Good to reset everything, if needed:

```
> restart;
```

Load the package VectorCalculus, which is very good for Math 2.

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

[&x, `*`, `+`, `-`, `.`, <,>, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal, Compatibility, ConvertVector, CrossProduct, Curl, Curvature, D, Del, DirectionalDiff, Divergence, DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames, GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, 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)**

Load the package plots, which is always very good.

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

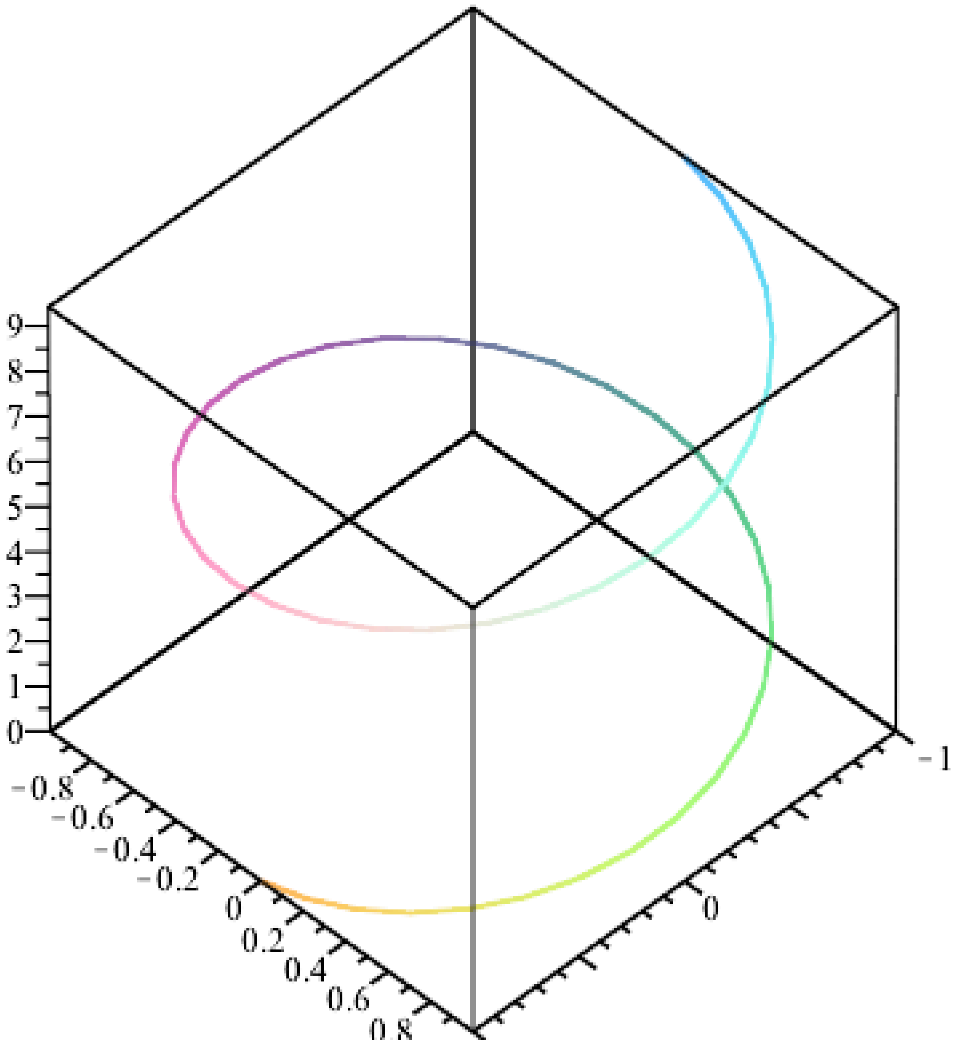
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, 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, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot] **(2)**

Define a function/expression in t (a helix curve).

```
> f(t) :=  <cos(t),sin(t),t>;
```

Plot the corresponding curve in space, on some interval, with boxed axes.

```
> spacecurve(f(t),t=0..3*Pi,axes=boxed);
```



Calculate (and simplify) some obvious entities.

```
> TangentVector(f(t),normalized);
```

$$1 \tag{3}$$

```
> PrincipalNormal(f(t),normalized);
```

$$h(t) := \begin{bmatrix} -\cos(t) \\ -\sin(t) \\ 0 \end{bmatrix} \tag{4}$$

```
> Curvature(f(t));
```

$$\frac{1}{4}\sqrt{2\cos(t)^2 + 2\sin(t)^2}\,\sqrt{2} \tag{5}$$

```
> simplify(%);
```

$$\frac{1}{2} \tag{6}$$

```
> Binormal(f(t),normalized);
```

$$j(t) := \begin{bmatrix} \frac{1}{2}\sqrt{2}\,\sin(t) \\ -\frac{1}{2}\sqrt{2}\,\cos(t) \\ \frac{1}{2}\sqrt{2} \end{bmatrix} \tag{7}$$

```
> Torsion(f(t));
```

$$\frac{1}{2}\,\frac{\sqrt{2}\,\sin(t)^2}{\sqrt{2\cos(t)^2+2\sin(t)^2}\,\sqrt{\left(\frac{1}{4}+\frac{1}{4}\sin(t)^2+\frac{1}{4}\cos(t)^2\right)\left(1+\sin(t)^2+\cos(t)^2\right)}} + \frac{1}{2}\left(\sqrt{2}\,\cos(t)^2\right) \Big/ \left(\sqrt{2\cos(t)^2+2\sin(t)^2}\,\sqrt{\left(\frac{1}{4}+\frac{1}{4}\sin(t)^2+\frac{1}{4}\cos(t)^2\right)\left(1+\sin(t)^2+\cos(t)^2\right)}\right) \tag{8}$$

```
> simplify(%);
```

$$\frac{1}{2} \tag{9}$$

Below, a different way of defining the same curve:

```
> helix:=PositionVector([cos(t),sin(t),t]);
```

$$helix := \begin{bmatrix} \cos(t) \\ \sin(t) \\ t \end{bmatrix} \tag{10}$$

Plotting the curve with the additional choices 'tangent', 'normal' and 'binormal' shows the geometrical meaning of the TNB-frame along a curve in three-dimensional space.

```
> PlotPositionVector(helix, t=0..Pi, points=[Pi/2], tangent,normal,
  binormal);
```