Maple Multivariate Calculus Commands

This is a very quick and dirty very short list of Maple commands you might find useful in Multivariate Calculus. If you don't understand what some of these commands do, don't worry; but remember you have this document later when we've defined all the ideas here.

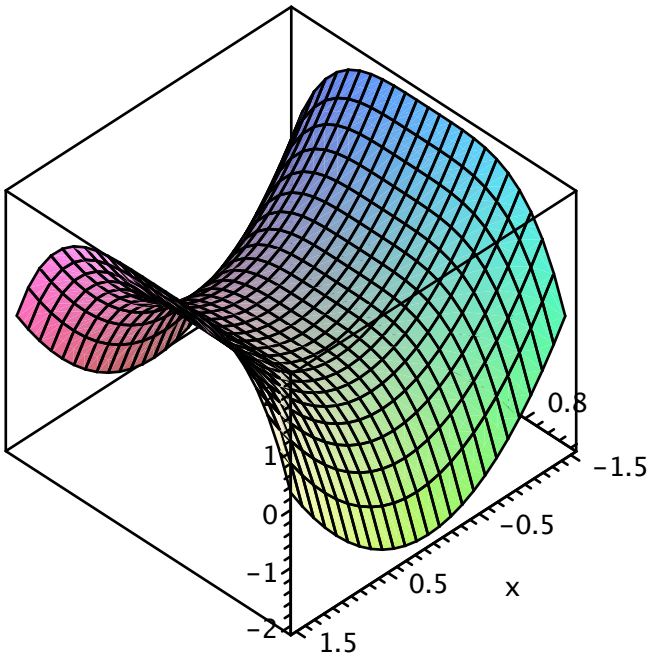
Plots

Most of what you want is in the **plots** package. Look at the new commands that get defined when you load **plots**, and ask for help on any that look interesting. To get axes and boxes to show in some of these plots, I had to click on the plot itself and use the contextual toolbar. You can also grab plots and rotate them to see them from different angles.

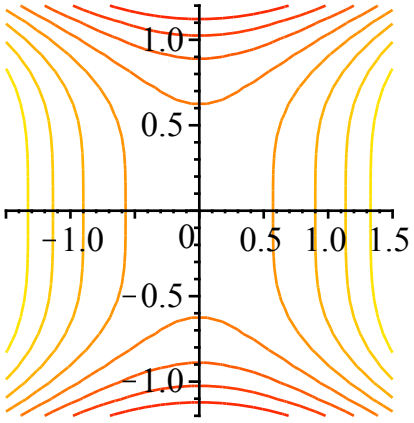
```
> with(plots);
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *densityplot*, *display*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *graphplot3d*, *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*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

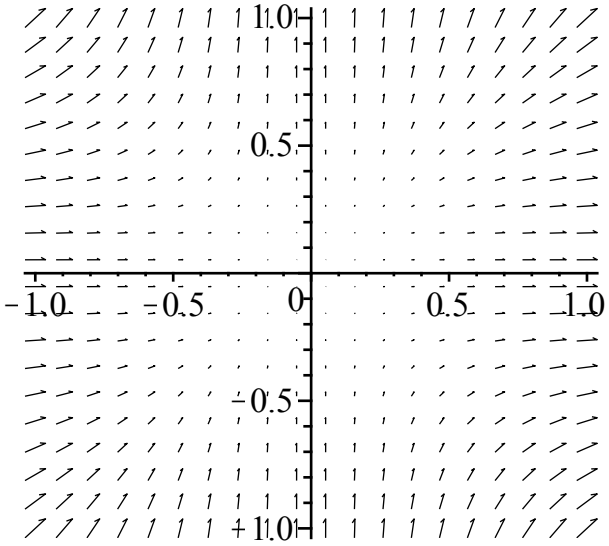
```
> plot3d(x^2-y^4, x=-1.5..1.5,y=-1.2..1.2);
```



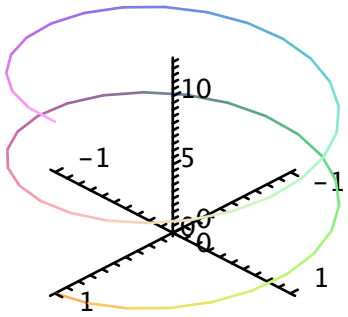
```
> contourplot(x^2-y^4, x=-1.5..1.5,y=-1.2..1.2);
```




```
> fieldplot([x^2,y^2], x=-1..1,y=-1..1);
```




```
> spacecurve([cos(t),sin(t),t],t=0..4*Pi);
```



Matrices and Vectors

These live in the package **LinearAlgebra**. Again, look at the names of the new functions and ask Maple for help if you need something less basic than the examples here.

