

Maple Tips

Chapter 10

Lesson 10-1

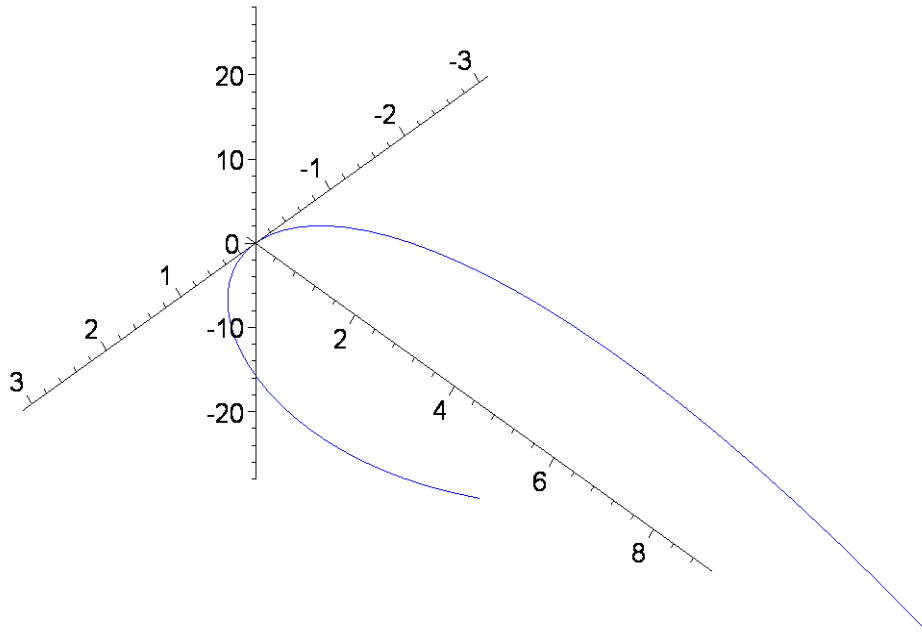
Graphing Curves in Space

The **spacecurve** command in the **plots** package can be used to graph curves in space, exactly like lines were created.

```
> restart: with(plots):
```

```
Warning, the name changecoords has been redefined
```

```
> spacecurve([t, t^2, t^3], t=-3..3, color=blue, axes=normal);
```



Lesson 10-2

Deriving Vector Functions

Similar to single variable functions, the **diff** command in the **VectorCalculus** package can be used to derive vector functions. Remember that the **eval** command can be used to find numerical derivatives.

```
> restart: with(VectorCalculus):
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, -, ., D,
```

Vector, diff, int, limit, series

```
> r:=t-><t,t^2,t^3>: r(t);
```

$$t \mathbf{e}_x + t^2 \mathbf{e}_y + t^3 \mathbf{e}_z$$

```
> diff(r(t),t);
```

$$\mathbf{e}_x + 2t \mathbf{e}_y + 3t^2 \mathbf{e}_z$$

```
> eval(diff(r(t),t), t=-2);
```

$$\mathbf{e}_x - 4 \mathbf{e}_y + 12 \mathbf{e}_z$$

Integrating Vector Functions

Likewise, the **int** command in the **VectorCalculus** package can be used to integrate vector functions.

```
> restart: with(VectorCalculus):
```

Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: *, +, -, ., D, Vector, diff, int, limit, series

```
> r:=t-><sin(t),cos(t),2*t>: r(t);
```

$$\sin(t) \mathbf{e}_x + \cos(t) \mathbf{e}_y + 2t \mathbf{e}_z$$

```
> int(r(t),t);
```

$$-\cos(t) \mathbf{e}_x + \sin(t) \mathbf{e}_y + t^2 \mathbf{e}_z$$

```
> int(r(t),t=0..Pi);
```

$$2 \mathbf{e}_x + \pi^2 \mathbf{e}_z$$

Lesson 10-3

Calculating the Arc Length of a Curve

The **ArcLength** command in the **VectorCalculus** package can be used to find the length of a curve over a specific interval.

```
> restart: with(VectorCalculus):
```

Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: *, +, -, ., D, Vector, diff, int, limit, series

```
> r1:=t-><cos(t), sin(t), t>: r1(t);
```

$$\cos(t) \mathbf{e}_x + \sin(t) \mathbf{e}_y + t \mathbf{e}_z$$

```
> ArcLength(r1(t), t=0..2*Pi);
```

$$2\sqrt{2}\pi$$

Calculating the Curvature of a Curve

The **Curvature** command in the **VectorCalculus** package can be used to find the curvature function of a curve for any parameter value. The **eval** command can be used with **Curvature** to evaluate the curvature at a particular point.

```
> restart: with(VectorCalculus):
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, -, ., D, Vector, diff, int, limit, series
```

```
> Curvature(<cos(t), sin(t), t>, t);
```

