Notebook 14: Vector-Valued Functions and Motion in Space

Vector-Valued Functions

The *VectorCalculus* package provides many tools for working with vectors and vector-valued functions.

> $with(VectorCalculus)$:
> $r(t) := \langle a(t), b(t), c(t) \rangle$

$$r := t \rightarrow VectorCalculus\text{:-}{<,>}(a(t), b(t), c(t))$$

The output is radically different than other function definitions. This is just Maple's way of showing that is using *VectorCalculus* commands ot define the function.

> $r(t)$

$$(a(t))e_x + (b(t))e_y + (c(t))e_z$$

Maple uses e_x, e_y, and e_z for the standard basis vectors **i**, **j**, and **k** (or ⟨1, 0, 0⟩, ⟨0, 1, 0⟩, and ⟨0, 0, 1⟩).

Differentiation is unchanged.

> $r'(t)$;
> $diff(r(t), t)$

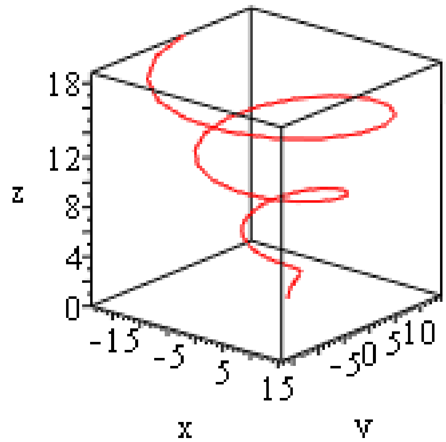
$$\left(\frac{d}{dt}\,a(t)\right)e_x + \left(\frac{d}{dt}\,b(t)\right)e_y + \left(\frac{d}{dt}\,c(t)\right)e_z$$
$$\left(\frac{d}{dt}\,a(t)\right)e_x + \left(\frac{d}{dt}\,b(t)\right)e_y + \left(\frac{d}{dt}\,c(t)\right)e_z$$

The *spacecurve* command in the *plots* package will graph a space curve.

> $with(plots)$:
> $r(t) := \langle \sin(t) - t\cdot\cos(t), \cos(t) + t\cdot\sin(t), t \rangle$:

Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

> $spacecurve(r(t), t = 0..6\pi, color = red, axes = boxed, labels = ["x", "y", "z"], orientation = [-50, 70]);$
$Curve := \% :$



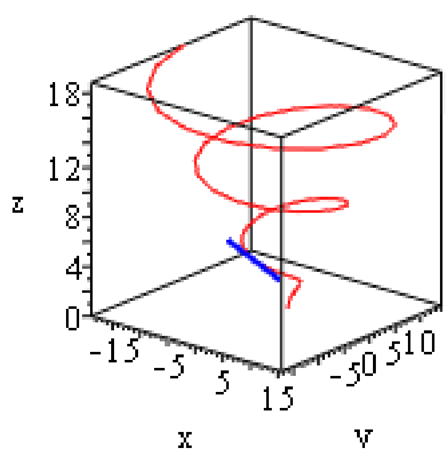
The tangent vector at $t_0 = \frac{3\pi}{2}$ is calculated below and then used to parameterize the line determined by the tangent vector at $r(t_0)$. The variable t_0 is used as the name of the number $\frac{3\pi}{2}$ because t must remain a free variable for plotting purposes.

> $t_0 := \frac{3\pi}{2} : VelocityVector = r'(t_0);$
$T(t) := r(t_0) + r'(t_0) \cdot t : T(t)$

$$VelocityVector = -\frac{3}{2}\pi e_x + e_z$$

$$\left(-1 - \frac{3}{2} t\pi\right)e_x - \frac{3}{2}\pi e_y + \left(\frac{3}{2}\pi + t\right)e_z$$

> $display([Curve, spacecurve(T(t), t=-1..1, color = blue, thickness = 2)])$



Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

T, N, B, Curvature, and Torsion

The *Normalize* command in the *VectorCalculus* package normalizes a given vector to unit length. The *Norm* command (also from the *VectorCalculus* package) will calculate the norm (or length) of a vector. The command to calculate the Euclidean length of a vector v is $Norm(v, 2)$. This may also be written as $\|v\|_2$ in 2D Input; the Norm template $\|x\|$ can be entered by typing Norm then pressing **[esc]/[enter]**, or the double vertical bars can be input either manually (as two vertical lines, |) or from the Common Symbols palette. Regardless of how the template is formed, the subscript 2 must be added manually. To put the cursor in the subscript position, press the underscore key _ (**[shift]-[minus]**).

