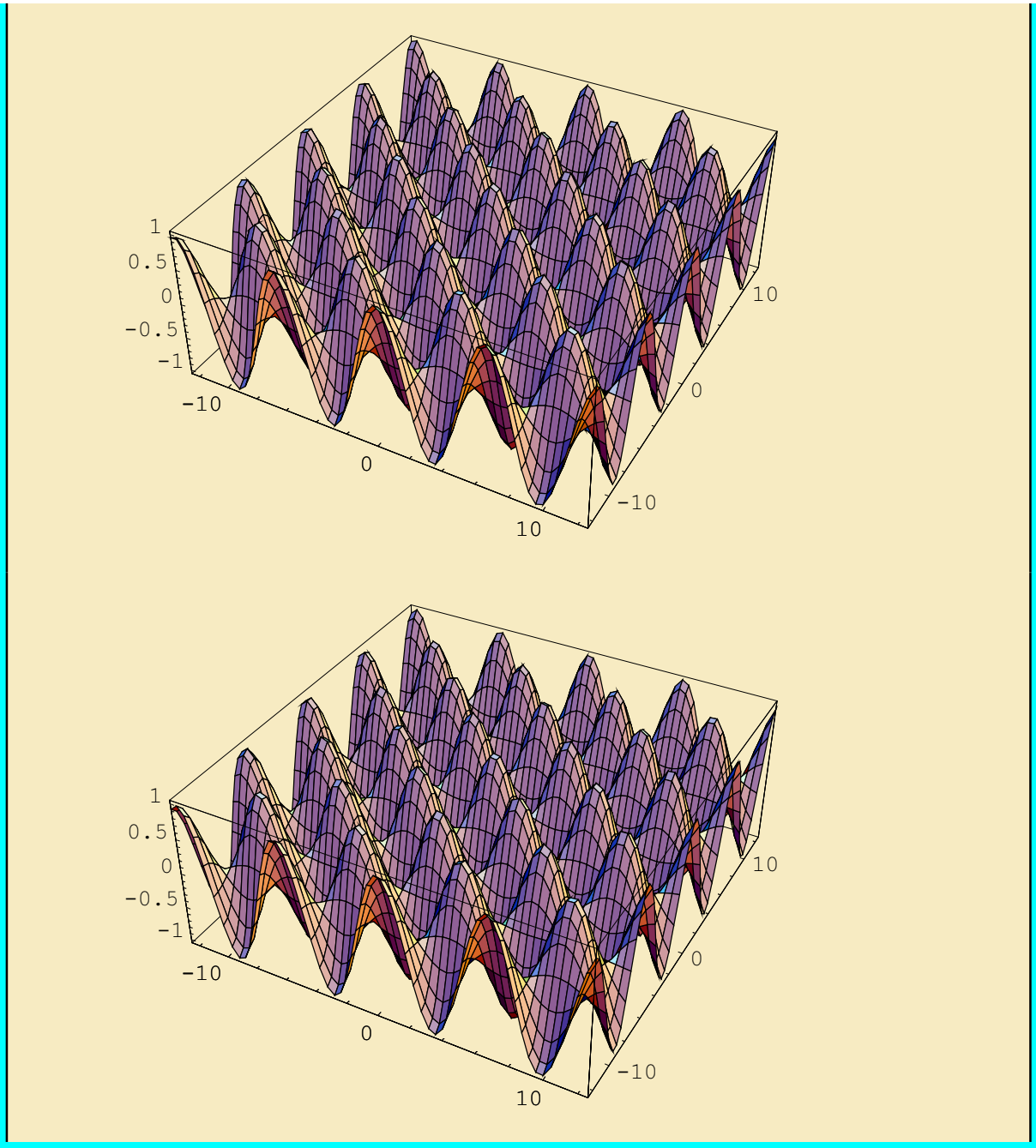

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

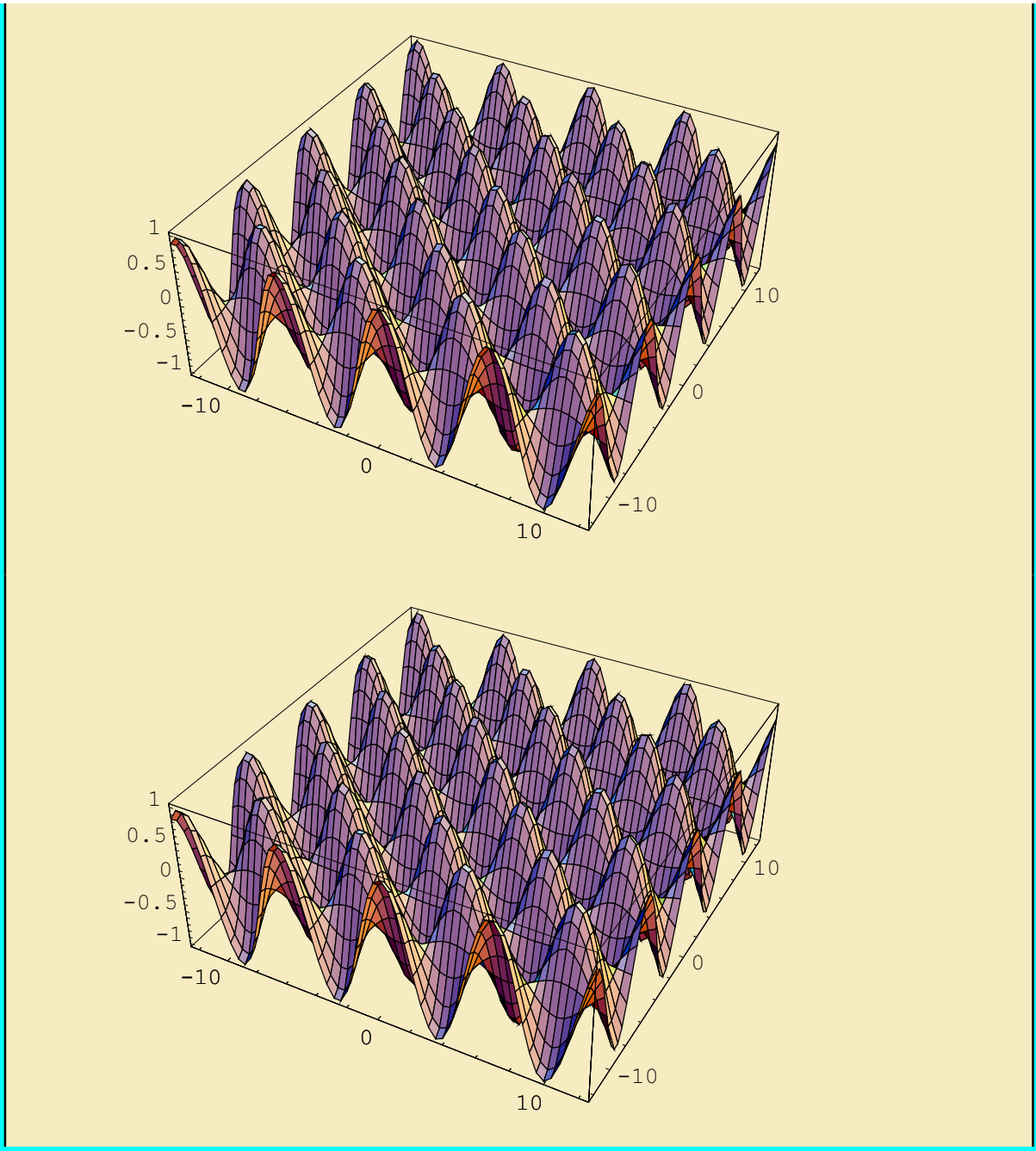

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

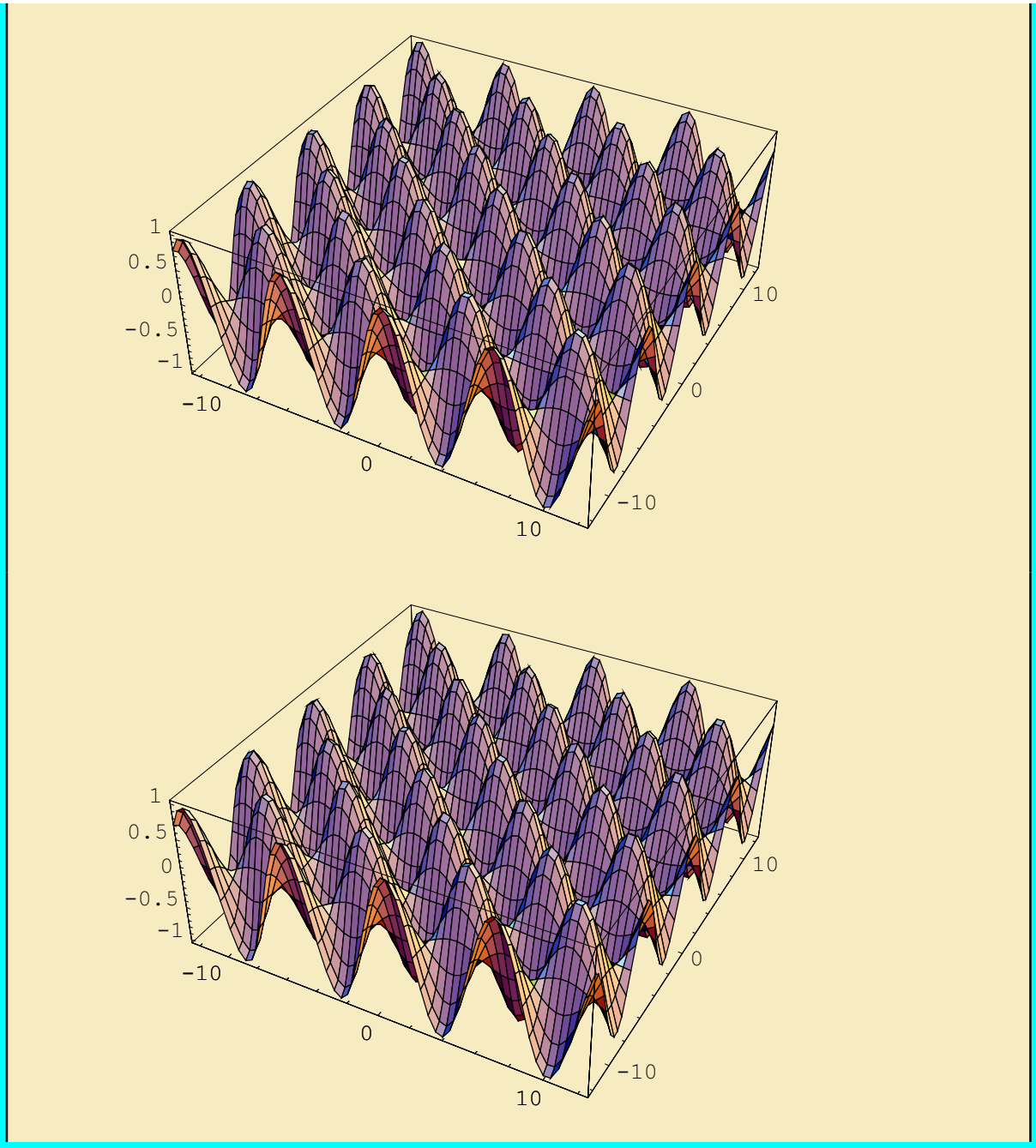

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

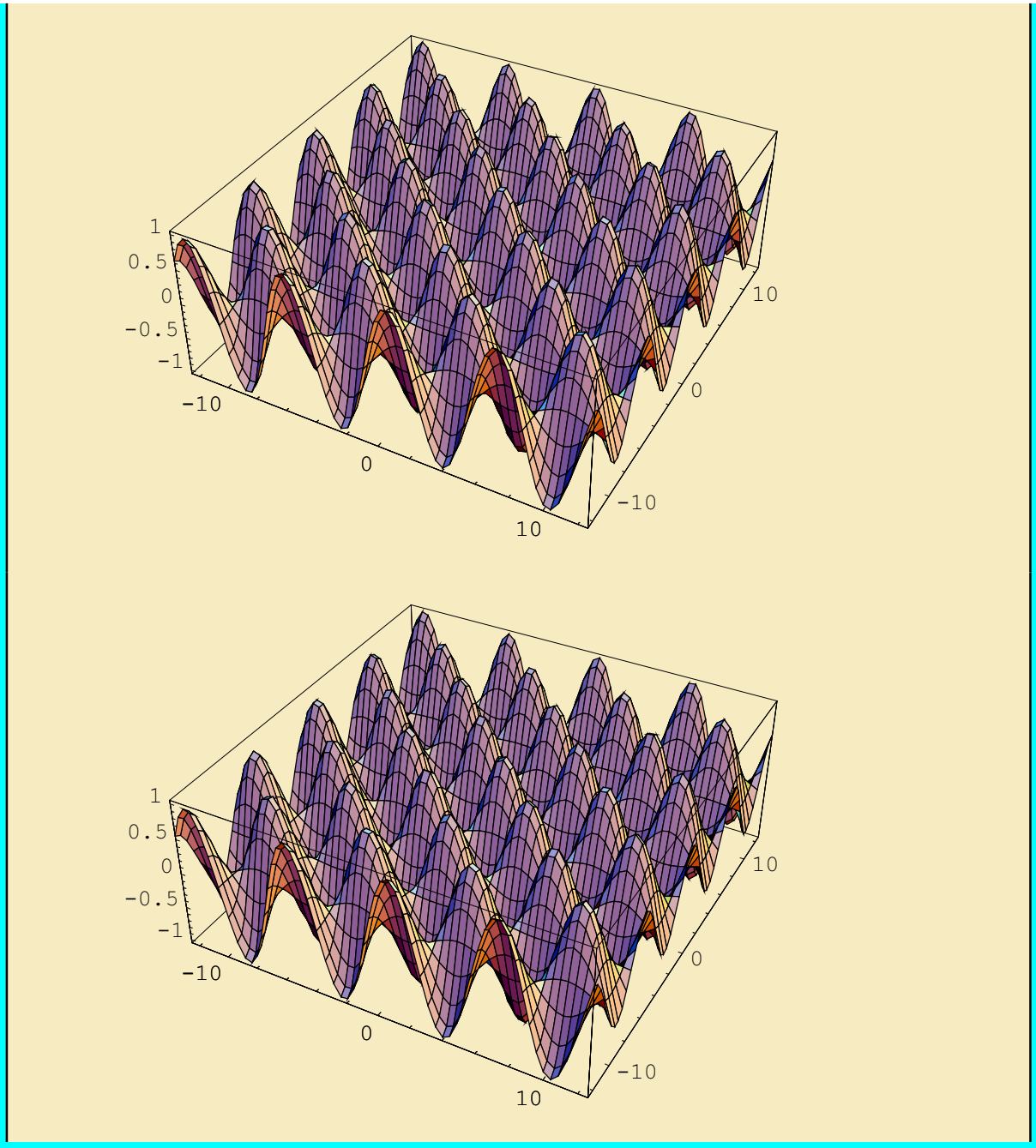

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

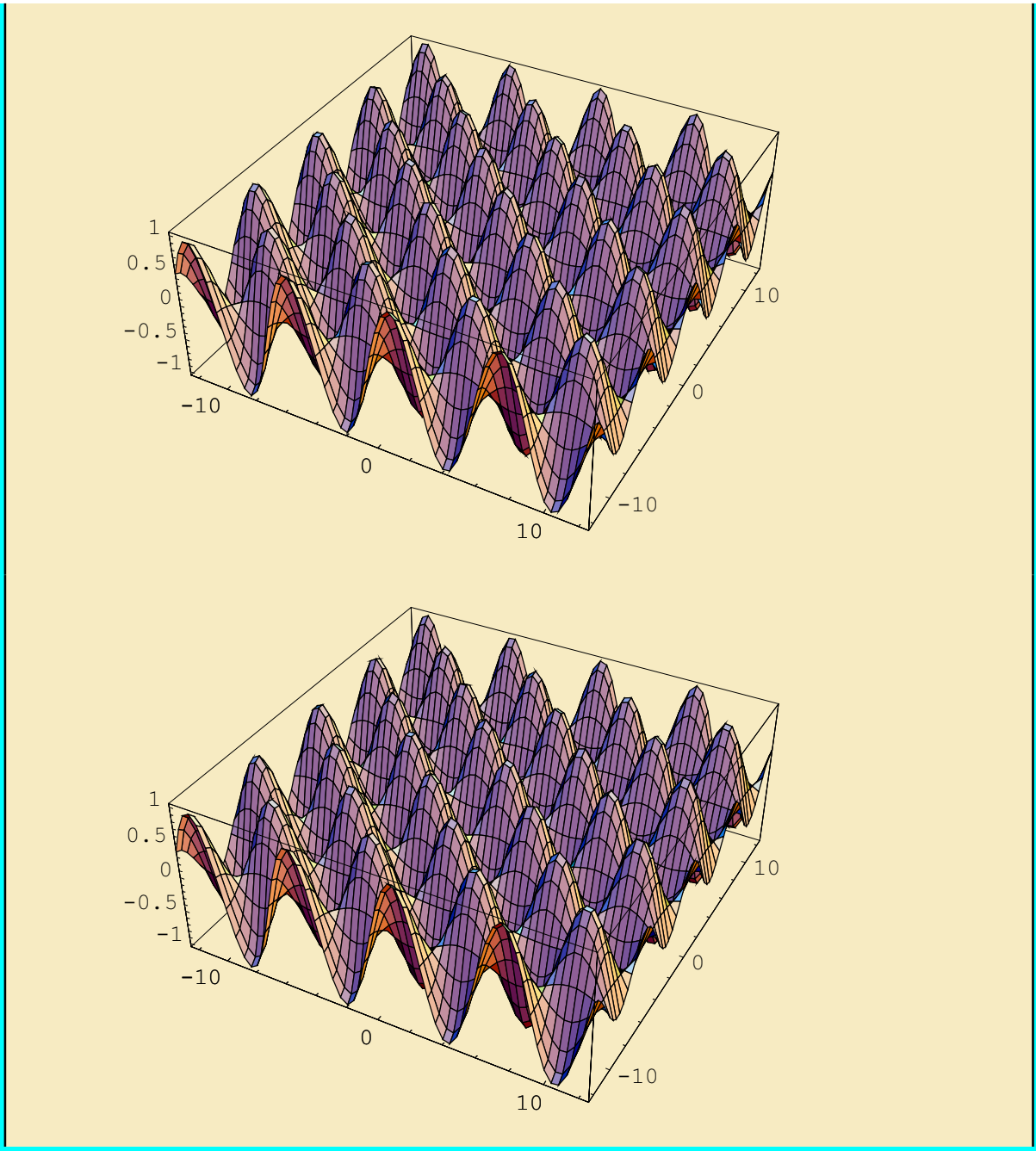

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10

