

Vector Fields

```
> restart:with(plots):with(DEtools):with(VectorCalculus):
  setoptions3d(axes=NORMAL,labels=["x","y","z"],orientation=[20,
  70]);
  BasisFormat(false):
```

Vector Fields

We use the command [fieldplot](#) to draw vector fields. We begin entering the simple vector field $\mathbf{F}(x, y) = \langle x, y \rangle = x \mathbf{i} + y \mathbf{j}$ using the [VectorField](#) command.

```
> vf:=VectorField(<x,y>,'cartesian'[x,y]);
```

$$vf := \begin{bmatrix} x \\ y \end{bmatrix}$$

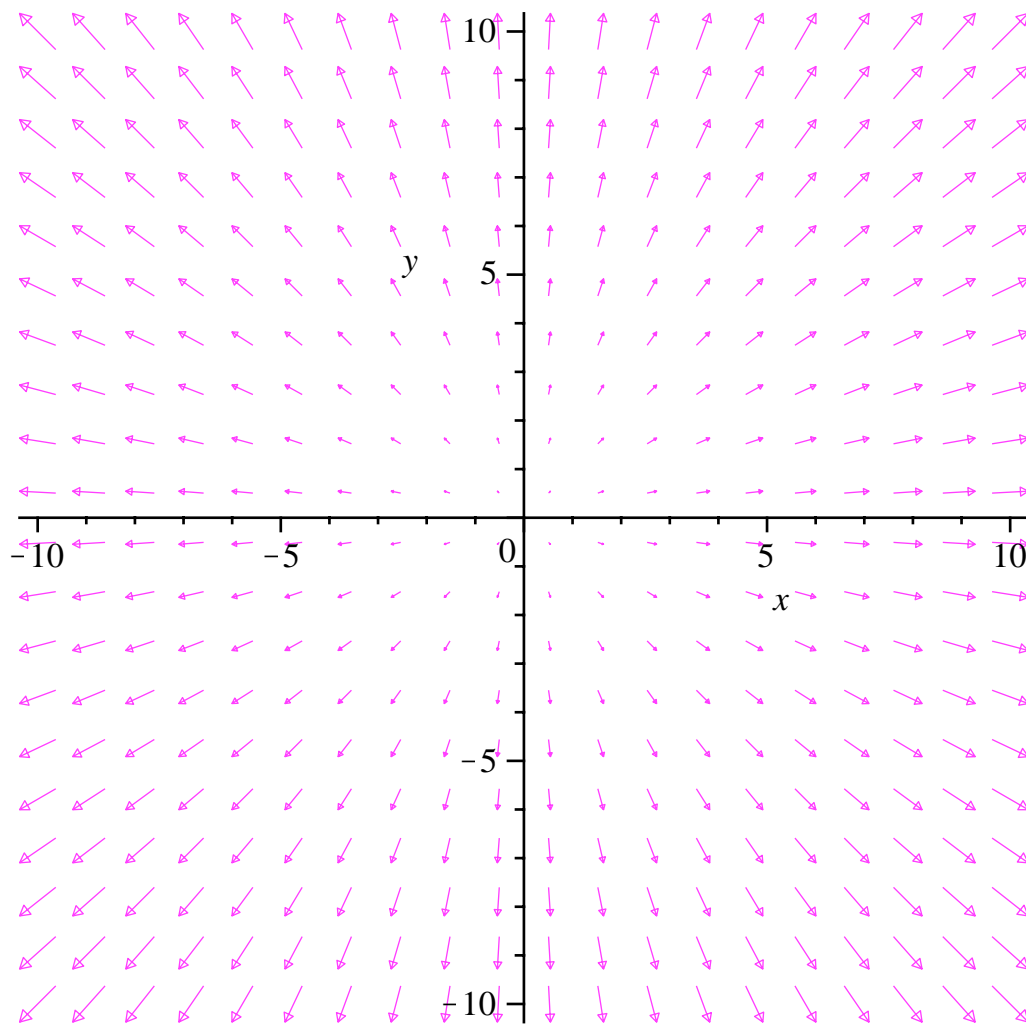
We can skip the last argument of the **fieldplot** command if we set the coordinate system using the [SetCoordinates](#) command.

```
> SetCoordinates('cartesian'[x,y]);
```

*cartesian*_{x,y}

We plot the vector field.

