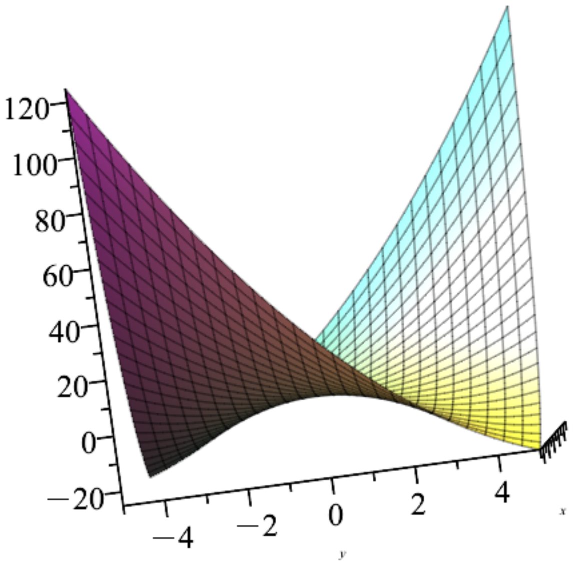
Constructing a plane defined by tangent lines to cross sections
We start by defining a function that we will work with. Let z=f(x,y) =x^2-3*x*y+y^2.
It is first of all useful to look at the graph.

```
> f := (x, y) → x^2−3*x*y + y^2;
```

$$f := (x, y) \to x^2 - 3\,x\,y + y^2 \qquad (1)$$

```
plot3d( f(x, y), x =−5 ..5, y =−5 ..5, axes = framed);
```

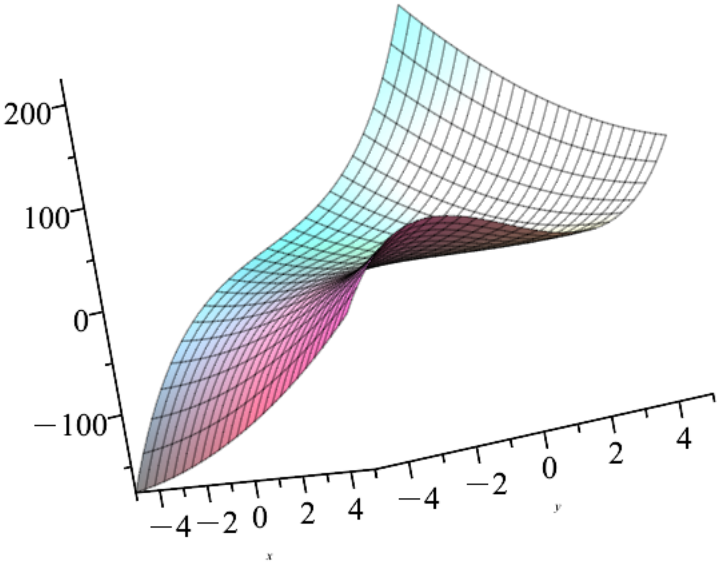


Notice that the graph goes up in one direction and down in another. We are at a saddle, so tangent planes will
tend to cut the surface, much as the tangent line to a cubic curve cuts the curve at the inflection point.

```
> g := (x, y) → x^2 − 3*x*y + y^3;
```

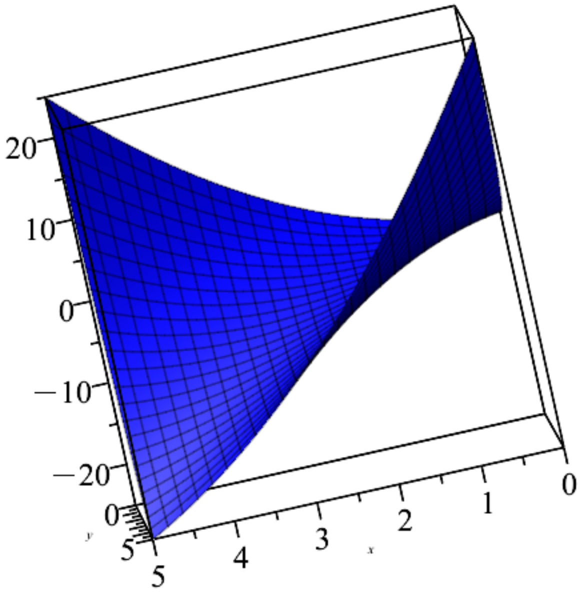
$$g := (x, y) \to x^2 - 3\,x\,y + y^3 \qquad (2)$$

```
> plot3d( g(x, y), x =−5 ..5, y =−5 ..5, axes = framed);
```



To find the tangent plane we considered cross sections.
We will start with the point (-1,3). We start by finding the z-value at the point on the graph.
Next compute the functions in one variable that we obtain by holding either x or y constant. i.e., f(x,y0) or f(x0,y)

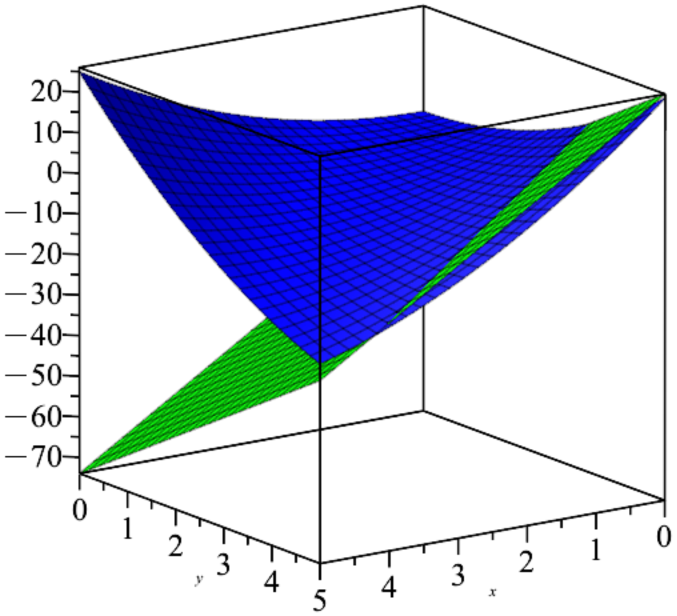
```
> graph := plot3d({f(x, y)}, x = 0 ..5, y = 0 ..5, color = blue);
```



