Units Updates in Maple 2022

There were several updates to working with units in Maple 2022.

▼ Visualization and Units

▼ The useunits option

- The useunits option to the plot and plot3d commands has been overhauled. It now rescales the plot data to conform to the units you specify (in most cases; there is one exception detailed on its help page).

```
> height := acceleration/2 * time^2 + initial_velocity * time;
```

$$height := \frac{1}{2}\, acceleration\, time^2 + initial\_velocity\, time$$

```
> acceleration := -9.81*Unit(m/s^2);
```

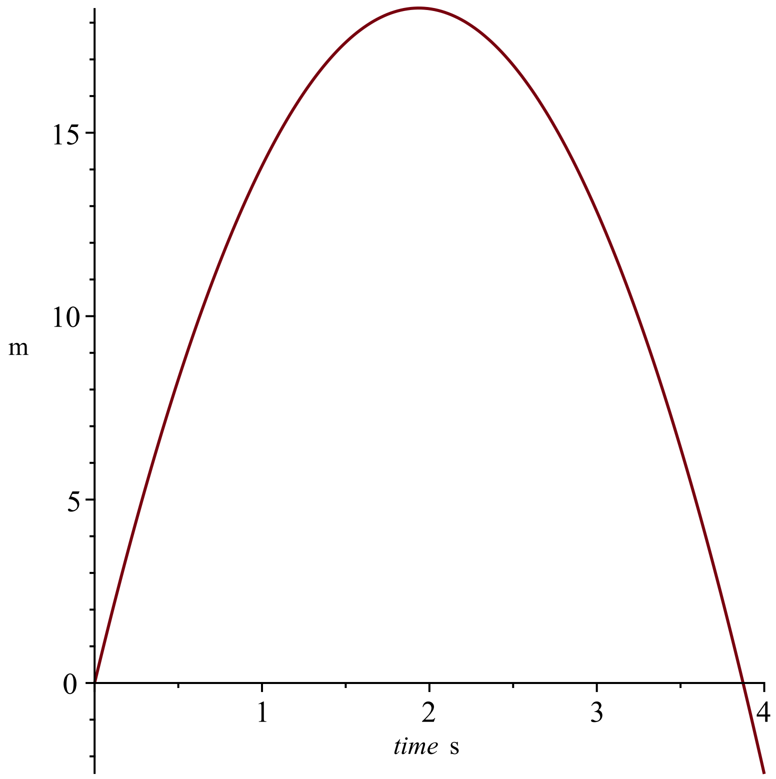
$$acceleration := 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$$

```
> initial_velocity := 19*Unit(m/s);
```

$$initial\_velocity := 19\,\frac{\mathrm{m}}{\mathrm{s}}$$

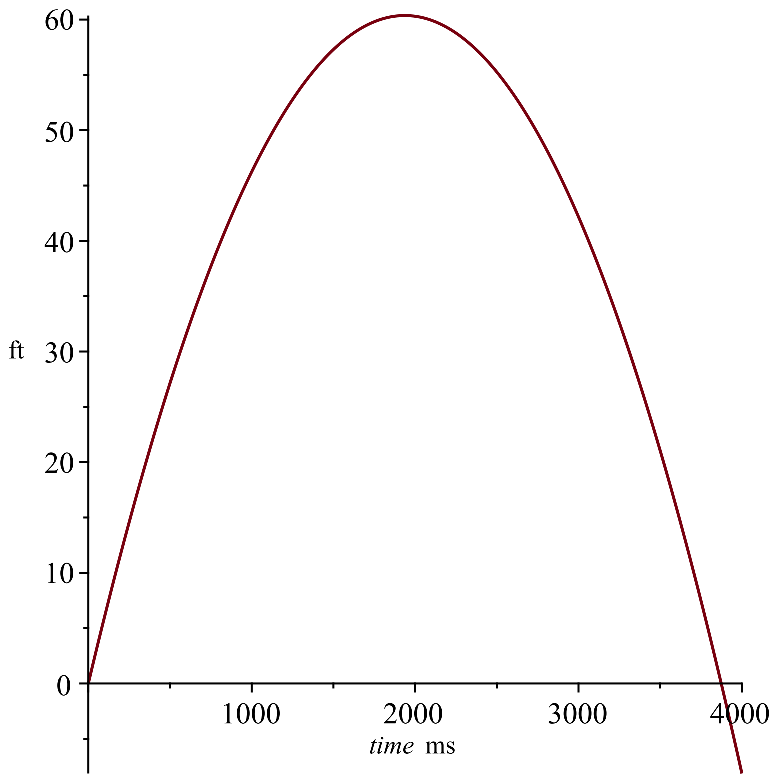
- The default units are seconds on the horizontal axis and meters on the vertical axis.

```
> plot(height, time = 0 .. 4*Unit(s));
```