> $v := Normalize(\langle 1, 1, 1\rangle);\ \|v\|_2$

$$v := \frac{1}{3}\sqrt{3}\,e_x + \frac{1}{3}\sqrt{3}\,e_y + \frac{1}{3}\sqrt{3}\,e_z$$
$$1$$

The cross product operator × can be found in the Common Symbols palette, or can be entered manually by typing times then pressing **[esc]/[enter]**.

> $\langle 1, 2, 3\rangle \times \langle 3, 2, 1\rangle$

$$-4e_x + 8e_y - 4e_z$$

The dot product of two vectors is obtained by placing a dot between the two vectors.

> $\langle 1, 2, 3\rangle.\langle 3, 2, 1\rangle$

$$10$$

Consider the trajectory $r(t)$ below and $t_0 = \ln(2)$.

> $r(t) := \langle e^t\cos(t), e^t\sin(t), e^t\rangle :'r(t)' = r(t);$
> $t_0 := \ln(2)$

$$r(t) = (e^t\cos(t))e_x + (e^t\sin(t))e_y + (e^t)e_z$$
$$t_0 := \ln(2)$$

Velocity, acceleration, and speed are calculated readily.

> $v(t) := r'(t) : a(t) := v'(t) : speed(t) := \|v(t)\| :$
> $'v(t)' = v(t); 'a(t)' = a(t); 'speed(t)' = speed(t)$

$$v(t) = (e^t\cos(t) - e^t\sin(t))e_x + (e^t\sin(t) + e^t\cos(t))e_y + (e^t)e_z$$
$$a(t) = -2\,e^t\sin(t)e_x + 2\,e^t\cos(t)e_y + (e^t)e_z$$
$$speed(t) = \sqrt{3}\,\sqrt{e^{2t}}$$

Evaluated at t_0, these quantities are

> $velocity = v(t_0);\ evalf[4](\%);$
> $acceleration = a(t_0);\ evalf[4](\%);$
> $speed = speed(t_0);\ evalf[4](\%)$

Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

$$velocity = (2\cos(\ln(2)) - 2\sin(\ln(2)))e_x + (2\sin(\ln(2)) + 2\cos(\ln(2)))e_y + 2e_z$$

$$velocity = (0.261)e_x + (2.817)e_y + (2.)e_z$$

$$acceleration = -4\sin(\ln(2))e_x + 4\cos(\ln(2))e_y + 2e_z$$

$$acceleration = (-2.556)e_x + (3.077)e_y + (2.)e_z$$

$$speed = 2\sqrt{3}$$

$$speed = 3.464$$

To define the unit tangent function, first normalize the velocity vector.

> $Normalize(r'(t))$

$$-\frac{1}{3}\frac{\sqrt{3}\,e^t(\sin(t) - \cos(t))}{\sqrt{e^{2t}}}e_x + \frac{1}{3}\frac{\sqrt{3}\,e^t(\cos(t) + \sin(t))}{\sqrt{e^{2t}}}e_y + \frac{1}{3}\frac{\sqrt{3}\,e^t}{\sqrt{e^{2t}}}e_z$$

This expression simplifies nicely if *t* is a real number. Maple must be told either to assume that *t* is real or to ignore complications of complex functions when simplifying. In Maple 13, the option *symbolic* in the *simplify* command takes care of this issue.

> $simplify(\%, symbolic)$

$$-\frac{1}{3}\sqrt{3}\,(\sin(t) - \cos(t))e_x + \frac{1}{3}\sqrt{3}\,(\cos(t) + \sin(t))e_y + \frac{1}{3}\sqrt{3}\,e_z$$

Now that the normalized velocity vector has been simplified, make it into the unit tangent function.

> $T := unapply(\%, t)$:'$T(t)$' = $T(t)$

$$T(t) = -\frac{1}{3}\sqrt{3}\,(\sin(t) - \cos(t))e_x + \frac{1}{3}\sqrt{3}\,(\cos(t) + \sin(t))e_y + \frac{1}{3}\sqrt{3}\,e_z$$

Next, define the unit normal and binormal functions

> $N := unapply(\,Normalize(T'(t)), t)$:'$N(t)$' = $N(t)$

$$N(t) = -\frac{1}{2}\sqrt{2}\,(\cos(t) + \sin(t))e_x - \frac{1}{2}\sqrt{2}\,(\sin(t) - \cos(t))e_y$$

