Lay Chapter 7, Symmetric Matrices and Quadratic Forms

Diagonalize a Matrix

```
Clear[a];

a = (3 1
     1 3);
```

Calculate the eigenvalues of a.

```
evals = Eigenvalues[a];
Λ = DiagonalMatrix[evals];
% // MatrixForm

(4 0
 0 2)
```

Calculate the eigenvectors of a.

```
evecs = Eigenvectors[a];
v = Transpose[evecs];
% // MatrixForm

(1 -1
 1  1)
```

Normalize the columns of v

```
v1 = Transpose[v][[1]];
v2 = Transpose[v][[2]];
```

$$p = \text{Transpose}\left[\left\{\frac{1}{\sqrt{v1.v1}}\ v1,\ \frac{1}{\sqrt{v2.v2}}\ v2\right\}\right];$$

```
% // MatrixForm
```

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Check.

```
Transpose[p] == Inverse[p];
a == p.Λ.Transpose[p]
```

```
True
```

Diagonalize a Matrix

```
Clear[a];
```

$$a = \begin{pmatrix} 5 & 2 & 9 & -6 \\ 2 & 5 & -6 & 9 \\ 9 & -6 & 5 & 2 \\ -6 & 9 & 2 & 5 \end{pmatrix};$$

Calculate the eigenvalues of a.

```
evals = Eigenvalues[a];
Λ = DiagonalMatrix[evals];
% // MatrixForm
```

$$\begin{pmatrix} 18 & 0 & 0 & 0 \\ 0 & -12 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

Calculate the eigenvectors of a.

```
evecs = Eigenvectors[a];
v = Transpose[evecs];
% // MatrixForm
```

$$\begin{pmatrix} -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Normalize the columns of v

```
v1 = Transpose[v][[1]];
v2 = Transpose[v][[2]];
v3 = Transpose[v][[3]];
v4 = Transpose[v][[4]];
```

$$\texttt{p = Transpose}\left[\left\{\frac{1}{\sqrt{\texttt{v1.v1}}}\,\texttt{v1},\ \frac{1}{\sqrt{\texttt{v2.v2}}}\,\texttt{v2},\ \frac{1}{\sqrt{\texttt{v3.v3}}}\,\texttt{v3},\ \frac{1}{\sqrt{\texttt{v4.v4}}}\,\texttt{v4}\right\}\right];$$

```
% // MatrixForm
```

$$\begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Check.

```
Transpose[p] == Inverse[p]
p.Λ.Transpose[p] == a

True
```

```
True
```

Matrix Representation of a Quadratic Form

Symmetric matrices represent quadratic forms in the following way.

```
Clear[a, x, x1, x2]

a = (1  2
     2  3);
x = {x1, x2};

x.a.x // Simplify

x1^2 + 4 x1 x2 + 3 x2^2
```

Here is the quadratic form generated by a diagonal matrix

```
Clear[a, x, x1, x2]

a = (4  0
     0  3);
x = {x1, x2};

x.a.x // Simplify

4 x1^2 + 3 x2^2
```

... and another generated by a symmetric matrix which is not a diagonal matrix.

```
Clear[a, x, x1, x2]

a = ( 3  -2
     -2   7);
x = {x1, x2};

x.a.x // Simplify

3 x1^2 - 4 x1 x2 + 7 x2^2
```

The difference is the "mixed" term.

Matrix Representation of Quadratic Forms

Write the matrix representing a particular quadratic form.

```
Clear[a, x, x1, x2, x3]

a = ( 5     -1/2  0
      -1/2   3    4
      0      4    2 );

x = {x1, x2, x3};

x.a.x // Simplify
```

$5\,x1^2 - x1\,x2 + 3\,x2^2 + 8\,x2\,x3 + 2\,x3^2$

Value of a Quadratic Form at a Point

Evaluate a particular quadratic form at distinct points.

```
Clear[a, x, x1, x2, q]

a = ( 1   -4
      -4  -5 );

x = {x1, x2};

q[x_] := x.a.x // Simplify
```

```
q[{-3, 1}]
q[{2, -2}]
q[{1, -3}]
```

28

16

-20

Change of Variable in a Quadratic Form

If x is a variable in $\mathbb{R}^n$, thena change of variable in a quadratic form is an equation of the form.

```
(* x = p y *)
```

where y is a variable in $\mathbb{R}^n$ and p is an invertible n x n matrix.

