DIFFERENTIAL EQUATION SOLVING WITH DSOLVE
## Contents

Introduction to Differential Equation Solving with DSolve ........................................... 1
Classification of Differential Equations ........................................................................... 7

### Ordinary Differential Equations (ODEs)
- Overview of ODEs ....................................................................................................... 11
- First-Order ODEs ....................................................................................................... 12
- Linear Second-Order ODEs ....................................................................................... 24
- Nonlinear Second-Order ODEs .................................................................................. 33
- Higher-Order ODEs ................................................................................................... 36
- Systems of ODEs ...................................................................................................... 42
- Lie Symmetry Methods for Solving Nonlinear ODEs .............................................. 50

### Partial Differential Equations (PDEs)
- Introduction to PDEs ................................................................................................ 55
- First-Order PDEs ...................................................................................................... 56
- Second-Order PDEs .................................................................................................. 66

### Differential-Algebraic Equations (DAEs)
- Introduction to DAEs ............................................................................................... 70
- Examples of DAEs ..................................................................................................... 71

### Initial and Boundary Value Problems
- Introduction to Initial and Boundary Value Problems ........................................... 76
- Linear IVPs and BVPs ............................................................................................. 77
- Nonlinear IVPs and BVPs ....................................................................................... 82
- IVPs with Piecewise Coefficients ........................................................................... 85

### Working with DSolve—A User’s Guide
- Introduction ............................................................................................................... 89
- Setting Up the Problem ............................................................................................ 90
- Verification of the Solution ...................................................................................... 94
- Plotting the Solution ............................................................................................... 98
- The GenerateParameters Options ....................................................................... 102
- Symbolic Parameters and Inexact Quantities ...................................................... 104
- Is the Problem Well-Posed? ................................................................................. 108

References .................................................................................................................... 111
Introduction to Differential Equation Solving with DSolve

The Mathematica function DSolve finds symbolic solutions to differential equations. (The Mathematica function NDSolve, on the other hand, is a general numerical differential equation solver.) DSolve can handle the following types of equations:

- **Ordinary Differential Equations** (ODEs), in which there is a single independent variable $t$ and one or more dependent variables $x_i(t)$. DSolve is equipped with a wide variety of techniques for solving single ODEs as well as systems of ODEs.

- **Partial Differential Equations** (PDEs), in which there are two or more independent variables and one dependent variable. Finding exact symbolic solutions of PDEs is a difficult problem, but DSolve can solve most first-order PDEs and a limited number of the second-order PDEs found in standard reference books.

- **Differential-Algebraic Equations** (DAEs), in which some members of the system are differential equations and the others are purely algebraic, having no derivatives in them. As with PDEs, finding exact solutions to DAEs is difficult, but DSolve can solve many examples of such systems that occur in applications.

DSolve @eqn, y[x], x@ solve a differential equation for $y[x]$

DSolve [{eqn1, eqn2, ...}, {y1 [x], y2 [x], ...}, x] solve a system of differential equations for $y_i[x]$

Finding symbolic solutions to ordinary differential equations.

DSolve returns results as lists of rules. This makes it possible to return multiple solutions to an equation. For a system of equations, possibly multiple solution sets are grouped together. You can use the rules to substitute the solutions into other calculations.

This finds the general solution for the given ODE. A rule for the function that satisfies the equation is returned.

In[1]:= DSolve [{y ' [x] == y [x]}, y [x], x ]

Out[1]= { {y[x] -> e^x C[1]}}
You can pick out a specific solution by using \( \text{ReplaceAll} \).

\[
\text{In}[2]:= \quad y[x] /. \text{DSolve}[\{y'[x] = y[x], \ y[x], \ x\}] \\
\text{Out}[2]= \{e^x c[1]\}
\]

A general solution contains arbitrary parameters \( c[i] \) that can be varied to produce particular solutions for the equation. When an adequate number of initial conditions are specified, \text{DSolve} returns particular solutions to the given equations.

Here, the initial condition \( y[0] = 1 \) is specified, and \text{DSolve} returns a particular solution for the problem.

\[
\text{In}[3]:= \quad \text{sol} = \text{DSolve}[\{y'[x] = y[x], \ y[0] = 1\}, \ y[x], \ x] \\
\text{Out}[3]= \{\{y[x] \to e^x\}\}
\]

This plots the solution. \text{ReplaceAll} (\( \text{/.} \)) is used in the \text{Plot} command to substitute the solution for \( y[x] \).

\[
\text{In}[4]:= \quad \text{Plot}[y[x] /. \text{sol}, \{x, -3, 2\}] \\
\text{Out}[4]=
\]

\begin{verbatim}
\text{DSolve}[eqn, y, x] \quad \text{solve a differential equation for } y \text{ as a pure function}
\text{DSolve}[\{eqn_1, eqn_2, \ldots\}, \{y_1, y_2, \ldots\}, x] \quad \text{solve a system of differential equations for the pure functions } y_i
\end{verbatim}

Finding symbolic solutions to ordinary differential equations as pure functions.

When the second argument to \text{DSolve} is specified as \( y \) instead of \( y[x] \), the solution is returned as a pure function. This form is useful for verifying the solution of the ODE and for using the solution in further work. More details are given in "Setting Up the Problem".
The solution to this differential equation is given as a pure function.

\[ \text{In[5]} := \text{eqn} = \{y'[x] = y[x]^2, y[0] = 1\}; \]
\[ \text{sol} = \text{DSolve[eqn, y, x]} \]
\[ \text{Out[6]} = \{\{y \to \text{Function}\{x\}, \frac{1}{1 - x}\}\} \]

This verifies the solution.

\[ \text{In[7]} := \text{eqn} /. \text{sol} \]
\[ \text{Out[7]} = \{8, True, True\} \]

This solves a system of ODEs. Each solution is labeled according to the name of the function (here, \(x\) and \(y\)), making it easier to pick out individual functions.

\[ \text{In[8]} := \text{eqns} = \{(t^2 + 1) \cdot x'[t] = -t \cdot x[t] + y[t] - \text{Sign}[t], (t^2 + 1) \cdot y'[t] = -x[t] - t \cdot y[t] + t \cdot \text{UnitStep}[t], x[0] = -1/2, y[0] = 2\}; \]
\[ \text{sol} = \text{DSolve[eqns, \{x, y\}, t]} \]
\[ \text{Out[9]} = \{\{x \to \text{Function}\{t\}, \frac{-1 + 4 t + 2 \left(\frac{\text{ArcTan}[t]}{-t} \text{True} + 2 t \left(\frac{\frac{1}{2} \text{Log}[1 + t^2]}{0} \text{True}\right)\}}{2 \left(1 + t^2\right)}\}
\]
\[ \{y \to \text{Function}\{t\}, \frac{4 + t - 2 t \left(\frac{\text{ArcTan}[t]}{-t} \text{True} + 2 t \left(\frac{\frac{1}{2} \text{Log}[1 + t^2]}{0} \text{True}\right)\}}{2 \left(1 + t^2\right)}\}\} \]

This substitutes a random value for the independent variable and shows that the solutions are correct at that point.

\[ \text{In[10]} := \text{eqns} /. \text{sol} /. \{t \to \text{RandomReal[\{}\}\}} \]
\[ \text{Out[10]} = \{\{\text{True}, \text{True}, \text{True}, \text{True}\}\} \]

This plots the solutions.

\[ \text{In[11]} := \text{Plot[\{x[t] /. sol, y[t] /. sol\}, \{t, -10, 10\}]} \]

\[ \text{Out[11]} = \]

\[
\text{DSolve[eqn, u[x, y], \{x, y\}]} \quad \text{solve a partial differential equation for } u[x, y]
\]

Finding symbolic solutions to partial differential equations.
While general solutions to ordinary differential equations involve arbitrary constants, general solutions to partial differential equations involve arbitrary functions. `DSolve` labels these arbitrary functions as $C[i]$.