> $B := unapply(simplify(T(t) \times N(t)), t)$:'$B(t)$' = $B(t)$

$$B(t) = \frac{1}{6}\sqrt{3}\sqrt{2}\,(\sin(t) - \cos(t))e_x - \frac{1}{6}\sqrt{3}\sqrt{2}\,(\cos(t) + \sin(t))e_y + \frac{1}{3}\sqrt{3}\sqrt{2}\,e_z$$

Evaluated at t_0, these quantities are

> $UnitTangent = T(t_0)$; $evalf[4](\%)$;
> $UnitNormal = N(t_0)$; $evalf[4](\%)$;
> $UnitBinormal = B(t_0)$; $evalf[4](\%)$

$$UnitTangent = -\frac{1}{3}\sqrt{3}\,(\sin(\ln(2)) - \cos(\ln(2)))e_x + \frac{1}{3}\sqrt{3}\,(\cos(\ln(2)) + \sin(\ln(2)))e_y + \frac{1}{3}\sqrt{3}\,e_z$$

$$UnitTangent = (0.07529)e_x + (0.8129)e_y + (0.5773)e_z$$

Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

$$UnitNormal = -\frac{1}{2}\sqrt{2}\left(\cos(\ln(2)) + \sin(\ln(2))\right)e_x - \frac{1}{2}\sqrt{2}\left(\sin(\ln(2)) - \cos(\ln(2))\right)e_y$$

$$UnitNormal = (-0.9955)e_x + (0.09220)e_y + (0.)e_z$$

$$UnitBinormal = \frac{1}{6}\sqrt{3}\sqrt{2}\left(\sin(\ln(2)) - \cos(\ln(2))\right)e_x - \frac{1}{6}\sqrt{3}\sqrt{2}\left(\cos(\ln(2)) + \sin(\ln(2))\right)e_y + \frac{1}{3}\sqrt{3}\sqrt{2}\,e_z$$

$$UnitBinormal = (-0.05323)e_x + (-0.5748)e_y + (0.8163)e_z$$

Note: The *TNBFrame* command in the *VectorCalculus* package can also be used to calculate the unit tangent, unit normal, and unit binormal vectors to a curve; however, the output from this command is in a slightly different format.

> $TNBFrame(r(t))$

$$\left[\begin{array}{c} -\frac{1}{3}\frac{\sqrt{3}\,e^t(\sin(t)-\cos(t))}{\sqrt{e^{2t}}} \\ \frac{1}{3}\frac{\sqrt{3}\,e^t(\cos(t)+\sin(t))}{\sqrt{e^{2t}}} \\ \frac{1}{3}\frac{\sqrt{3}\,e^t}{\sqrt{e^{2t}}} \end{array}\right], \left[\begin{array}{c} -\frac{1}{2}\frac{\sqrt{2}\,e^t(\cos(t)+\sin(t))}{\sqrt{e^{2t}}} \\ -\frac{1}{2}\frac{\sqrt{2}\,e^t(\sin(t)-\cos(t))}{\sqrt{e^{2t}}} \\ 0 \end{array}\right], \left[\begin{array}{c} \frac{1}{6}\sqrt{3}\sqrt{2}(\sin(t)-\cos(t)) \\ -\frac{1}{6}\sqrt{3}\sqrt{2}(\cos(t)+\sin(t)) \\ \frac{1}{3}\sqrt{3}\sqrt{2} \end{array}\right]$$

> $T = simplify(\ TNBFrame(r(t), output = ['T']), symbolic)$

$$T = \left[\begin{array}{c} -\frac{1}{3}\sqrt{3}\left(\sin(t) - \cos(t)\right) \\ \frac{1}{3}\sqrt{3}\left(\cos(t) + \sin(t)\right) \\ \frac{1}{3}\sqrt{3} \end{array}\right]$$

> $N = simplify(\ TNBFrame(r(t), output = ['N']), symbolic)$

$$N = \left[\begin{array}{c} -\frac{1}{2}\sqrt{2}\left(\cos(t) + \sin(t)\right) \\ \frac{1}{2}\sqrt{2}\left(-\sin(t) + \cos(t)\right) \\ 0 \end{array}\right]$$

> $B = simplify(\ TNBFrame(r(t), output = ['B']), symbolic)$

$$B = \left[\begin{array}{c} -\frac{1}{6}\sqrt{3}\sqrt{2}\left(-\sin(t) + \cos(t)\right) \\ -\frac{1}{6}\sqrt{3}\sqrt{2}\left(\cos(t) + \sin(t)\right) \\ \frac{1}{3}\sqrt{3}\sqrt{2} \end{array}\right]$$

Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

The curvature and torsion functions can be defined either using the respective formulas, or the *Curvature* and *Torsion* commands in the *VectorCalculus* package.