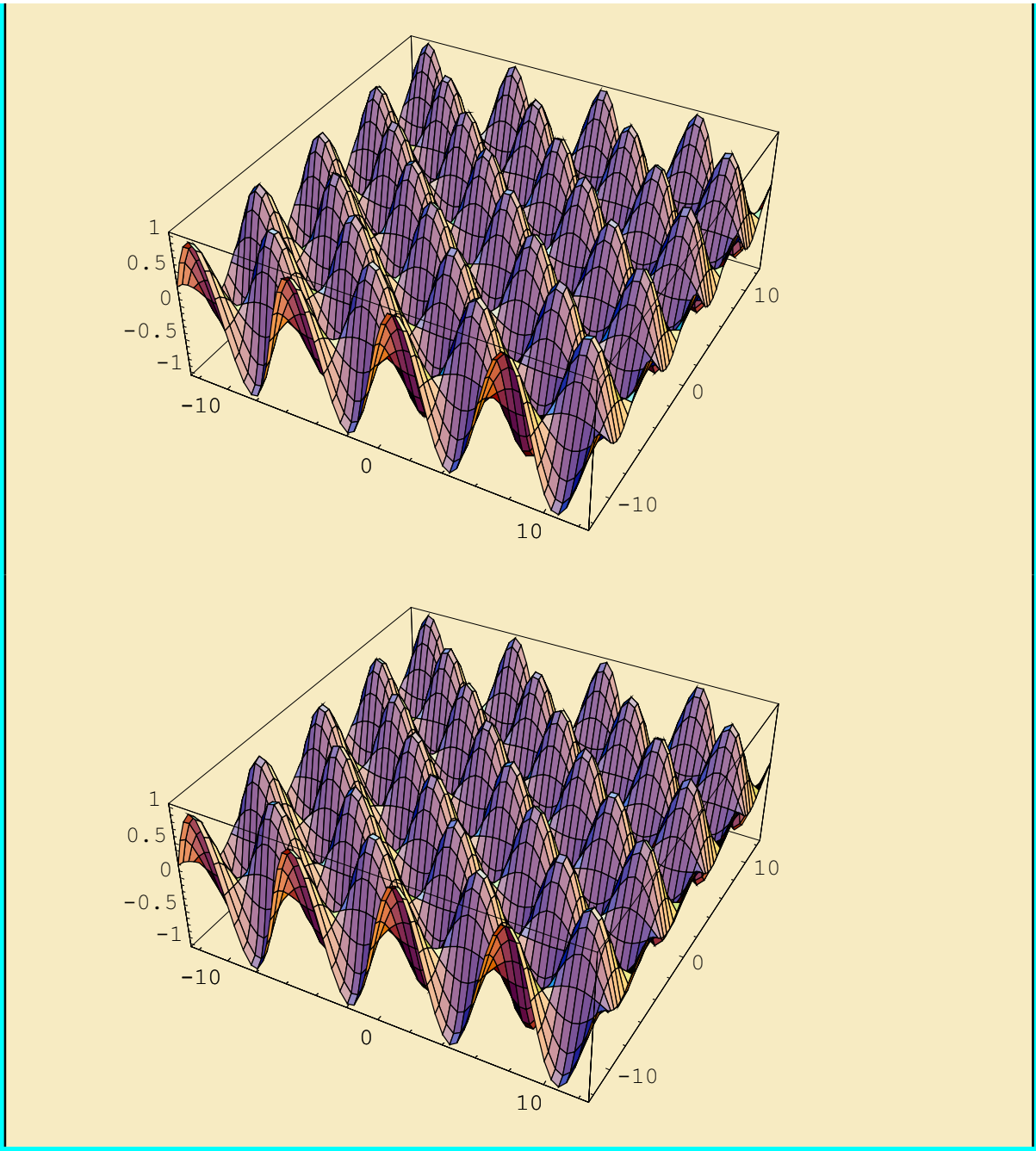

1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10


1
0.5
0
-0.5
-1
-10
0
10
-10
0
10
1
0.5
0
-0.5
-1
-10
0
10
-10
0
10


1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10


1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10


1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10


1
0.5
0
-0.5
-1
-10
0
10
10
0
-10
1
0.5
0
-0.5
-1
-10
0
10
10
0
-10

```
{- SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -}
```