Diagonalize the following quadratic form.

```
Clear[a, x, x1, x2, q, evals, evecs, d, p]

a = (1   -4
     -4  -5);
x = {x1, x2};

qx[x_] := x.a.x // Simplify
```

First, calculate the corresponding diagonal matrix.

```
evals = Eigenvalues[a];
d = DiagonalMatrix[evals];
% // MatrixForm
```

$$\begin{pmatrix} -7 & 0 \\ 0 & 3 \end{pmatrix}$$

Then the columns of the matrix p are the orthonormal eigenvectors of a, in the proper order.

```
evecs = Eigenvectors[a];
d1 = evecs[[1]];
d2 = evecs[[2]];

p = Transpose[{1/Sqrt[d1.d1] d1, 1/Sqrt[d2.d2] d2}];
% // MatrixForm
```

$$\begin{pmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{pmatrix}$$

Check the diagonalization.

```
a == p.d.Inverse[p]

True
```

Check that the two quadratic forms agree for x = {2,-2}.
First, here is qx applied to x = {2,-2}.

```
qx[{2, -2}]

16
```

Now x = p.y implies that y = Inverse[p].x
Here is qy applied to y = Inverse[p].x

```
qy[y_] := y.d.y

qy[Inverse[p].{2, -2}]

16
```

Colorful Vectors and Curves
Contains *Mathematica* Code Only -- No Solved Exercises

Principal Axes

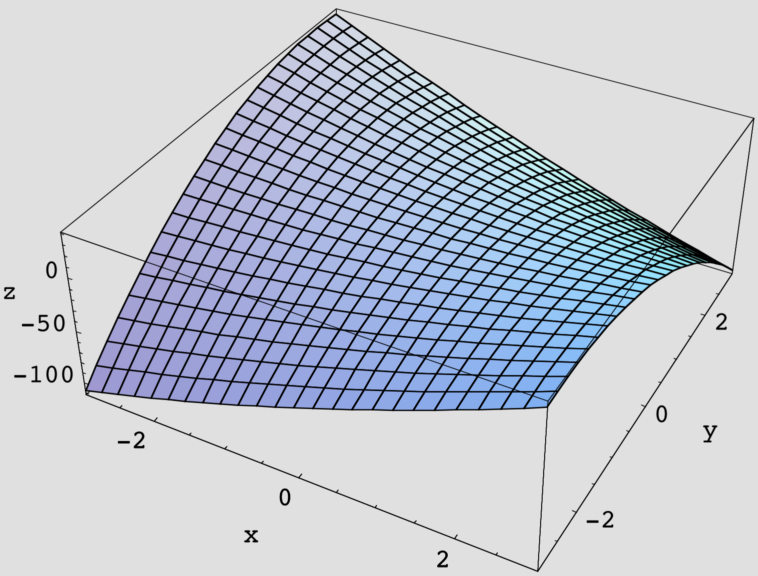
Let's recast a quadratic form generated by a symmetric 2 x 2 matrix as a function of two variables.
We use the matrix a from the previous example.

```
Clear[qa, qd, a, d, x1, x2, y1, y2]

a = ( 1  -4
     -4  -5 );

qa[x1_, x2_] := {x1, x2}.a. {x1, x2}
```

```
qaSurface = Plot3D[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```

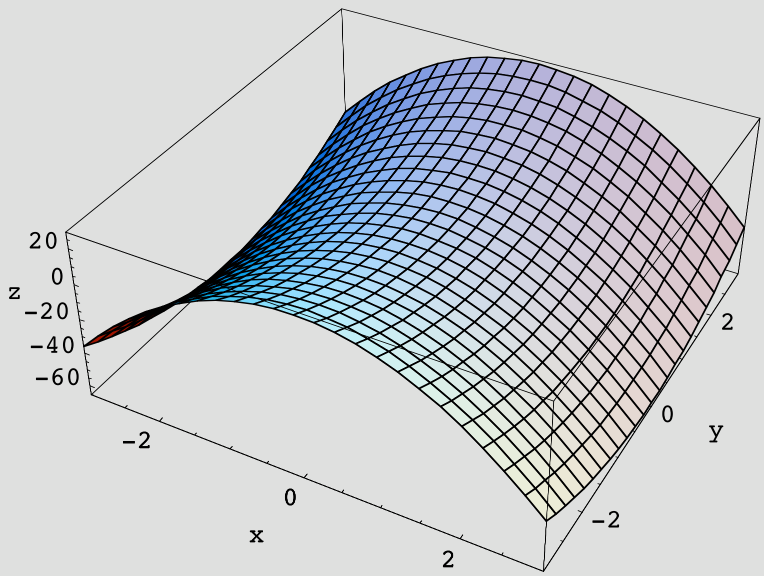


Now do the same thing for the matrix d.