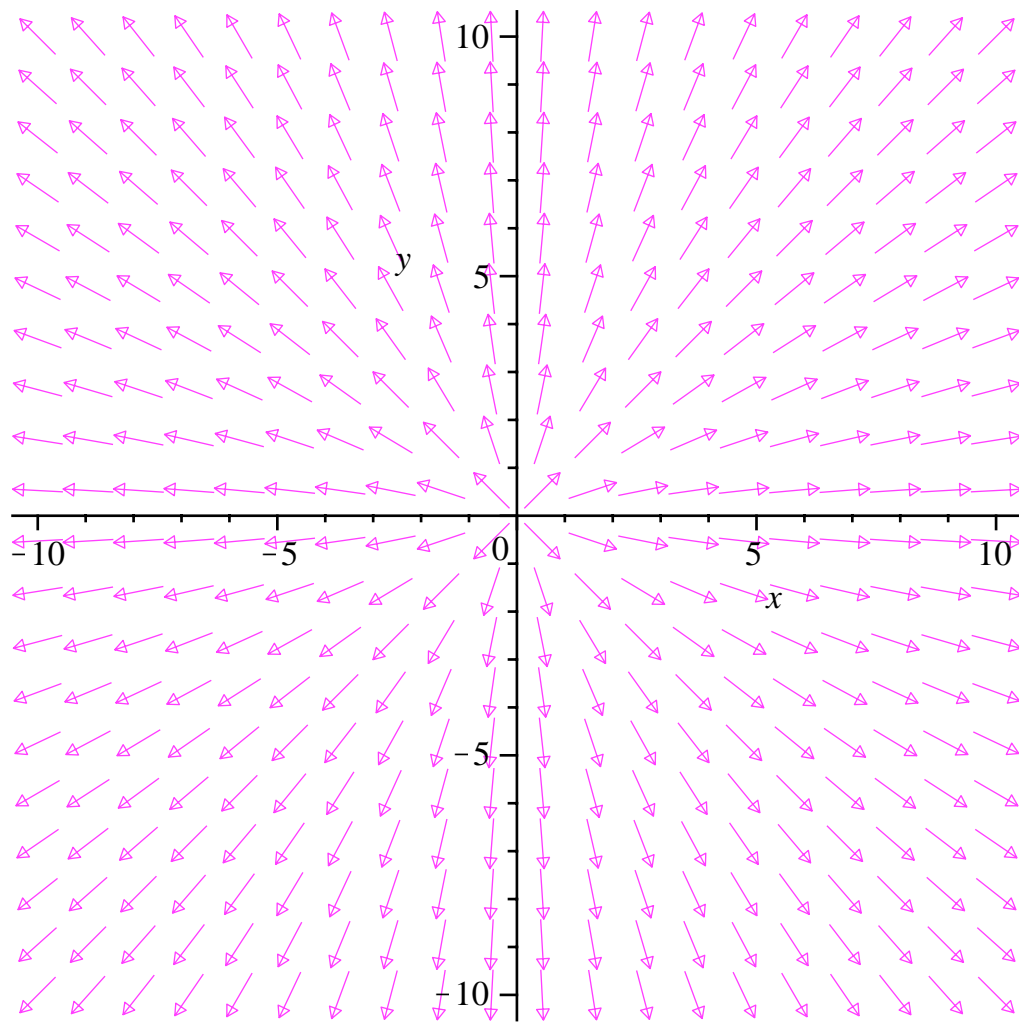
```
> fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);
```



We notice that Maple scales the vector field proportionally when the vectors are too long to give a good picture otherwise. Next we look at the vector field $\mathbf{F}(x, y) = \left\langle \frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}} \right\rangle = \frac{x}{\sqrt{x^2 + y^2}} \mathbf{i} + \frac{y}{\sqrt{x^2 + y^2}} \mathbf{j}$ that is the same as that above except for the fact that all vectors are unit vectors.