Exercice 2

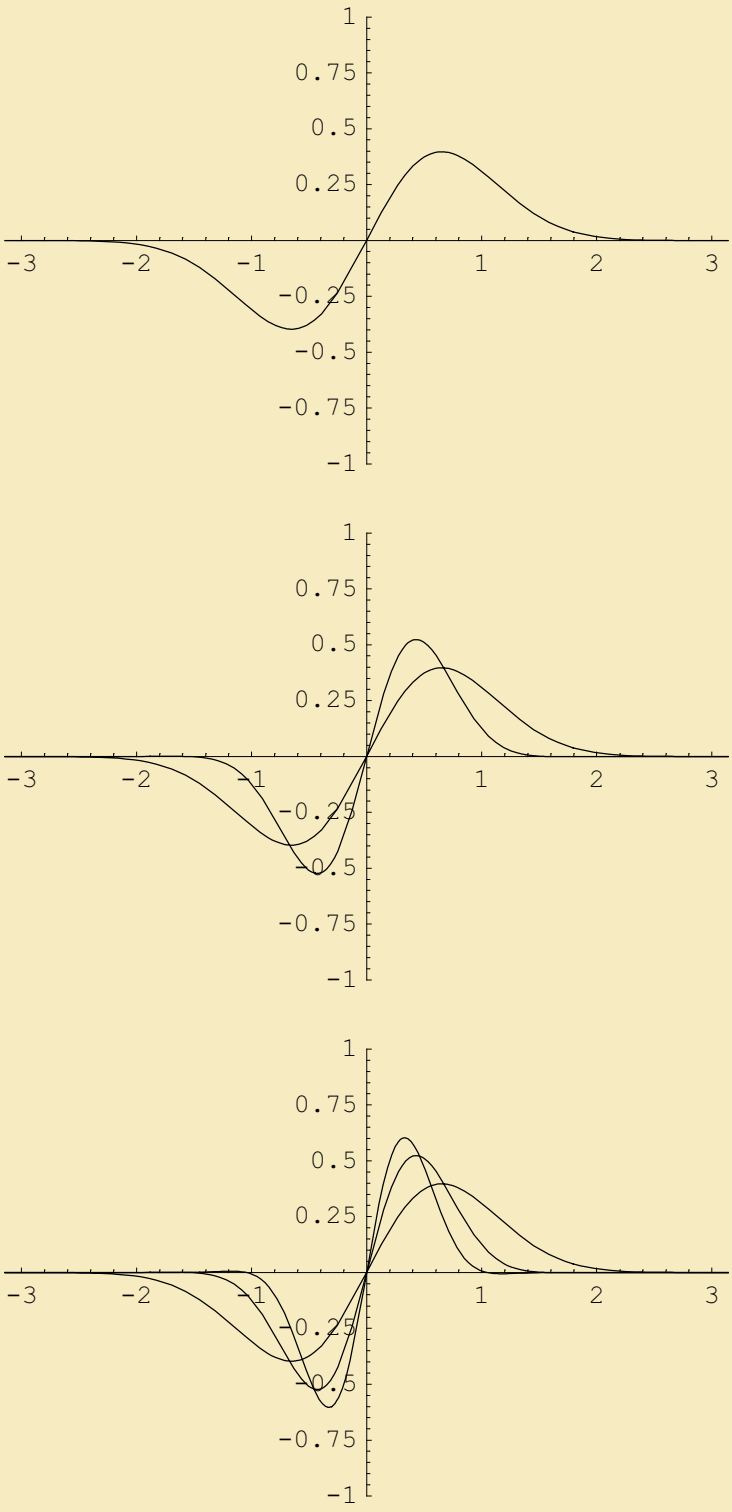
```
Clear[q, u, gg, ggg, x, n, k]
```

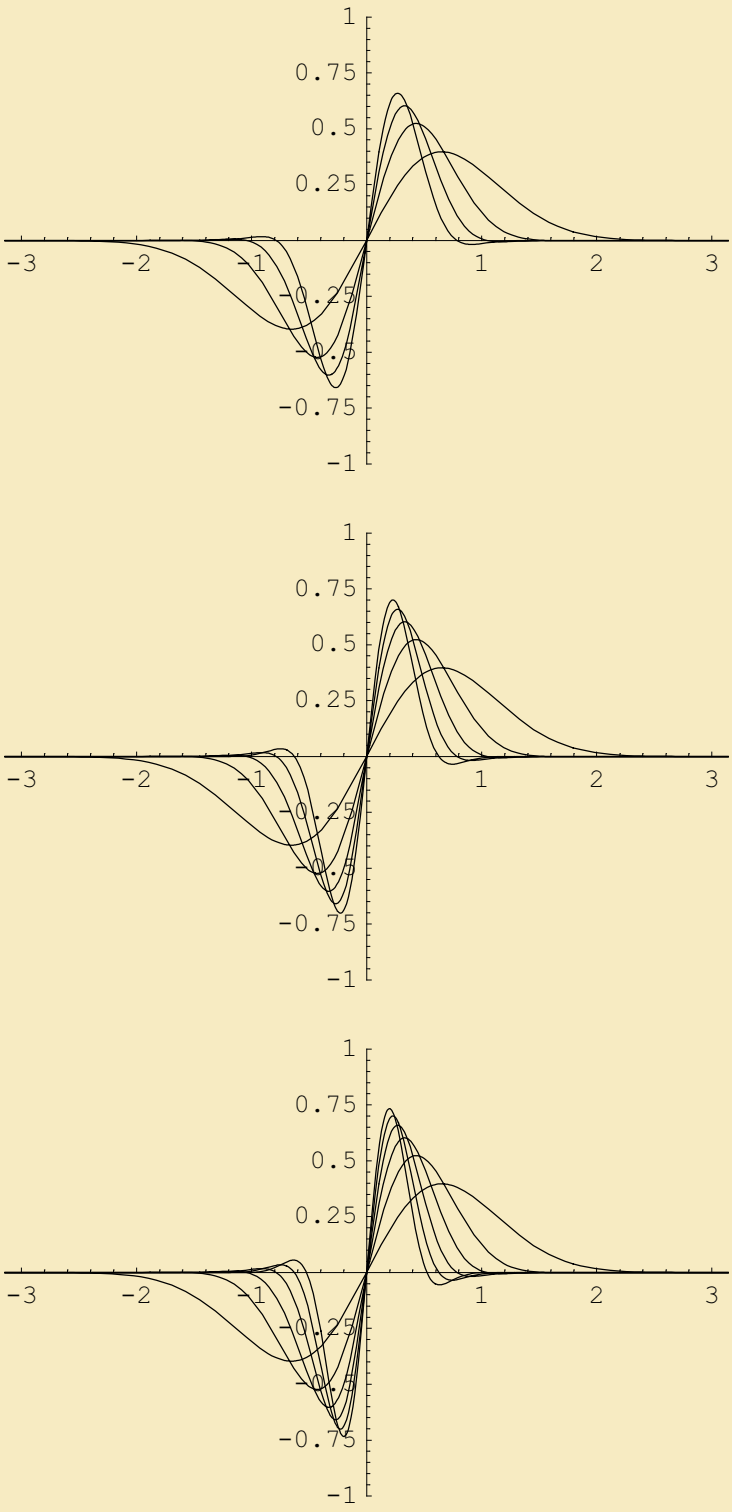
```
u[k_, x_] := Sin[k x] Exp[- k x^2]
```

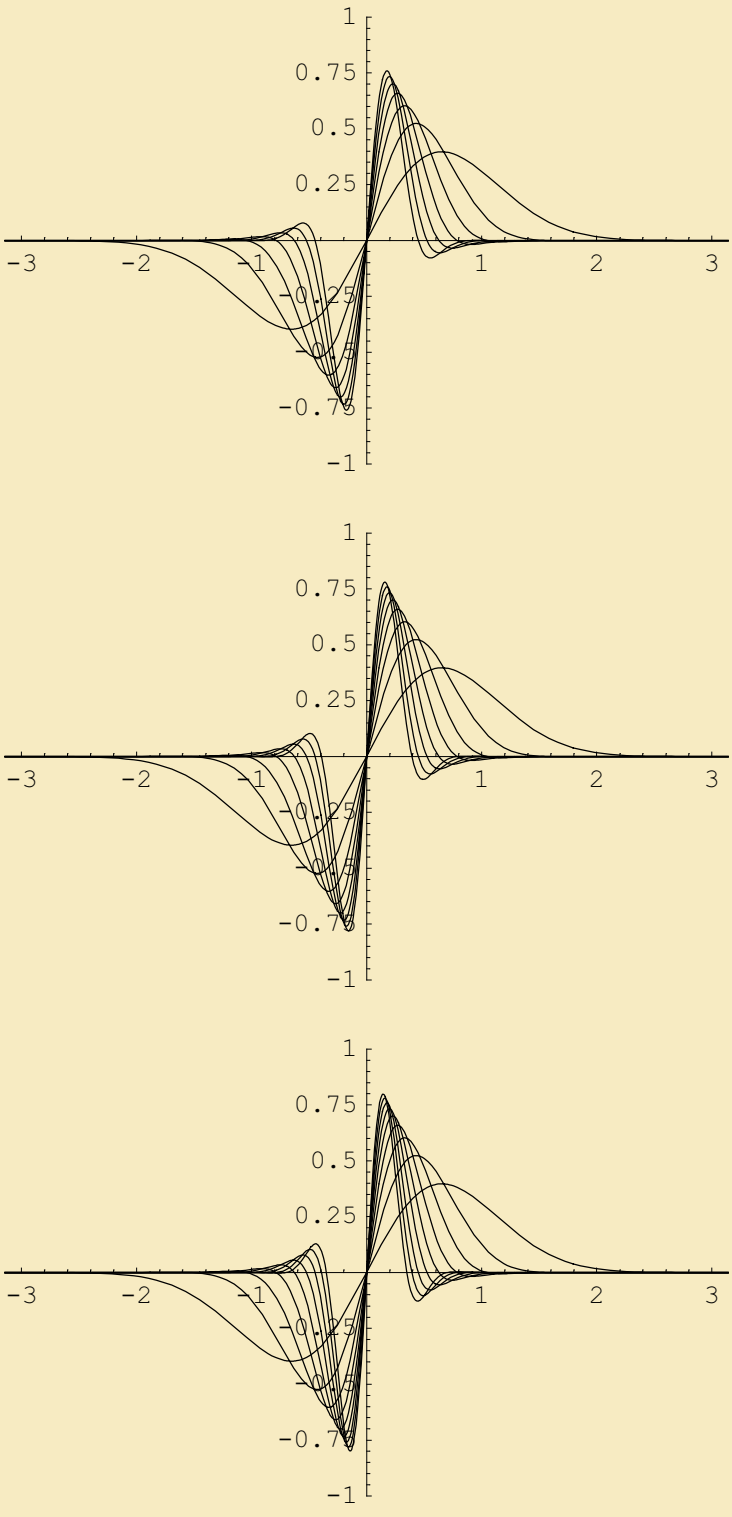
```
gg[n_] := Plot[u[n, x], {x, -Pi, Pi},
  PlotRange → {{-Pi, Pi}, {-1, 1}}]
```

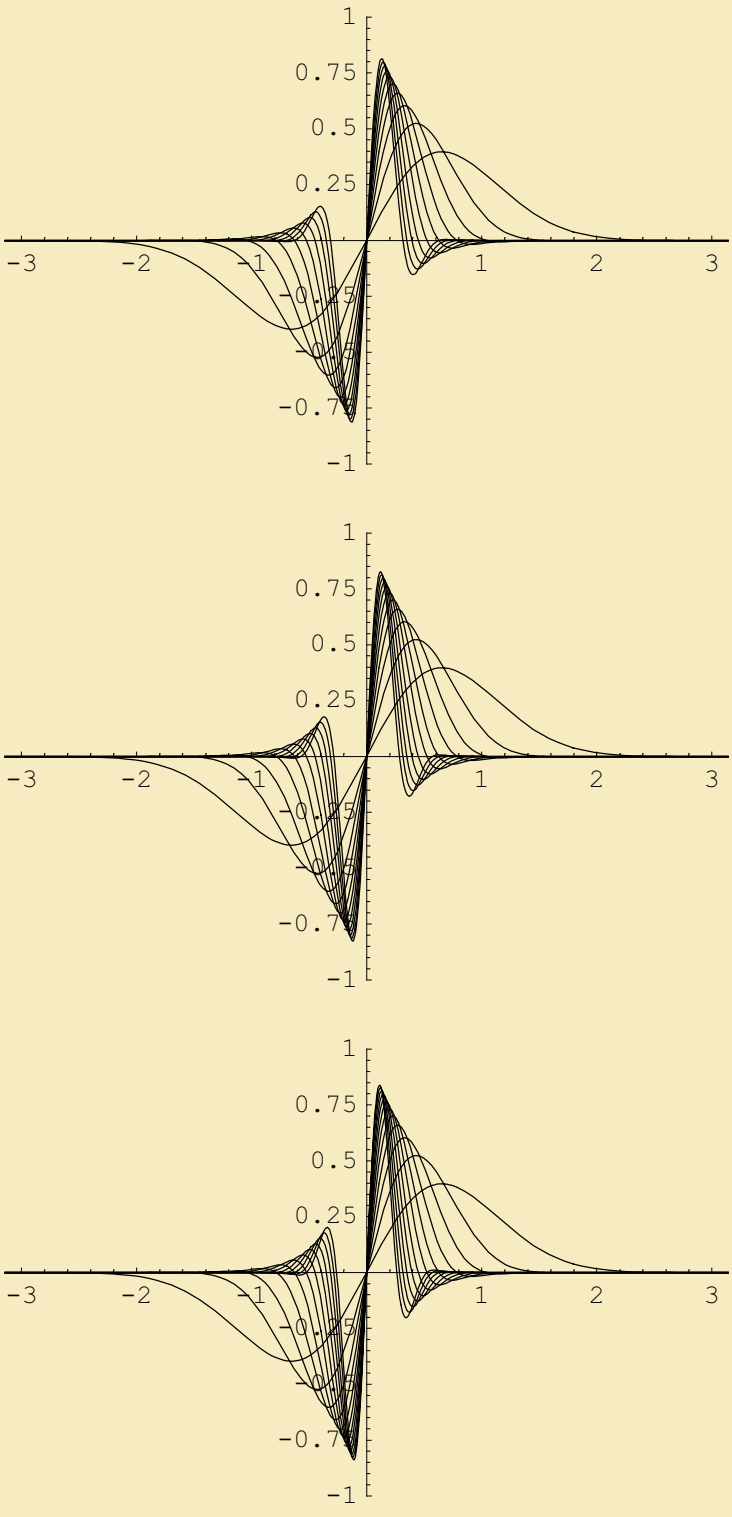
```
ggg[n_] :=
 Plot[Evaluate[Table[N[u[q, x]], {q, 1, n}]],
  {x, -Pi, Pi}, PlotRange → {{-Pi, Pi}, {-1, 1}}]
```