```
d = ( -7  0
       0  3 );

qd[y1_, y2_] := {y1, y2}.d. {y1, y2}
```

```
qdSurface = Plot3D[qd[y1, y2], {y1, -3, 3}, {y2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```



The second surface s obtained from the first by a rotation.
The y1- and y2-axes in the second figure correspond to the **principal axes** in the first.
The principal axes are the directions indicated by the columns of the matrix p used in describing the change of variables: x = p y.

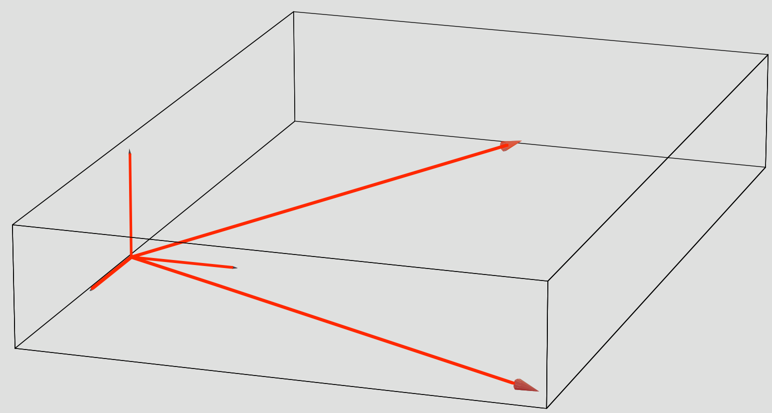
```
origin = {0, 0, 0};
p1 = {Transpose[p][[1]], 0} // Flatten
p2 = {Transpose[p][[2]], 0} // Flatten
```

$\left\{\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}, 0\right\}$

$\left\{-\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}, 0\right\}$

Let's see those two vectors. We will exaggerate their lengths in order to see them better.
Note the orientation of the standard coordinate axes.

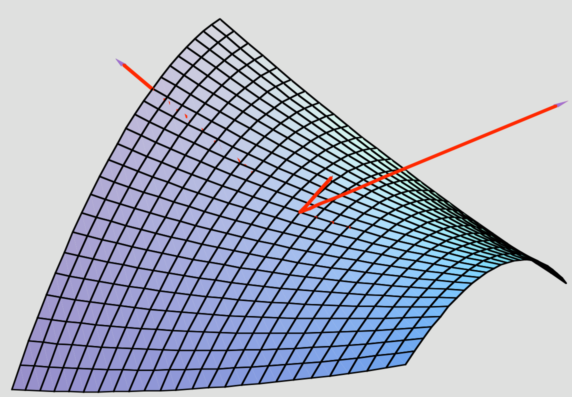
```
showColorful3DVectors[{{origin, 5 p1}, {origin, 5 p2}},
  Red, Tomato, PlotRange → All];
```