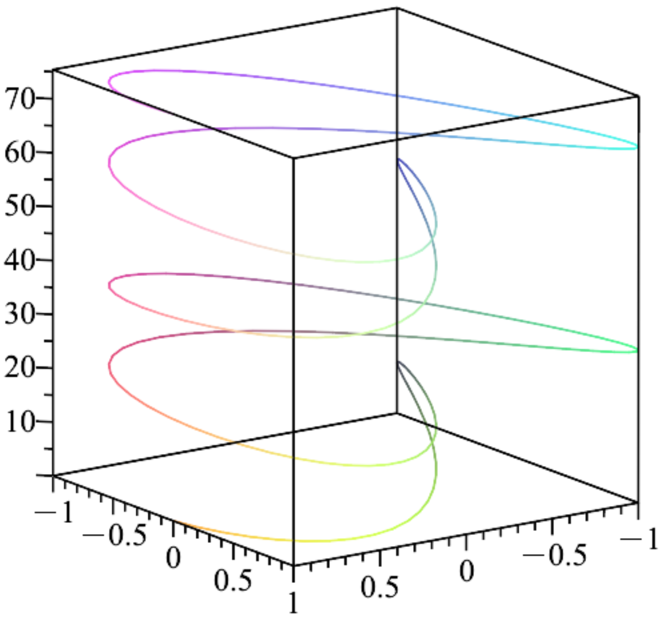
```
> tanPlane := plot3d( {19−11 * (x + 1) + 9 * (y−3) }, x = 0 ..5, y = 0 ..5, color = green) :
```

The equation of tangent plane at (x0,y0) is f(x0,y0)+Df_x(x-x0)+Df_y(y-y0);

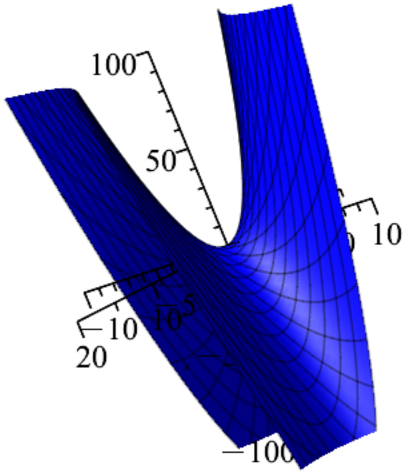
```
> plots[display](graph, tanPlane);
```



```
> ?plots,spacecurve
with(plots) :
spacecurve([cos(2· t), sin(3· t), 6· t], t = 0 ..4 * Pi);
```



```
plot3d(f(x, y), x =−15 ..20, y =−10 ..10, view =−100 ..100, axes = normal, color = blue);
```



```
> x0 := −1; y0 := 3;
  z0 := f(x0, y0);
```

```
x0 := −1
```

