Differential Equations in Maple.

Using the dsolve and DEplot commands.

The exponential growth model is $y'(t) = k\,y(t)$.

```
> de := diff(y(t),t) = k*y(t);
```

$$de := \frac{d}{dt}\,y(t) = k\,y(t)$$

To solve a differential equation in Maple use the dsolve command

```
> dsolve( de, y(t) );
```

$$y(t) = \_C1\,e^{k\,t}$$

That is the general solution. Maple uses _C1 instead of c for the constant of integration. Here is how you specify an initial value e.g. $y(0) = 5$ to obtain a particular solution.

```
> dsolve( {de,y(0)=5}, y(t) );
```

$$y(t) = 5\,e^{\frac{1}{10}\,t}$$

Note, the solve command does not work

```
> solve( de, y(t) );
Error, (in solve) cannot solve expressions with diff(y(t), t) for y(t)
```

The differential equation for Newton's law of cooling is $T(t) = k\cdot(Am - T(t))$ where
$T(t)$ is the temperature of the body at time t,
Am is the Ambient temperature (assumed to be constant) and
k is the cooling rate constant.

```
> NLC := diff(T(t),t) = k*(Am-T(t));
```

$$NLC := \frac{d}{dt}\,T(t) = k\,(Am - T(t))$$

```
> dsolve( NLC, T(t) );
```

$$T(t) = Am + e^{-k\,t}\,\_C1$$

```
> dsolve( { NLC, T(0)=60 }, T(t) );
```

$$T(t) = Am + e^{-k\,t}\,(60 - Am)$$

To graph the solution we need to fix values for the parameters

```
> Am  := 20;
  k := 0.1;
```

$$Am := 20$$

$$k := 0.1$$

```
> NLC;
```

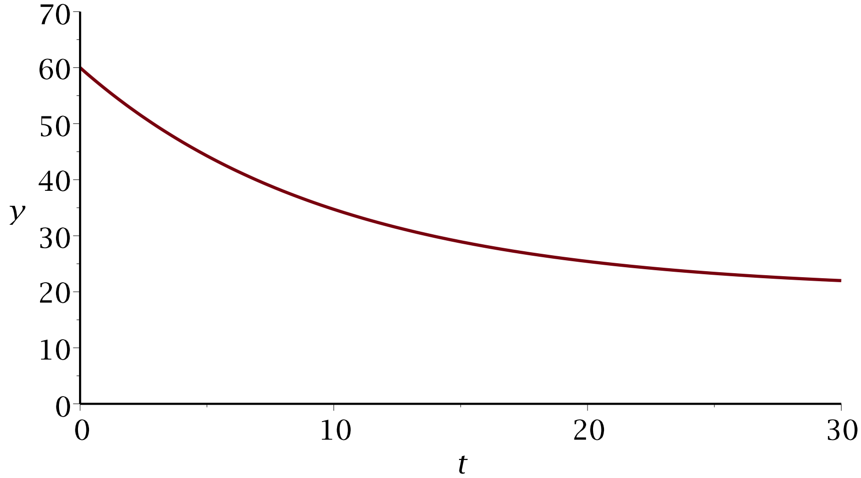
$$\frac{d}{dt}\,T(t) = 2.0 - 0.1\,T(t)$$

```
> sol := dsolve( {NLC,T(0)=60}, T(t) );
```

$$sol := T(t) = 20 + 40\,e^{-\frac{1}{10}\,t}$$

Notice that dsolve returns the solution as an equation. To graph the solution we need to extract the right-hand-side of the equation.

```
> plot( rhs(sol), t=0..30, y=0..70 );
```



Let's generate a plot for several different initial values.

```
> initTemps := [0,10,20,30,40,50,60];
```

$$initTemps := [0, 10, 20, 30, 40, 50, 60]$$

```
> for i to nops(initTemps) do
      initval := initTemps[i];
      sol[i] := dsolve({NLC,T(0)=initval}, T(t)) ;
  od;
```

