MATH529 Lesson02

First-order differential equations

Exact equations

The equation $z(x, y) = c$ is an implicit definition of a curve C and the differential $dz = M\,dx + N\,dy$ evaluates to zero on the curve C ($M = \partial_x z$, $N = \partial_y z$). This implies that $z(x, y) = c$ is a solution to the ODE
$\frac{dy}{dx} = -\frac{M}{N}$, or $y'(x) = -N/M = f(x, y)$. A solution to the ODE exists if f is continuous and differentiable, hence $z(x, y)$ has continuous second derivatives, in which case $z_{xy} = z_{yx}$.

Criterion for an exact differential

In general, the differential form $M(x, y)\,dx + N(x, y)\,dy$ is said to be exact if there exists some $z(x, y)$ such that $z_x = M(x, y)$, $z_y = N(x, y)$. When $z(x, y)$ is twice differentiable, the condition for an exact differential form is
$M_y = N_x$ or $\partial_y M = \partial_x N$.

Example 1: Exact differential corresponds to ODE solution

Curves defined by $z(x, y) = c$.

In[ ]:=
```
z[x_, y_] = x^3 - 5 x y - y^2
```

Out[ ]=
```
x^3 - 5 x y - y^2
```

In[ ]:=
```
cplt = ContourPlot[z[x, y], {x, 0, 2}, {y, -1, 5},
  Contours → Table[c, {c, 0, 2, 0.2}], ContourShading → None]
```

$y'(x) = -N/M = f(x, y)$

Construct the ODE $y'(x) = -N/M = f(x, y)$, choose an initial condition and superimpose on the above contours

In[∘]:=
```
zM[x_, y_] = ∂x z[x, y]
```
Out[∘]=

$$3x^2 - 5y$$

In[∘]:=
```
zN[x_, y_] = ∂y z[x, y]
```
Out[∘]=

$$-5x - 2y$$

$y'(x) = -N/M = f(x, y)$

In[∘]:=
```
f[x_, y_] = -zM[x, y] / zN[x, y]
```
Out[∘]=

$$\frac{-3x^2 + 5y}{-5x - 2y}$$

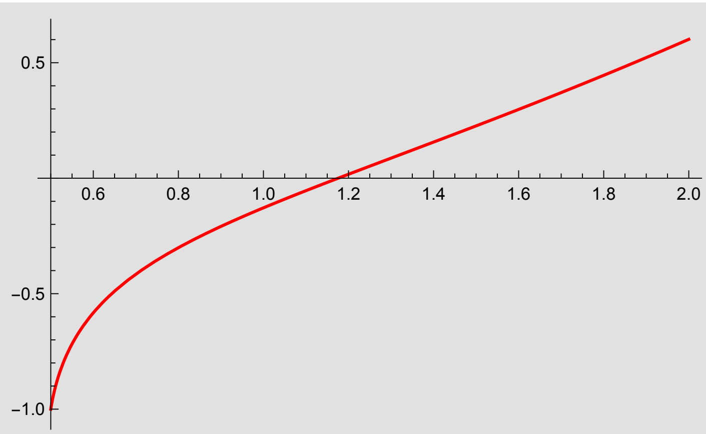
In[∘]:=
```
sol = DSolve[{y'[x] == f[x, y[x]], y[0.5] == -1}, y[x], x][[1, 1]]
```

DSolve : For some branches of the general solution, the given boundary conditions lead to an empty solution.