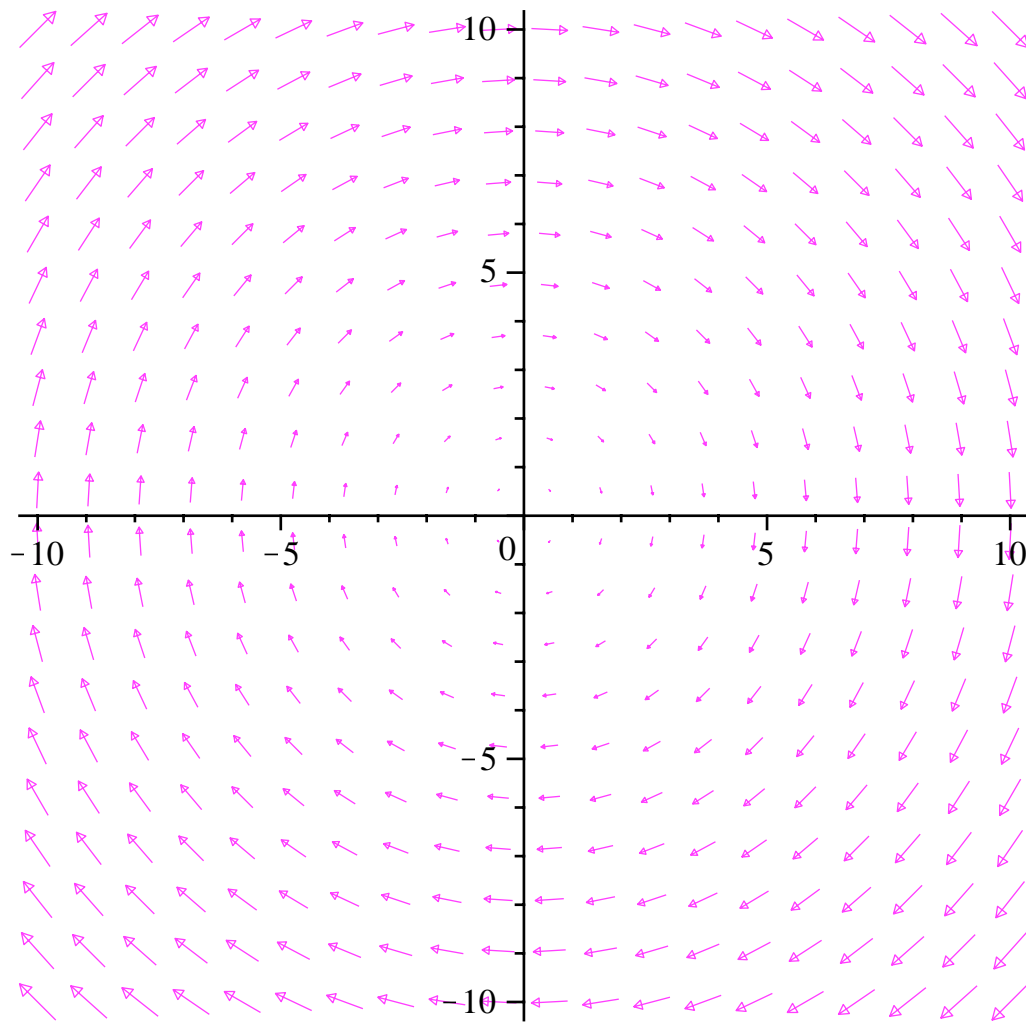
> **vf:=VectorField(<x/sqrt(x^2+y^2),y/sqrt(x^2+y^2)>);
fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);**

$$vf := \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \end{bmatrix}$$



Next we look at the "spin field" given by $\mathbf{F}(x, y) = (y, -x) = y \mathbf{i} - x \mathbf{j}$.

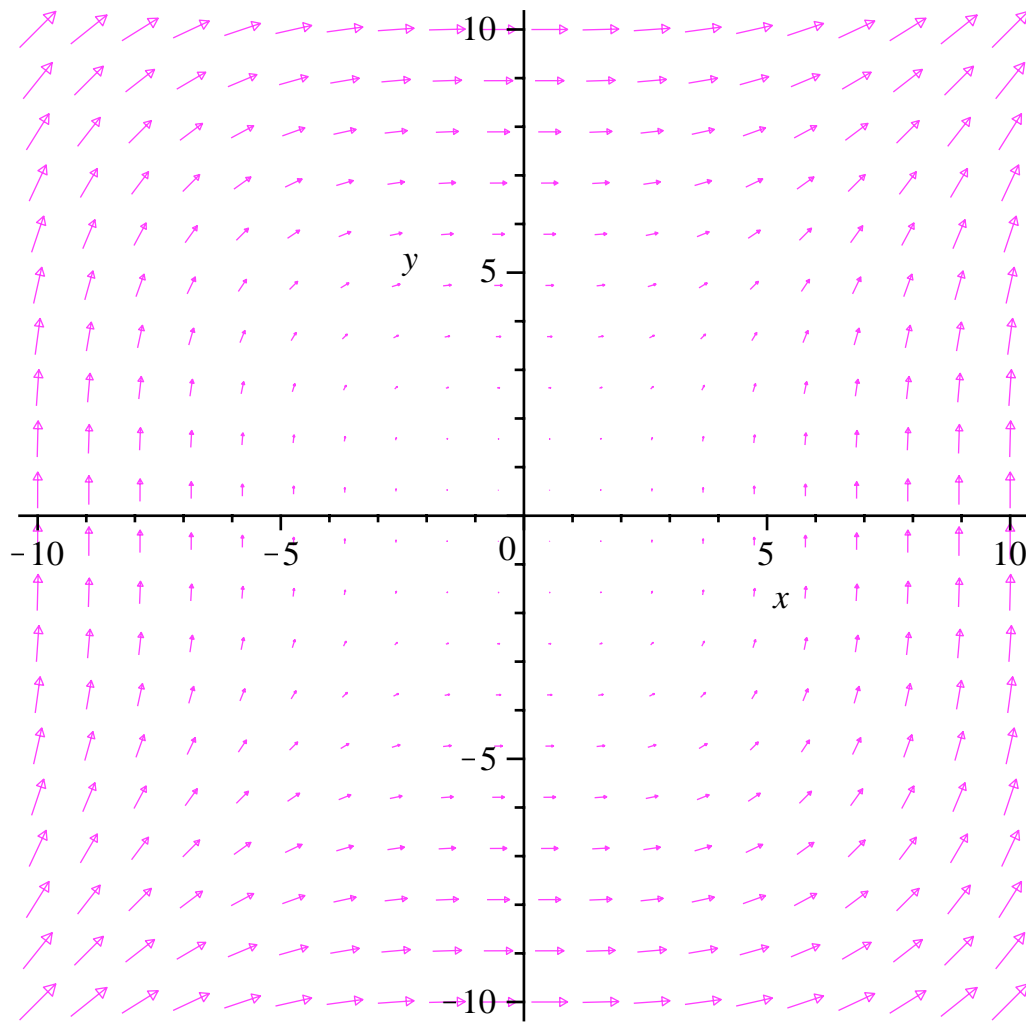
```
> vf:=VectorField(<y,-x>,'cartesian'[x,y]);
  fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);
      vf:=(y)ex-xey
```