$$initval := 0$$

$$sol_1 := T(t) = 20 - 20\,e^{-\frac{1}{10}t}$$

$$initval := 10$$

$$sol_2 := T(t) = 20 - 10\,e^{-\frac{1}{10}t}$$

$$initval := 20$$

$$sol_3 := T(t) = 20$$

$$initval := 30$$

$$sol_4 := T(t) = 20 + 10\,e^{-\frac{1}{10}t}$$

$$initval := 40$$

$$sol_5 := T(t) = 20 + 20\,e^{-\frac{1}{10}t}$$

$$initval := 50$$

$$sol_6 := T(t) = 20 + 30\,e^{-\frac{1}{10}t}$$

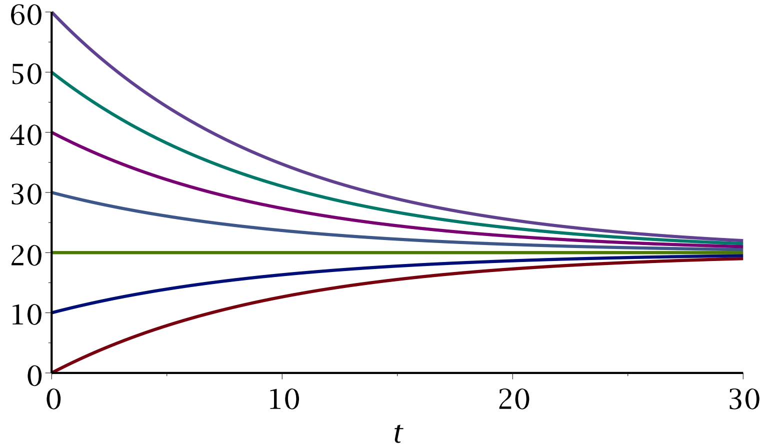
$$initval := 60$$

$$sol_7 := T(t) = 20 + 40\,e^{-\frac{1}{10}t}$$

```
> sols := [seq( rhs(sol[i]), i=1..nops(initTemps) )];
```

$$sols := \left[20 - 20\,e^{-\frac{1}{10}t},\ 20 - 10\,e^{-\frac{1}{10}t},\ 20,\ 20 + 10\,e^{-\frac{1}{10}t},\ 20 + 20\,e^{-\frac{1}{10}t},\ 20 + 30\,e^{-\frac{1}{10}t},\ 20 + 40\,e^{-\frac{1}{10}t}\right]$$

```
> plot( sols, t=0..30 );
```



This plot together with the field plot can be generated using the DEplot command in the DEtools package.