• We can override this with the useunits option. Note that the plot data are scaled.

```
> plot(height, time = 0 .. 4*Unit(s), useunits=[ms, ft]);
```



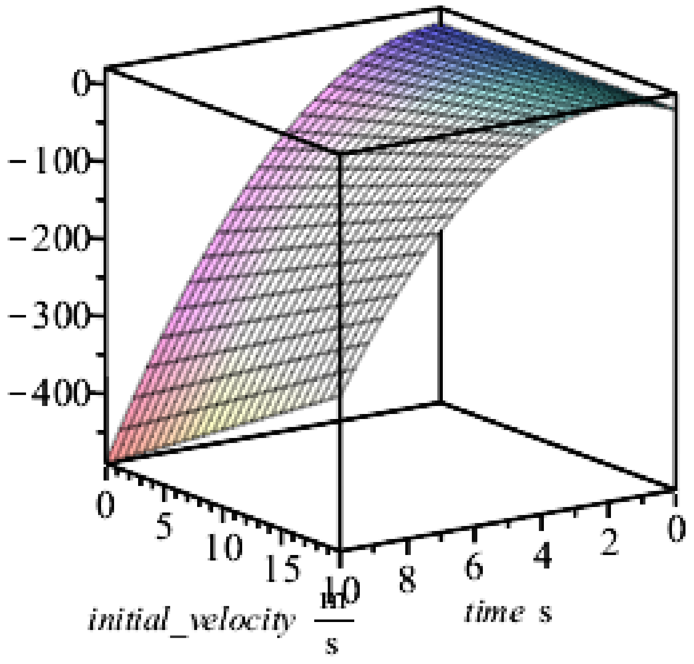
• If you specify an incompatible unit, Maple issues an error.

```
> plot(height, time = 0 .. 4*Unit(s), useunits=[ms, ft/s]);
Error, (in plot) unable to convert `m` to `ft/s`
```

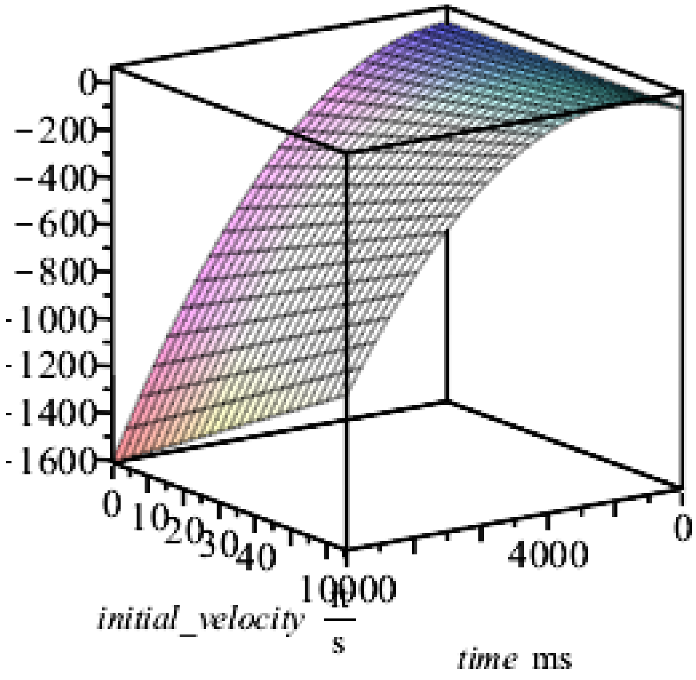
- This also works with the plot3d command.

```
> unassign('initial_velocity');
```

```
> plot3d(height, time = 0 .. 10*Unit(s), initial_velocity = 0 ..
  20*Unit(m/s));
```



```
> plot3d(height, time = 0 .. 10*Unit(s), initial_velocity = 0 ..
  20*Unit(m/s), useunits=[ms, ft/s, ft]);
```