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

```
> r2:=t-><t, t^2, t^3>: r2(t);
```

$$t\mathbf{e}_x + t^2\mathbf{e}_y + t^3\mathbf{e}_z$$

```
> eval(Curvature(r2(t),t), t=0);
```

$$\frac{\sqrt{16}}{2}$$

Finding the Unit Tangent, Unit Normal, and Binormal Vectors

The **TNBFrame** command in the **VectorCalculus** package can be used to find any combination of the unit tangent, unit normal, and binormal vectors to a curve as a function of the parameter. The **eval** command can be used with **TNBFrame** to find the specific vectors at a particular point.

```
> restart: with(VectorCalculus):
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, -, ., D, Vector, diff, int, limit, series
```

```
> r1:=t-><cos(t), sin(t), t>: r1(t);
```

$$\cos(t)\mathbf{e}_x + \sin(t)\mathbf{e}_y + t\mathbf{e}_z$$

```
> TNBFrame(r1(t),t);
```

$$-\frac{1}{2}\sqrt{2}\sin(t)\mathbf{e}_x + \frac{1}{2}\sqrt{2}\cos(t)\mathbf{e}_y + \frac{\sqrt{2}}{2}\mathbf{e}_z, -\cos(t)\mathbf{e}_x - \sin(t)\mathbf{e}_y, \\ \frac{1}{2}\sqrt{2}\sin(t)\mathbf{e}_x - \frac{1}{2}\sqrt{2}\cos(t)\mathbf{e}_y + \left(\frac{1}{2}\sqrt{2}\sin^2(t) + \frac{1}{2}\sqrt{2}\cos^2(t)\right)\mathbf{e}_z$$

```
> TNBFrame(r1(t),t, output=[T]); TNBFrame(r1(t),t, output=[N]);
TNBFrame(r1(t),t, output=[B]);
```

$$-\frac{1}{2}\sqrt{2}\sin(t)\mathbf{e}_x + \frac{1}{2}\sqrt{2}\cos(t)\mathbf{e}_y + \frac{\sqrt{2}}{2}\mathbf{e}_z$$

$$-\cos(t)\mathbf{e}_x - \sin(t)\mathbf{e}_y$$

$$\frac{1}{2}\sqrt{2}\sin(t)\mathbf{e}_x - \frac{1}{2}\sqrt{2}\cos(t)\mathbf{e}_y + \left(\frac{1}{2}\sqrt{2}\sin(t)^2 + \frac{1}{2}\sqrt{2}\cos(t)^2\right)\mathbf{e}_z$$

```
> eval(TNBFrame(r1(t),t, output=[T]), t=Pi);
```

$$-\frac{\sqrt{2}}{2}\mathbf{e}_y + \frac{\sqrt{2}}{2}\mathbf{e}_z$$

Lesson 10-4

Finding the Velocity, Speed, and Acceleration of an Object in Space

The `diff` command in the `VectorCalculus` package can be used to calculate the velocity, speed, and acceleration to a position vector.

```
> restart: with(VectorCalculus):
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: *, +, -, ., D, Vector, diff, int, limit, series

```
> r:=t-><t^2, exp(t), t*exp(t)>: r(t);
```

$$t^2\mathbf{e}_x + e^t\mathbf{e}_y + te^t\mathbf{e}_z$$

```
> v:=t->diff(r(t), t): v(t);
```

$$2t\mathbf{e}_x + e^t\mathbf{e}_y + (e^t + te^t)\mathbf{e}_z$$

```
> s:=t->Norm(v(t)): s(t);
```

$$\sqrt{4t^2 + (e^t)^2 + (e^t + te^t)^2}$$

```
> a:=t->diff(v(t), t): a(t);
```

$$2\mathbf{e}_x + e^t\mathbf{e}_y + (2e^t + te^t)\mathbf{e}_z$$

Lesson 10-5

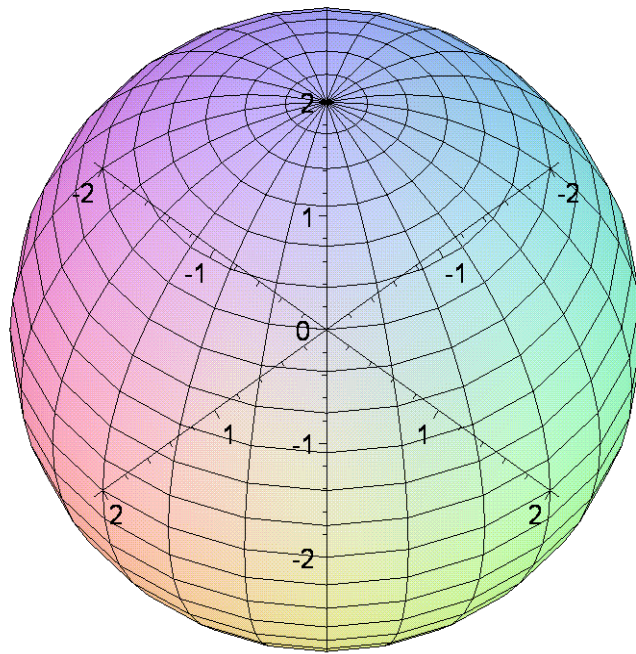
Graphing Parametric Surfaces

The `plot3d` command in the `plots` package can be used to graph surfaces defined parametrically.

```
> restart: with(plots):
```

```
Warning, the name changecoords has been redefined
```

```
> plot3d([2*cos(theta)*sin(phi), 2*sin(theta)*sin(phi), 2*cos(phi)],  
theta=0..2*Pi, phi=0..Pi, axes=normal);
```



```
>
```