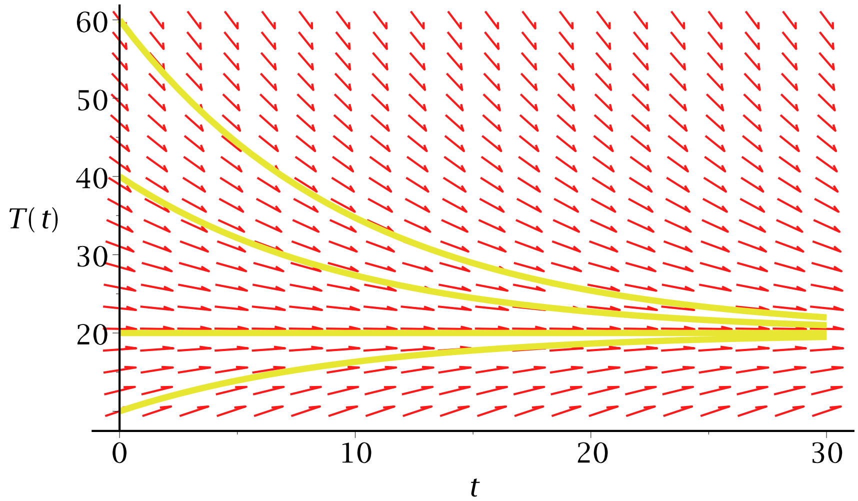
```
> with(DEtools);
```

[*AreSimilar*, *Closure*, *DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *DFactorLCLM*, *DFactorsols*, *Dchangevar*, *Desingularize*, *FunctionDecomposition*, *GCRD*, *Gosper*, *Heunsols*, *Homomorphisms*, *IVPsol*, *IsHyperexponential*, *LCLM*, *MeijerGsols*, *MultiplicativeDecomposition*, *ODEInvariants*, *PDEchangecoords*, *PolynomialNormalForm*, *RationalCanonicalForm*, *ReduceHyperexp*, *RiemannPsols*, *Xchange*, *Xcommutator*, *Xgauge*, *Zeilberger*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *caseplot*, *casesplit*, *checkrank*, *chinisol*, *clairautsol*, *constcoeffsols*, *convertAlg*, *convertsys*, *dalembertsol*, *dcoeffs*, *de2diffop*, *dfieldplot*, *diff_table*, *diffop2de*, *dperiodic_sols*, *dpolyform*, *dsubs*, *eigenring*, *endomorphism_charpoly*, *equinv*, *eta_k*, *eulersols*, *exactsol*, *expsols*, *exterior_power*, *firint*, *firtest*, *formal_sol*, *gen_exp*, *generate_ic*, *genhomosol*, *gensys*, *hamilton_eqs*, *hypergeomsols*, *hyperode*, *indicialeq*, *infgen*, *initialdata*, *integrate_sols*, *intfactor*, *invariants*, *kovacicsols*, *leftdivision*, *liesol*, *line_int*, *linearsol*, *matrixDE*, *matrix_riccati*, *maxdimsystems*, *moser_reduce*, *muchange*, *mult*, *mutest*, *newton_polygon*, *normalG2*, *ode_int_y*, *ode_y1*, *odeadvisor*, *odepde*, *parametricsol*, *particularsol*, *phaseportrait*, *poincare*, *polysols*, *power_equivalent*, *rational_equivalent*, *ratsols*, *redode*, *reduceOrder*, *reduce_order*, *regular_parts*, *regularsp*, *remove_RootOf*, *riccati_system*, *riccatisol*, *rifread*, *rifsimp*, *rightdivision*, *rtaylor*, *separablesol*, *singularities*, *solve_group*, *super_reduce*, *symgen*, *symmetric_power*, *symmetric_product*, *symtest*, *transinv*, *translate*, *untranslate*, *varparam*, *zoom*]

```
> initTemps :=[T(0)=10, T(0)=20, T(0)=40, T(0)=60];
```

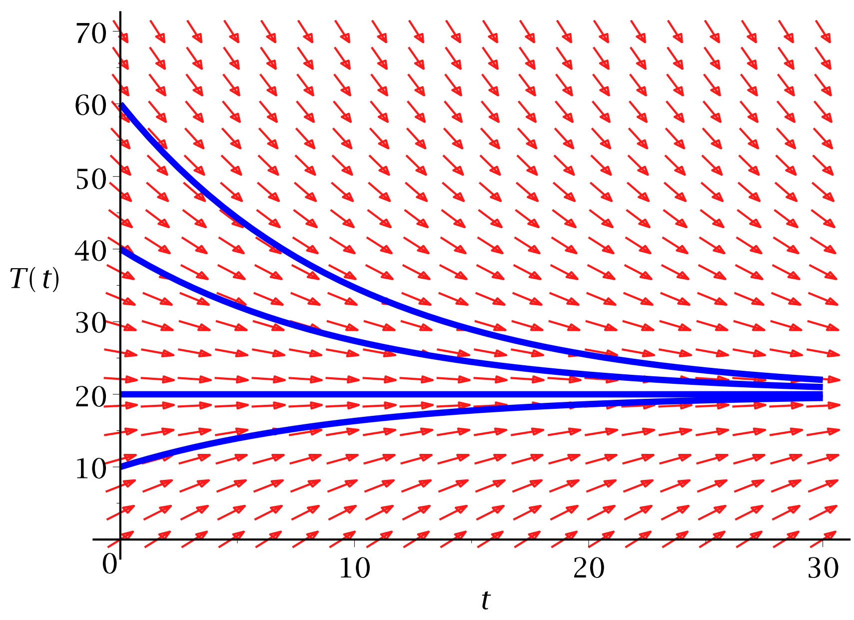
$$initTemps := [T(0) = 10, T(0) = 20, T(0) = 40, T(0) = 60]$$

```
> DEplot( NLC, T(t), t=0..30, initTemps  );
```



There are many options. Three are illustrated here

```
> DEplot( NLC, T(t), t=0..30, T=0..70, initTemps, linecolor=blue,
  arrows=medium );
```



You can also get an animation by specifying animatecurves = true

```
> DEplot( NLC, T(t), t=0..30, T=0..70, initTemps, linecolor=blue,
  arrows=medium, animatecurves=true );
```

Here is the Logistic growth equation

Ym is the carrying capacity of the population

a is the constant such that $k = a \cdot Ym$ is the natural growth rate.

```
> LG := diff(y(t),t) = a*y(t)*(Ym-y(t));
```

$$LG := \frac{d}{dt}\, y(t) = a\, y(t)\, (Ym - y(t))$$

```
> a := 0.01;
  Ym := 20;
```