Lists of procedures in the plot and plot3d commands

- As of Maple 2022, Maple accepts units in calls to the plot and plot3d commands when the object to be plotted is a list of procedures.

```
> f_expr := 2.5*sin(t)*Unit(m);
```

$$f\_expr := 2.5 \sin(t) \ \mathrm{m}$$

```
> f_proc := t -> 2.5*sin(t)*Unit(m);
```

$$f\_proc := t \quad 2.5 \cdot \sin(t) \cdot \mathrm{m}$$

```
> g_expr := 1400*cos(t)*Unit(mm);
```

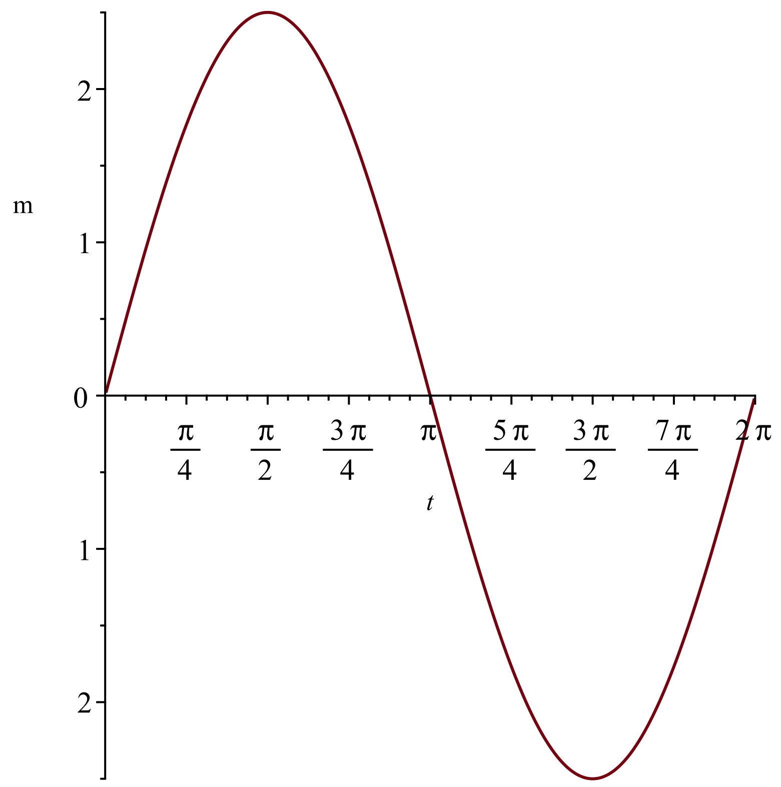
$$g\_expr := 1400 \cos(t) \ \mathrm{mm}$$

```
> g_proc := t -> 1400*cos(t)*Unit(mm);
```