We perceive a definite counter-clockwise flow. For a different effect, we look at $\mathbf{F}(x, y) = \langle y^2, x^2 \rangle = y^2 \mathbf{i} + x^2 \mathbf{j}$.

```
> vf:=VectorField(<y^2,x^2>);
   fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);
```

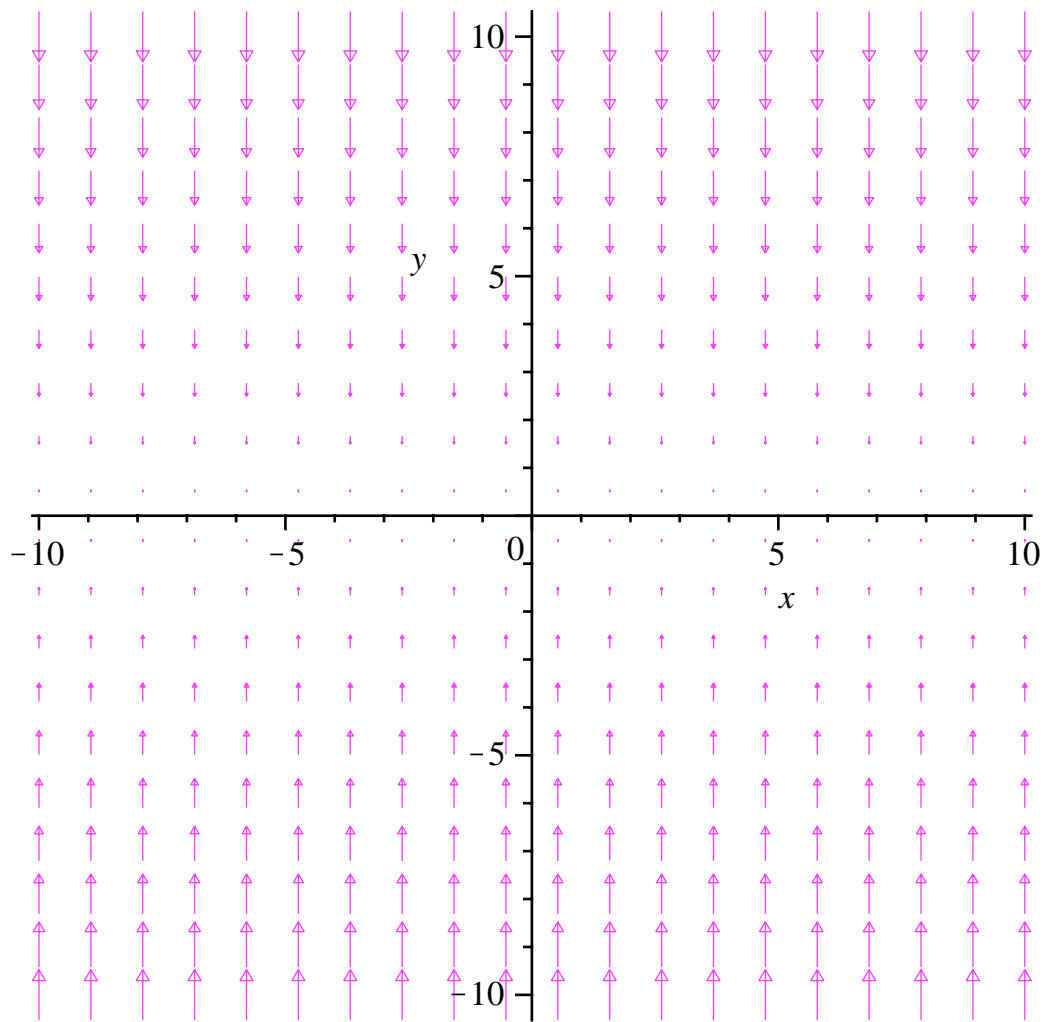
$$vf := \begin{bmatrix} y^2 \\ x^2 \end{bmatrix}$$