Here is the general solution to a linear first-order PDE. In the solution, $C[1]$ labels an arbitrary function of $-x + y$.

```
In[12]:= eqn = x^2*D[u[x, y], x] + y^2*D[u[x, y], y] - (x + y)*u[x, y];
sol = DSolve[eqn == 0, u, {x, y}]
Out[13]= {{u -> Function[{x, y}, -x y C[1][(-x + y)/x y]]}}
```

This obtains a particular solution to the PDE for a specific choice of $C[1]$.

```
In[14]:= fn = u[x, y] /. sol[[1]] /. {C[1][t_] -> Sin[t^2] + (t/10)}
Out[14]= -x y (-x + y)/(10 x y) + Sin[-x + y]^2/x^2 y^2
```

Here is a plot of the surface for this solution.

```
In[15]:= Plot3D[fn, {x, -5, 5}, {y, -5, 5}]
```

`DSolve` can also solve differential-algebraic equations. The syntax is the same as for a system of ordinary differential equations.

This solves a DAE.

```
In[16]:= eqns = {f''[x] == g[x], f[x] + g[x] == 3 Sin[x], f[Pi] == 1, f'[Pi] == 0};
sol = DSolve[eqns, {f, g}, x]
Out[17]= {{f -> Function[{x}, 1/2 (-2 Cos[x] + 3 π Cos[x] - 3 x Cos[x] + 3 Sin[x])],
g -> Function[{x}, 1/2 (2 Cos[x] - 3 π Cos[x] + 3 x Cos[x] + 3 Sin[x])]}]
```
This verifies the solutions.

\[
\text{In[18]}:= \text{Simplify}[\text{eqns/.sol}]
\]

\[
\text{Out[18]}= \{\text{True, True, True, True}\}
\]

A plot of the solutions shows that their sum satisfies the algebraic relation 
\[f[x] + g[x] = 3 \sin[x].\]

\[
\text{In[19]}:= \text{Plot}[\{f[x]/.sol, g[x]/.sol, f[x]+g[x]/.sol\}, \{x, -5, 5\}]
\]

\[
\text{Out[19]}=
\]

**Goals of Differential Equation Solving with DSolve Tutorials**

The design of DSolve is modular: the algorithms for different classes of problems work independently of one another. Once a problem has been classified (as described in "Classification of Differential Equations"), the available methods for that class are tried in a specific sequence until a solution is obtained. The code has a hierarchical structure whereby the solution of complex problems is reduced to the solution of relatively simpler problems, for which a greater variety of methods is available. For example, higher-order ODEs are typically solved by reducing their order to 1 or 2.

The process described is done internally and does not require any intervention from the user. For this reason, these tutorials have the following basic goals.

- **To provide enough information and tips so that users can pose problems to DSolve in the most appropriate form and apply the solutions in their work.** This is accomplished through a substantial number of examples. A summary of this information is given in "Working with DSolve".

- **To give a catalog of the kinds of problems that can be handled by DSolve as well as the nature of the solutions for each case.** This is provided in the tutorials on ODEs, PDEs, DAEs, and boundary value problems (BVPs).
The author hopes that these Differential Equation Solving with DSolve tutorials will be useful in acquiring a basic knowledge of DSolve and also serve as a ready reference for information on more advanced topics.
Classification of Differential Equations

While differential equations have three basic types—ordinary (ODEs), partial (PDEs), or differential-algebraic (DAEs), they can be further described by attributes such as order, linearity, and degree. The solution method used by DSolve and the nature of the solutions depend heavily on the class of equation being solved.

The order of a differential equation is the order of the highest derivative in the equation.

This is a first-order ODE because its highest derivative is of order 1.

\[ \text{In[1]} := \text{DSolve}[x^2 (1 - x^2) \cdot y'[x] = (x - 3 x^3 - y[x]) y[x], y[x], x] \]

\[ \text{Out[1]} = \left\{ \left\{ y[x] \rightarrow \frac{-x + x^3}{C[1] - \text{Log}[x]} \right\} \right\} \]

Here is the general solution to a fourth-order ODE.

\[ \text{In[2]} := \text{DSolve}[y''''[x] - 16 \cdot y[x] = x^2, y[x], x] \]

\[ \text{Out[2]} = \left\{ \left\{ y[x] \rightarrow \frac{x^2}{16} + e^x C[1] + e^{-2x} C[2] + C[3] \cdot \text{Cos}[2x] + C[4] \cdot \text{Sin}[2x] \right\} \right\} \]

A differential equation is linear if the equation is of the first degree in \( y \) and its derivatives, and if the coefficients are functions of the independent variable.

This is a nonlinear second-order ODE that represents the motion of a circular pendulum. It is nonlinear because \( \text{Sin}[y[x]] \) is not a linear function of \( y[x] \). The Solve::ifun warning message appears because Solve uses JacobiAmplitude (the inverse of EllipticF) to find an expression for \( y[x] \).

\[ \text{In[3]} := \text{sol = DSolve}[y''[x] + 3 \cdot \text{Sin}[y[x]] = 0, y, x] \]

\[ \text{Out[3]} = \left\{ \left\{ y \rightarrow \text{Function} \left\{ x \right\}, -2 \cdot \text{JacobiAmplitude} \left[ \frac{1}{2} \cdot \sqrt{(6 + C[1]) \cdot (x + C[2])^2}, \frac{12}{6 + C[1]} \right] \right\} \right\}, \right\{ \left\{ y \rightarrow \text{Function} \left\{ x \right\}, 2 \cdot \text{JacobiAmplitude} \left[ \frac{1}{2} \cdot \sqrt{(6 + C[1]) \cdot (x + C[2])^2}, \frac{12}{6 + C[1]} \right] \right\} \right\} \]
This plots the solutions. The discontinuity in the graphs at \(x = -3\) results from the choice of inverse functions used by Solve.

\[
\text{In}[4]:= \text{Plot}[y[x] /. \text{sol}[[1]] /.(C[1] \rightarrow -1, C[2] \rightarrow 3), \\
\quad y[x] /. \text{sol}[[2]] /.(C[1] \rightarrow -1, C[2] \rightarrow 3)], \{x, -5, 5\}
\]

\[
\text{Out}[4]= -4 -2 2 4 -1.0 -0.5 0.5 1.0
\]

It should be noted that sometimes the solutions to fairly simple nonlinear equations are only available in implicit form. In these cases, \texttt{DSolve} returns an unevaluated \texttt{Solve} object.

This nonlinear differential equation only has an implicit solution. The \texttt{Solve::tdep} messages can be ignored; they appear because \texttt{Solve} cannot find an explicit expression for \(y'\) because non-algebraic functions (\texttt{ArcTan} and \texttt{Log}) are involved.

\[
\text{In}[5]:= \text{DSolve}[\frac{d}{dx}(y[x] + x - 1) * y'[x] - y[x] + 2 x + 3 = 0, y[x], x]
\]

\[
\text{Solve::tdep : The equations appear to involve the variables to be solved for in an essentially non–algebraic way. \[\]}
\]

\[
\text{Solve::tdep : The equations appear to involve the variables to be solved for in an essentially non–algebraic way. \[\]}
\]

\[
\text{Out}[5]= \text{Solve}\left[\frac{2}{3} \sqrt{2} \text{ArcTan}\left[\frac{-2 + \frac{2 (2 x)}{1-y(x)}}{\sqrt{2}}\right] - \log\left[\left(-1+x+y(x)\right)^2 \left(3+\frac{\frac{2 (2 x)}{1+y(x)}}{1-x+y(x)}\right)\right]}{(2+3 x)^2}\right] = C[1] + \frac{4}{3} \log[2+3 x], y\]
\]

When the coefficients of a linear ODE do not depend on \(x\), the ODE is said to have constant coefficients.