$$y0 := 3$$

$$z0 := 19 \tag{3}$$

```
fx := f(x, y0);
```

$$x^2 - 9\,x + 9 \tag{4}$$

```
fy := f(x0, y);
```

$$y^2 + 3\,y + 1 \tag{5}$$

```
Dfx := diff(fx, x);
```

$$2\,x - 9 \tag{6}$$

```
> mx := subs(x = x0, Dfx);
```

$$mx := -11 \tag{7}$$

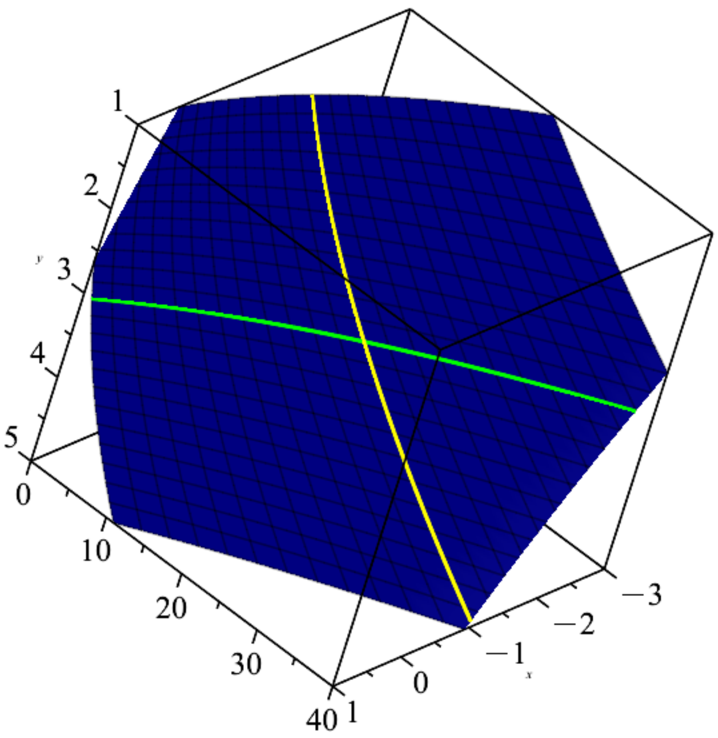
```
Dfy := diff(fy, y);
```

$$2\,y + 3 \tag{8}$$

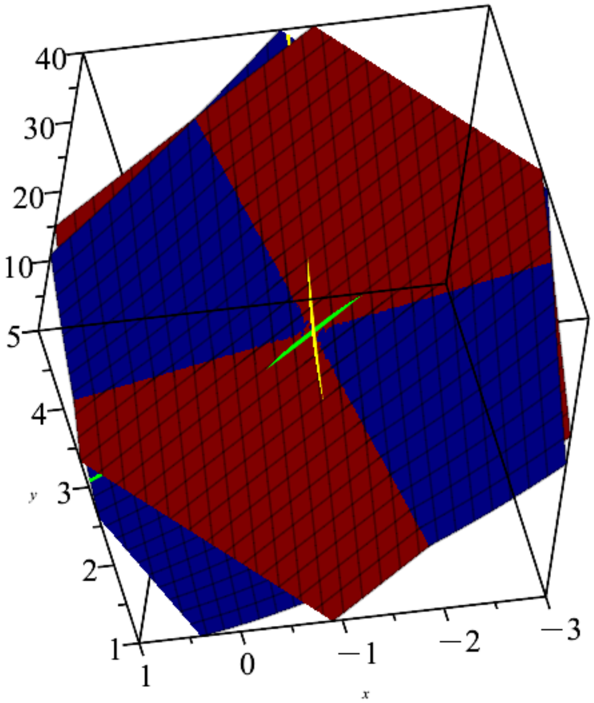
```
my := subs(y = y0, Dfy);
```

$$9 \tag{9}$$

```
s1 := plot3d(f(x, y), x = -3 ..1, y = 1 ..5,
   view = 0 ..40, color = blue) :
s2 := plot3d(19 - 11 * (x + 1) + 9 * (y - 3), x = -3 ..1, y = 1 ..5,
   view = 0 ..40, color = red) :
c1 := spacecurve([x, y0, f(x, y0)], x = -3 ..1,
   color = green, thickness = 3) :
c2 := spacecurve([x0, y, f(x0, y)], y = 1 ..5,
   color = yellow, thickness = 3) :
display3d({s1, c1, c2}, axes = boxed);
```



```
> s1 ≔ plot3d(f(x, y), x =−3 ..1, y = 1 ..5,
    view = 0 ..40, color = blue) :
  s2 ≔ plot3d(19 − 11 * (x + 1) + 9 * (y − 3), x =−3 ..1, y = 1 ..5,
    view = 0 ..40, color = red) :
  c1 ≔ spacecurve([x, y0, f(x, y0)], x =−3 ..1,
    color = green, thickness = 3) :
  c2 ≔ spacecurve([x0, y, f(x0, y)], y = 1 ..5,
    color = yellow, thickness = 3) :
  display3d({s1, s2, c1, c2}, axes = boxed);
```



We can set up a block of code that does the work of the example all in one step.

```
> f := (x, y) → x^2 − 3 * x * y + y^2;
  x0 := −1;    y0 := 3;
  width := 5 :
  z0 := f(x0, y0);
  fx := f(x, y0);
  Dfx := diff(fx, x);
  xslope := subs(x = x0, Dfx);
  fy := f(x0, y);
  Dfy := diff(fy, y);
  yslope := subs(y = y0, Dfy);
  fsurface := plot3d(f(x, y), x = x0 − width ..x0 + width, y = y0 − width ..y0 + width,
     color = red) :
  ftanplane := plot3d(z0 + xslope * (x − x0) + yslope * (y − y0), x = x0 − width ..x0 + width,
     y = y0 − width ..y0 + width, color = blue) :
  xcurve := spacecurve([x, y0, f(x, y0)], x = x0 − width ..x0 + width,
     color = green, thickness = 3) :
  ycurve := spacecurve([x0, y, f(x0, y)], y = y0 − width ..y0 + width,
     color = yellow, thickness = 3) :
  display3d({fsurface, ftanplane, xcurve, ycurve}, axes = boxed);
```

$$f := (x, y) \rightarrow x^2 - 3\,x\,y + y^2$$

$$x0 := -1$$

$$y0 := 3$$

$$z0 := 19$$

$$fx := x^2 - 9\,x + 9$$

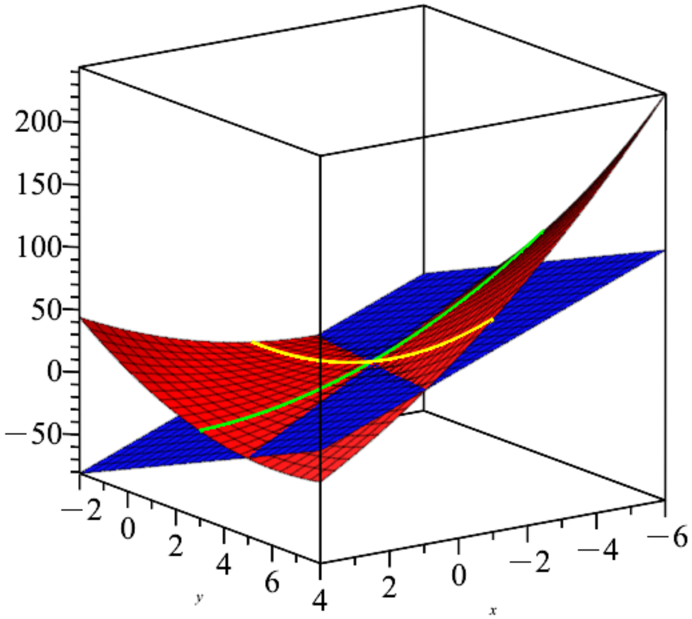
$$Dfx := 2\,x - 9$$

$$xslope := -11$$

$$fy := y^2 + 3\,y + 1$$

$$Dfy := 2\,y + 3$$

$$yslope := 9$$



EXERCISES

Compute the tangent plane of the following surfaces at the given point, and plot both of them in one frame.