In this image as well, the lengths of the vectors indicating the principal axes are exagerated in order to see them better. One of the vectors is mostly underneath the surface.

```
principalAxes = showColorful3DVectors[{{origin, 5 p1}, {origin, 5 p2}},
   Red, Orchid, PlotRange → All, DisplayFunction → Identity];

Show[qaSurface, principalAxes, PlotRange → All, Axes → False,
  Boxed → False, DisplayFunction → $DisplayFunction];
```



Using Contour Plots to Investigate Quadratic Forms and their Principal Axes

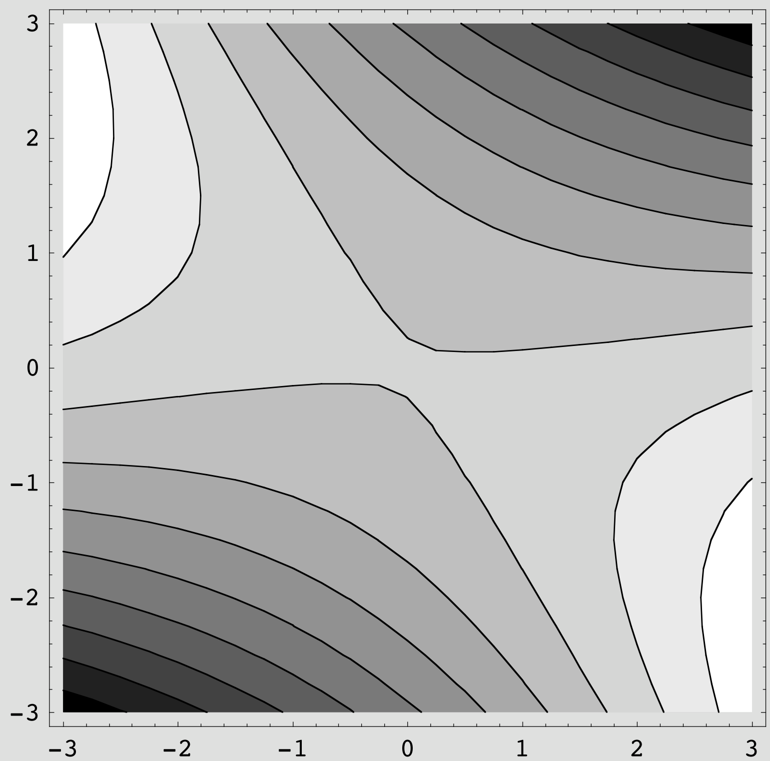
Let's see a contour plot of the quadratic form qa.

```
Clear[qa, qd, a, d, x1, x2, y1, y2]

a = (1  -4
     -4  -5);

qa[x1_, x2_] := {x1, x2}.a.{x1, x2}

contours = ContourPlot[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3}];
```