This is an ODE with constant coefficients.

\[
\text{In}[6]:= \text{eqn} = y'''[x] + 3 * y'[x] - 25 * y'[x] + 21 * y[x]; \\
\text{sol} = \text{DSolve}[\text{eqn} = 0, y[x], x]
\]

\[
\]

The previous equation is also homogeneous: all terms contain \(y\) or derivatives of \(y\) and its right-hand side is zero. Adding a function of the independent variable makes the equation inhomogeneous. The general solution to an inhomogeneous equation with constant coefficients is obtained by adding a particular integral to the solution to the corresponding homogeneous equation.
Here, $x^2$ is added to the right-hand side of the previous equation, making the new equation inhomogeneous. The general solution to this new equation is the sum of the previous solution and a particular integral.

```
In[8] := sol2 = DSolve[eqn == x^2, y[x], x]
```

```
Out[8] = 
  {y[x] -> (1124 + 1050 x + 441 x^2)/9261 + e^{-7 x} C[1] + e^x C[2] + e^{3 x} C[3]}
```

When the coefficients of an ODE depend on $x$, the ODE is said to have variable coefficients. Since equations with variable coefficients that are rational functions of $x$ have singularities that are easily classified, there are sophisticated algorithms available for solving them.

The coefficients of this equation are rational functions of $x$.

```
In[9] := sol = DSolve[{y'[x] - ((1 / x) - (3 / (16 x^2))) * y[x] == 0, y[1] == 1, y'[1] == 4}, y[x], x]
```

```
Out[9] = 
  {y[x] -> -1/8 e^{-2 - 2/Sqrt[x]} (-11 e^4 + 19 e^{4/Sqrt[x]} x^{1/4})}
```

There is a close relationship between functions and differential equations. Starting with a function of almost any type, it is possible to construct a differential equation satisfied by that function. Conversely, any differential equation gives rise to one or more functions, in the form of solutions to that equation. In fact, many special functions from classical analysis have their origins in the solution of differential equations. Mathieu functions are one such class of special functions. Mathieu was interested in studying the vibrations of elliptical membranes. The eigenfunctions for the wave equation that describes this motion are given by products of Mathieu functions.

This linear second-order ODE with rational coefficients has a general solution given by Mathieu functions.

```
In[10] := DSolve[(t - 1) (t + 1) * y''[t] + t * y'[t] + (-2 - 6 * t^2) * y[t] == 0, y[t], t]
```

```
Out[10] = 
```

The presence of $\text{ArcCos}[t]$ in the previous solution suggests that the equation can be given a simpler form using trigonometric functions. This is the form in which these equations were introduced by Mathieu in 1868.

```
In[11] := DSolve[y''[x] + (3 \text{Cos}[2 x] + 5) y[x] == 0, y, x]
```

```
Out[11] = 
```
This plots the surface for a particular product of solutions to this equation.

\[
\text{In[12]:= } \text{Plot3D} [\text{MathieuC}[5, -\frac{3}{2}, x] * \text{MathieuS}[5, -\frac{3}{2}, y], \{x, -3, 3\}, \{y, -3, 3\}]
\]

The degree of a differential equation is the highest power of the highest-order derivative in the equation.

This is a first-order ODE of degree 2.

\[
\text{In[13]:= } \text{sol = DSolve}[\{y'[x]^2 = 1 - x^2, y[0] = 2\}, y, x]
\]

\[
\text{Out[13]= } \{\{y \rightarrow \text{Function}[[x], \frac{1}{2} \left(4 - x \sqrt{1 - x^2} - \text{ArcSin}[x]\right)]\},
\{y \rightarrow \text{Function}[[x], \frac{1}{2} \left(4 + x \sqrt{1 - x^2} + \text{ArcSin}[x]\right)]\}\}
\]

The higher degree leads to non-uniqueness of the solution.

\[
\text{In[14]:= } \text{Plot}[[y[x] /. \text{sol[[1]]}], y[x] /. \text{sol[[2]]}], \{x, 0, 1\}]
\]

The examples in this tutorial have focused on the classification of ODEs. The classification of PDEs is similar but more involved. PDEs can also be classified by linearity or nonlinearity, order, degree, and constant or variable coefficients. More important is the classification that identifies a PDE as hyperbolic, parabolic, or elliptic. These classifications are discussed in further detail in "Second-Order PDEs".
Ordinary Differential Equations (ODEs)

Overview of Ordinary Differential Equations (ODEs)

There are four major areas in the study of ordinary differential equations that are of interest in pure and applied science.

- Exact solutions, which are closed-form or implicit analytical expressions that satisfy the given problem.

- Numerical solutions, which are available for a wider class of problems, but are typically only valid over a limited range of the independent variables.

- Qualitative theory, which is concerned with the global properties of solutions and is particularly important in the modern approach to dynamical systems.

- Existence and uniqueness theorems, which guarantee that there are solutions with certain desirable properties provided a set of conditions is fulfilled by the differential equation.

Of these four areas, the study of exact solutions has the longest history, dating back to the period just after the discovery of calculus by Sir Isaac Newton and Gottfried Wilhelm von Leibniz. The following table introduces the types of equations that can be solved by DSolve.
<table>
<thead>
<tr>
<th>name of equation</th>
<th>general form</th>
<th>date of discovery</th>
<th>mathematician</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separable</td>
<td>$y'(x) = f(x) , g(y)$</td>
<td>1691</td>
<td>G. Leibniz</td>
</tr>
<tr>
<td>Homogeneous</td>
<td>$y'(x) = f\left(\frac{x}{y(x)}\right)$</td>
<td>1691</td>
<td>G. Leibniz</td>
</tr>
<tr>
<td>Linear first-order ODE</td>
<td>$y'(x) + P(x) , y(x) = Q(x)$</td>
<td>1694</td>
<td>G. Leibniz</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>$y'(x) + P(x) , y(x) = Q(x)$</td>
<td>1695</td>
<td>James Bernoulli</td>
</tr>
<tr>
<td>Exact first-order ODE</td>
<td>$M , \frac{dx}{dx} + N , \frac{dy}{dy} = 0$ with $\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}$</td>
<td>1734</td>
<td>L. Euler</td>
</tr>
<tr>
<td>Clairaut</td>
<td>$y(x) = x , y'(x) + f(y'(x))$</td>
<td>1734</td>
<td>A-C. Clairaut</td>
</tr>
<tr>
<td>Linear with constant coefficients</td>
<td>$y^{(n)}(x) + a_{n-1} , y^{(n-1)}(x) + \ldots + a_0 , y(x) = P(x)$</td>
<td>1743</td>
<td>L. Euler</td>
</tr>
<tr>
<td>Hypergeometric</td>
<td>$x(1 - x) , y''(x) + (c - (a + b + 1) , x) , y'(x) - \frac{a , b , y(x)}{x} = 0$</td>
<td>1769</td>
<td>L. Euler</td>
</tr>
<tr>
<td>Legendre</td>
<td>$(1 - x^2) , y''(x) - 2 , x , y'(x) + n(n + 1) , y(x) = 0$</td>
<td>1785</td>
<td>M. Legendre</td>
</tr>
<tr>
<td>Bessel</td>
<td>$x^2 , y'''(x) + x , y''(x) + (x^2 - n^2) , y(x) = 0$</td>
<td>1824</td>
<td>F. Bessel</td>
</tr>
<tr>
<td>Mathieu</td>
<td>$y'''(x) + (a - 2 , q , \cos(2 , x)) , y(x) = 0$</td>
<td>1868</td>
<td>E. Mathieu</td>
</tr>
<tr>
<td>Abel</td>
<td>$y'(x) = f(x) + g(x) , y(x) + h(x) , y(x)^2 + k(x) , y(x)^3$</td>
<td>1834</td>
<td>N. H. Abel</td>
</tr>
<tr>
<td>Chini</td>
<td>$y'(x) = f , (x) + g , (x) , y(x) + h , (x) , y(x)^n$</td>
<td>1924</td>
<td>M. Chini</td>
</tr>
</tbody>
</table>

Examples of ODEs belonging to each of these types are given in other tutorials (clicking a link in the table will bring up the relevant examples).