```
> 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*]

```
> V := Vector([3,1,4]);
```

$$V := \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$$

```
> V[1];
```

$$3$$

```
> V[3];
```

$$4$$

```
> DotProduct(V, Vector([1,-3,2]));
```

$$8$$

```
> CrossProduct(Vector([3,1,4]), Vector([1,-3,2]));
```

$$\begin{bmatrix} 14 \\ -2 \\ -10 \end{bmatrix}$$

```
> M := Matrix([[1,2,3],[3,1,4],[-1,1,1]]);
```

$$M := \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 4 \\ -1 & 1 & 1 \end{bmatrix}$$

```
> M[2,3];
```

$$4$$

```
> Determinant(M);
```

$$-5$$

Multivariate Functions, Divergence, Gradient, and Curl

Here are some examples of these concepts. They are defined within the **VectorCalculus** package, so you need to load it before calling these functions.

```
> with(VectorCalculus);
```

[*&x*, `*`, `+`, `-`, `.`, *<,>*, *<|>*, *About*, *AddCoordinates*, *ArcLength*, *BasisFormat*, *Binormal*, *Compatibility*, *ConvertVector*, *CrossProd*, *CrossProduct*, *Curl*, *Curvature*, D, *Del*, *DirectionalDiff*, *Divergence*, *DotProd*, *DotProduct*, *Flux*, *GetCoordinateParameters*, *GetCoordinates*, *GetPVDescription*, *GetRootPoint*, *GetSpace*, *Gradient*, *Hessian*, *Jacobian*, *Laplacian*, *LineInt*, *MapToBasis*, *Nabla*, *Norm*, *Normalize*, *PathInt*, *PlotPositionVector*, *PlotVector*, *PositionVector*, *PrincipalNormal*, *RadiusOfCurvature*, *RootedVector*, *ScalarPotential*, *SetCoordinateParameters*, *SetCoordinates*, *SpaceCurve*, *SurfaceInt*, *TNBFrame*, *Tangent*, *TangentLine*, *TangentPlane*, *TangentVector*, *Torsion*, *Vector*, *VectorField*, *VectorPotential*, *VectorSpace*, *Wronskian*, *diff*, *eval*, *evalVF*, *int*, *limit*, *series*] **(1)**

We want to work in Cartesian coordinates.

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

$$cartesian_{x,\,y,\,z} \quad \textbf{(2)}$$

```
> f := x -> [cos(x), sin(x), x];
```

$$f := x \rightarrow [\cos(x), \sin(x), x]$$

```
> f(Pi);
```

$$[-1, 0, \pi]$$

```
> diff(f(x),x);
```

$$[-\sin(x), \cos(x), 1]$$

```
> F := (x,y,z) -> x^2 + y*sin(z);
```

$$F := (x, y, z) \rightarrow x^2 + y \sin(z)$$

```
> F(1,2,z);
```

$$1 + 2 \sin(z)$$

```
> Gradient(F(x,y,z),[x,y,z]);
```

$$2x\bar{e}_x + (\sin(z))\bar{e}_y + (y\cos(z))\bar{e}_z$$

Here e_x, e_y, and e_z are the unit vectors in the x, y, and z directions, resp.

```
> G := VectorField(<-y,x,z^2>);
```

$$G := -y\bar{e}_x + (x)\bar{e}_y + (z^2)\bar{e}_z$$

```
> Divergence(G);
```

$$2z$$

```
> Curl(G);
```

$$2\bar{e}_z$$

This is only a very slight beginning. There are lots more functions in all these opackages. There are also subpackages of the **Student** package called **Student[LinearAlgebra]** and **Student [VectorCalculus]**. A wealth of useful stuff can be found on the help pages for these packages.