> $Curvature(r(t))$: $simplify(\%, symbolic)$

$$\frac{1}{3}\sqrt{2}\,e^{-t}$$

> $\kappa := unapply\left(simplify\left(\frac{\|T'(t)\|_2}{\|r'(t)\|_2}, symbolic\right), t\right)$

$$\kappa := t \rightarrow \frac{1}{3}\sqrt{2}\,e^{-t}$$

> $Torsion(r(t))$: $simplify(\%, symbolic)$

$$\frac{1}{3}e^{-t}$$

> $\tau := unapply\left(simplify\left(-\frac{(B'(t).N(t))}{\|v(t)\|}, symbolic\right), t\right);$

$$\tau := t \rightarrow \frac{1}{3}e^{-t}$$

The curvature and torsion at t_0.

> $curvature = \kappa(t_0)$; $evalf[4](\%)$;
> $torsion = \tau(t_0)$; $evalf[4](\%)$

$$curvature = \frac{1}{6}\sqrt{2}$$

$$curvature = 0.2357$$

$$torsion = \frac{1}{6}$$

$$torsion = 0.1667$$

The tangential and normal components of acceleration are easily calculated at this point, as well.

> $aT(t) := simplify(a(t).T(t))$: $'aT(t)' = aT(t)$;
> $aN(t) := simplify(a(t).N(t))$: $'aN(t)' = aN(t)$

$$aT(t) = e^t\sqrt{3}$$

$$aN(t) = e^t\sqrt{2}$$

The Osculating Circle

Consider the 2-dimensional trajectory curve

> $r(t) := \langle e^{-t}\cos(t), e^{-t}\sin(t)\rangle$: $'r(t)' = r(t)$

$$r(t) = (e^{-t}\cos(t))e_x + (e^{-t}\sin(t))e_y$$

The unit normal and curvature functions are needed to parameterize the osculating circle.

Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.

> $T := unapply(\,Normalize(r'(t)),t)$:
$simplify(\,Normalize(T'(t)),symbolic)$:
$N := unapply(\%, t)$:'$N(t)$' = $N(t)$

$$N(t) = \frac{1}{2}\sqrt{2}\,(\sin(t) - \cos(t))e_x - \frac{1}{2}\sqrt{2}\,(\cos(t) + \sin(t))e_y$$

> $\kappa := unapply(\,simplify(Curvature(r(t)), symbolic), t)$

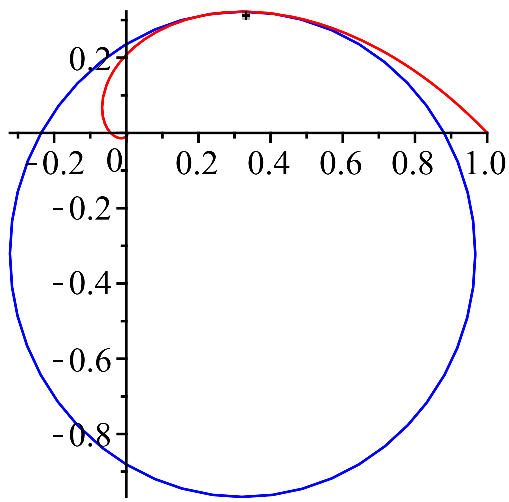
$$\kappa := t \rightarrow \frac{1}{2}\,e^t\sqrt{2}$$

The following is a parameterization of the osculating circle to $r(t)$ at $r\left(\frac{\pi}{4}\right)$. Notice that since the trajectory curve and the osculating circle are 2-dimensional vectors, they can be entered as sets of parametric equations and plotted using the *plot* command.

> $t_0 := \frac{\pi}{4}$ $:oc := unapply\left(r(t_0) + \frac{N(t_0)}{\kappa(t_0)} + \frac{\langle\cos(t), \sin(t)\rangle}{\kappa(t_0)}, t\right)$ $: Point := convert(r(t_0), list)$;

$$Point := \left[\frac{1}{2}\,e^{-\frac{1}{4}\pi}\sqrt{2}, \frac{1}{2}\,e^{-\frac{1}{4}\pi}\sqrt{2}\right]$$

> $plot(\,[\,[oc(t)[1], oc(t)[2], t = 0..2\pi], [r(t)[1], r(t)[2], t = 0..6\pi], [Point]\,]$,
$style = [line\$2, point], color = [blue, red, black])$



Copyright © 2010 Pearson Education, Inc. Publishing as Addison-Wesley.