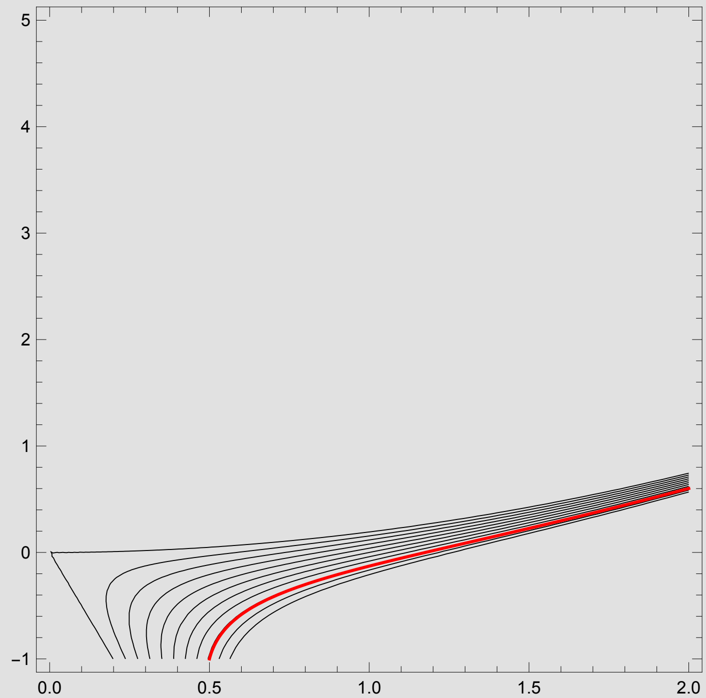
Out[∘]=

$$y[x] \to \frac{1}{2}\left(-5x + \sqrt{2}\sqrt{-3.25 + \frac{25x^2}{2} + 2x^3}\right)$$

In[∘]:=
```
splt = Plot[y[x] /. sol, {x, 0.5, 2}, PlotStyle → Red]
```
Out[∘]=



In[ ]:=
```
Show[{cplt, splt}]
```
Out[ ]=



Example 2: Verification of exact differential form

Consider the differential form $\left(e^{2y} - y \cos x\, y\right) dx + \left(2\, x\, e^{2y} - x \cos x\, y + 2\, y\right) dy$

In[ ]:=
```
zM[x_, y_] = e^(2 y) - y Cos[x y]; zN[x_, y_] = 2 x e^(2 y) - x Cos[x y] + 2 y;
```

Verify that it is exact

In[ ]:=
```
∂y zM[x, y] == ∂x zN[x, y]
```
Out[ ]=
```
True
```

Solving exact differential equations

Given $M(x, y)\, dx + N(x, y)\, dy = 0$, the function $z(x, y)$ that satisfies $z_x = M(x, y), z_y = N(x, y)$, is recovered by:

1. Integration with respect to x holding y constant

$z(x, y) = \int M(x, y)\, dx + g(y)$

2. To find $g(y)$, differentiate result with respect to y, and set equal to $N(x, y)$
$\partial_y z = N(x, y) = \frac{\partial}{\partial y} \int M(x, y)\, dx + g'(y)$

Integrating the result gives $g(y)$
$g(y) = \int N(x, y)\, dy - \int M(x, y)\, dx$

Example 3: Solving an exact differential equation

Consider the differential equation $\left(e^{2y} - y \cos x y\right) dx + \left(2 x e^{2y} - x \cos x y + 2 y\right) dy = 0$

In[∘]:=
```
zM[x_, y_] = e^(2 y) - y Cos[x y]; zN[x_, y_] = 2 x e^(2 y) - x Cos[x y] + 2 y;
```

It is exact

In[∘]:=
```
∂y zM[x, y] == ∂x zN[x, y]
```

Out[∘]=
```
True
```

1. Integration with respect to x holding y constant
$z(x, y) = \int M(x, y)\, dx + g(y)$

In[∘]:=
```
z[x_, y_] = Integrate[zM[x, y], x] + g[y]
```

Out[∘]=
```
e^(2 y) x + g[y] - Sin[x y]
```

2. Find $g(y) = \int N(x, y)\, dy - \int M(x, y)\, dx$

In[∘]:=
```
g[y_] = Integrate[zN[x, y], y] - Integrate[zM[x, y], x]
```

Out[∘]=
```
y^2
```

Since $g(y)$ is now defined, it will be substituted in the definition of $z(x, y)$

In[∘]:=
```
z[x, y]
```

Out[∘]=
```
e^(2 y) x + y^2 - Sin[x y]
```

Verify

In[ ]:=
```
{D[z[x, y], x] == zM[x, y], D[z[x, y], y] == zN[x, y]}
```
Out[ ]=
```
{True, True}
```

Solution by substitutions

In $y' = f(x, y)$ the dependent variable $y(x)$ may be replaced by $y = g(x, u)$.