## First-Order ODEs

### Straight Integration

This equation is solved by simply integrating the right-hand side with respect to $x$.

In[1]:= \[sol = DSolve[y'[x] == x^2 + Sin[x] + Sqrt[1 + x^2], y, x]\]

Out[1]= \[\left\{\left[y \to Function\left[x, \frac{1}{2} \, x \, \sqrt{1 + x^2} + \frac{\text{ArcSinh}[x]}{2} + C[1] - (-2 + x^2) \, \text{Cos}[x] + 2 \, x \, \text{Sin}[x]\right]\right]\]
Here is a plot of the integral curves for different values of the arbitrary constant C[1].

\[ \text{In[2]:=} \quad \text{tab = Table}[y[x] \/. \text{sol[[1]]} \/. \{C[1] \rightarrow k\}, \{k, -80, 80, 40\}]; \]

\[ \text{In[3]:=} \quad \text{Plot}[\text{Evaluate}[\text{tab}], \{x, 3, 18\}] \]

\[ \text{Out[3]=} \]

![Plot of integral curves](image)

### Separable Equations

The general solution to this equation is found by separation of variables.

\[ \text{In[1]:=} \quad \text{DSolve}[y'[x] = \frac{x^2 y[x]^2}{\sqrt{3 - x^2}}, y, x] \]

\[ \text{Out[1]=} \quad \{\{y \rightarrow \text{Function}[[x], \frac{2}{x \sqrt{3 - x^2} - 3 \text{ArcSin}\left[\frac{x}{\sqrt{3}}\right] - 2 C[1]}]} \}

Even when variables can be separated, the final solution might be accompanied by a warning message from Solve, or it might only be given as an InverseFunction object.

Solving this ODE generates a warning message because Solve obtains an expression for \(y[x]\) using Log, the inverse of Exp. This warning message can be ignored.

\[ \text{In[2]:=} \quad \text{DSolve}[y'[x] = \frac{x^2 \text{Exp}[y[x]]}{\sqrt{3 - x^2}}, y, x] \]

\[ \text{Out[2]=} \quad \{\{y \rightarrow \text{Function}[[x], -\text{Log}\left[\frac{1}{2} x \sqrt{3 - x^2} - \frac{3}{2} \text{ArcSin}\left[\frac{x}{\sqrt{3}}\right] - C[1]\right]} \}} \]
The solution to this equation is given as an InverseFunction object, in order to get an explicit expression for \( y[x] \).

\[
\text{In[3]:= } \text{sol = DSolve}\left[ y'[x] = \frac{x^2}{\sqrt{9 - x^2 \log[y[x]] \cdot \sin[y[x]]}} , y[x], x \right]
\]

InverseFunction::ifun : Inverse functions are being used. Values may be lost for multivalued inverses. 

InverseFunction::ifun : Inverse functions are being used. Values may be lost for multivalued inverses. 

Solve::tdep : The equations appear to involve the variables to be solved for in an essentially non-algebraic way. 

\[
\text{Out[3]= } \{ \{ y[x] \rightarrow \text{InverseFunction}[\text{CosIntegral}[x] - \text{Cos}[x] \log[x] \} & \}
\]

\[
- \left( \frac{1}{2} - \frac{1}{\sqrt{3 + x}} x \sqrt{3 + x} + 9 \text{ArcSin}\left[ \frac{\sqrt{3 + x}}{\sqrt{6}} \right] \right) + C[1] \}} \}
\]

### Homogeneous Equations

Here is a homogeneous equation in which the total degree of both the numerator and the denominator of the right-hand side is 2. The two parts of the solution list give branches of the integral curves in the form \( y = f(x) \).

\[
\text{In[1]:=} \text{eqn = } y'[x] = - (x^2 - 3 y[x]^2) / (x \cdot y[x]); \\
\text{sol = DSolve}\left[ \text{eqn, y, x} \right]
\]

\[
\text{Out[2]= } \{ \{ y[x] \rightarrow \text{Function}[\{ x, - \frac{x^2}{2} x^6 C[1] \}] \}, \{ y[x] \rightarrow \text{Function}[\{ x, \sqrt[3]{x^6 C[1]} \}] \}\}
\]

This plots both branches together, showing the complete integral curves \( y^2 = C[1] x^6 + \frac{x^2}{2} \) for several values of \( C[1] \).

\[
\text{In[3]:=} \text{tab1 = Table}[y[x] /. \text{sol[[1]]} /. C[1] \rightarrow k, \{ k, 0, 3, 0.5 \}]; \\
\text{In[4]:=} \text{tab2 = Table}[y[x] /. \text{sol[[2]]} /. C[1] \rightarrow k, \{ k, 0, 3, 0.5 \}]; \\
\text{In[5]:=} \text{Plot}[\text{Evaluate}[\text{Join[tab1, tab2]], \{x, 0, 1.7\}]]
\]

![Plot of integral curves](image-url)
If an initial condition is specified, DSolve picks the branch that passes through the initial point. The DSolve::bvnul message indicates that one branch of the general solution (the lower branch in the previous graph) did not give a solution satisfying the given initial condition y[1] = 3.

\[ \text{In[6]:=} \text{DSolve}\{\text{eqn, y[1] = 3}, \ y[x], \ x\} \]

\[ \text{DSolve::bvnul : For some branches of the general solution, the given boundary conditions lead to an empty solution.} \]

\[ \text{Out[6]} = \left\{ \left\{ y[x] \rightarrow \frac{\sqrt{x^2 (1 + 17 x^4)}}{\sqrt{2}} \right\} \right\} \]

**Linear First-Order Equations**

The following is a linear first-order ODE because both y[x] and y'[x] occur in it with power 1 and y'[x] is the highest derivative. Note that the solution contains the imaginary error function Erfi.

\[ \text{In[1]:=} \text{DSolve}\{y'[x] + x y[x] = \text{Exp}[3 x], \ y[x], \ x\} \]

\[ \text{Out[1]} = \left\{ \left\{ y[x] \rightarrow e^{-\frac{x^2}{2}} C[1] + e^{-\frac{x^2}{2}} \sqrt{\frac{\pi}{2}} \text{Erfi}\left[\frac{3 + x^2}{\sqrt{2}}\right] \right\} \right\} \]

Here is the solution for a more general linear first-order ODE. The K variables are used as dummy variables for integration. The Erfi term in the previous example comes from the integral in the second term of the general solution as follows.

\[ \text{In[2]:=} \text{sol = DSolve}\{y'[x] + y[x] = Q[x], \ y[x], \ x\} \]

\[ \text{Out[2]} = \left\{ \left\{ y[x] \rightarrow e^{-x} C[1] + e^{-x} \int_{1}^{x} e^{K[1]} Q[K[1]] \, dK[1] \right\} \right\} \]

A more traditional form of the solution can be obtained by replacing K[1] with a variable such as t.

\[ \text{In[3]:=} \text{sol /. \{K[1] \rightarrow t\}} \]

\[ \text{Out[3]} = \left\{ \left\{ y[x] \rightarrow e^{-x} C[1] + e^{-x} \int_{1}^{x} e^{t} Q[t] \, dt \right\} \right\} \]

**Inverse Linear Equations**

It may happen that a given ODE is not linear in y(x) but can be viewed as a linear ODE in x(y). In this case, it is said to be an inverse linear ODE.
This is an inverse linear ODE. It is constructed by interchanging $x$ and $y$ in an earlier example.

```math
In[1]:= DSolve[y'[x] == 1 / (-x*y[x] + Exp[3*y[x]]), y[x], x]
```

Solve::tdep:
The equations appear to involve the variables to be solved for in an essentially non-algebraic way.

```math
Out[1]= Solve\left\{ x = e^{\frac{1}{2} y[x]^2} C[1] + e^{\frac{1}{2} y[x]^2} \sqrt{\frac{\pi}{2}} \text{Erfi}\left(\frac{3 + y[x]}{\sqrt{2}}\right), y[x] \right\}
```

**Bernoulli Equations**

A Bernoulli equation is a first-order equation of the form

$$y'(x) + P(x) y(x) = Q(x)y(x)^n.$$  

The problem of solving equations of this type was posed by James Bernoulli in 1695. A year later, in 1696, G. Leibniz showed that it can be reduced to a linear equation by a change of variable.