Next, let us consider the vector field $\mathbf{F}(x, y) = \langle 0, -3y \rangle = -3y\mathbf{j}$.

```
> vf:=VectorField(<0,-3*y>);
   fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);
```

$$vf := \begin{bmatrix} 0 \\ -3y \end{bmatrix}$$



Velocity Vector Fields

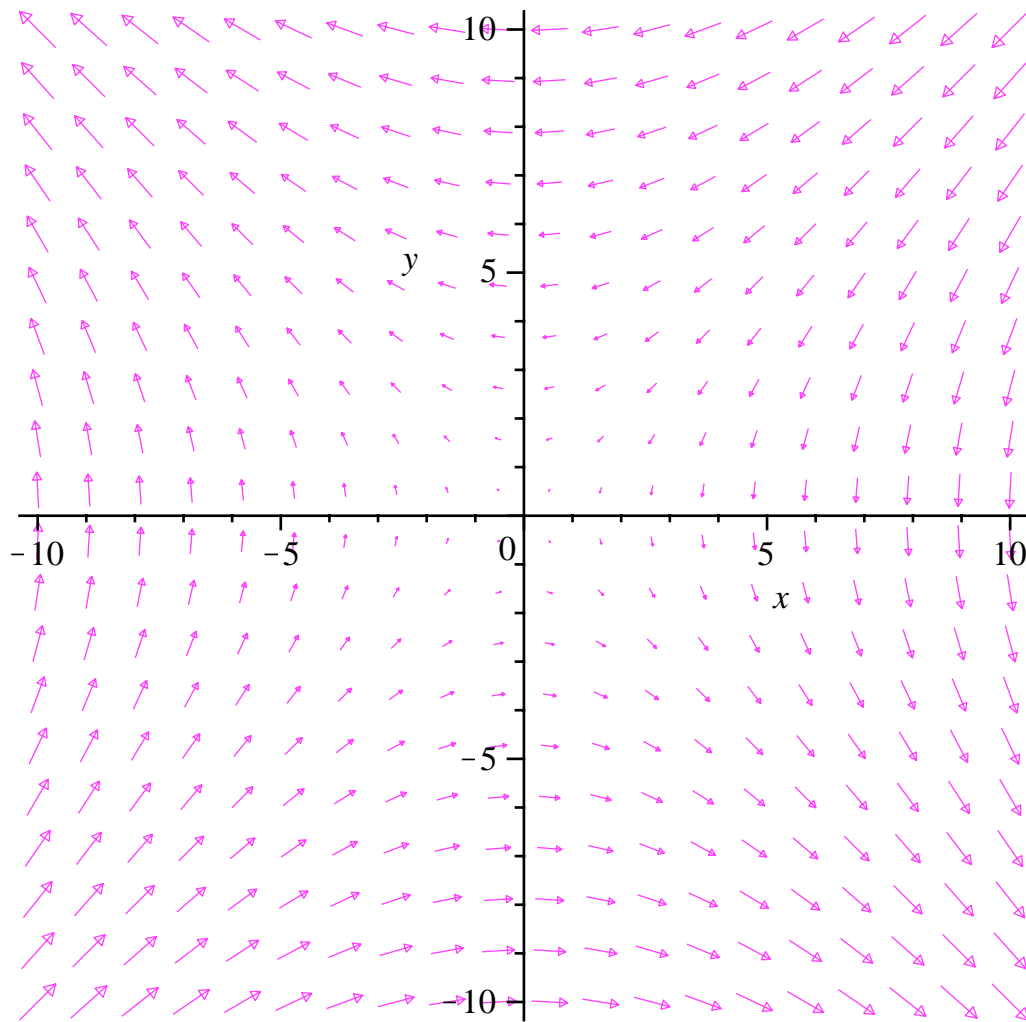
For a **velocity vector field**, $\mathbf{F}(x(t), y(t))$ gives the velocity of a "particle" at the point $(x(t), y(t))$ at the time t . Then the curve traced out by $\langle x(t), y(t) \rangle$ is a **flow line** where $x'(t) = f_1((x(t), y(t)))$ and $y'(t) = f_2((x(t), y(t)))$. In this case the velocity vector field is also a vector field for the differential

$$\text{equation } \frac{d}{dx} y = \frac{f_2((x(t), y(t)))}{f_1((x(t), y(t)))}.$$