Homogeneous functions

A function $f(x, y)$ is homogeneous if $f(s\,x,\ s\,y) = s^n f(x, y)$ with n the degree of the homogeneous function. A differential equation $M(x, y)\,dx + N(x, y)\,dy = 0$ where $M(x, y),\ N(x, y)$ are homogeneous functions of the same degree can be reduced to a separable equation by the substitution $y = u\,x$.

$M(x, y)\,dx + N(x, y)\,dy = M(x, u\,x)\,dx + N(x, u\,x)\,d(u\,x) = x^n[M(1, u)\,dx + N(1, u)\,d(u\,x)] = 0 \Rightarrow$
$[M(1, u) + u\,N(1, u)]\,dx + x\,N(1, u)\,du = 0 \Rightarrow$
$\frac{dx}{x} + \frac{N(1,u)\,du}{M(1,u)+u\,N(1,u)} = \frac{dx}{x} + F(u)\,du = 0 \Rightarrow \ln x + \int F(u)\,du = \ln c \Rightarrow x = c\exp\left[-\int F(u)\,du\right] = G(u)$

Example

Solve $(x^2 + y^2)\,dx + (x^2 - x\,y)\,dy = 0$

In[ ]:=
```
zM[x_, y_] = x^2 + y^2; zN[x_, y_] = x^2 - x y;
```
In[ ]:=
```
F[u_] = Simplify[zN[1, u] / (zM[1, u] + u zN[1, u])]
```
Out[ ]=
$$\frac{1-u}{1+u}$$

In[ ]:=
```
Integrate[F[u], u]
```
Out[ ]=
```
-u + 2 Log[1 + u]
```

In[ ]:=
```
G[u_] = Exp[Integrate[-F[u], u]]
```
Out[ ]=

$$\frac{e^{u}}{(1+u)^{2}}$$

In[ ]:=
```
sol = x == c G[y / x]
```
Out[ ]=

$$x == \frac{c\, e^{\frac{y}{x}}}{\left(1+\frac{y}{x}\right)^{2}}$$

Bernoulli’s equation

In $y' + p(x)\, y = f(x)\, y^{n}$, replace $u = y^{1-n}$ to obtain a linear ODE

Example

Solve $x\, y' + y = x^{2}\, y^{2}$

In[ ]:=
```
DE = x y'[x] + y[x] == x^2 (y[x])^2
```
Out[ ]=

$$y[x] + x\, y'[x] == x^{2}\, y[x]^{2}$$

Substitute $u = y^{1-n} = y^{1-2} = y^{-1}$

In[ ]:=
```
sub = y[x] → 1 / u[x]
```
Out[ ]=

$$y[x] \to \frac{1}{u[x]}$$

Differentiate on both sides of substitution

In[ ]:=
```
dsub = Simplify[D[sub, x]]
```
Out[ ]=

$$y'[x] \to -\frac{u'[x]}{u[x]^{2}}$$

Use substitution in original equation

In[∘]:=
```
uDE = Assuming[{n > 1, u[x] > 0}, FullSimplify[DE /. {sub, dsub}]]
```
Out[∘]=
```
u[x] == x (x + u′[x])
```

The DE is now linear

In[∘]:=
```
usol = DSolve[uDE, u[x], x][[1, 1]]
```
Out[∘]=
```
u[x] → -x^2 + x c1
```

Verify original equation

In[∘]:=
```
y[x_] = 1 / u[x] /. usol
```
Out[∘]=

$$\frac{1}{-x^2 + x\, c_1}$$

In[∘]:=
```
Simplify[DE]
```
Out[∘]=
```
True
```

Linear Models

In many models the rate of growth is proportional to the current state
$x'(t) = k\, x,\ x(0) = x_0$

Bacterial growth

Let $P(t)$ denote current population, k the growth rate

In[∘]:=
```
DE = P'[t] - k P[t] == 0; IC = P[0] == P0;
sol = DSolve[{DE, IC}, P[t], t][[1, 1]]
```
Out[∘]=
```
P[t] → e^(k t) P0
```