Here is an example of a Bernoulli equation.

```math
In[1]:= eqn = y'[x] + 11 x*y[x] == x^3*y[x]^3
```

```math
Out[1]= 11 x y[x] + y'[x] = x^3 y[x]^3
```

```math
In[2]:= sol = DSolve[eqn, y, x]
```

```math
Out[2]= \left\{ \left\{ y \rightarrow \text{Function}\left[\{x\}, \frac{11}{\sqrt{1 + 11 x^2 + 121 e^{11 x^2} C[1]}}\right] \right\}, \left\{ y \rightarrow \text{Function}\left[\{x\}, \frac{11}{\sqrt{1 + 11 x^2 + 121 e^{11 x^2} C[1]}}\right] \right\} \right\}
```

This verifies that the solution is correct.

```math
In[3]:= eqn /. sol // Simplify
```

```math
Out[3]= \{True, True\}
```

In general, the solution to a Bernoulli equation will consist of $n - 1$ branches, where $n$ is the degree of $y(x)$ in the equation.
Here is an example of a Bernoulli equation with \( n = 5 \). The solution has four branches.

```math
In[4]:= DSolve[3 x * y'[x] - 7 x * Log[x] * y[x] ^ 5 - y[x] = 0, y[x], x]
```

```math
Out[4]= \{\{y[x] \to \frac{7^{1/4} x^{1/3}}{(12 x^{7/3} + 7 C[1] - 28 x^{7/3} Log[x])^{1/4}}\}, \{y[x] \to \frac{i 7^{1/4} x^{1/3}}{(12 x^{7/3} + 7 C[1] - 28 x^{7/3} Log[x])^{1/4}}\}, \{y[x] \to \frac{7^{1/4} x^{1/3}}{(12 x^{7/3} + 7 C[1] - 28 x^{7/3} Log[x])^{1/4}}\}\}
```

### Riccati Equations

A Riccati equation is a first-order equation of the form

\[
y'(x) = f(x) + g(x) y(x) + h(x) y(x)^2.
\]

This equation was used by Count Riccati of Venice (1676-1754) to help in solving second-order ordinary differential equations.

Solving Riccati equations is considerably more difficult than solving linear ODEs.

Here is a simple Riccati equation for which the solution is available in closed form.

```math
In[1]:= DSolve[y'[x] + (2 / x^2) - 3 y[x] ^ 2 = 0, y[x], x] // Simplify
```

```math
Out[1]= \{\{y[x] \to \frac{-3 x^2 - 2 C[1]}{3 x^6 + 3 x C[1]}\}\}
```

Any Riccati equation can be transformed to a second-order linear ODE. If the latter can be solved explicitly, then a solution for the Riccati equation can be derived.

Here is an example of a Riccati equation and the corresponding second-order ODE, which is Legendre's equation.

```math
In[2]:= DSolve[u'[x] = ((2 x) / (1 - x ^ 2)) * u[x] - ((15 / 4) / (1 - x ^ 2)) - u[x] ^ 2, u[x], x] // Simplify
```

```math
Out[2]= \{\{u[x] \to \frac{\frac{3}{2} - x C[1] \text{LegendreP}\left[\frac{3}{2}, x\right] + C[1] \text{LegendreP}\left[\frac{5}{2}, x\right] - x \text{LegendreQ}\left[\frac{3}{2}, x\right] + \text{LegendreQ}\left[\frac{5}{2}, x\right]\}}{2 (1 + x^2) \left(C[1] \text{LegendreP}\left[\frac{3}{2}, x\right] + \text{LegendreQ}\left[\frac{3}{2}, x\right]\right)\}}\}
```

```math
In[3]:= DSolve[\(\(1 - x ^ 2\) * y''[x] - 2 x * y'[x] + (15 / 4) * y[x] = 0, y[x], x\]
```

```math
Out[3]= \{\{y[x] \to C[1] \text{LegendreP}\left[\frac{3}{2}, x\right] + C[2] \text{LegendreQ}\left[\frac{3}{2}, x\right]\}\}
```
Finally, consider the following Riccati equation. It is integrable because the sum of the coefficients of the terms on the right-hand side is 0.

\[ \text{In[4]} := \text{eqn} = y' + 3x + 5y - (3x + 5) y^2; \]

\[ \text{In[5]} := \text{RightHandSideCoeffs} = \{3x, 5, -(3x + 5)\}; \]

\[ \text{In[6]} := \text{Total}[\text{RightHandSideCoeffs}] \]

\[ \text{Out[6]} = 0 \]

\[ \text{In[7]} := \text{sol} = \text{DSolve}[\text{eqn}, y, x] \]

\[ \text{Out[7]} = \left\{ \left\{ y \rightarrow \text{Function}\left[ \{x\}, 1 + \frac{e^{-5x - 3x^2}}{C[1] + \frac{1}{\sqrt{3}} \left( -6 e^{-x (5 + 3x)} + 5 e^{25/12} \text{Erf}\left[ \frac{5 + 6x}{2 \sqrt{3}} \right] \right)} \right] \right\} \right\} \]

This verifies the solution.

\[ \text{In[8]} := \text{eqn} /. \text{sol} // \text{Simplify} \]

\[ \text{Out[8]} = \{\text{True}\} \]

**Exact Equations**

Here is an example of an exact ODE.

\[ \text{In[1]} := \text{P}[x_, y_] := -\left(5x^2 - 2y^2 + 11\right) \]

\[ \text{In[2]} := \text{Q}[x_, y_] := (\text{Sin}[y] + 4x \cdot y + 3) \]

\[ \text{In[3]} := \text{Simplify}[\text{D}[\text{P}[x, y], y] - \text{D}[\text{Q}[x, y], x]] \]

\[ \text{Out[3]} = 0 \]

\[ \text{In[4]} := \text{eqn} = y'[x] == -\text{P}[x, y[x]] / \text{Q}[x, y[x]] \]

\[ \text{Out[4]} = y'[x] = \frac{11 + 5x^2 - 2y[x]^2}{3 \text{Sin}[y[x]] + 4x y[x]} \]

\[ \text{In[5]} := \text{sol} = \text{DSolve}[\text{eqn}, y[x], x] \]

\[ \text{Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\[ \text{Out[5]} = \text{Solve}\left[ -11 x - \frac{5x^3}{3} - \text{Cos}[y[x]] + 3 y[x] + 2xy[x]^2 = C[1], y[x] \right] \]
This verifies the solution.

\[
\text{In[6]} := \text{Solve[D[\text{sol}[[1]], \text{x}], \text{y}'[\text{x}]]} \hspace{1pt} \text{// Simplify}
\]

\[
\text{Out[6]} = \left\{ \left\{ \text{y}'[\text{x}] \rightarrow \frac{11 + 5 \text{x}^2 - 2 \text{y}[\text{x}]^2}{3 + \text{Sin}[\text{y}[\text{x}]] + 4 \text{x} \text{y}[\text{x}]} \right\} \right\}
\]

Here is a contour plot of the solution.

\[
\text{In[7]} := \text{ContourPlot[Evaluate[\text{sol}[[1, 1]]} \hspace{1pt} \text{// \{y[\text{x}] \rightarrow \text{y}\}}, \{\text{x}, -5, 5\}, \{\text{y}, -5, 5\}]
\]

If an equation is not exact, it may be possible to find an integrating factor (a multiplier for the functions \(P\) and \(Q\), defined previously) that converts the equation into exact form. \text{DSolve} tries a variety of techniques to automatically find integrating factors in such situations.