Explain what you find.

f(x,y)=x*y+2*y+3*x at (x0,y0)=(1,-1)

f(x,y)=x^2-y^2 at (x0,y0)=(2,2) and also at (2,-2)

f(x,y)=x^3-x*y+y^3-1 at (1,-1);

f(x,y)=$3\,x - x^3 + y^3 - 3\,y$ $\quad$ *at* $\{(-1,-1), (-1,1), (1,-1), (1,1)\}$

```
> GradientTutor( );
```

$$GradientTutor(\ ) \tag{10}$$

with(*Student*[*MultivariateCalculus*]);

[*&x*, `.`, *Angle*, *ApproximateInt*, *ApproximateIntTutor*, *AreOrthogonal*, *AreParallel*, *AreSkew*, *BoxProduct*, *CenterOfMass*, *ChangeOfVariables*, *Contains*, *CrossProduct*, *CrossSection*, *CrossSectionTutor*, *Del*, *DirectionalDerivative*, *DirectionalDerivativeTutor*, *Distance*, *DotProduct*, *Equal*, *FunctionAverage*, *GetDimension*, *GetDirection*, *GetIntersection*, **(11)** *GetNormal, GetPlot, GetPoint, GetRepresentation, Gradient, GradientTutor, Intersects, Jacobian, LagrangeMultipliers, Line, MultiInt, Nabla, Norm, Normalize, Plane, Projection, Revert, SecondDerivativeTest, SurfaceArea, TaylorApproximation, TaylorApproximationTutor, TripleScalarProduct, diff*]

```
> GradientTutor( );
```

```
> Gradient(x * y / (x^2 + y^2 + 1, [x, y] = [1, 1]);
Error, unable to match delimiters
                              Gradient(x * y / (x^2 + y^2 + 1, [x, y] = [1, 1]);
```

```
> Gradient(x^2+x*y+y^2,[x,y]=[0,1]);
```

$$\left[\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right] \qquad (12)$$

```
> Gradient(x^3+y^3,[x,y]=[[-1,1],[0,1]]);
```

$$\left[\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}\right] \qquad (13)$$

```
>
```

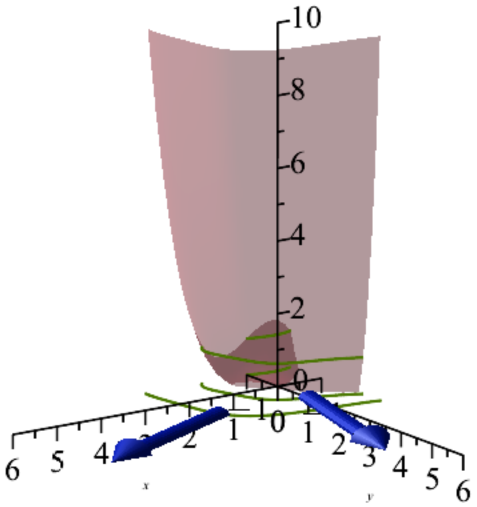
```
> with(Student[MultivariateCalculus]) :
```

```
> Gradient(x^2+y^2,[x,y]=[[0,1],[2,1]], output=dplot);
Error, invalid input: Student:-MultivariateCalculus:-Del expects
value for keyword parameter output to be of type identical(value,
plot, gradplot), but received dplot
```

```
> Gradient(x^2+y^2,[x,y]=[[0,1],[2,1]], output=gradplot);
Error, invalid input: Student:-MultivariateCalculus:-Del expects
value for keyword parameter output to be of type identical(value,
plot, gradplot), but received plots:-gradplot
```

```
> Gradient(x^3+y^4+x*y,[x,y]=[[0,1],[2,1]], x=-1..6, y=-1..6, z=0.
  .10, output=plot);
```

Gradient Vectors



For the function $f(x, y) = y^4 + x^3 + x\,y$, level curves, their prc

```
> restart;
```

Plotting Vector Fields

```
>
```

In Math courses up through Calculus II, we studied functions where elements of both the domain (input values) and the range (output values) are numbers. In the firsct half of this course we have looked at functions that take vectors to numbers. In the last chapter we looked at parameterizations of curves (taking a number to a vector) and surfaces (taking a vector to a vector with a different number of coordinates). In those case we tended to graph just the output rather than the input and output together.

We are now ready to look at functions that take vectors to vectors of the same dimension. The standard example of this are functions that take exery point to a force or velocity at the same point. We can plot these vector fields with the fieldplot command in the plots package.

```
> with(plots);
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,        (14)
    conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
    display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d,
    inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d,
    listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto,
    plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,
    polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions,
setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
tubeplot]
```