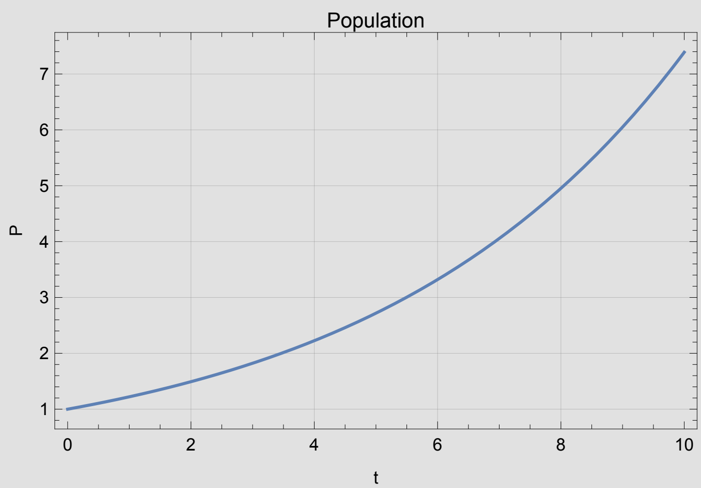
Define some common plot options for reuse

In[ ]:=
```
pltopt =
  {GridLines → Automatic, PlotLabel → title, Frame → True, FrameLabel → xylabel};
```

Plot the solution

In[ ]:=
```
Plot[P[t] /. sol /. {k → 0.2, P0 → 1}, {t, 0, 10},
 Evaluate[pltopt /. {title → "Population", xylabel → {"t", "P"}}]]
```

Out[ ]=



Radioactive decay

Let $A(t)$ denote current number of isotopes, k the decay rate

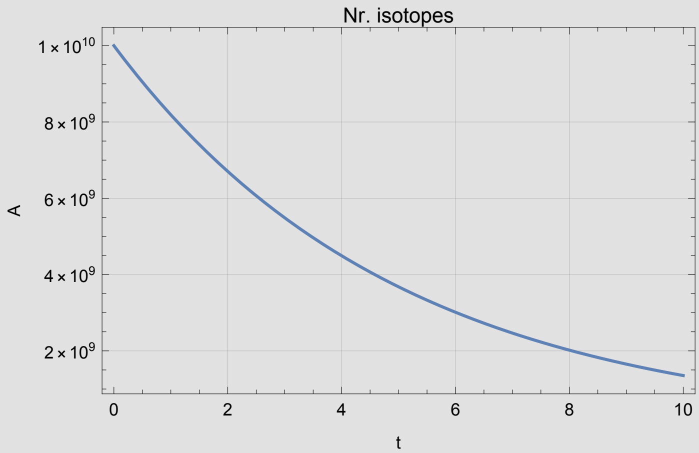
In[ ]:=
```
DE = A'[t] + k A[t] == 0; IC = A[0] == A0;
sol = DSolve[{DE, IC}, A[t], t][[1, 1]]
```

Out[ ]=

```
A[t] → A0 e^(-k t)
```

In[•]:=
```
Plot[A[t] /. sol /. {k → 0.2, A0 → 10^10}, {t, 0, 10},
 Evaluate[pltopt /. {title → "Nr. isotopes", xylabel → {"t", "A"}}]]
```

Out[•]=



LR circuit

An in-series electrical circuit of source $e(t)$, resistor R, and inductor L gives DE

$$L\,i' + R\,i = e\,(t)$$

In[•]:=
```
Plot[A[t] /. sol /. {k → 0.2, A0 → 10^10}, {t, 0, 10},
 Evaluate[pltopt /. {title → "Nr. isotopes", xylabel → {"t", "A"}}]]
```