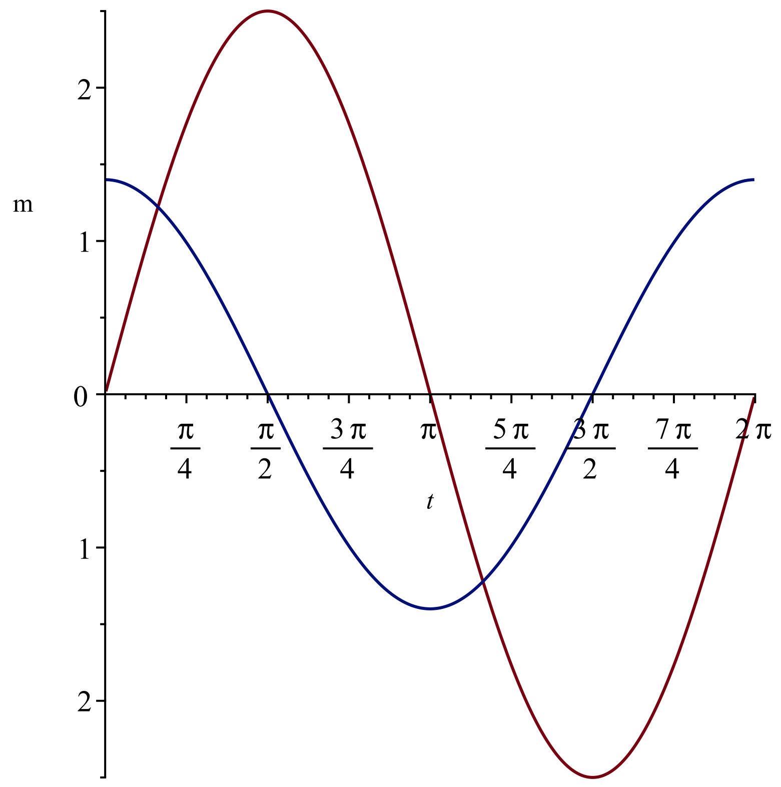
$$g\_proc := t \quad 1400 \cdot \cos(t) \cdot \mathrm{mm}$$

• The following three calls to plot already worked as expected before Maple 2022.

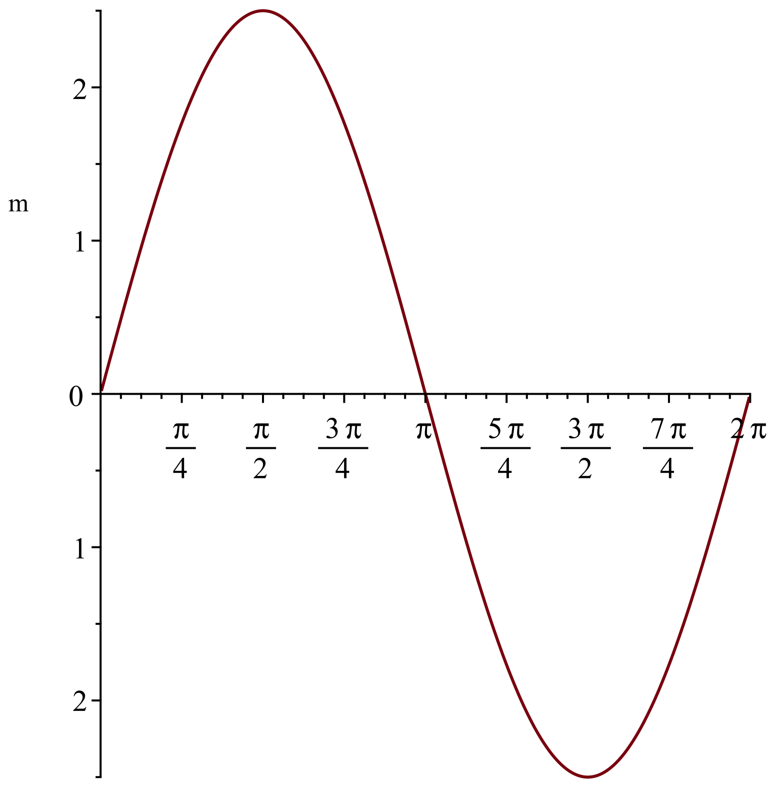
```
> plot(f_expr, t=0..2*Pi);
```



```
> plot([f_expr, g_expr], t=0..2*Pi);
```