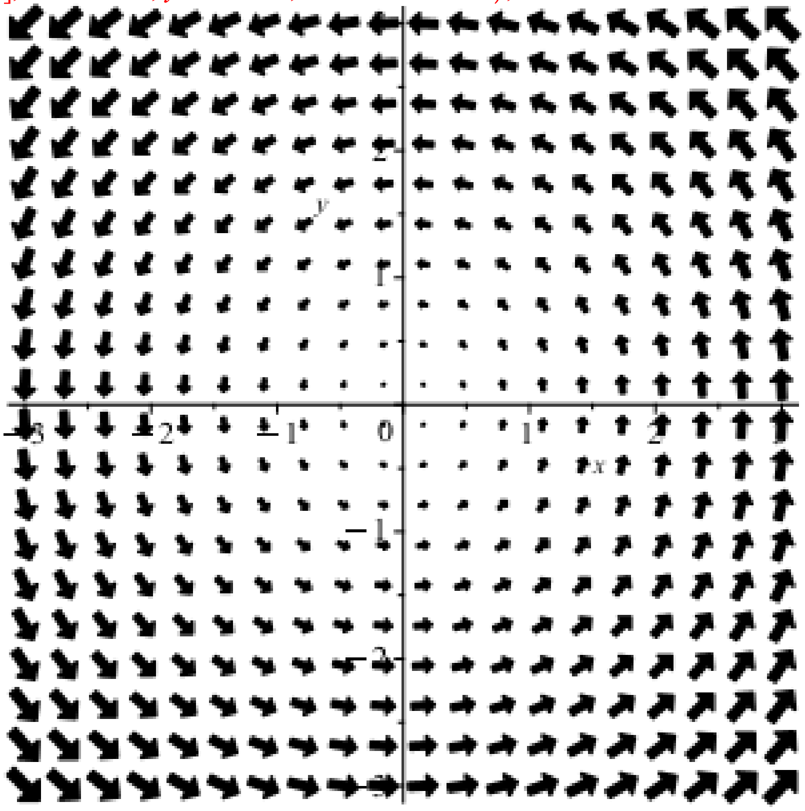
Vector Fields in 2 *dimensions*

Whenever we look at a new command in Maple it is probably a good idea to look at the help page for it.

```
> ?fieldplot
```

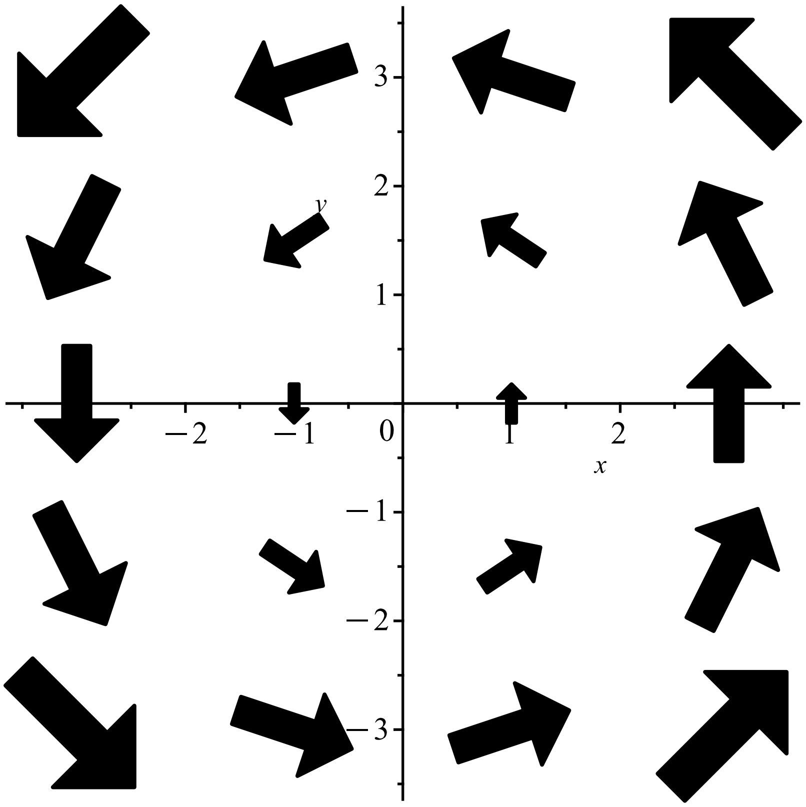
We are ready to plot a simple vector field.

```
> fieldplot([−y, x], x =−3 ..3, y =−3 ..3, arrows = thick);
```



Notice that the plot reduces the scale of the arrows so that we do not get two much overlap. We can control the number of arrows plotted by using the grid option. (The default grid is [20, 20].) You may also want to box the axes to get them from under the arrows.

```
> fieldplot([−y, x], x =−3 ..3, y =−3 ..3, arrows = thick, grid = [4, 5]);
```



Notice that Maple is centering the arrows over the input point rather than starting them at the point.

Exercises

1) Plot the vector fields given by the formulas below. Exercises 2, 4, 5, and 7 from Section 17.1.)

(a) F(x, y) = 2i + 3j.

(b) F(x, y) = 2xi +xj.

(c) F(x, y) = [x + y, x- y].

(d) F(r) = r.

2) Use fieldplot to verify that you have found the correct formulas in 2 for the fields described below.

(a) All vectors are vertical. They proportionally increase in length, always pointing away from the y-axis.

(b) All vectors point radially our from the origin. The magnitude of any vector is proportional to the distance of the starting point from the origin.

(c) All vectors are of length 1 and point radially out from the origin.

(d) The vectors form a counterclockwise swirl growing as you move out from the origin.