**Clairaut Equations**

A Clairaut equation is a first-order equation of the form

\[
y(x) = x \, y'(x) + f(y'(x)).
\]

A remarkable feature of this nonlinear equation is that its general solution has a very simple form.
This is an example of a Clairaut equation. The warning message from Solve can be ignored. It is given because DSolve first tries to find an expression for \( y'[x] \) from the given ODE.

\[
\text{In}[1]:= \quad \text{sol} = \text{DSolve}[y'[x] == x \cdot y'[x] + y'[x]^2 + \text{Exp}[y'[x]], y[x], x]
\]

\[
\text{Out}[1]= \quad \left\{ \{y[x] \rightarrow e^{c[1]} + x \cdot c[1] + c[1]^2\} \right\}
\]

The general solution to Clairaut equations is simply a family of straight lines.

This plots the solution for several values of \( c[1] \).

\[
\text{In}[2]:= \quad \text{Plot[Evaluate[Table[y[x] /. sol /. \{c[1] \rightarrow 1 / k\}, \{k, -5, 5, 2\}], \{x, 1, 5\}]]}
\]

\[
\text{Out}[2]=
\]

**Abel Equations**

An Abel ODE is a first-order equation of the form

\[
y'(x) = f(x) + g(x) y(x) + h(x) y(x)^2 + k(x) y(x)^3.
\]

This equation arose in the context of the studies of Niels Henrik Abel on the theory of elliptic functions, and represents a natural generalization of the Riccati equation.

Associated with any Abel ODE is a sequence of expressions that is built from the coefficients of the equation \( \{f_0, f_1, f_2, f_3\} \) and invariant under certain coordinate transformations of the independent variable and the dependent variable. These invariants characterize each equation and can be used for identifying integrable classes of Abel ODEs. In particular, Abel ODEs with zero or constant invariants can be integrated easily and constitute an important integrable class of these equations.
Here is the construction of a particular invariant with value 0 and the solution of the corresponding Abel ODE.

\[\text{In}[1] := \begin{align*}
f_0 &= 0; & f_1 &= -1; & f_2 &= -3; & f_3 &= x;
\end{align*}\]

\[\text{In}[2] := \begin{align*}
\text{Invariant} &= f_0 f_3^2 + \frac{1}{3} \left(\frac{2 f_2^3}{9} - f_1 f_3 f_2 - \partial_x f_3 f_2 + f_3 \partial_x f_2\right)
\end{align*}\]

\[\text{Out}[2] = 0\]

\[\text{In}[3] := \text{AbelODE} = y'[x] = f_0 + f_1 y[x] + f_2 y[x]^2 + f_3 y[x]^3\]

\[\text{Out}[3] = \frac{y'[x]}{x} = -3 y[x]^2 + x y[x]^3\]

\[\text{In}[4] := \text{sol} = \text{DSolve}[\text{AbelODE, } y, x]\]

\[\text{Out}[4] = \begin{Bmatrix}
\{y \rightarrow \text{Function}\{x\}, \frac{1}{x} + \frac{1}{x^2 \sqrt{\frac{1}{x^2} + C[1]}}\},
\{y \rightarrow \text{Function}\{x\}, -\frac{1}{x} + \frac{1}{x^2 \sqrt{\frac{1}{x^2} + C[1]}}\}\end{Bmatrix}\]

\[\text{In}[5] := \text{AbelODE} /. \text{sol} \text{ // } \text{Simplify}\]

\[\text{Out}[5] = \{\text{True, True}\}\]

\[\text{In}[6] := \text{Clear}[f_0, f_1, f_2, f_3]\]

Another important class of integrable Abel ODEs are those that can be reduced to inverse linear first-order ODEs using a nonlinear coordinate transformation.

This Abel ODE is solved by transforming it to an inverse linear first-order ODE. The \text{ExpIntegralEi} term in the solution to this equation comes from solving the linear ODE.

\[\text{In}[7] := \text{DSolve}\left[y'[x] = y[x]^3 - \frac{x y[x]^2}{x - 1}, y[x], x\right]\]

\[\text{Out}[7] = \text{Solve}\left\{\frac{e^{\frac{1}{y[x]}} - 1}{y[x]} + C[1] + \text{ExpIntegralEi}\left[1 - \frac{1}{y[x]}\right] = 0, y[x]\right\}\]

Another important class of integrable Abel ODEs consists of those that can be transformed to an inverse Riccati equation. Since Riccati equations can be transformed to second-order linear ODEs, the solutions for this class are usually given in terms of special functions such as \text{AiryAi} and \text{BesselJ}. 
This Abel ODE is solved by reducing it to an inverse Riccati equation.

\[
\text{In}[8]:= \text{AbelODE} = y'[x] == \frac{y[x]^3}{8 x^2} - y[x]^2
\]

\[
\text{Out}[8]= y'[x] = -y[x]^2 + \frac{y[x]^3}{8 x^2}
\]

\[
\text{In}[9]:= \text{sol} = \text{DSolve}[\text{AbelODE}, y'[x], x]
\]

InverseFunction::ifun: Inverse functions are being used. Values may be lost for multivalued inverses. >>
InverseFunction::ifun: Inverse functions are being used. Values may be lost for multivalued inverses. >>
InverseFunction::ifun: Inverse functions are being used. Values may be lost for multivalued inverses. >>
General::stop: Further output of InverseFunction::ifun will be suppressed during this calculation. >>

Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way. >>

\[
\text{Out}[9]= \text{Solve}
\]

\[
C[1] + \left\{\text{AiryAiPrime}\left[\frac{-(-1)^{2/3}}{2 \times 2^{1/3} x} + \left(-2\right)^{1/3} x + \left(-2\right)^{1/3} \frac{y[x]}{y[x]}\right]\right\} + \text{AiryAi}\left[\frac{-(-1)^{2/3}}{2 \times 2^{1/3} x} + \left(-2\right)^{1/3} x + \left(-2\right)^{1/3} \frac{y[x]}{y[x]}\right] + \\
\text{AiryBi}\left[\frac{-(-1)^{2/3}}{2 \times 2^{1/3} x} + \left(-2\right)^{1/3} x + \left(-2\right)^{1/3} \frac{y[x]}{y[x]}\right]\right\} = 0, y[x]
\]

This verifies the solution.

\[
\text{In}[10]:= \text{Solve[D[\text{sol}[[1]], x], y'[x]] // \text{FullSimplify}}
\]

\[
\text{Out}[10]= \left\{y'[x] \rightarrow \frac{1}{8} y[x]^2 \left(-8 + \frac{y[x]}{x^2}\right)\right\}
\]

The Abel ODEs considered so far are said to be of the \textit{first kind}. Abel ODEs of the \textit{second kind} are given by the following general formula.

\[
y'(x) = \frac{f(x) + g(x) y(x) + h(x) y(x)^2 + k(x) y(x)^3}{a(x) + b(x) y(x)}
\]

An Abel ODE of the second kind can be converted to an equation of the first kind with a coordinate transformation. Thus, the solution methods for this kind of Abel ODE are identical to the methods for equations of the first kind.
Here is the solution for an Abel ODE of the second kind.

\[
\text{In[11]} := \text{sol = DSolve}\left[ y'[x] = \frac{-\frac{2x}{9} + x^3 + y[x]}{y[x]}, y[x], x \right]
\]

\text{Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\text{Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\text{Out[11]} = \text{Solve}\left[ C[1] = \left( -1 + \left( \frac{\sqrt{2}}{3} \frac{y[x]}{x^2} - \frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right) ^2 \right) ^{1/4} \right.