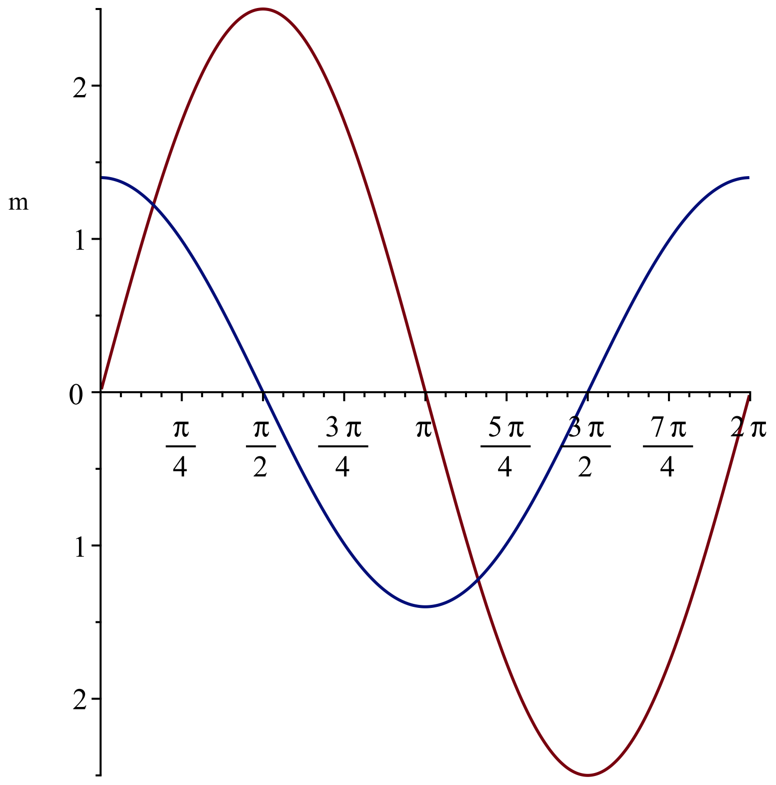


```
> plot(f_proc, 0..2*Pi, useunits);
```



• The following one, however, works only from Maple 2022 onwards.

```
> plot([f_proc, g_proc], 0..2*Pi, useunits);
```



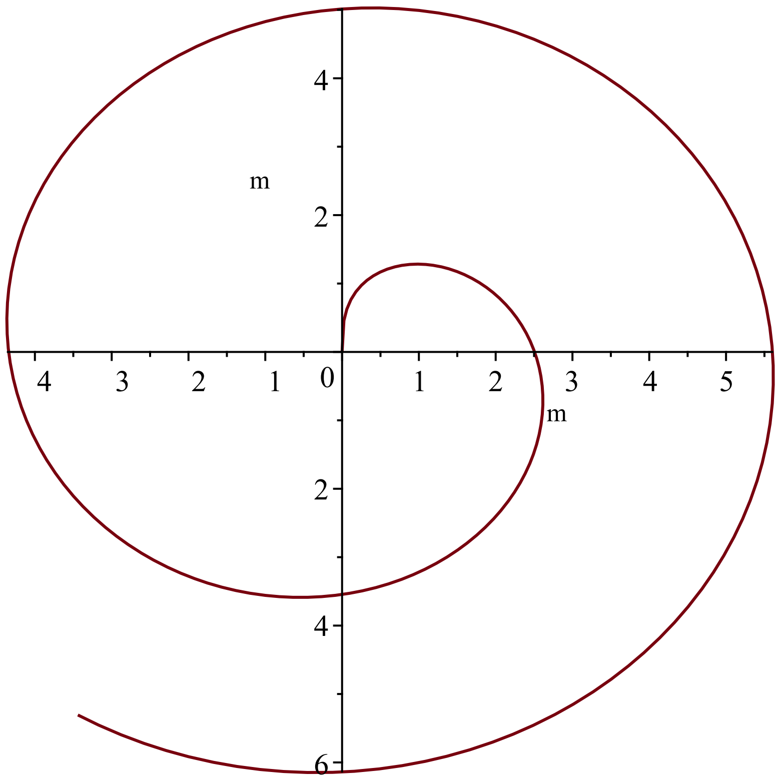
Parametric plots with units

- The plot and plot3d commands have supported parametric plots for many, many releases. They have supported units since Maple 13. However, before Maple 2022, parametric plots have not supported units. Now they do. Here is a 2-D example with a spiral:

```
> radius := 2*sqrt(t/Unit(s)) * Unit(m);
```

$$radius := 2\sqrt{\frac{t}{\mathrm{s}}}\ \mathrm{m}$$

```
> plot([radius * sin(t/Unit(s)), radius * cos(t/Unit(s)), t = 0 ..
  10*Unit(s)]);
```