(e) The vectors form a clockwise swirl. All vectors are of length 1.

```
> GradientTutor( );
```

$$GradientTutor(\,) \qquad \textbf{(15)}$$

```
with(Student[MultivariateCalculus]);
```

[&x, `.`, Angle, ApproximateInt, ApproximateIntTutor, AreOrthogonal, AreParallel, AreSkew, BoxProduct, CenterOfMass, ChangeOfVariables, Contains, CrossProduct, CrossSection, CrossSectionTutor, Del, DirectionalDerivative, DirectionalDerivativeTutor, Distance, DotProduct, Equal, FunctionAverage, GetDimension, GetDirection, GetIntersection, GetNormal, GetPlot, GetPoint, GetRepresentation, Gradient, GradientTutor, Intersects, Jacobian, LagrangeMultipliers, Line, MultiInt, Nabla, Norm, Normalize, Plane, Projection, Revert, SecondDerivativeTest, SurfaceArea, TaylorApproximation, TaylorApproximationTutor, TripleScalarProduct, diff] **(16)**

```
> GradientTutor( );
> Gradient(x*y/(x^2 + y^2 + 1, [x, y] = [1, 1]);
Error, unable to match delimiters
                    Gradient(x*y/(x^2 + y^2 + 1, [x, y] = [1, 1]);
> Gradient(x^2+x*y+y^2,[x,y]=[0,1]);
```

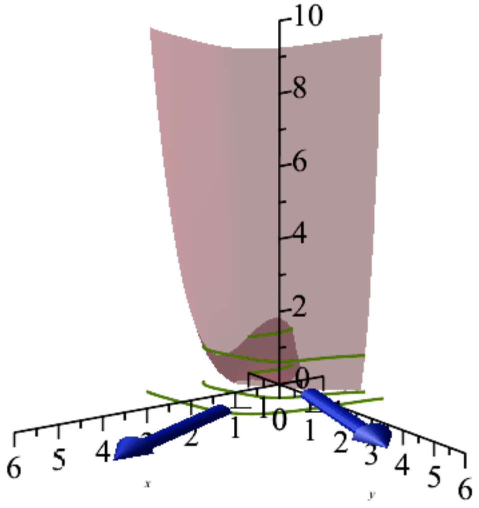
$$\left[\left[\begin{array}{c}1\\2\end{array}\right]\right] \qquad \textbf{(17)}$$

```
> Gradient(x^3+y^3,[x,y]=[[-1,1],[0,1]]);
```

$$\left[\left[\begin{array}{c}3\\3\end{array}\right],\left[\begin{array}{c}0\\3\end{array}\right]\right] \qquad \textbf{(18)}$$

```
>
> with(Student[MultivariateCalculus]) :
> Gradient(x^2+y^2,[x,y]=[[0,1],[2,1]], output=dplot);
Error, invalid input: Student:-MultivariateCalculus:-Del expects
value for keyword parameter output to be of type identical(value,
plot, gradplot), but received dplot
> Gradient(x^2+y^2,[x,y]=[[0,1],[2,1]], output=gradplot);
Error, invalid input: Student:-MultivariateCalculus:-Del expects
value for keyword parameter output to be of type identical(value,
plot, gradplot), but received plots:-gradplot
> Gradient(x^3+y^4+x*y,[x,y]=[[0,1],[2,1]], x=-1..6, y=-1..6, z=0.
  .10, output=plot);
```