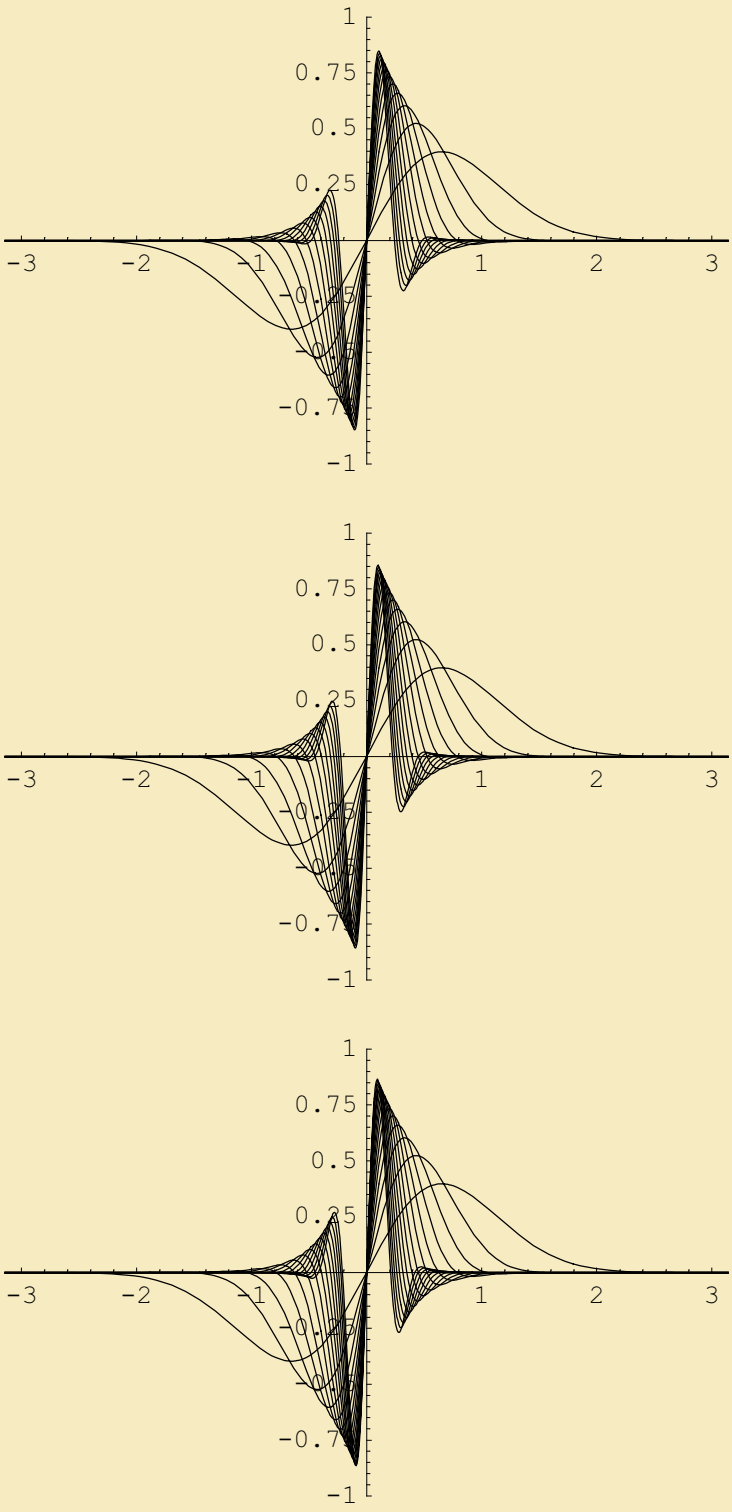
```
Table[ggg[k], {k, 1, 30}]
```

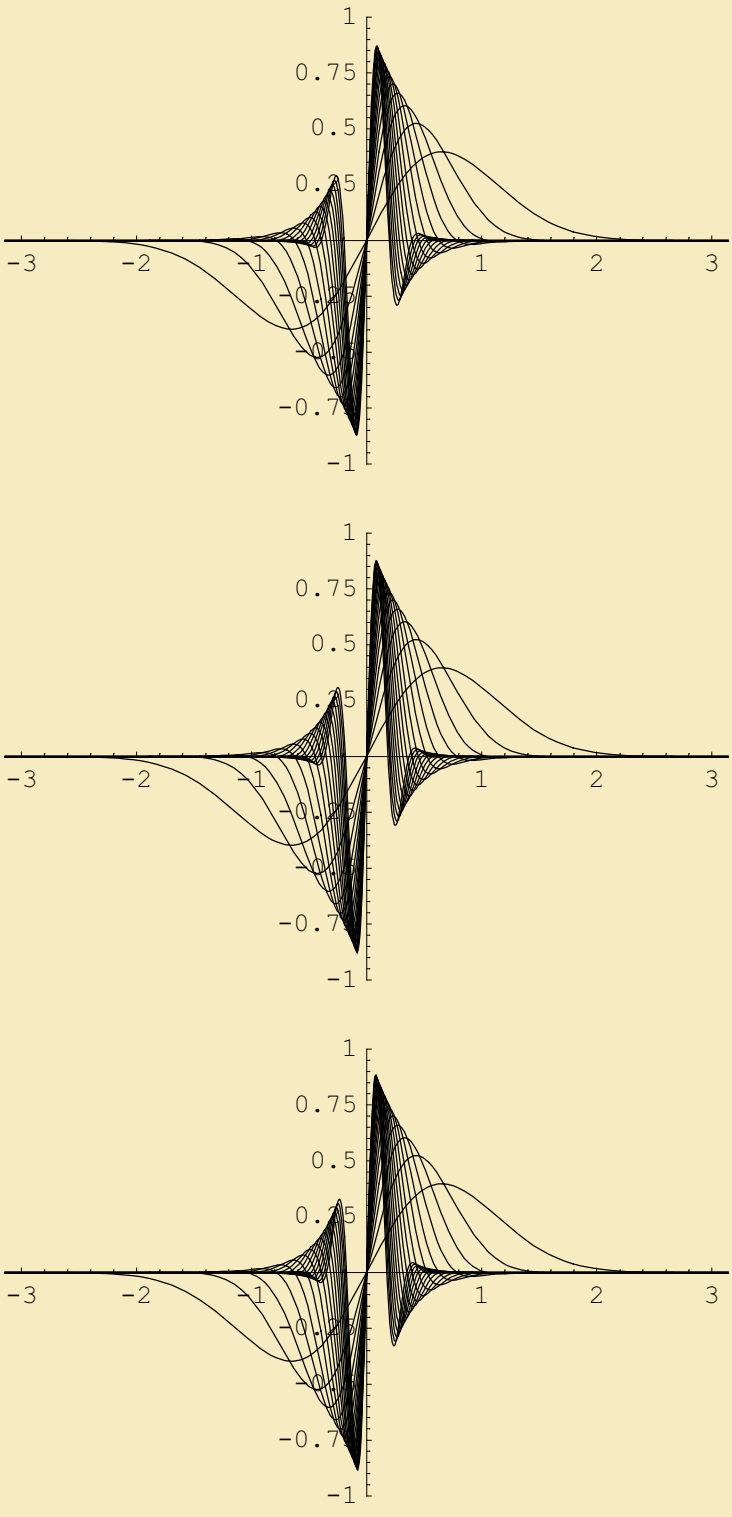

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1

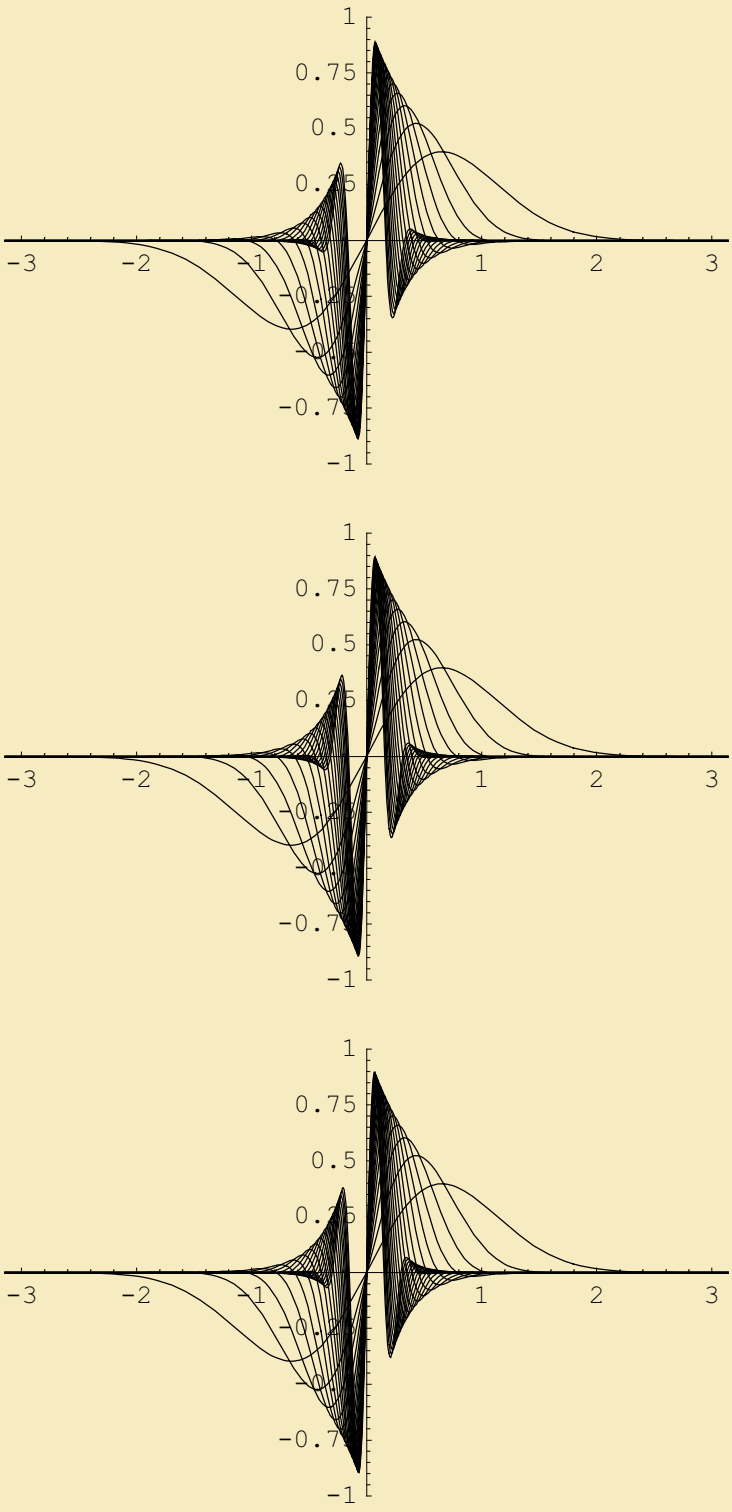

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1

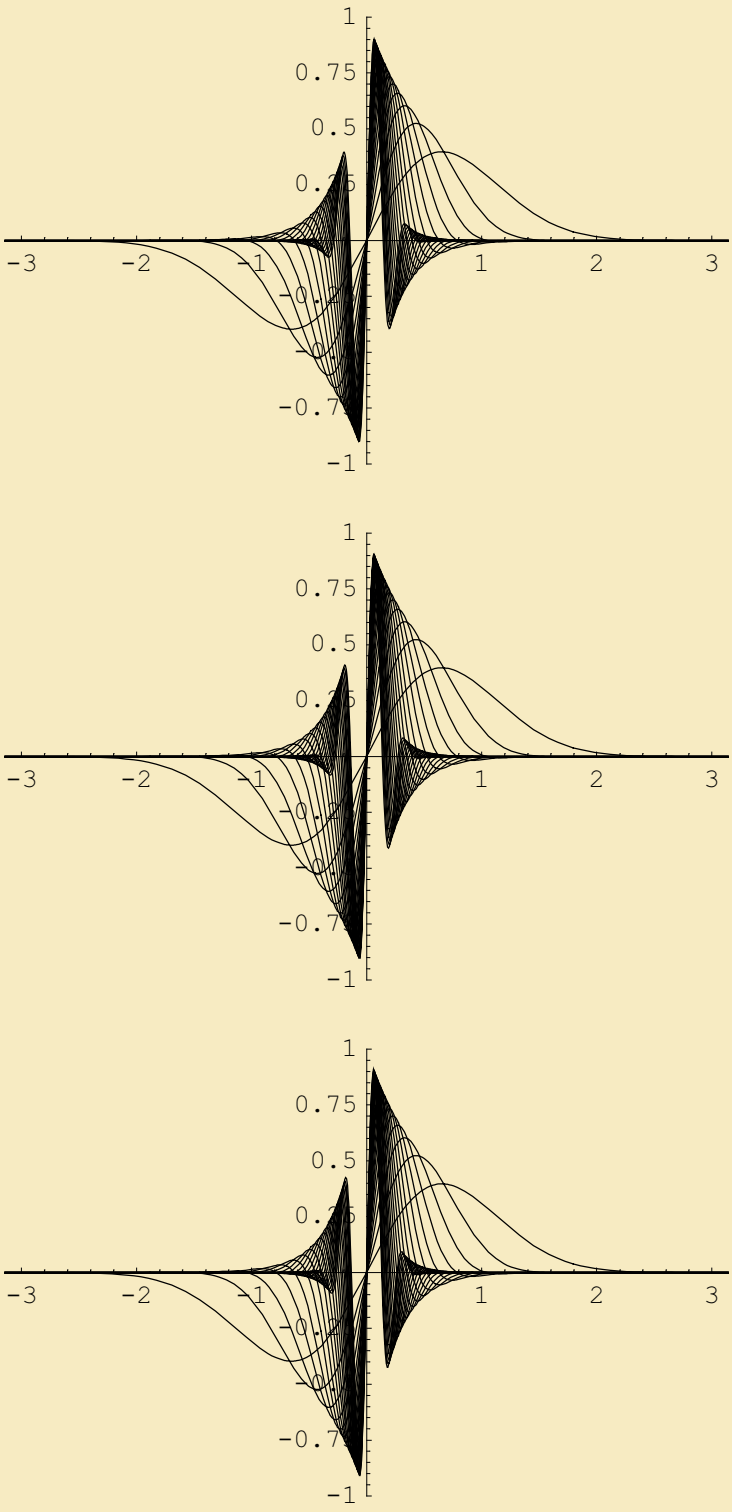

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1

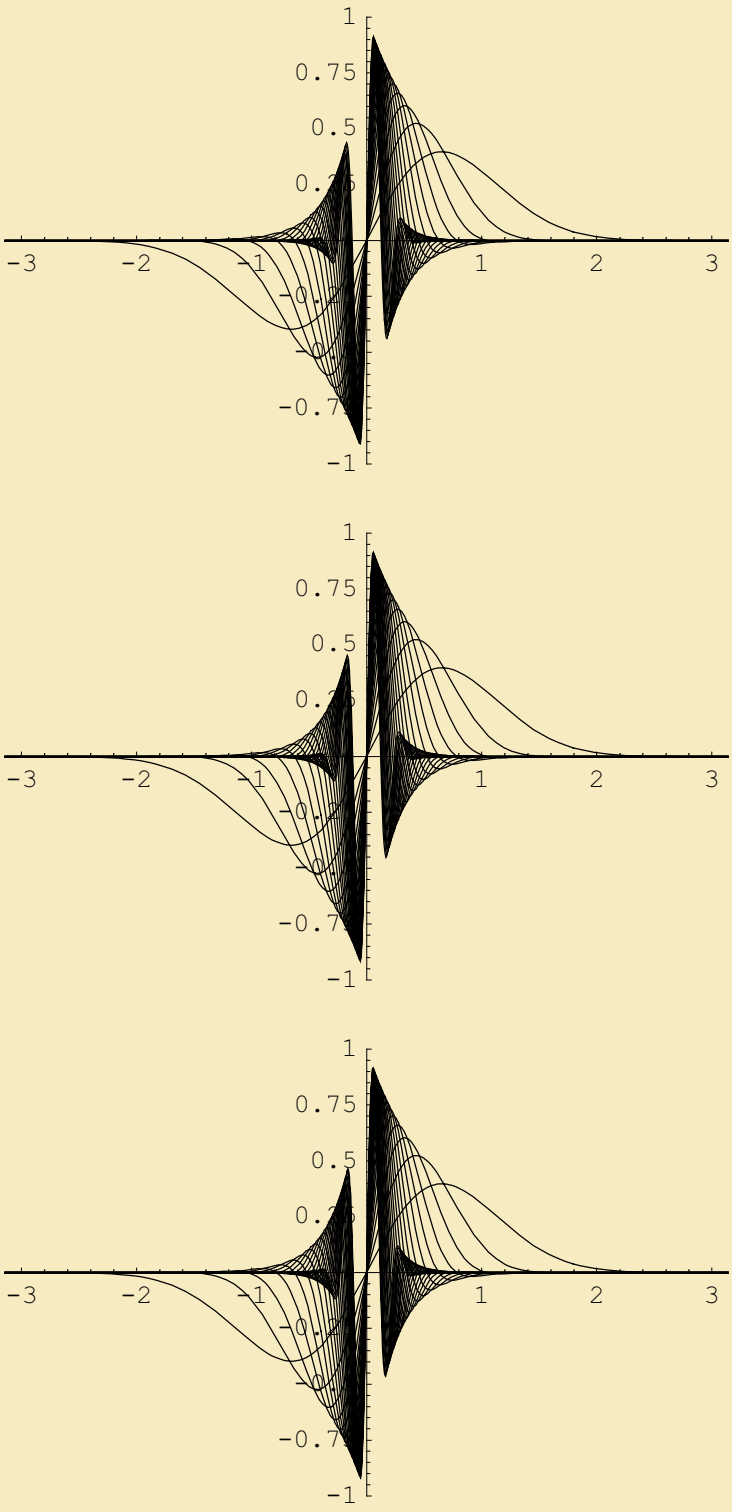

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1

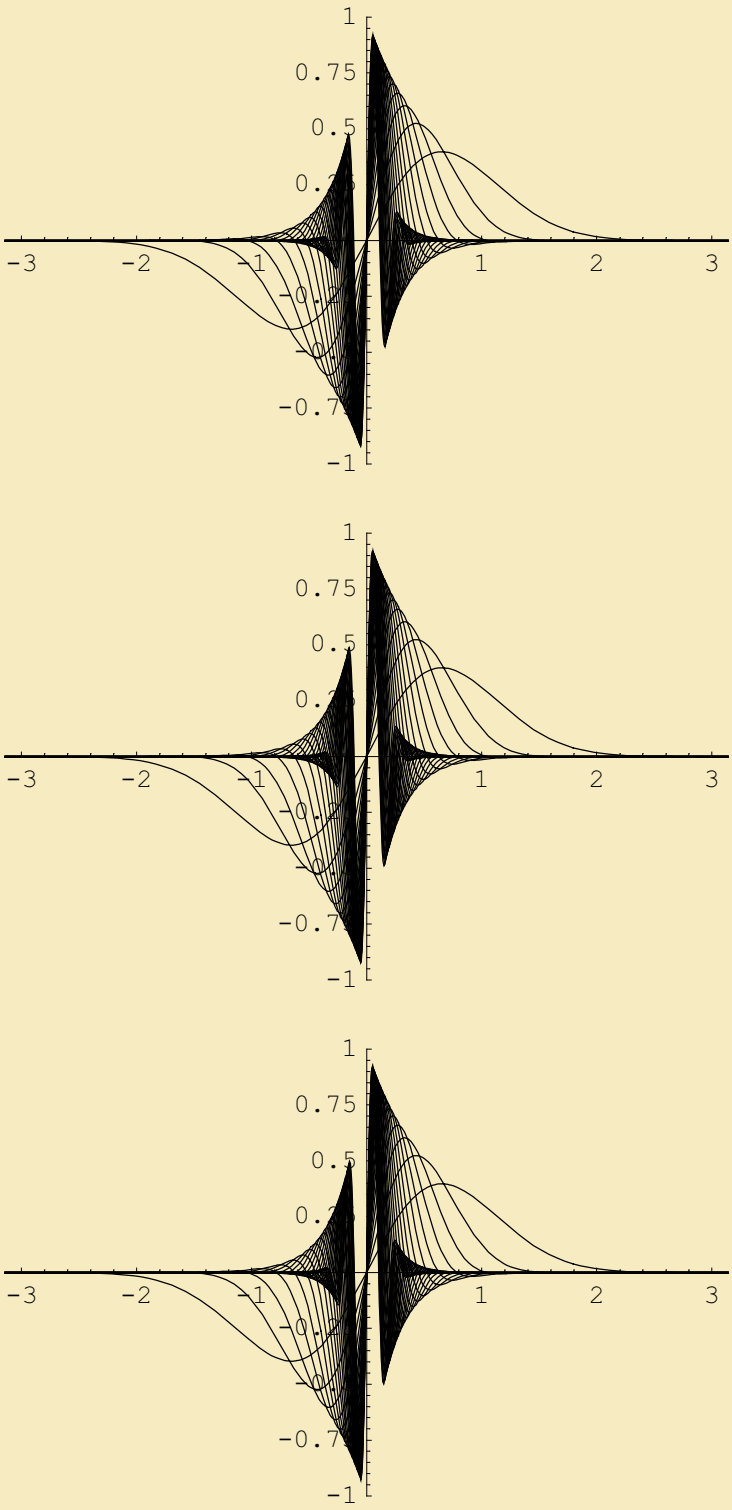

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.75