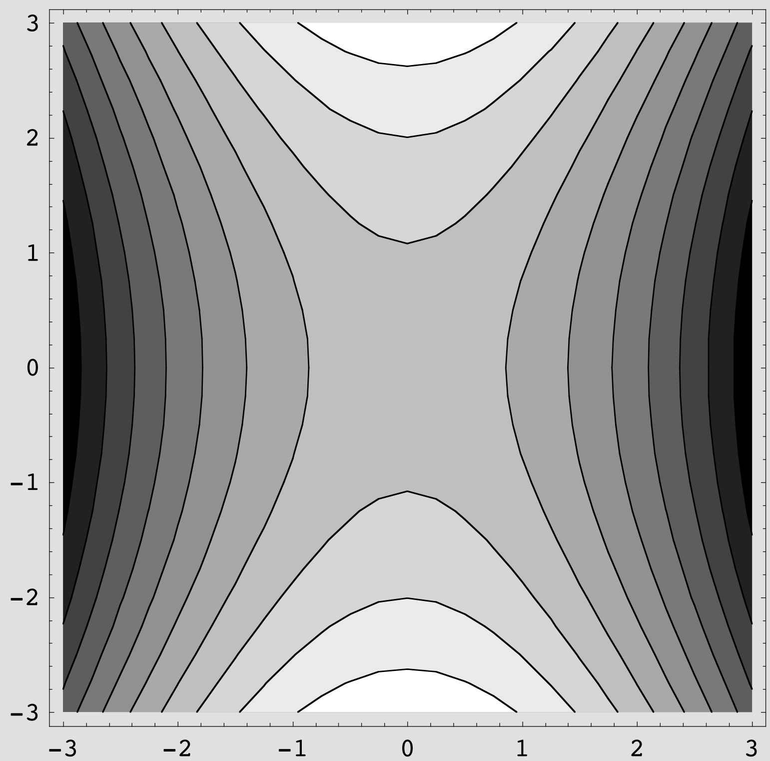


... and a contour plot of the quadratic form qd.
It seems clear that the second is obtained by a rotation of the first.

```
d = ( -7  0 );
    (  0  3 )

qd[y1_, y2_] := {y1, y2}.d.{y1, y2}

ContourPlot[qd[y1, y2], {y1, -3, 3}, {y2, -3, 3}];
```

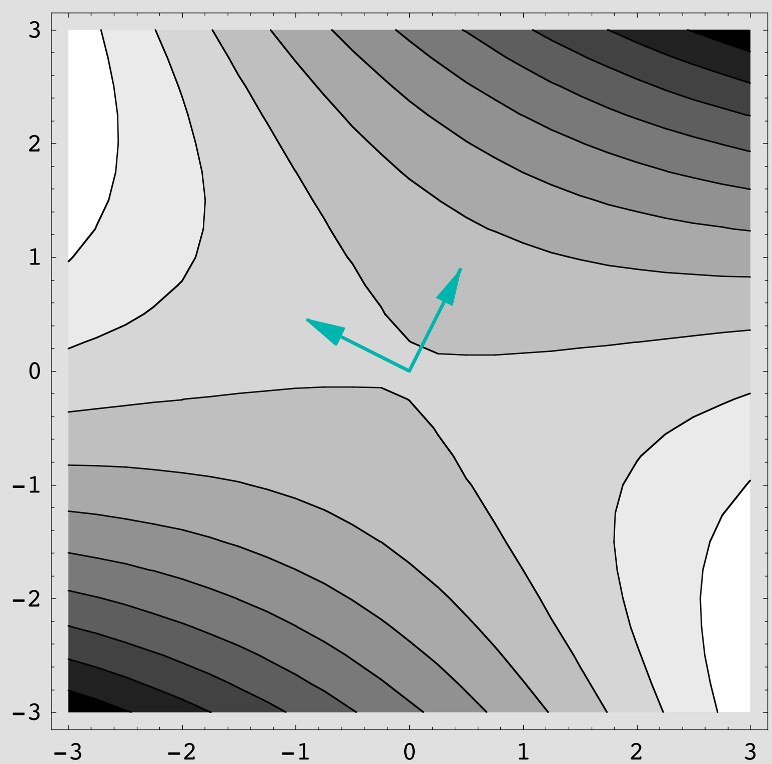


Now let's see those **principal axes** superimposed on the contour plot of qa.

```
origin = {0, 0};
pa1 = {origin, Transpose[p][[1]]};
pa2 = {origin, Transpose[p][[2]]};

vecs = showColorfulVectors[{pa1, pa2},
   ManganeseBlue, DisplayFunction → Identity];

Show[contours, vecs, DisplayFunction → $DisplayFunction];
```