- This 3-D example also uses the useunits option discussed above; it also applies to parametric plots.

```
> c1 := [cos(x) - 2*cos(0.4*y*Unit(Hz)), sin(x) - 2*sin(0.4*y*Unit
  (Hz)), y];
```

$$c1 := [\cos(x) \quad 2\cos(0.4\,y\ \mathrm{Hz}), \sin(x) \quad 2\sin(0.4\,y\ \mathrm{Hz}), y]$$

```
> c2 := [cos(x) + 2*cos(0.4*y*Unit(Hz)), sin(x) + 2*sin(0.4*y*Unit
  (Hz)), y];
```

$$c2 := [\cos(x) + 2\cos(0.4\,y\ \mathrm{Hz}), \sin(x) + 2\sin(0.4\,y\ \mathrm{Hz}), y]$$

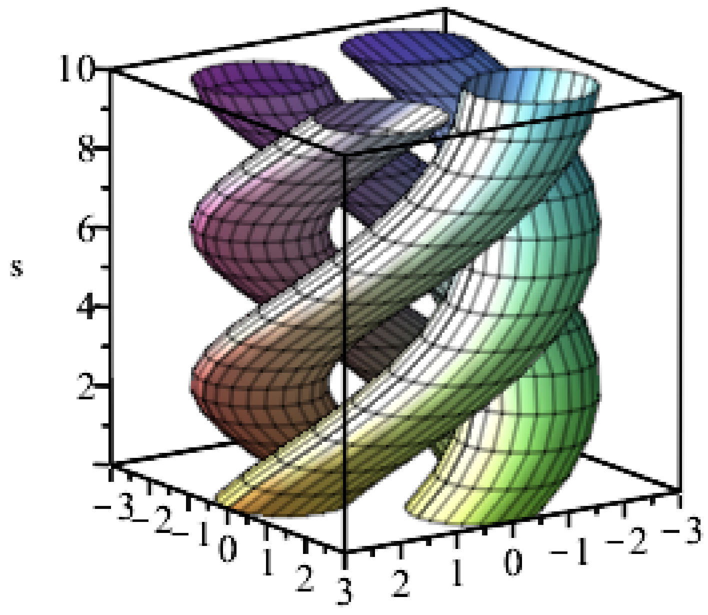
```
> c3 := [cos(x) + 2*sin(0.4*y*Unit(Hz)), sin(x) - 2*cos(0.4*y*Unit
  (Hz)), y];
```

$$c3 := [\cos(x) + 2\sin(0.4\, y\, \mathrm{Hz}), \sin(x) - 2\cos(0.4\, y\, \mathrm{Hz}), y]$$