\left\{ \begin{array}{c}
\frac{3x}{\sqrt{2}} - \frac{\sqrt{2} \text{Hypergeometric2F1}\left[ \frac{1}{2}, 3, 1, -\frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right] \left( \frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right)^2 \right) \left( \frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right)^2 \right. \\
2 \left( 1 - \left( \frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right)^2 \right)^{1/4} \\
\frac{\sqrt{2} \sqrt{y[x]}}{x^2} \right) \right\}

\text{This verifies the solution.}

\text{In[12]} := \text{DSolve}\left[ \text{D[sol[[1]], x]}, y'[x] \right] \text{// FullSimplify}

\text{Out[12]} = \left\{ y'[x] \to 1 + \frac{-\frac{2x}{9} + x^3}{y[x]} \right\}

\textbf{Chini Equations}

Chini equations are a generalization of Abel and Riccati equations.

This solves a Chini equation.

\text{In[1]} := \text{DSolve}\left[ y'[x] = 5 \times y[x] ^ 4 + 3 \times x ^ (-4/3), y[x], x \right]

\text{Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\text{Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\text{Out[1]} = \text{Solve}\left[ -45 \text{RootSum}\left[ -45 + \frac{3^{1/4} 5^{3/4}}{y[x]} - 45 \text{Root}[\_11 + \left( \frac{5}{3} \right)^{1/4} \left( x^{4/3} \right)^{1/4} y[x]] & \right] = \frac{\text{Log}\left[ -\text{Root}\left[ \_11 + \left( \frac{5}{3} \right)^{1/4} \left( x^{4/3} \right)^{1/4} y[x] \right] \right]}{3^{1/4} 5^{3/4} - 180 \_11^3} \times \frac{3^{1/4} 5^{1/4} \left( x^{4/3} \right)^{1/4} \text{Log}[x]}{\_11^{1/3}}, y[x] \right]
Linear Second-Order ODEs

Overview of Linear Second-Order ODEs

Solving linear first-order ODEs is straightforward and only requires the use of a suitable integrating factor. In sharp contrast, there are a large number of methods available for handling linear second-order ODEs, but the solution to the general equation belonging to this class is still not available. Therefore the linear case is discussed in detail before moving on to nonlinear second-order ODEs.

The general linear second-order ODE has the form

$$y''(x) + P(x) y'(x) + Q(x) y(x) = R(x).$$

Here, $P(x)$, $Q(x)$ and $R(x)$ are arbitrary functions of $x$. The term "linear" refers to the fact that the degree of each term in $y(x)$, $y'(x)$ and $y''(x)$ is 1. (Thus, terms like $y(x)^2$ or $y(x) y''(x)$ would make the equation nonlinear.)

Linear Second-Order Equations with Constant Coefficients

The simplest type of linear second-order ODE is one with constant coefficients.

This linear second-order ODE has constant coefficients.

$$In[1]:= \text{sol} = \text{DSolve}[y''[x] + 5 y'[x] - 6 y[x] = 0, y, x]$$

Notice that the general solution is a linear combination of two exponential functions. The arbitrary constants $C[1]$ and $C[2]$ can be varied to produce particular solutions.

This is one particular solution to the equation.

$$In[2]:= \text{soll} = y[x] /. \text{sol}[[1]] /. \{C[1] \rightarrow 2, C[2] \rightarrow 3\}$$

The exponents $-6$ and $1$ in the basis $\{e^{-6x}, e^x\}$ are obtained by solving the associated quadratic equation. This quadratic equation is called the auxiliary or characteristic equation.
This solves the auxiliary equation.

\[\text{In}[3]:= \text{Solve}[m^2 + 5 m - 6 = 0, m]\]

\[\text{Out}[3]= \{\{m \to -6\}, \{m \to 1\}\}\]

The roots are real and distinct in this case. There are two other cases of interest: real and equal roots, and imaginary roots.

This example has real and equal roots.

\[\text{In}[4]:= \text{sol = DSolve[y''[x] - 6 \, y'[x] + 9 \, y[x] = 0, y, x}\]

\[\text{Out}[4]= \{\{y \to \text{Function}[\{x\}, e^{3 \, x} \, C[1] + e^{3 \, x} \, C[2]]\}\}\]

\[\text{In}[5]:= \text{sol2 = y[x] /. \text{sol[[1]]} /. \{C[1] \to 2, C[2] \to 3\}\]

\[\text{Out}[5]= 2 \, e^{3 \, x} + 3 \, e^{3 \, x} \, x\]

This example has roots with nonzero imaginary parts.

\[\text{In}[6]:= \text{sol = DSolve[y''[x] - 6 \, y'[x] + 9 \, y[x] = 0, y, x}\]

\[\text{Out}[6]= \{\{y \to \text{Function}[\{x\}, e^{x/2} \, C[1] \, \cos\left(\frac{\sqrt{3} \, x}{2}\right) + e^{x/2} \, C[2] \, \sin\left(\frac{\sqrt{3} \, x}{2}\right)\}\}\}\]

\[\text{In}[7]:= \text{sol3 = y[x] /. \text{sol[[1]]} /. \{C[1] \to 2, C[2] \to 3\}\]

\[\text{Out}[7]= 2 \, e^{x/2} \, \cos\left(\frac{\sqrt{3} \, x}{2}\right) + 3 \, e^{x/2} \, \sin\left(\frac{\sqrt{3} \, x}{2}\right)\]

Here is a plot of the three solutions.

\[\text{In}[8]:= \text{Plot}\{\{\text{sol1, sol2, sol3}\}, \{x, -0.5, 0.5\}\}

\[\text{Out}[8]=\]
Euler and Legendre Equations

An *Euler equation* has the general form

\[ x^2 y''(x) + a x y'(x) + b y(x) = 0. \]

Euler equations can be solved by transforming them to equations with constant coefficients.

This is an example of an Euler equation.

```math
In[1]:= DSolve[x^2 * y''[x] + 5 * x * y'[x] + 6 * y[x] == 0, y[x], x]
```

```
```

The *Legendre linear equation* is a generalization of the Euler equation. It is an ODE of the form

\[ (c x + d)^2 y''(x) + a (c x + d) y'(x) + b y(x) = 0. \]

Here is an example of a Legendre linear equation.

```math
In[2]:= DSolve[(3 x + 1)^2 * y''[x] + 5 * (3 x + 1) * y'[x] + 6 * y[x] == 0, y[x], x]
```

```
```

Exact Linear Second-Order Equations

A linear second-order ordinary differential equation

\[ a_0(x) y''(x) + a_1(x) y'(x) + a_2(x) y(x) = 0 \]

is said to be *exact* if

\[ a_0''(x) - a_1'(x) + a_2(x) = 0. \]

An exact linear second-order ODE is solved by reduction to a linear first-order ODE.