Classification of Quadratic Forms

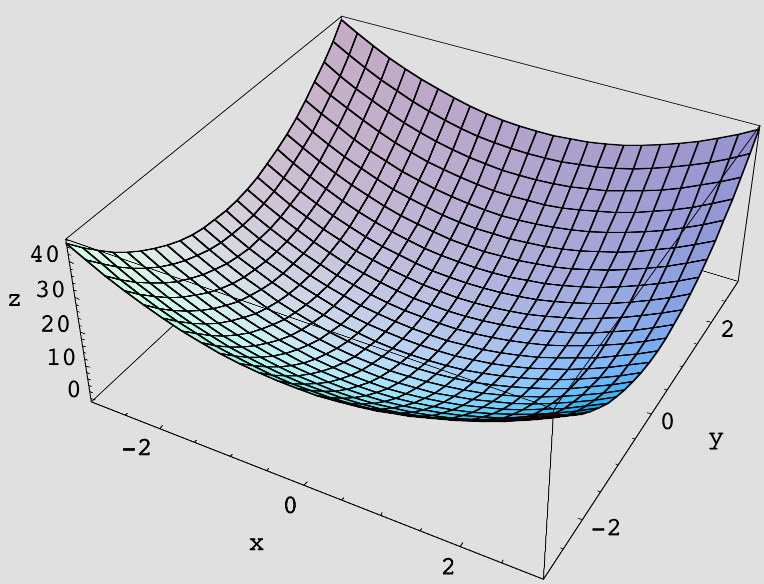
■ Positive definite

```
Clear[a, qa, qd]
```

$$a = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix};$$

```
qa[x1_, x2_] := {x1, x2}.a.{x1, x2}

qaSurface = Plot3D[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```



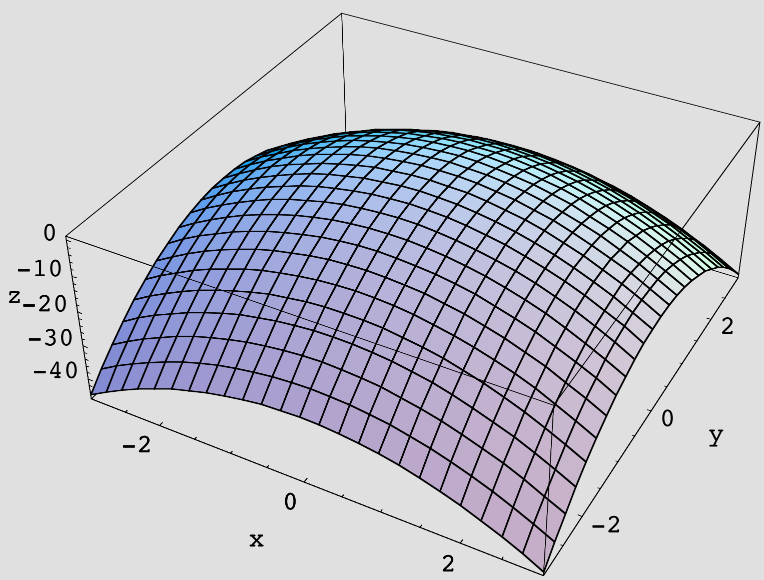
■ Negative definite

```
Clear[a, qa, qd]

a = (-2   0
      0  -3);