$$a := 0.01$$

$$Ym := 20$$

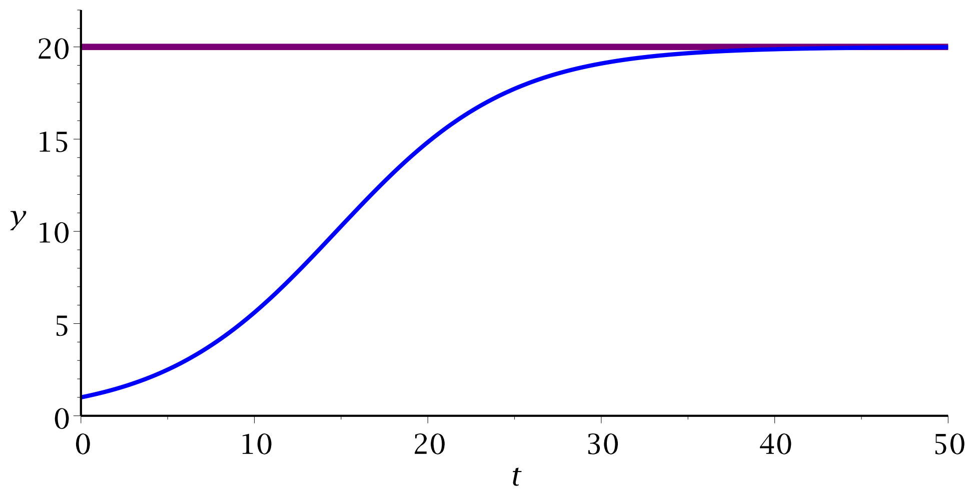
```
> dsolve( LG, y(t) );
```

$$y(t) = \frac{20}{1 + 20\, e^{-\frac{1}{5}\, t}\, \_C1}$$

```
> sol := dsolve( {LG,y(0)=1}, y(t) );
```

$$sol := y(t) = \frac{20}{1 + 19\, e^{-\frac{1}{5}\, t}}$$

```
> plot( [Ym,rhs(sol)], t=0..50, y=0..22, color=[purple,blue], thickness=
  [3,2] );
```



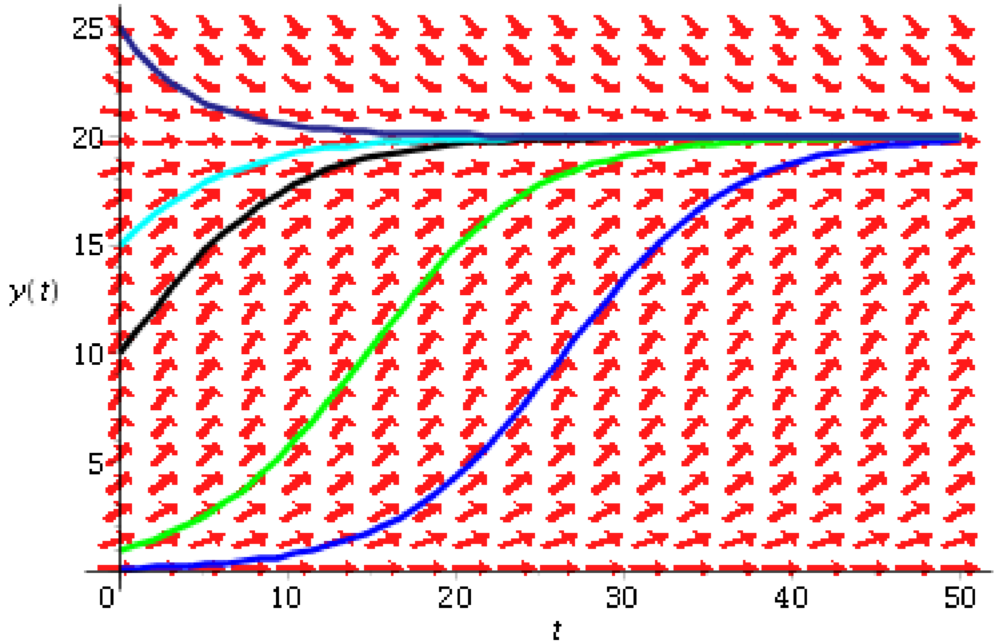

```
> initPops := [ y(0)=0.1, y(0)=1, y(0)=10, y(0)=15, y(0)=25 ];
```

$$initPops := [y(0) = 0.1, y(0) = 1, y(0) = 10, y(0) = 15, y(0) = 25]$$

```
> cols := [blue,green,black,cyan,navy];
```

$$cols := [blue, green, black, cyan, navy]$$

```
> DEplot( LG, y(t), t=0..50, initPops, linecolor=cols, arrows=large );
```



25
20
15
10
5
0
y(t)
10
20
30
40
50
t