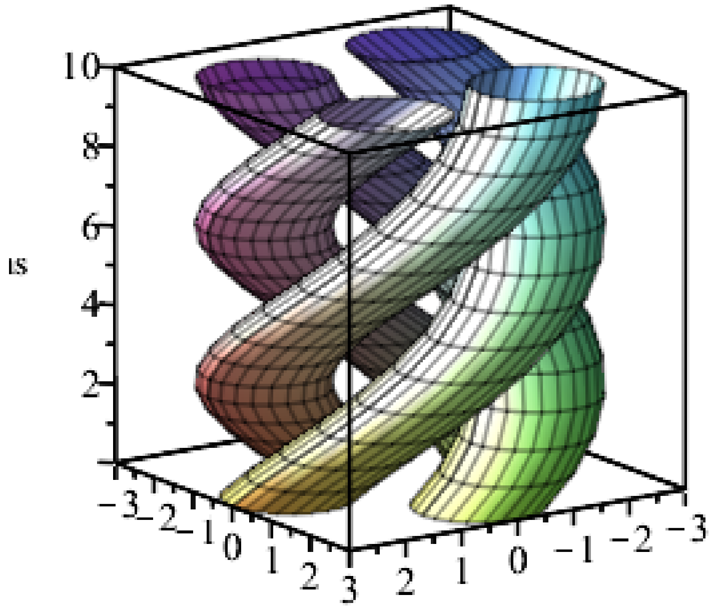
```
> c4 := [cos(x) - 2*sin(0.4*y*Unit(Hz)), sin(x) + 2*cos(0.4*y*Unit
  (Hz)), y];
```

$$c4 := [\cos(x) - 2\sin(0.4\, y\, \mathrm{Hz}), \sin(x) + 2\cos(0.4\, y\, \mathrm{Hz}), y]$$

```
> plot3d({c1, c2, c3, c4}, x = 0 .. 2*Pi, y = 0 .. 10*Unit(s), grid
  = [25, 15]);
```



```
> plot3d({c1, c2, c3, c4}, x = 0 .. 2*Pi, y = 0 .. 10*Unit(s), grid
  = [25, 15], useunits=[default, default, Unit(ms)]);
```



The piecewise Command Now Understands Units

- The piecewise command is used to construct piecewise-defined expressions. For Maple 2022, we made this command understand units in the conditions that govern switching between the branches of the piecewise command.

```
> acceleration := 2*Unit(m/s^2);
```

$$acceleration := 2\,\frac{\mathrm{m}}{\mathrm{s}^2}$$

```
> expr := piecewise(t < 2*Unit(s), 0, t < 3*Unit(s), acceleration/2 *
  (t - 2*Unit(s))^2, acceleration*(t - 5/2*Unit(s))*Unit(s));
```

$$expr := \begin{cases} 0 & t < 2\,\mathrm{s} \\ (t - 2\,\mathrm{s})^2\,\frac{\mathrm{m}}{\mathrm{s}^2} & t < 3\,\mathrm{s} \\ 2\left(t - \frac{5}{2}\,\mathrm{s}\right)\frac{\mathrm{m}}{\mathrm{s}^2}\,\mathrm{s} & \mathit{otherwise} \end{cases}$$

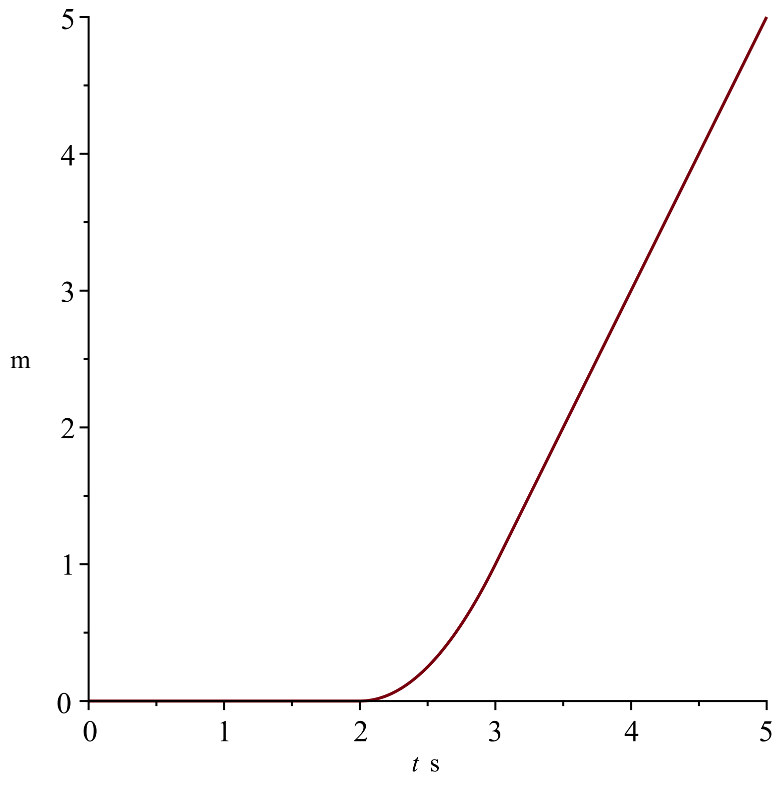
```
> combine(eval(expr, t = 2200*Unit(ms)), units);
```