qa[x1_, x2_] := {x1, x2}.a. {x1, x2}

qaSurface = Plot3D[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```



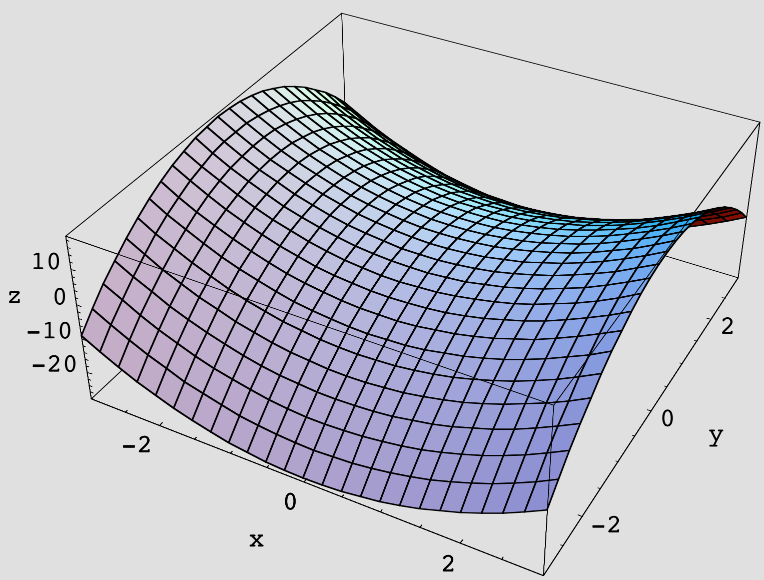
■ Indefinite

```
Clear[a, qa, qd]

a = (2   0
     0  -3);

qa[x1_, x2_] := {x1, x2}.a.{x1, x2}

qaSurface = Plot3D[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```



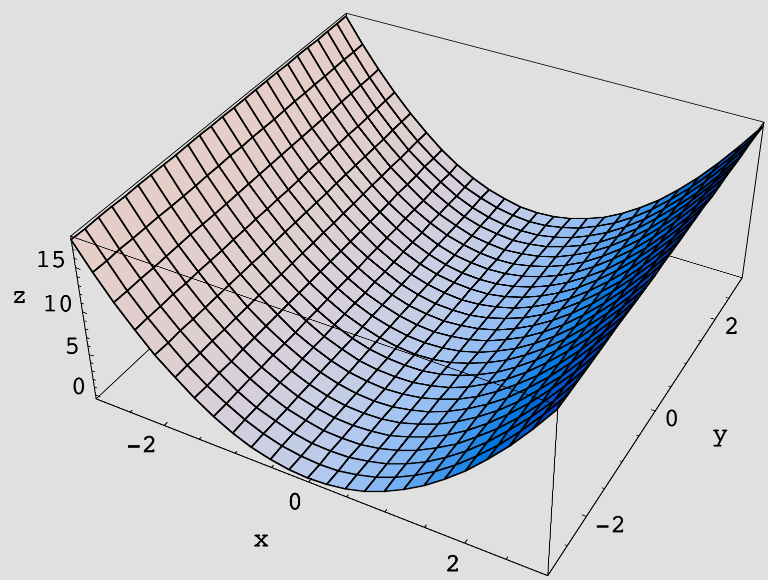
■ Positive Semidefinite

```
Clear[a, qa, qd]
```

$$a = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix};$$

```
qa[x1_, x2_] := {x1, x2}.a.{x1, x2}

qaSurface = Plot3D[qa[x1, x2], {x1, -3, 3}, {x2, -3, 3},
   AxesLabel → {"x", "y", "z"}];
```



Diagonalization a Quadratic Form

Diagonalize the following quadratic form.

```
Clear[a, x, x1, x2, x3, x4, q, evals, evecs, d, p]
```

$$a = \begin{pmatrix} 1 & 9/2 & 0 & -6 \\ 9/2 & 1 & 6 & 0 \\ 0 & 6 & 1 & 9/2 \\ -6 & 0 & 9/2 & 1 \end{pmatrix};$$

```
x = {x1, x2, x3, x4};

qx[x_] = x.a.x // Simplify
```

$$x1^2 + 9\, x1\, x2 + x2^2 + 12\, x2\, x3 + x3^2 - 12\, x1\, x4 + 9\, x3\, x4 + x4^2$$

First, calculate the corresponding diagonal matrix.

```
evals = Eigenvalues[a];
d = DiagonalMatrix[evals];
% // MatrixForm
```

$$\begin{pmatrix} \frac{17}{2} & 0 & 0 & 0 \\ 0 & \frac{17}{2} & 0 & 0 \\ 0 & 0 & -\frac{13}{2} & 0 \\ 0 & 0 & 0 & -\frac{13}{2} \end{pmatrix}$$

Use d to construct the equivalent (diagonal) quadratic form.
This quadratic form is **Lay 7.2.17**.