Suppose the vector field is $\mathbf{F}(x(t), y(t)) = \langle -y, -x \rangle$.

```
> vf:=VectorField(<-y,-x>);
   fieldplot(vf,x=-10..10,y=-10..10,color=magenta,arrows=SLIM);
```

$$vf := \begin{bmatrix} -y \\ -x \end{bmatrix}$$



This is a velocity vector field for the differential equation $\frac{\partial}{\partial x} y = \frac{y}{x}$. Suppose we want to find the flow line through the point (3,4). This corresponds to the initial condition $y(3) = 4$. We enter the differential equation and the initial condition.

```
> de:=diff(y(x),x)=x/y(x);
```

$$de := \frac{d}{dx} y(x) = \frac{x}{y(x)}$$

```
> ic:=y(3)=4;
```

$$ic := y(3) = 4$$

We find the solution of the differential equation.

```
> soln:=dsolve({de,ic},y(x));
```

$$soln := y(x) = \sqrt{x^2 + 7}$$

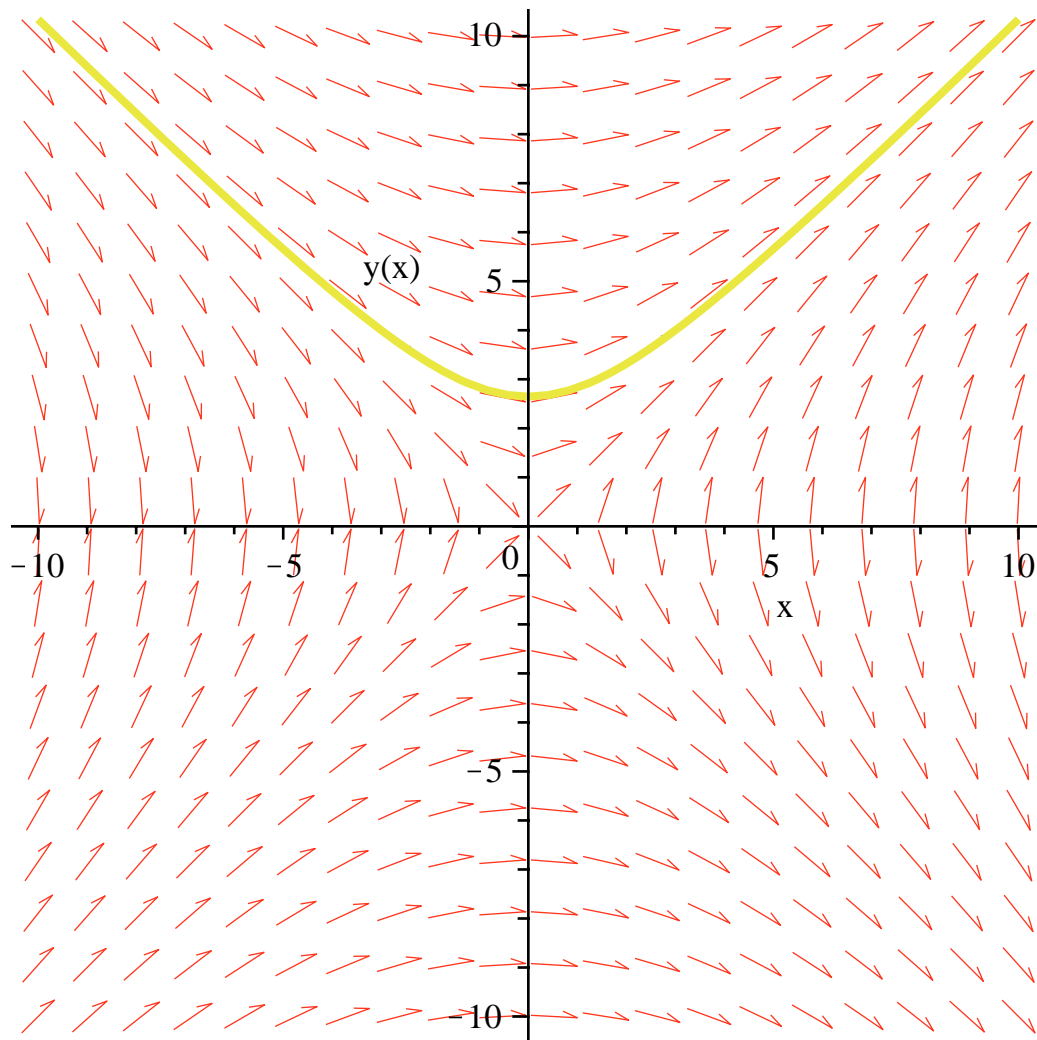
We use the command [DEplot](#) from the [DEtools](#) package, which we load, to plot the direction field (showing direction, but not relative velocity) with the solution or flow line through (3,4).

```
> with(DEtools);
```

[[AreSimilar](#), [DENormal](#), [DEplot](#), [DEplot3d](#), [DEplot_polygon](#), [DFactor](#), [DFactorLCLM](#), [DFactorsols](#), [Dchangevar](#), [FunctionDecomposition](#), [GCRD](#), [Gosper](#), [Heunsols](#), [Homomorphisms](#),

IsHyperexponential, LCLM, MeijerGsols, MultiplicativeDecomposition, PDEchangecoords, PolynomialNormalForm, RationalCanonicalForm, ReduceHyperexp, RiemannPsols, Xchange, Xcommutator, Xgauge, Zeilberger, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diff_table, diffop2de, dperiodic_sols, dpolyform, dsubs, eigenring, endomorphism_charpoly, equinv, η k, eulersols, exactsol, expsols, exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic, genhomosol, gensys, hamilton_eqs, hypergeomsols, hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor, invariants, kovacicols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom]