-1


1
0.75
0.5
-3
-2
-1
1
2
3
-0.7
-1


1
0.75
0.5
-3
-2
-1
1
2
3
-1

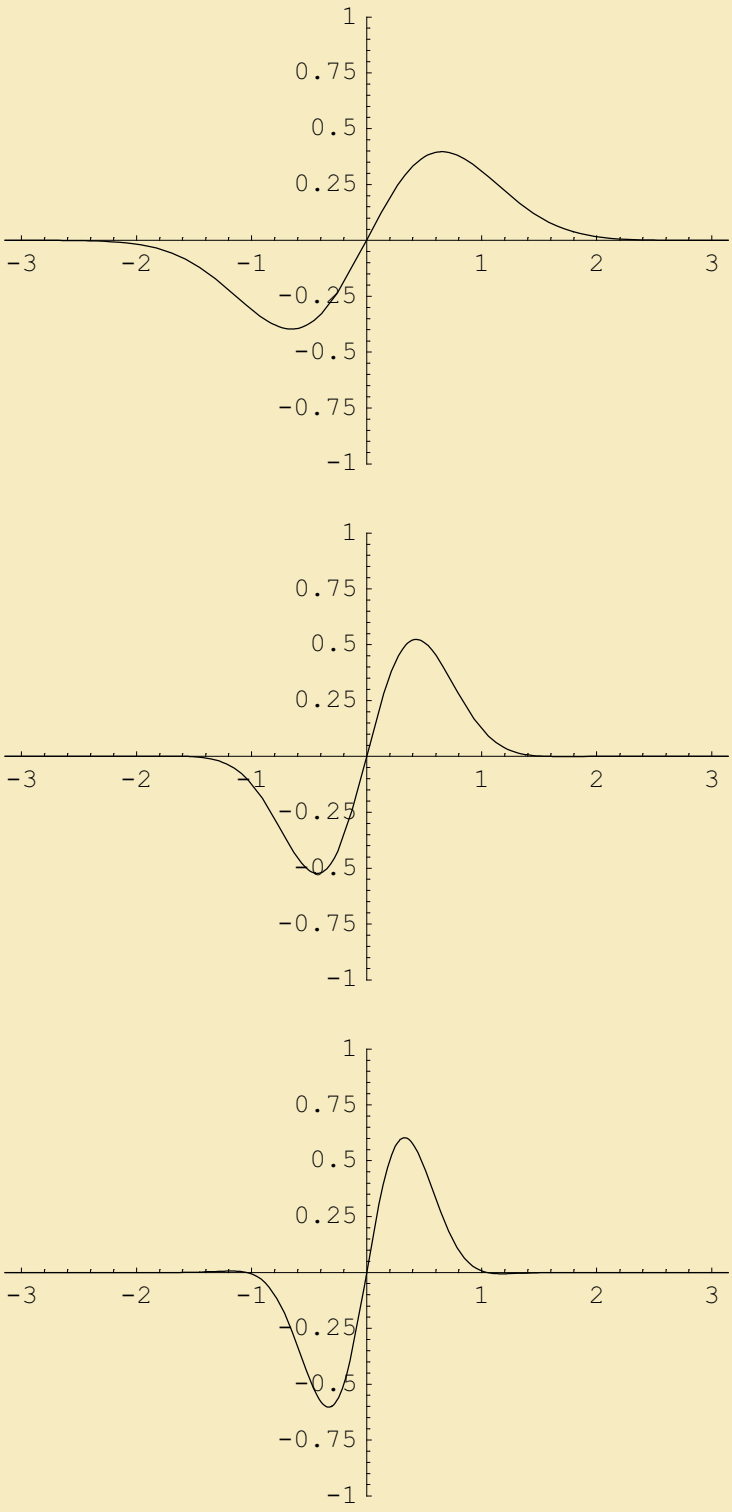

1
0.75
0.5
-3
-2
-1
1
2
3
-1

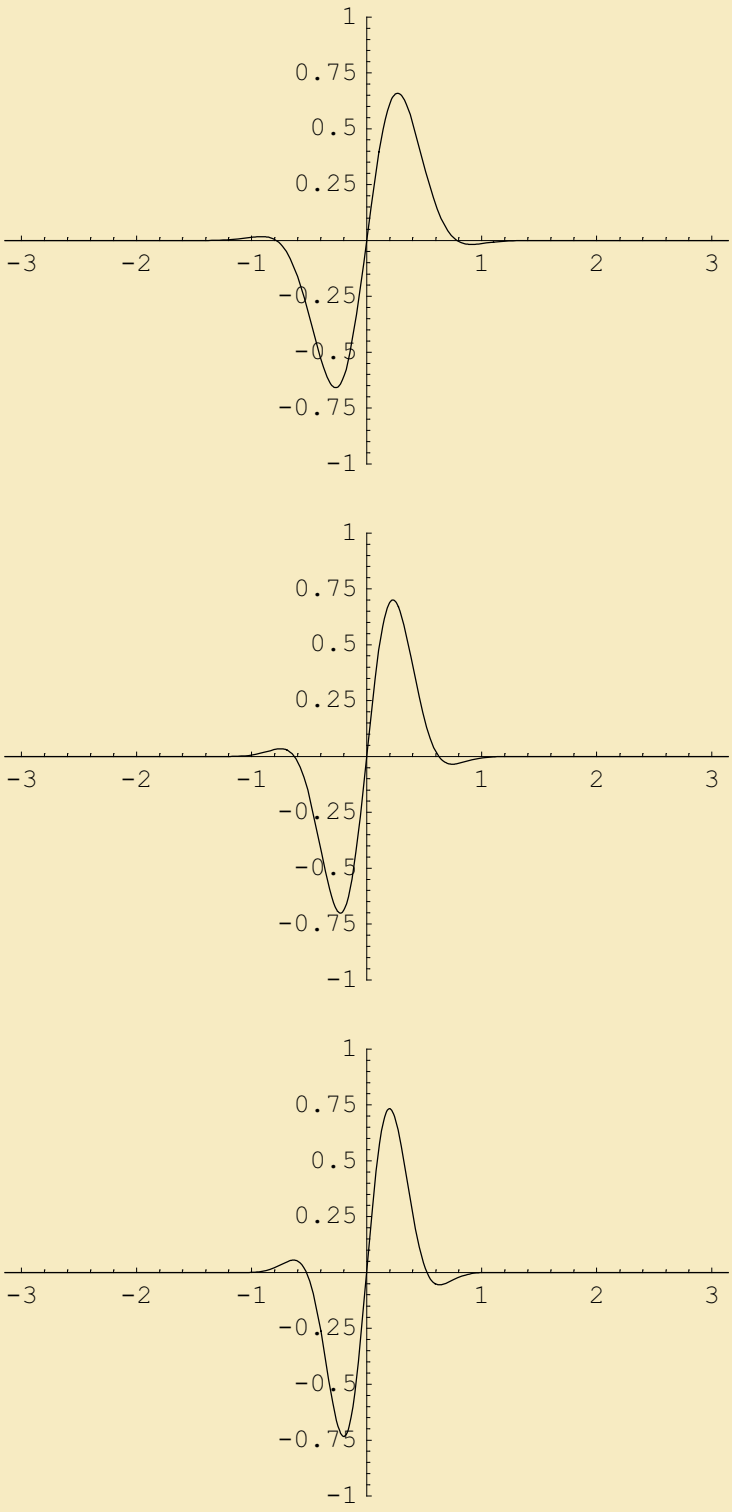

1
0.75
0.5
-3
-2
-1
1
2
3
-1

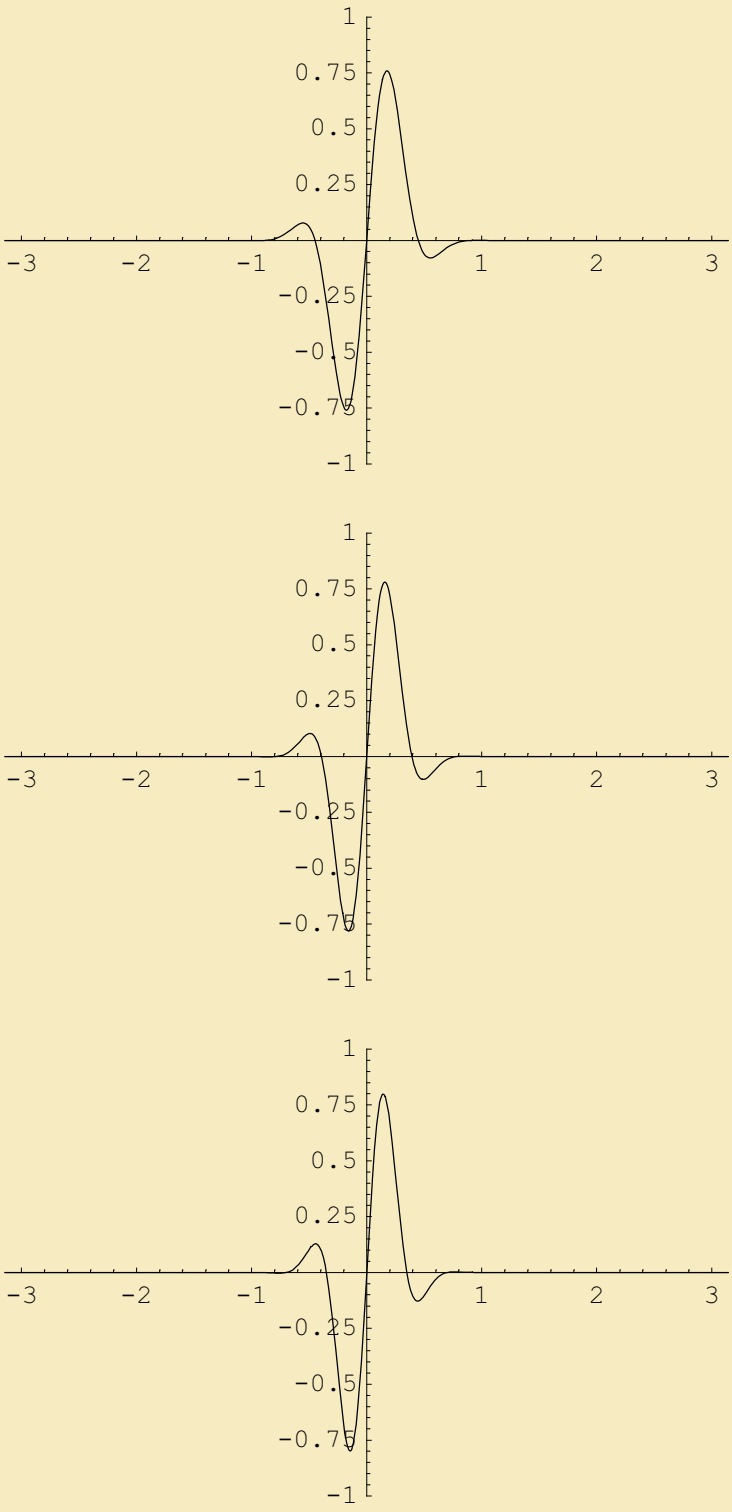


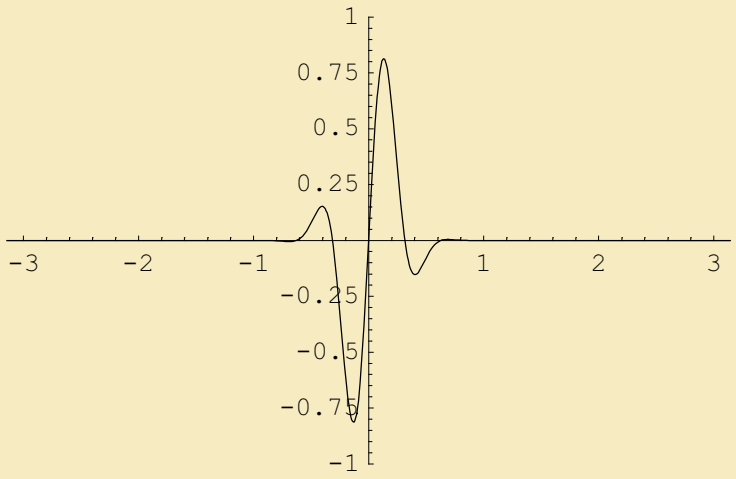

```
{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}
```

```
Table[gg[k], {k, 1, 10}]
```



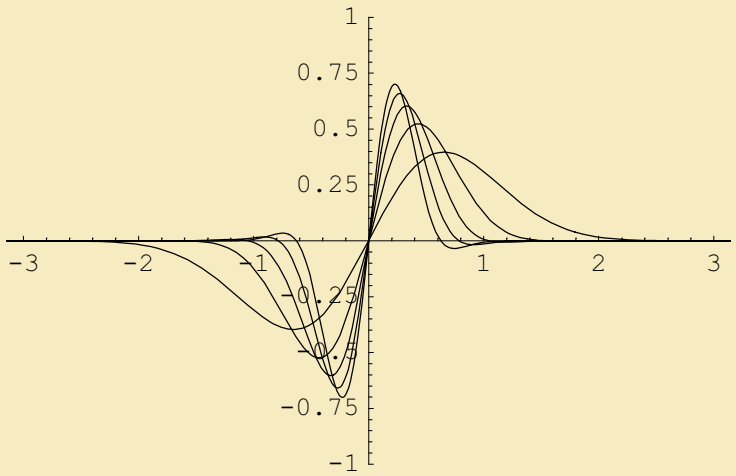

1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1


1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1
1
0.75