```
qd[y1_, y2_, y3_, y4_] = {y1, y2, y3, y4}.d. {y1, y2, y3, y4}
```

$$\frac{17\, y1^2}{2} + \frac{17\, y2^2}{2} - \frac{13\, y3^2}{2} - \frac{13\, y4^2}{2}$$

Now relate the two quadratic forms by calculating the change of basis matrix p such that x = p y.
The columns of the matrix p are the orthonormal eigenvectors of a, in the proper order.

```
evecs = Eigenvectors[a];
d1 = evecs[[1]];
d2 = evecs[[2]];
d3 = evecs[[3]];
d4 = evecs[[4]];

p = Transpose[{1/Sqrt[d1.d1] d1, 1/Sqrt[d2.d2] d2, 1/Sqrt[d3.d3] d3, 1/Sqrt[d4.d4] d4}];
% // MatrixForm
```

$$\begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{3}{5\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{3}{5\sqrt{2}} \\ -\frac{3}{5\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{3}{5\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{2\sqrt{2}}{5} & 0 & \frac{2\sqrt{2}}{5} \\ \frac{2\sqrt{2}}{5} & 0 & \frac{2\sqrt{2}}{5} & 0 \end{pmatrix}$$

Check the diagonalization.

```
a == p.d.Inverse[p]

True
```

Constrained Optimization

Consider the following quadratic form:

```
Clear[a, x, x1, x2, x3, evals, evecs, d, p]
```

$$a = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix};$$

```
x = {x1, x2, x3};

x.a.x // Simplify
```

$$3\,x1^2 + 3\,x2^2 + 2\,x2\,x3 + 4\,x3^2 + 2\,x1\,(2\,x2 + x3)$$

Calculate the corresponding diagonal matrix.

```
evals = Eigenvalues[a];
d = DiagonalMatrix[evals];
% // MatrixForm
```

$$\begin{pmatrix} 6 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The columns of the matrix p are the orthonormal eigenvectors of a, taken in the proper order.

```
evecs = Eigenvectors[a];
d1 = evecs[[1]];
d2 = evecs[[2]];
d3 = evecs[[3]];
```

$$\text{p = Transpose}\left[\left\{\frac{1}{\sqrt{\text{d1.d1}}}\,\text{d1},\ \frac{1}{\sqrt{\text{d2.d2}}}\,\text{d2},\ \frac{1}{\sqrt{\text{d3.d3}}}\,\text{d3}\right\}\right];$$

```
% // MatrixForm
```

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \sqrt{\frac{2}{3}} & 0 \end{pmatrix}$$

Check the diagonalization.

```
a == p.d.Inverse[p]

True
```

Constrained Optimization

Consider the following quadratic form.

```
Clear[a, x, x1, x2, evals, evecs, d, p]

a = (3  1
     1  3);
x = {x1, x2};

x.a.x // Simplify
```

$3\, x1^2 + 2\, x1\, x2 + 3\, x2^2$

Calculate the corresponding diagonal matrix.

```
evals = Eigenvalues[a];
d = DiagonalMatrix[evals];
% // MatrixForm
```

$$\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$$

Then the columns of the matrix p are the orthonormal eigenvectors of a, taken in the proper order.

```
evecs = Eigenvectors[a];
d1 = evecs[[1]];
d2 = evecs[[2]];
```

$$p = \text{Transpose}\left[\left\{\frac{1}{\sqrt{d1.d1}}\, d1,\ \frac{1}{\sqrt{d2.d2}}\, d2\right\}\right];$$

```
% // MatrixForm
```

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Check the diagonalization.

```
a == p.d.Inverse[p]

True
```

We can graph this surface.
Stretch out the graph for a better view.

```
Clear[x1, x2]

q[x1_, x2_] = If [x1^2 + x2^2 ≤ 1, {x1, x2}.a.{x1, x2}, -1];

Plot3D[q[x1, x2], {x1, -1.1, 1.1}, {x2, -1.1, 1.1},
  PlotRange → {0, 5},
  Mesh → False, PlotPoints → 100, AxesLabel → {x1, x2, None}];
```