> **DPlot(de, y(x), x=-10..10, y=-10..10, [[3,4]]);**



Gradient Vector Fields

The gradient of a function is also a vector field. We consider the function $z = f(x, y) = \frac{x^2}{4} + \frac{y^2}{9}$.

```
> z:=x^2/4+y^2/9;
```

$$z := \frac{1}{4}x^2 + \frac{1}{9}y^2$$

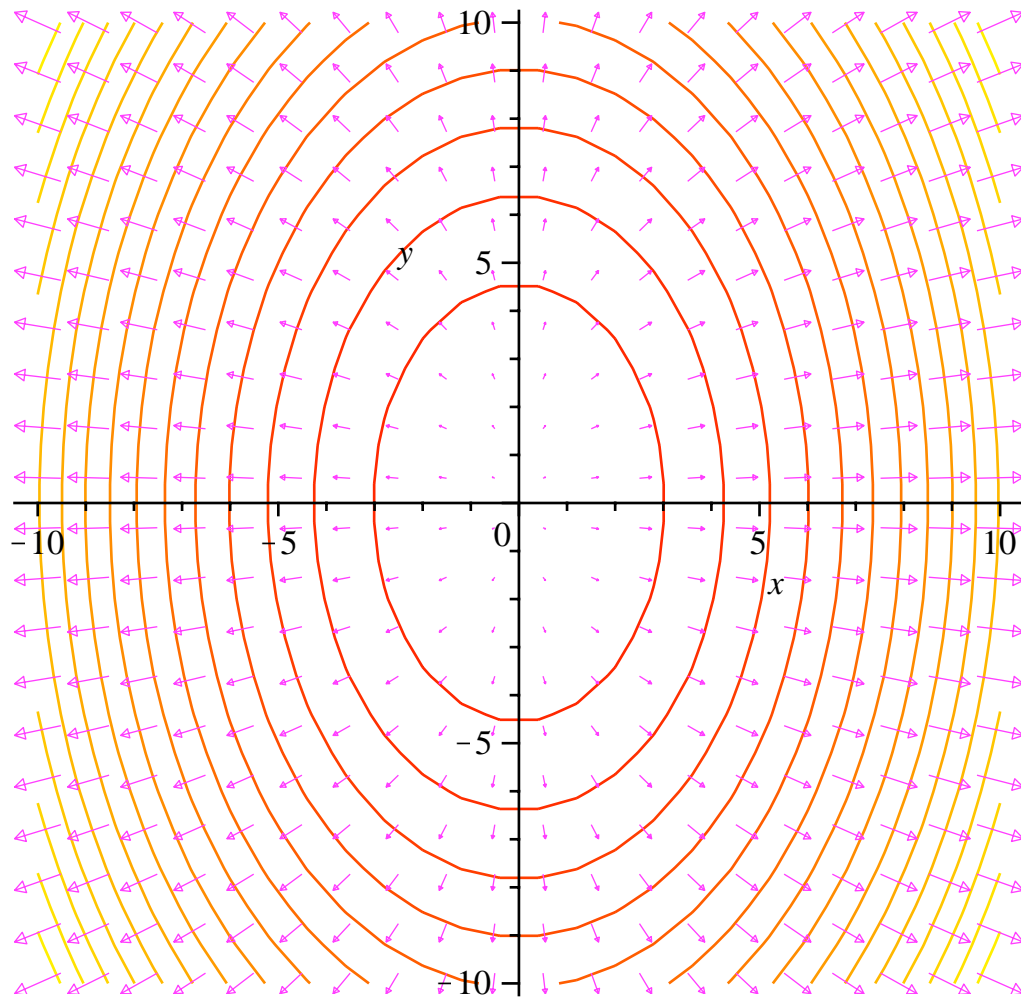
We find the gradient of the function using the [Gradient](#) command.

```
> gradf:=Gradient(z,[x,y]);
```

$$\text{gradf} := \begin{bmatrix} \frac{1}{2}x \\ \frac{2}{9}y \end{bmatrix}$$

Recalling that the vectors of the gradient field are perpendicular to its level curves, we plot the gradient field of the above function along with a contour plot.

```
> p0:=contourplot(z, x=-10..10,y=-10..10,contours=15):
p1:=gradplot(z, x=-10..10,y=-10..10,arrows=slim,color=magenta):
display(p0,p1,scaling=constrained);
```