0.5
0.25
-3
-2
-1
1
2
3
-0.25
-0.5
-0.75
-1




```
{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}
```

ggg[5]




```
- Graphics -
```

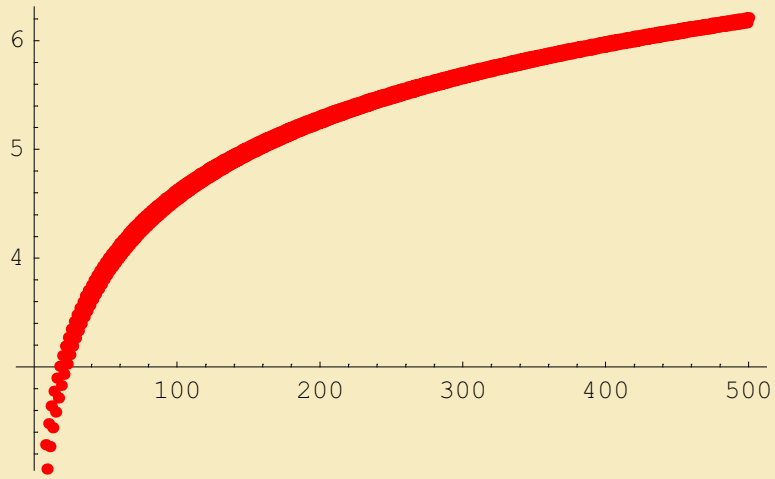
Table[N[u[q, Pi / 4]], {q, 1, 5}]

```
{0.381584, 0.291213, 0.111122, 0., -0.0323602}
```

Exercice 3

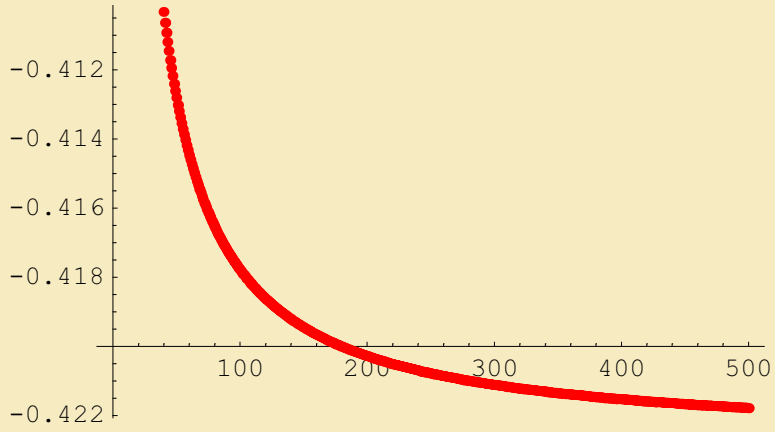
$$\text{ListPlot}\left[\text{Table}\left[\sum_{n=1}^{k}\left(\frac{1}{n}+\frac{(-1)^n}{\sqrt{n}}\right),\ \{k,\ 1,\ 500\}\right],\right.$$

$$\left.\text{PlotStyle} \to \{\text{RGBColor}[1,\ 0,\ 0],\ \text{PointSize}[.015]\}\right]$$