$$\frac{1}{25}\ \mathrm{m}$$

```
> combine(eval(expr, t = 0.5*Unit(min)), units);
```

$$55.00000000\ \mathrm{m}$$

```
> plot(expr, t = 0 .. 5*Unit(s));
```



Performance of the Units:-Simple package

- The Units:-Simple package was introduced in Maple 2017. It provides an easy and powerful way to work with units.
- Some computations slow down somewhat when using the Units:-Simple package. This is unavoidable, because units-aware operations need to do more work than the default operations. However, for Maple 2022, the units-aware operations in the Units:-Simple package have been sped up considerably for some cases.
- The computation that follows comes from a slope stability application developed by one of our in-house engineers. To set it up, we load the Units:-Simple package and initialize some variables:

```
> with(Units:-Simple):
```

```
> n := 1000:
```

```
> X := Vector(n, i -> i * x__max / n);
```

$$X := \begin{bmatrix} \frac{x_{\max}}{1000} \\ \frac{x_{\max}}{500} \\ \frac{3\,x_{\max}}{1000} \\ \frac{x_{\max}}{250} \\ \frac{x_{\max}}{200} \\ \frac{3\,x_{\max}}{500} \\ \frac{7\,x_{\max}}{1000} \\ \frac{x_{\max}}{125} \\ \frac{9\,x_{\max}}{1000} \\ \frac{x_{\max}}{100} \\ \vdots \end{bmatrix}$$

1000 element Vector[column]

```
> y__slope := piecewise(x < 33.948*Unit(m), .50076*x, 17 * Unit(m));
```

$$y_{slope} := \begin{cases} 0.50076\,x & x < 33.948\ \mathrm{m} \\ 17\ \mathrm{m} & \mathit{otherwise} \end{cases}$$

```
> y__fail := evalf(50*Unit(m) - sqrt(2401*Unit(m^2) - x^2));
```

$$y_{fail} := 50.\ \mathrm{m} - 1.\sqrt{2401.\ \mathrm{m}^2 - 1.\,x^2}$$

```
> rho := 20000*Unit(N/m^3);
```

$$\rho := 20000\ \frac{\mathrm{N}}{\mathrm{m}^3}$$

```
> x__max := 4*sqrt(82)*Unit(m);
```

$$x_{\max} := 4\sqrt{82}\ \mathrm{m}$$

- The time consuming part of the computation now follows.

```
> result := Vector(n, i -> x__max / n * eval(y__slope - y__fail, x = X
  [i]) * rho);
```

$$result := \begin{bmatrix} 80\sqrt{82}\left(0.002003040000\sqrt{82} - 1.00001339\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.004006080000\sqrt{82} - 1.00005355\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.006009120000\sqrt{82} - 1.00012049\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.008012160000\sqrt{82} - 1.00021420\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.01001520000\sqrt{82} - 1.00033470\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.01201824000\sqrt{82} - 1.00048196\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.01402128000\sqrt{82} - 1.00065600\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.01602432000\sqrt{82} - 1.00085682\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.01802736000\sqrt{82} - 1.00108442\right)\frac{\mathrm{N}}{\mathrm{m}} \\ 80\sqrt{82}\left(0.02003040000\sqrt{82} - 1.00133879\right)\frac{\mathrm{N}}{\mathrm{m}} \\ \vdots \end{bmatrix}$$

1000 element Vector[column]

- This computation was sped up by about a factor 3, comparing to Maple 2021. It also uses about half as much memory in Maple 2022.