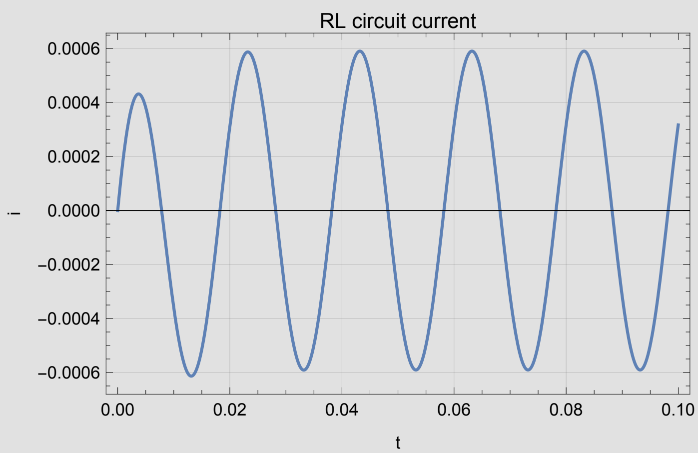
In[•]:=
```
DE = L i'[t] + R i[t] == e[t]; IC = i[0] == 0;
sol = DSolve[{DE, IC} /. e[t] → 110 Cos[100 π t], i[t], t]〚1, 1〛
```

Out[•]=

$$i[t] \to \frac{110\, e^{-\frac{R t}{L}} \left(-R + e^{\frac{R t}{L}} R \,\mathrm{Cos}[100\,\pi\, t] + 100\, e^{\frac{R t}{L}} L\, \pi\, \mathrm{Sin}[100\,\pi\, t]\right)}{10\,000\, L^2 \pi^2 + R^2}$$

```
In[ ]:= Plot[i[t] /. sol /. {R → 10^5, L → 500}, {t, 0, 0.1},
  Evaluate[pltopt /. {title → "RL circuit current", xylabel → {"t", "i"}}],
  PlotRange → All]
```

Out[ ]=



After initial transient, the circuit stabilizes at current imposed by source.

Nonlinear models

Variable population growth rate (logistic equation)

Environmental constraints on population growth can be modeled by the logistic equation

$$P' = r(1 - P/K)P$$

where r is the growth rate when $P \ll K$, and K is maximum population

```
In[ ]:= DE = P'[t] == r (1 - P[t] / K) P[t]; IC = P[0] == P0;
sol = DSolve[{DE, IC}, P[t], t][[1, 1]]
```

Solve : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

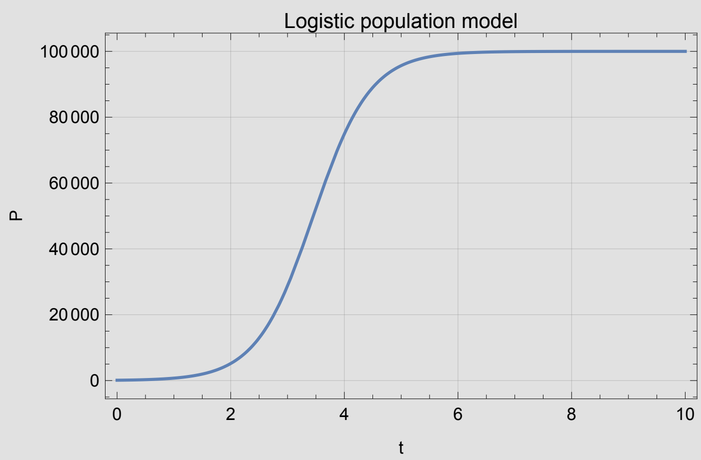
Out[ ]=

$$P[t] \to \frac{e^{r\,t}\,K\,P0}{K - P0 + e^{r\,t}\,P0}$$

Growth from population less than environment carrying capacity

In[ ]:=
```
Plot[P[t] /. sol /. {K → 10^5, r → 2, P0 → 100}, {t, 0, 10},
 Evaluate[pltopt /. {title → "Logistic population model", xylabel → {"t", "P"}}],
 PlotRange → All]
```