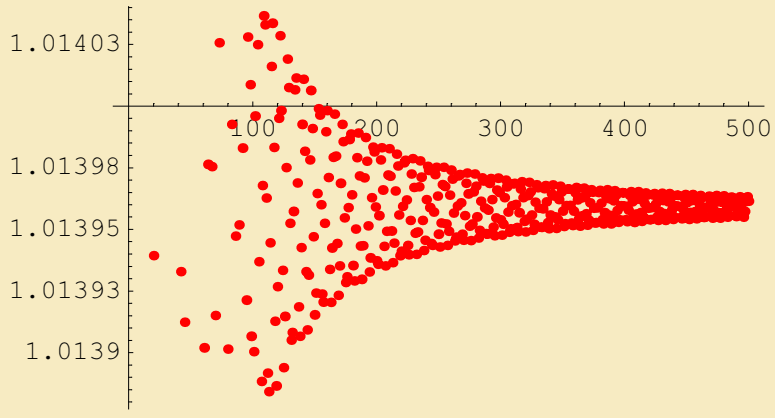
- Graphics -

```
ListPlot[Table[Sum[1/n + Log[(n - 1)/n], {n, 2, k}], {k, 1, 500}],
 PlotStyle → {RGBColor[1, 0, 0], PointSize[.015]}]
```



- Graphics -

```
ListPlot[Table[Sum[Sin[n]/n^2, {n, 1, k}], {k, 1, 500}],
 PlotStyle → {RGBColor[1, 0, 0], PointSize[.015]}]
```



- Graphics -

Exercice 4

```
Clear[m, n, k, p, f, g, h]
```

Au début de chaque exercice, penser à effacer de la mémoire de travail les symboles que l'on va utiliser et qui pourraient avoirété définis précédemment.

■ Création des données, Exportation et importation

```
m[n_] :=
 Table[N[Sin[Binomial[k, p]]], {k, 0, n}, {p, 0, n}]
```

Avant d'exporter des données, toujours vérifier le répertoire

```
Directory[]
```

```
E:\WOLFRA~1\MATHEM~1\4.0
```

```
SetDirectory["F:\\"]
```

```
F:\
```

```
Directory[]
```

```
F:\
```

```
Export["m19.dat", m[19]]
```

```
m19.dat
```

Attention : sous Unix, remplacer les commandes "Import" et "Export" par "get" et "Put".

```
donnees1 := Import["m19.dat"]
```

```
donnees1
```

```
{{0.841471, 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0.}, {0.841471, 0.909297, 0.841471, 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.14112, 0.14112, 0.841471, 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.756802, -0.279415, -0.756802, 0.841471, 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.958924, -0.544021, -0.544021, -0.958924,
  0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.279415, 0.650288, 0.912945, 0.650288, -0.279415,
  0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.656987, 0.836656, -0.428183, -0.428183, 0.836656,
  0.656987, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.989358, 0.270906, -0.521551, 0.773891, -0.521551,
  0.270906, 0.989358, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.412118, -0.991779, 0.73319, 0.329991, 0.329991, 0.73319,
  -0.991779, 0.412118, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.544021, 0.850904, 0.580611, 0.467719, 0.623012,
  0.467719, 0.580611, 0.850904, -0.544021, 0.841471, 0., 0.,
  0., 0., 0., 0., 0., 0., 0.}, {0.841471, -0.99999, -0.999755,
  0.997797, -0.132382, -0.184811, -0.184811, -0.132382, 0.997797,
  -0.999755, -0.99999, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.536573, -0.0265512, 0.0883987, -0.980234,
  0.313286, 0.363256, 0.313286, -0.980234, 0.0883987,
  -0.0265512, -0.536573, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.420167, 0.513978, -0.114815, -0.958907, -0.868906,
  0.636854, 0.636854, -0.868906, -0.958907, -0.114815,
  0.513978, 0.420167, 0.841471, 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.990607, 0.105988, -0.412091, 0.919991, -0.721163,
  -0.354685, 0.982009, -0.354685, -0.721163, 0.919991,
  -0.412091, 0.105988, 0.990607, 0.841471, 0., 0., 0., 0., 0.},
 {0.841471, 0.650288, -0.970535, 0.50634, 0.999758, -0.354685,
  -0.428564, 0.851188, 0.851188, -0.428564, -0.354685, 0.999758,
  0.50634, -0.970535, 0.650288, 0.841471, 0., 0., 0., 0.},
 {0.841471, -0.287903, 0.580611, 0.714919, -0.850983, 0.926954,
  -0.0802396, -0.993995, 0.89351, -0.993995, -0.0802396, 0.926954,
  -0.850983, 0.714919, 0.580611, -0.287903, 0.841471, 0., 0., 0.},
 {0.841471, -0.961397, -0.790433, 0.988041, -0.970484, -0.806097,
  -0.954069, 0.99957, 0.348571, 0.348571, 0.99957, -0.954069, -0.806097,
  -0.970484, 0.988041, -0.790433, -0.961397, 0.841471, 0., 0.},
 {0.841471, -0.750987, 0.806401, -0.727101, 0.0886389,
  -0.768718, -0.323106, -0.327429, 0.947099, 0.65342,
  0.947099, -0.327429, -0.323106, -0.768718, 0.0886389,
  -0.727101, 0.806401, -0.750987, 0.841471, 0.},
 {0.841471, 0.149877, 0.976591, 0.983604, -0.663386, -0.822384, 0.934141,
  0.00457091, 0.999976, 0.50724, 0.50724, 0.999976, 0.00457091, 0.934141,
  -0.822384, -0.663386, 0.983604, 0.976591, 0.149877, 0.841471}}
```

Pour s'assurer que les données soient dans le ficheir Mathematica, et éviter les relecture de fichiers de données externes, copier le ficheir importé dans le clipboard puis dans une cellule définissant le résultat comme les données.

In[1]:=
```
donnees :=
 {{0.8414709848078965`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, 0.8414709848078965`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, 0.9092974268256817`,
   0.8414709848078965`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, 0.1411200080598672`,
   0.1411200080598672`, 0.8414709848078965`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, -0.7568024953079282`,
   -0.27941549819892586`, -0.7568024953079282`,
   0.8414709848078965`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, -0.9589242746631385`,
   -0.5440211108893698`, -0.5440211108893698`,
   -0.9589242746631385`, 0.8414709848078965`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, -0.27941549819892586`,
   0.6502878401571168`, 0.9129452507276277`,
   0.6502878401571168`, -0.27941549819892586`,
   0.8414709848078965`, 0.`, 0.`, 0.`, 0.`,
   0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
  {0.8414709848078965`, 0.6569865987187891`,
```

```
 0.8366556385360561`, -0.428182669496151`,
 -0.428182669496151`, 0.8366556385360561`,
 0.6569865987187891`, 0.8414709848078965`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`,
 0.`, 0.`, 0.`, 0.`}, {0.8414709848078965`,
 0.9893582466233818`, 0.27090578830786904`,
 -0.5215510020869119`, 0.7738906815578891`,
 -0.5215510020869119`, 0.27090578830786904`,
 0.9893582466233818`, 0.8414709848078965`, 0.`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, 0.4121184852417566`,
 -0.9917788534431158`, 0.7331903200732922`,
 0.329990825673782`, 0.329990825673782`,
 0.7331903200732922`, -0.9917788534431158`,
 0.4121184852417566`, 0.8414709848078965`, 0.`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, -0.5440211108893698`,
 0.8509035245341184`, 0.5806111842123143`,
 0.46771851834275896`, 0.623012211003653`,
 0.46771851834275896`, 0.5806111842123143`,
 0.8509035245341184`, -0.5440211108893698`,
 0.8414709848078965`, 0.`, 0.`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, -0.9999902065507035`,
 -0.9997551733586199`, 0.9977972794498907`,
 -0.13238162920545193`, -0.18481136973170906`,
 -0.18481136973170906`, -0.13238162920545193`,
 0.9977972794498907`, -0.9997551733586199`,
 -0.9999902065507035`, 0.8414709848078965`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, -0.5365729180004349`,
 -0.02655115402396679 4`,
 0.08839871248753149`, -0.9802336960326743`,
 0.3132860367040136`, 0.363255622392 4401`,
```

```
 0.3132860367040136`, -0.9802336960326743`,
 0.08839871248753149`, -0.02655115402396679 4`,
 -0.5365729180004349`, 0.8414709848078965`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, 0.4201670368266409`,
 0.5139784559875352`, -0.11481475884166603`,
 -0.9589071712941251`, -0.8689060794851441`,
 0.6368542184798601`, 0.6368542184798601`,
 -0.8689060794851441`, -0.9589071712941251`,
 -0.11481475884166603`, 0.5139784559875352`,
 0.4201670368266409`, 0.8414709848078965`,
 0.`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, 0.9906073556948704`,
 0.10598751175115685`, -0.41209101962194344`,
 0.9199905975863218`, -0.7211630200865586`,
 -0.35468470978178057`, 0.9820091362699451`,
 -0.35468470978178057`, -0.7211630200865586`,
 0.9199905975863218`, -0.41209101962194344`,
 0.10598751175115685`, 0.9906073556948704`,
 0.8414709848078965`, 0.`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, 0.6502878401571168`,
 -0.9705352835374847`, 0.5063396468349013`,
 0.999757834077975`, -0.35468470978178057`,
 -0.42856400839508124`, 0.8511884381115083`,
 0.8511884381115083`, -0.42856400839508124`,
 -0.35468470978178057`,
 0.999757834077975`, 0.5063396468349013`,
 -0.9705352835374847`, 0.6502878401571168`,
 0.8414709848078965`, 0.`, 0.`, 0.`, 0.`},
{0.8414709848078965`, -0.2879033166650653`,
 0.5806111842123143`, 0.7149185852567496`,
 -0.8509826923271134`, 0.9269542787057038`,
 -0.0802396466914345`, -0.9939946731716932`,
 0.8935098892685653`, -0.9939946731716932`,
```

```
 -0.0802396466914345`, 0.9269542787057038`,
 -0.8509826923271134`, 0.7149185852567496`,
 0.5806111842123143`, -0.2879033166650653`,
 0.8414709848078965`, 0.`, 0.`, 0.`},
{0.8414709848078965`, -0.9613974918795568`,
 -0.7904332067228887`, 0.9880409219176677`,
 -0.9704844170480315`, -0.8060974251403136`,
 -0.9540692627477192`, 0.9995701395620273`,
 0.34857144256216915`, 0.34857144256216915`,
 0.9995701395620273`, -0.9540692627477192`,
 -0.8060974251403136`, -0.9704844170480315`,
 0.9880409219176677`, -0.7904332067228887`,
 -0.9613974918795568`, 0.8414709848078965`,
 0.`, 0.`}, {0.8414709848078965`,
 -0.750987246771676`, 0.8064005807754863`,
 -0.7271011113730546`, 0.08863892066173254`,
 -0.7687175819228536`, -0.3231062290724931`,
 -0.3274285931649403`, 0.9470986777407293`,
 0.6534196271223813`, 0.9470986777407293`,
 -0.3274285931649403`, -0.3231062290724931`,
 -0.7687175819228536`, 0.08863892066173254`,
 -0.7271011113730546`, 0.8064005807754863`,
 -0.750987246771676`, 0.8414709848078965`, 0.`},
{0.8414709848078965`, 0.14987720966295234`,
 0.9765908679435658`, 0.9836040582900275`,
 -0.6633858009184382`, -0.8223841968488909`,
 0.9341406337390213`, 0.004570910776757183`,
 0.9999764950498864`, 0.5072396797589778`,
 0.5072396797589778`, 0.9999764950498864`,
 0.004570910776757183`, 0.9341406337390213`,
 -0.8223841968488909`, -0.6633858009184382`,
 0.9836040582900275`, 0.9765908679435658`,
 0.14987720966295234`, 0.8414709848078965`}}
```