Gradient Vectors



For the function $f(x, y) = y^4 + x^3 + x\,y$, level curves, their prc

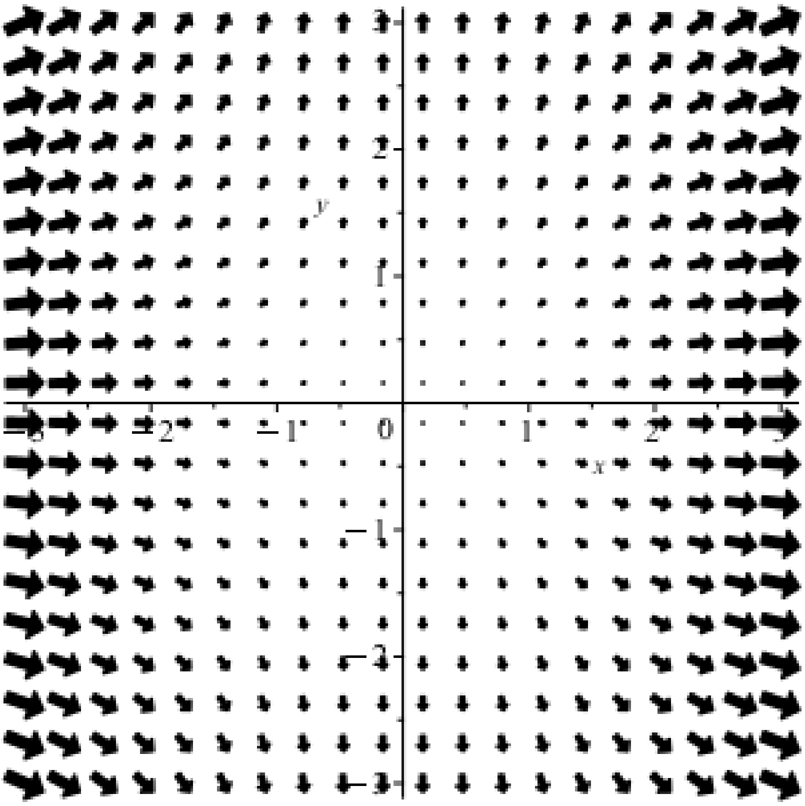
Gradient fields in 2 dimensions

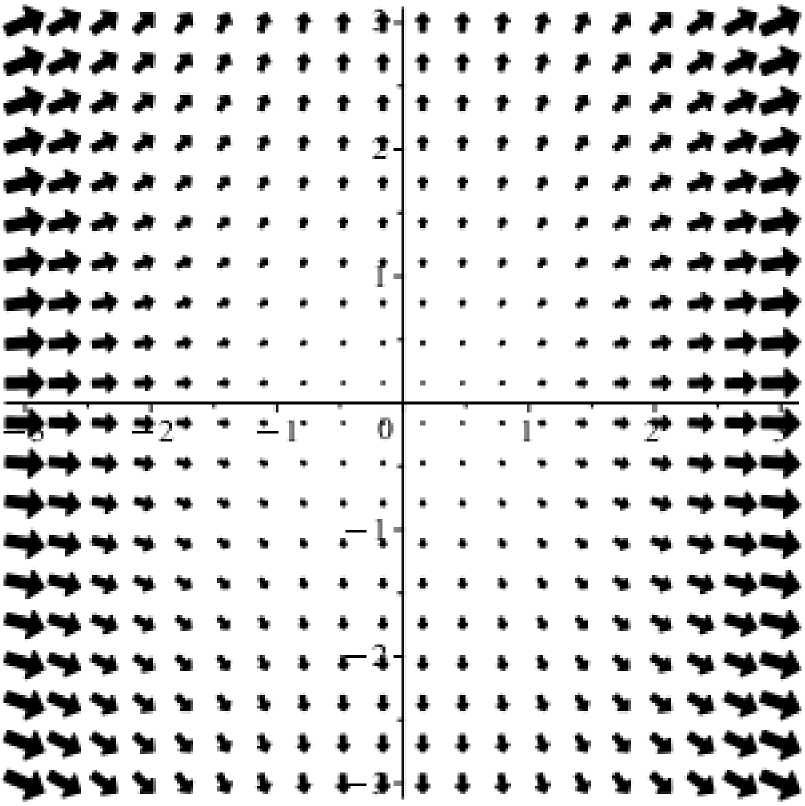
The most common kind of vector field we will be interested in plotting are vector fields that are produced as the gradients of a multivariable function.
We can either computer the gradient and then plot it or use the gradplot command. (Maple has a special command for plotting gradient fields.)

```
> ?gradplot
> f := (x, y) -> x^3 + 2*y^2;
  gradf := [diff(f(x,y),x), diff(f(x,y),y)];
  fieldplot(gradf, x=-3..3, y=-3..3, arrows=thick);
  gradplot(f(x,y), x=-3..3, y=-3..3, arrows=thick);
```

$$f := (x, y) \rightarrow x^3 + 2\,y^2$$

$$gradf := \left[3\,x^2, 4\,y\right]$$





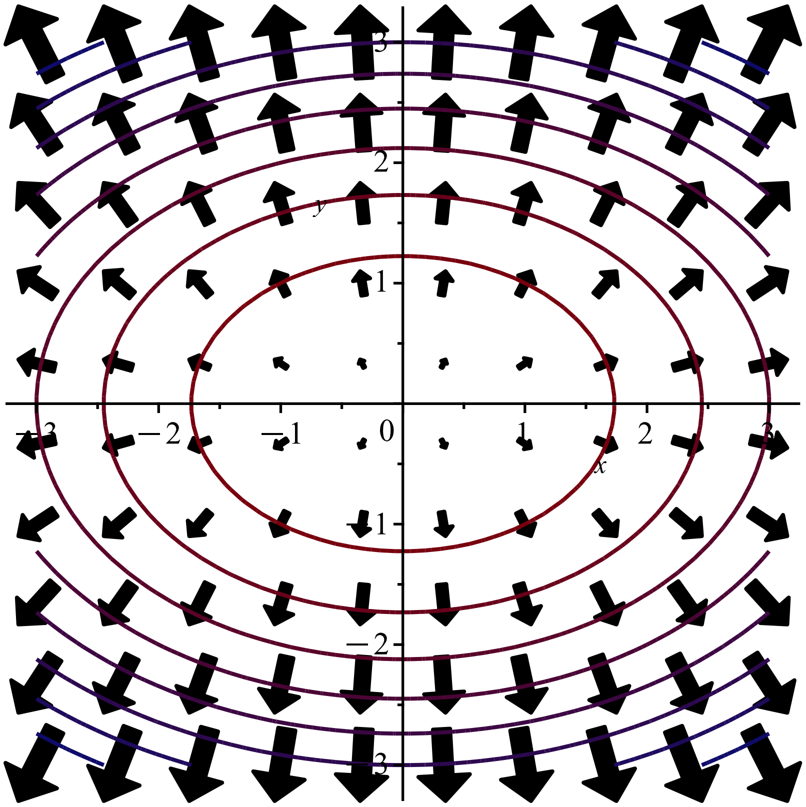
>

You should recall that the gradient vector is always perpendicular to the contours of the function. It is worthwhile to put a contourplot together with the gradplot.

(Note that we are putting 2 plots together with the display command. When creating the plots we use colons rather than semicolons.)

```
> f := (x, y) -> x^2 + 2*y^2;
  gradf := gradplot(f(x,y), x=-3..3, y=-3..3, arrows=thick, grid=
  [10, 10]):
  contourf := contourplot(f(x,y), x=-3..3, y=-3..3):
  display({gradf, contourf});
```

$$f := (x, y) \rightarrow x^2 + 2\,y^2$$



Exercises:

3) Plot the gradient field for the function x^2-y^3 over the window [-10, 10]x[-10, 10].

4) Plot the gradient field of x^2 - 2x + y^2 +4y - 8 togethr with a contour map of the same function on the region [-10, 10]x[-10, 10].