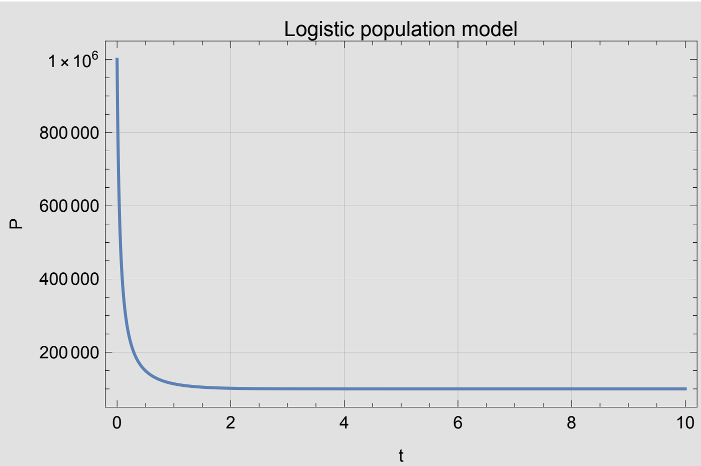
Out[ ]=



Decrease when population is greater than K

In[ ]:=
```
Plot[P[t] /. sol /. {K → 10^5, r → 2, P0 → 10^6}, {t, 0, 10},
 Evaluate[pltopt /. {title → "Logistic population model", xylabel → {"t", "P"}}],
 PlotRange → All]
```

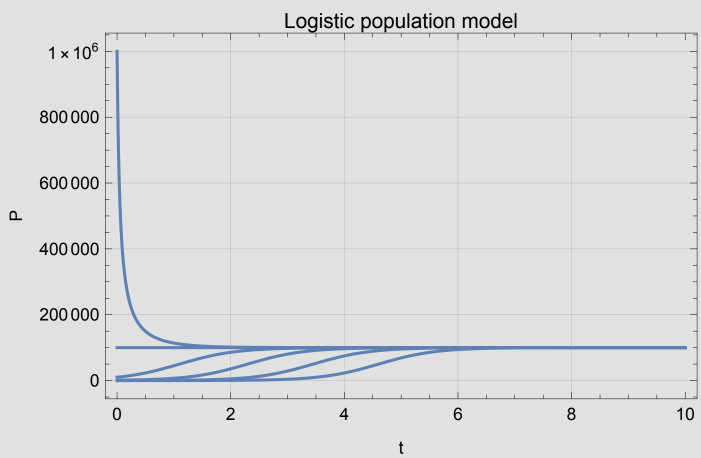
Out[ ]=



Multiple initial conditions rendered on same plot

In[•]:=
```
Plot[Table[P[t] /. sol /. {K → 10^5, r → 2, P0 → 10^p}, {p, 1, 6}], {t, 0, 10},
 Evaluate[pltopt /. {title → "Logistic population model", xylabel → {"t", "P"}}],
 PlotRange → All]
```

Out[•]=



Systems of first-order differential equations

Predator-prey model

Foxes $f(t)$ prey on rabbits $r(t)$, but lacking prey the fox population would decrease. The rabbit population increases (vegetation is freely available) in absence of foxes. The maximum number of rabbit- fox encounters is $f(t)\,r(t)$. The resulting (Lotka-Volterra) model is

$$f' = -a\,f + b\,f\,r,\; r' = c\,r - d\,f\,r$$

In[•]:=
```
DE = {f'[t] == -a f[t] + b f[t] × r[t], r'[t] == c r[t] - d f[t] × r[t]};
IC = {f[0] == 4, r[0] == 4};
IVP = Flatten[{DE, IC}];
params = {a → 0.16, b → 0.08, c → 4.5, d → 0.9};
```

An analytical solution of the above non-linear system is tedious to obtain. Compute a numerical approximation

In[ ]:=
```
sol = NDSolve[IVP /. params, {f[t], r[t]}, {t, 0, 20}][[1]]
```
Out[ ]=
```
{f[t] → InterpolatingFunction[ Domain: {{0., 20. }} Output: scalar ][t],
 r[t] → InterpolatingFunction[ Domain: {{0., 20. }} Output: scalar ][t]}
```

In[ ]:=
```
Plot[{f[t] /. sol, r[t] /. sol}, {t, 0, 20},
 Evaluate[pltopt /. {title → "Lotka–Volterra model", xylabel → {"t", "f,r"}}],
 PlotLegends → {"Foxes", "Rabbits"}]
```
Out[ ]=