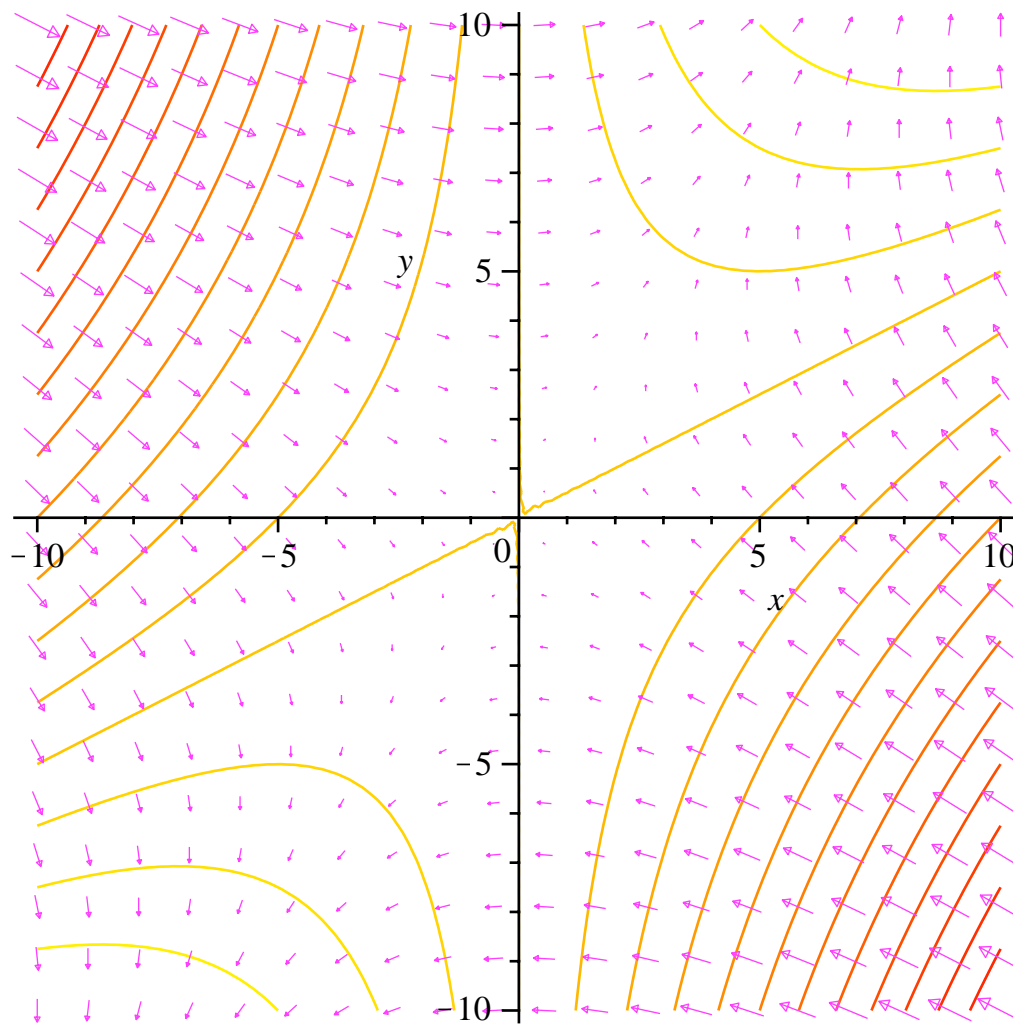


Next we look at $z = f(x, y) = 2xy - x^2$.

```
> z:=2*x*y-x^2;
   gradf:=Gradient(z,[x,y]);
   p0:=contourplot(z, x=-10..10,y=-10..10,contours=15,numpoints=
5000):
   p1:=gradplot( z, x=-10..10,y=-10..10,arrows=slim,color=magenta):
   display(p0,p1);
```

$$z := 2xy - x^2$$

$$\text{gradf} := \begin{bmatrix} 2y - 2x \\ 2x \end{bmatrix}$$



Three Dimensions

The earth's gravitational vector field can be given by the vector function $\mathbf{F}(\mathbf{r}) = -k \mathbf{r} / \|\mathbf{r}\|^3$. We plot the three-dimensional vector field with $k = 1$.

```
> z:='z';
   SetCoordinates('cartesian'[x,y,z]);
   r:=VectorField(<x,y,z>);
```

```
z:=z
cartesianx,y,z
```

$$r := \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

For our next step, we need the version of [Norm](#) found in the [LinearAlgebra](#) package.

```
> with(LinearAlgebra);
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm,
 BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension,
 ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix,
```

ConstantVector, Copy, CreatePermutation, CrossProduct, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, LA_Main, LUDecomposition, LeastSquares, LinearSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

> **F:=-1/(Norm(r,2)^3)*r;**

$$F := \begin{bmatrix} -\frac{x}{(|x|^2 + |y|^2 + |z|^2)^{3/2}} \\ -\frac{y}{(|x|^2 + |y|^2 + |z|^2)^{3/2}} \\ -\frac{z}{(|x|^2 + |y|^2 + |z|^2)^{3/2}} \end{bmatrix}$$

> **fieldplot3d(F,x=-5..5,y=-5..5,z=-5..5,color=magenta,arrows=SLIM);**