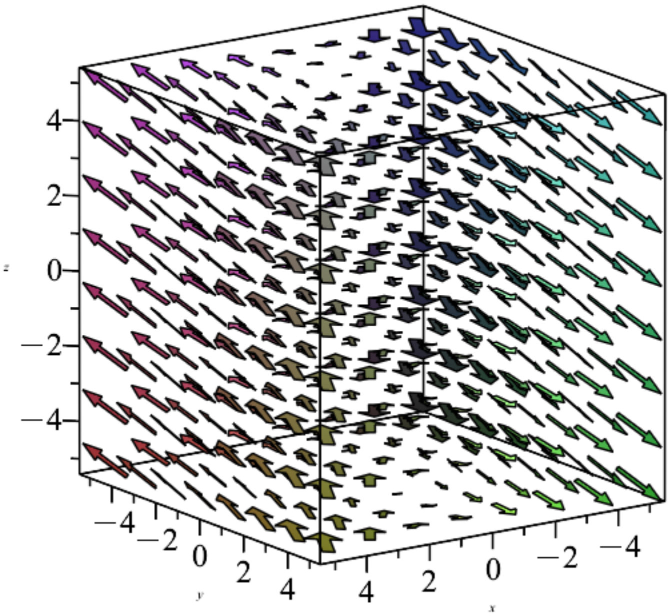
```
>
```

Vector and Gradient Fields in 3 dimensions

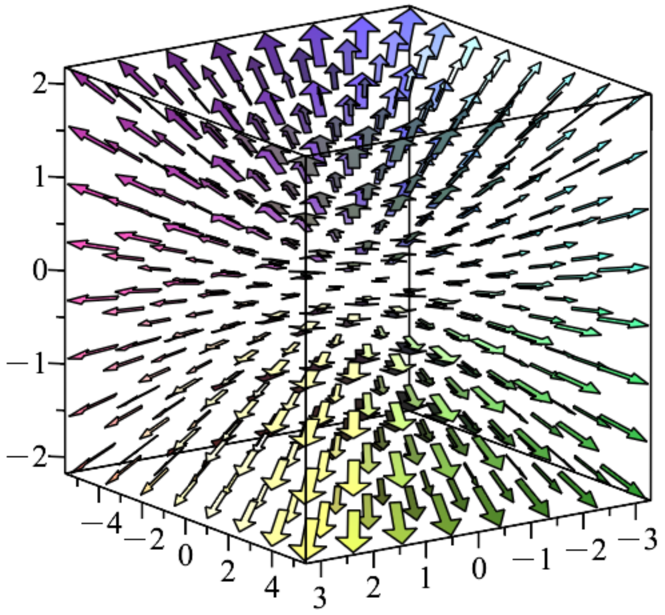
When we go to 3 dimensions, the theory remains the same. We simply change the commands to the ones appropriate for 3 dimenstions. The commands fieldplot and gradplot become fieldplot3d and gradplot3d.

To illoustrate we plot the gradient field of x^2 + y^2 + z^2 both as a vector field that we have computed and also as gradplot.

```
> fieldplot3d([2*x, 2*y, 2*x], x = −5..5, y = −5..5, z = −5..5,
     arrows = THICK, axes = boxed);
```



```
> gradplot3d(x^2 + y^2 + z^2, x = -3 ..3, y = -5 ..5, z = -2 ..2,
    arrows = THICK, axes = boxed);
```



Exercises:

5) Graph the following vector fields. (Problems 7 and 8 from section 18.2.) Descibe the vector field

in words.

(a) F(x, y, z) = xi + 2zyj + xk.

(b) F(x, y, z) = [x^3, y^2, z].

6) Plot the gradient fields of the following functions.

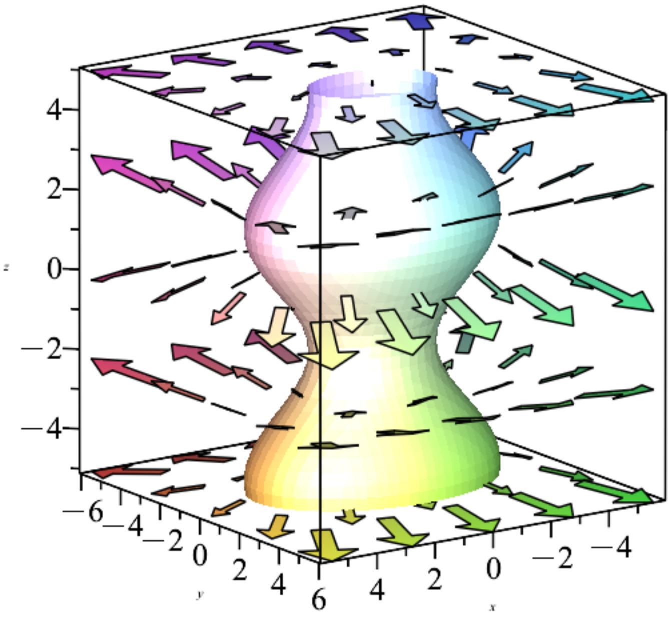
(a) f(x, y, z) = -x^2 - y^2 + z^2.

(b) f(x, y, z) = (x - 1)^2 + y^2 + z^2.

If we want to plot the gradient field together with the level surface of a function things get a little more complicated. The display command is replaced by the display3d command. Unfortuneatly, the contourplot3d does not give level surfaces. Instead we need to use implicitplot3d. Maple warns that implicitplot3d is memory intensive.

```
> ?implicitplot3d
> g := (x, y, z) -> x^2 + 2*y^2 - (3 + sin(z))^2;
  gradg := gradplot3d( g(x, y, z), x=-5..5, y=-5..5, z=-5..5,
      arrows=THICK, grid=[5, 5, 5]):
  surfg := implicitplot3d( g(x, y, z)=0, x=-5..5, y=-5..5, z=-5..5,
  style=patchnogrid, lightmodel=light1, grid=[33,33,33]):
  display3d({gradg, surfg});
```

$$g := (x, y, z) \rightarrow x^2 + 2\,y^2 - (3 + \sin(z))^2$$



```
>
```