■ Création des fonctions d'indexation, f, g et h

In[2]:=
```
f1[i_, j_] := Flatten[donnees][[10 (i - 1) + j]]
```

In[3]:=
```
Table[f1[i, j], {i, 1, 40}, {j, 1, 10}]
```

Out[3]=
```
{{0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.909297, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.14112, 0.14112, 0.841471, 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, -0.756802,
  -0.279415, -0.756802, 0.841471, 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, -0.958924,
  -0.544021, -0.544021, -0.958924, 0.841471, 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, -0.279415,
  0.650288, 0.912945, 0.650288, -0.279415, 0.841471, 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, 0.656987, 0.836656,
  -0.428183, -0.428183, 0.836656, 0.656987, 0.841471, 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, 0.989358, 0.270906,
  -0.521551, 0.773891, -0.521551, 0.270906, 0.989358, 0.841471, 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, 0.412118, -0.991779,
  0.73319, 0.329991, 0.329991, 0.73319, -0.991779, 0.412118, 0.841471},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}, {0.841471, -0.544021, 0.850904,
  0.580611, 0.467719, 0.623012, 0.467719, 0.580611, 0.850904, -0.544021},
 {0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.99999, -0.999755, 0.997797, -0.132382,
  -0.184811, -0.184811, -0.132382, 0.997797, -0.999755},
 {-0.99999, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, -0.536573, -0.0265512, 0.0883987, -0.980234,
  0.313286, 0.363256, 0.313286, -0.980234, 0.0883987},
 {-0.0265512, -0.536573, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.420167, 0.513978, -0.114815, -0.958907,
  -0.868906, 0.636854, 0.636854, -0.868906, -0.958907},
 {-0.114815, 0.513978, 0.420167, 0.841471, 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.990607, 0.105988, -0.412091, 0.919991,
  -0.721163, -0.354685, 0.982009, -0.354685, -0.721163},
 {0.919991, -0.412091, 0.105988, 0.990607, 0.841471, 0., 0., 0., 0., 0.},
 {0.841471, 0.650288, -0.970535, 0.50634, 0.999758,
  -0.354685, -0.428564, 0.851188, 0.851188, -0.428564},
 {-0.354685, 0.999758, 0.50634, -0.970535, 0.650288, 0.841471,
  0., 0., 0., 0.}, {0.841471, -0.287903, 0.580611, 0.714919,
  -0.850983, 0.926954, -0.0802396, -0.993995, 0.89351, -0.993995},
 {-0.0802396, 0.926954, -0.850983, 0.714919, 0.580611, -0.287903,
  0.841471, 0., 0., 0.}, {0.841471, -0.961397, -0.790433, 0.988041,
  -0.970484, -0.806097, -0.954069, 0.99957, 0.348571, 0.348571},
 {0.99957, -0.954069, -0.806097, -0.970484, 0.988041,
```

```
 -0.790433, -0.961397, 0.841471, 0., 0.},
{0.841471, -0.750987, 0.806401, -0.727101, 0.0886389,
 -0.768718, -0.323106, -0.327429, 0.947099, 0.65342},
{0.947099, -0.327429, -0.323106, -0.768718, 0.0886389,
 -0.727101, 0.806401, -0.750987, 0.841471, 0.},
{0.841471, 0.149877, 0.976591, 0.983604, -0.663386,
 -0.822384, 0.934141, 0.00457091, 0.999976, 0.50724},
{0.50724, 0.999976, 0.00457091, 0.934141, -0.822384,
 -0.663386, 0.983604, 0.976591, 0.149877, 0.841471}}
```

In[4]:=
```
f2[i_, j_] := Flatten[donnees][[40 (i - 1) + j]]
```

In[5]:=
```
Table[f1[i, j], {i, 1, 10}, {j, 1, 40}]
```

Out[5]=
```
{{0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0.841471, 0.841471, 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, 0.841471, 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0.841471, 0.909297, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0.841471, 0.909297, 0.841471, 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, 0.909297, 0.841471,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0.841471, 0.14112, 0.14112, 0.841471, 0., 0., 0., 0., 0., 0.},
 {0.841471, 0.909297, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0.841471, 0.14112, 0.14112, 0.841471,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, 0.14112,
  0.14112, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0.841471, -0.756802,
  -0.279415, -0.756802, 0.841471, 0., 0., 0., 0., 0.},
 {0.841471, 0.14112, 0.14112, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0.841471, -0.756802, -0.279415, -0.756802,
  0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, -0.756802,
  -0.279415, -0.756802, 0.841471, 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, -0.958924,
  -0.544021, -0.544021, -0.958924, 0.841471, 0., 0., 0., 0.},
 {0.841471, -0.756802, -0.279415, -0.756802, 0.841471, 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471,
  -0.958924, -0.544021, -0.544021, -0.958924, 0.841471,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, -0.958924,
  -0.544021, -0.544021, -0.958924, 0.841471, 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.841471, -0.279415,
  0.650288, 0.912945, 0.650288, -0.279415, 0.841471, 0., 0., 0.}}
```

In[6]:=
```
g[i_, j_, k_] :=
 Flatten[donnees][[80 (i - 1) + 10 (j - 1) + k]]
```

In[7]:=
```
Table[g[i, j, k], {i, 1, 5}, {j, 1, 8}, {k, 1, 10}]
```

Out[7]=
```
{{{0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.909297, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.14112, 0.14112, 0.841471, 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}},
 {{0.841471, -0.756802, -0.279415, -0.756802, 0.841471, 0.,
   0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, -0.958924, -0.544021, -0.544021, -0.958924, 0.841471,
   0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, -0.279415, 0.650288, 0.912945, 0.650288, -0.279415,
   0.841471, 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.656987, 0.836656, -0.428183, -0.428183, 0.836656,
   0.656987, 0.841471, 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}},
 {{0.841471, 0.989358, 0.270906, -0.521551, 0.773891,
   -0.521551, 0.270906, 0.989358, 0.841471, 0.},
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.412118, -0.991779, 0.73319, 0.329991, 0.329991, 0.73319,
   -0.991779, 0.412118, 0.841471}, {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, -0.544021, 0.850904, 0.580611, 0.467719,
   0.623012, 0.467719, 0.580611, 0.850904, -0.544021},
  {0.841471, 0., 0., 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, -0.99999, -0.999755, 0.997797, -0.132382,
   -0.184811, -0.184811, -0.132382, 0.997797, -0.999755},
  {-0.99999, 0.841471, 0., 0., 0., 0., 0., 0., 0., 0.}},
 {{0.841471, -0.536573, -0.0265512, 0.0883987, -0.980234,
   0.313286, 0.363256, 0.313286, -0.980234, 0.0883987},
  {-0.0265512, -0.536573, 0.841471, 0., 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.420167, 0.513978, -0.114815, -0.958907,
   -0.868906, 0.636854, 0.636854, -0.868906, -0.958907},
  {-0.114815, 0.513978, 0.420167, 0.841471, 0., 0., 0., 0., 0., 0.},
  {0.841471, 0.990607, 0.105988, -0.412091, 0.919991,
   -0.721163, -0.354685, 0.982009, -0.354685, -0.721163},
  {0.919991, -0.412091, 0.105988, 0.990607, 0.841471, 0., 0., 0., 0., 0.},
  {0.841471, 0.650288, -0.970535, 0.50634, 0.999758, -0.354685,
   -0.428564, 0.851188, 0.851188, -0.428564}, {-0.354685, 0.999758,
   0.50634, -0.970535, 0.650288, 0.841471, 0., 0., 0., 0.}},
 {{0.841471, -0.287903, 0.580611, 0.714919, -0.850983,
   0.926954, -0.0802396, -0.993995, 0.89351, -0.993995},
  {-0.0802396, 0.926954, -0.850983, 0.714919, 0.580611, -0.287903,
   0.841471, 0., 0., 0.}, {0.841471, -0.961397, -0.790433, 0.988041,
   -0.970484, -0.806097, -0.954069, 0.99957, 0.348571, 0.348571},
  {0.99957, -0.954069, -0.806097, -0.970484, 0.988041,
```

```
   -0.790433, -0.961397, 0.841471, 0., 0.},
  {0.841471, -0.750987, 0.806401, -0.727101, 0.0886389,
   -0.768718, -0.323106, -0.327429, 0.947099, 0.65342},
  {0.947099, -0.327429, -0.323106, -0.768718, 0.0886389,
   -0.727101, 0.806401, -0.750987, 0.841471, 0.},
  {0.841471, 0.149877, 0.976591, 0.983604, -0.663386,
   -0.822384, 0.934141, 0.00457091, 0.999976, 0.50724},
  {0.50724, 0.999976, 0.00457091, 0.934141, -0.822384,
   -0.663386, 0.983604, 0.976591, 0.149877, 0.841471}}}
```

In[8]:=
```
h[i_, j_, k_, l_] := Flatten[
   donnees][[100 (i - 1) + 20 (j - 1) + 2 (k - 1) + l]]
```

In[9]:=
```
Table[h[i, j, k, l], {i, 1, 4},
 {j, 1, 5}, {k, 1, 10}, {l, 1, 2}]
```

Out[9]=
```
{{{{0.841471, 0.}, {0., 0.}, {0., 0.}, {0., 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, 0.841471}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, 0.909297}, {0.841471, 0.}, {0., 0.}, {0., 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, 0.14112}, {0.14112, 0.841471}, {0., 0.}, {0., 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, -0.756802}, {-0.279415, -0.756802}, {0.841471, 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}}},
 {{{0.841471, -0.958924}, {-0.544021, -0.544021}, {-0.958924, 0.841471},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, -0.279415}, {0.650288, 0.912945}, {0.650288, -0.279415},
   {0.841471, 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.},
   {0., 0.}}, {{0.841471, 0.656987}, {0.836656, -0.428183},
   {-0.428183, 0.836656}, {0.656987, 0.841471}, {0., 0.},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, 0.989358}, {0.270906, -0.521551}, {0.773891, -0.521551},
   {0.270906, 0.989358}, {0.841471, 0.}, {0., 0.}, {0., 0.}, {0., 0.},
   {0., 0.}, {0., 0.}}, {{0.841471, 0.412118}, {-0.991779, 0.73319},
   {0.329991, 0.329991}, {0.73319, -0.991779}, {0.412118, 0.841471},
   {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}}},
 {{{0.841471, -0.544021}, {0.850904, 0.580611}, {0.467719, 0.623012},
   {0.467719, 0.580611}, {0.850904, -0.544021},
   {0.841471, 0.}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, -0.99999}, {-0.999755, 0.997797}, {-0.132382, -0.184811},
   {-0.184811, -0.132382}, {0.997797, -0.999755},
   {-0.99999, 0.841471}, {0., 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, -0.536573}, {-0.0265512, 0.0883987},
   {-0.980234, 0.313286}, {0.363256, 0.313286}, {-0.980234, 0.0883987},
   {-0.0265512, -0.536573}, {0.841471, 0.}, {0., 0.}, {0., 0.}, {0., 0.}},
  {{0.841471, 0.420167}, {0.513978, -0.114815}, {-0.958907, -0.868906},
   {0.636854, 0.636854}, {-0.868906, -0.958907}, {-0.114815, 0.513978},
```

```
    {0.420167, 0.841471}, {0., 0.}, {0., 0.}, {0., 0.}},
   {{0.841471, 0.990607}, {0.105988, -0.412091}, {0.919991, -0.721163},
    {-0.354685, 0.982009}, {-0.354685, -0.721163}, {0.919991, -0.412091},
    {0.105988, 0.990607}, {0.841471, 0.}, {0., 0.}, {0., 0.}}},
  {{{0.841471, 0.650288}, {-0.970535, 0.50634}, {0.999758, -0.354685},
    {-0.428564, 0.851188}, {0.851188, -0.428564}, {-0.354685, 0.999758},
    {0.50634, -0.970535}, {0.650288, 0.841471}, {0., 0.}, {0., 0.}},
   {{0.841471, -0.287903}, {0.580611, 0.714919}, {-0.850983, 0.926954},
    {-0.0802396, -0.993995}, {0.89351, -0.993995},
    {-0.0802396, 0.926954}, {-0.850983, 0.714919},
    {0.580611, -0.287903}, {0.841471, 0.}, {0., 0.}},
   {{0.841471, -0.961397}, {-0.790433, 0.988041}, {-0.970484, -0.806097},
    {-0.954069, 0.99957}, {0.348571, 0.348571}, {0.99957, -0.954069},
    {-0.806097, -0.970484}, {0.988041, -0.790433}, {-0.961397, 0.841471},
    {0., 0.}}, {{0.841471, -0.750987}, {0.806401, -0.727101},
    {0.0886389, -0.768718}, {-0.323106, -0.327429},
    {0.947099, 0.65342}, {0.947099, -0.327429}, {-0.323106, -0.768718},
    {0.0886389, -0.727101}, {0.806401, -0.750987}, {0.841471, 0.}},
   {{0.841471, 0.149877}, {0.976591, 0.983604}, {-0.663386, -0.822384},
    {0.934141, 0.00457091}, {0.999976, 0.50724}, {0.50724, 0.999976},
    {0.00457091, 0.934141}, {-0.822384, -0.663386},
    {0.983604, 0.976591}, {0.149877, 0.841471}}}}
```