Here is an example. The appearance of the unevaluated integral in the solution is explained here.

```math
In[1]:= a[0] = 1;
In[2]:= a[1] = Log[x];
```
\[ In[3] := \quad a_2 = 1/x; \]
\[ In[4] := \quad \text{eqn} = a_0 \cdot y''[x] + a_1 \cdot y'[x] + a_2 \cdot y[x] = 0; \]
\[ In[5] := \quad \text{conditionforexactness} = (D[a0, \{x, 2\}] - D[a1, x] + a2 = 0) \]
\[ Out[5] = \text{True} \]

\[ In[6] := \quad \text{sol} = \text{DSolve}[\text{eqn}, y, x] \]
\[ Out[6] = \left\{\left[y \to \text{Function}\left[\{x\}, e^{-x \log[x]} C[2] + e^{x \log[x]} \int_1^x e^{-1.K[1].1.K[1]} \log[K[1]] C[1] dK[1]\right]\right]\right\} \]

This verifies the solution.
\[ In[7] := \quad \text{eqn} /. \text{sol} // \text{FullSimplify} \]
\[ Out[7] = \{\text{True}\} \]
\[ In[8] := \quad \text{Clear}[a0, a1, a2] \]

**Linear Second-Order Equations with Solutions Involving Special Functions**

\text{DSolve} can find solutions for most of the standard linear second-order ODEs that occur in applied mathematics.

Here is the solution for Airy’s equation.
\[ In[1] := \quad \text{DSolve}[y''[x] - x \cdot y[x] = 0, y[x], x] \]
\[ Out[1] = \left\{\{y[x] \to \text{AiryAi}[x]C[1] + \text{AiryBi}[x]C[2]\}\right\} \]

Here is a plot that shows the oscillatory behavior of the Airy functions for large negative values of \(x\).
\[ In[2] := \quad \text{Plot}[\{\text{AiryAi}[x], \text{AiryBi}[x]\}, \{x, -15, -12\}] \]
The solution to this equation is given in terms of the derivatives of the Airy functions, AiryAiPrime and AiryBiPrime.

\[ a x y' - (a (a x + b))^2 y = 0, \quad y, \quad x \]

Out[3]= 
\[ \{y \to \text{Function}[[x], \text{AiryAiPrime}[b + a x] C[1] + \text{AiryBiPrime}[b + a x] C[2]]\} \]

Here is the solution for Bessel’s equation with \( n = 4 \). Note that the solution is given in terms of Bessel functions of the first kind, BesselJ, as well as those of the second kind, BesselY.

\[ x^2 y'' + x y' + (x^2 - 16) y = 0, \quad y[x], \quad x \]

Out[4]= 
\[ \{y[x] \to \text{BesselJ}[4, x] C[1] + \text{BesselY}[4, x] C[2]\} \]

Here is a plot of the BesselJ functions for specific values of \( n \).

\[ \text{Plot}[[\text{BesselJ}[1, x], \text{BesselJ}[3, x], \text{BesselJ}[4, x]], \{x, 0, 10\}] \]

Out[5]=

\[ \text{Plot image} \]

Here is the general solution for Legendre’s equation with \( n = 7 \).

\[ 16 (1 - x^2) y'' - 32 x y' + 21 y = 0, \quad y[x], \quad x \]

Out[6]= 
\[ \{y[x] \to C[1] \text{LegendreP}[\frac{3}{4}, x] + C[2] \text{LegendreQ}[\frac{3}{4}, x]\} \]

These special functions can be expressed in terms of elementary functions for certain values of their parameters. Mathematica performs this conversion automatically wherever it is possible.

These are some of these expressions that are automatically converted.

\[ \text{In}[7]= \{\text{BesselJ}[1/2, x], \text{LegendreP}[4, x], \text{HermiteH}[5, x]\} \]

Out[7]= 
\[ \left\{ \frac{\sqrt{2} \sin[x]}{\sqrt{x}}, \frac{1}{8} \left(3 - 30 x^2 + 35 x^4\right), 120 x - 160 x^3 + 32 x^5\right\} \]

As a result of these conversions, the solutions of certain ODEs can be partially expressed in terms of elementary functions. Hermite’s equation is one such ODE.
Here is the solution for Hermite’s equation with arbitrary $n$.

\[ \text{In[8]} := \text{DSolve}[y''[x] - 2 x * y'[x] + 2 n * y[x] == 0, y[x], x] \]

\[ \text{Out[8]} = \left\{ \left\{ y[x] \to C[1] \text{HermiteH}[n, x] + C[2] \text{Hypergeometric1F1}\left[-\frac{n}{2}, -\frac{1}{2}, x^2\right] \right\} \right\} \]

With $n$ set to 5, the solution is given in terms of polynomials, exponentials, and Erfi.

\[ \text{In[9]} := \text{Collect[Simplify[PowerExpand[} \]

\[ \frac{1}{8} \left( 960 x - 1280 x^3 + 256 x^5 \right) C[1] + \frac{1}{8} C[2] \left( 8 e^{x^2} - 18 e^{x^2} x^2 + 4 e^{x^2} x^3 + \sqrt{\pi} x (-15 + 20 x^2 - 4 x^4) \text{ Erfi}[x] \right) \]

\[ \text{Out[9]} = \frac{1}{8} \left( 960 x - 1280 x^3 + 256 x^5 \right) C[1] + \frac{1}{8} C[2] \left( 8 e^{x^2} - 18 e^{x^2} x^2 + 4 e^{x^2} x^3 + \sqrt{\pi} x (-15 + 20 x^2 - 4 x^4) \text{ Erfi}[x] \right) \]

**Linear Second-Order ODEs with Rational Coefficients**

The hypergeometric functions play a unifying role in mathematical analysis since many important functions, such as the Bessel functions and Legendre functions, are special cases of them. Each hypergeometric function is associated with a linear ODE having rational coefficients.

Here is the ODE for the Hypergeometric2F1 function.

\[ \text{In[1]} := \text{DSolve}[\left( x^2 - x \right) \ y''[x] + (((a + b + 1) - x - c) \ y'[x] + b + a \ y[x] == 0, y[x], x] \]

\[ \text{Out[1]} = \left\{ \left\{ y[x] \to C[1] \text{Hypergeometric2F1}[a, b, c, x] + \right. \right. \]

\[ (-1)^{1-c} x^{1-c} C[2] \text{Hypergeometric2F1}[1 + a - c, 1 + b - c, 2 - c, x] \}

\[ \text{Out[1]} = \left\{ \left\{ y[x] \to C[1] \text{Hypergeometric2F1}[a, b, c, x] + \right. \right. \]

\[ (-1)^{1-c} x^{1-c} C[2] \text{Hypergeometric2F1}[1 + a - c, 1 + b - c, 2 - c, x] \}

\[ \text{Out[1]} = \left\{ \left\{ y[x] \to C[1] \text{Hypergeometric2F1}[a, b, c, x] + \right. \right. \]

\[ (-1)^{1-c} x^{1-c} C[2] \text{Hypergeometric2F1}[1 + a - c, 1 + b - c, 2 - c, x] \}

\[ \text{Out[1]} = \left\{ \left\{ y[x] \to C[1] \text{Hypergeometric2F1}[a, b, c, x] + \right. \right. \]

\[ (-1)^{1-c} x^{1-c} C[2] \text{Hypergeometric2F1}[1 + a - c, 1 + b - c, 2 - c, x] \}

\[ \text{DSolve can solve a large class of second-order linear ODEs by reducing them to the ODEs for hypergeometric functions. The reduction involves coordinate transformations of both the independent and dependent variables.} \]

This equation is equivalent to the ODE for Hypergeometric2F1.

\[ \text{In[2]} := \text{sol} = \text{DSolve}[64 x^2 (x - 1)^2 y''[x] + 32 x (x - 1) (3 x - 1) y'[x] + (5 x - 21) y[x] == 0, y, x] \]

\[ \text{Out[2]} = \left\{ \left\{ y \to \text{Function}\left[\left\{ x \right\}, \frac{\frac{1}{e^{\frac{1}{4} (-2 \text{Log}[1-x] + \text{Log}[x])}} C[1] \text{Hypergeometric2F1}\left[-\frac{7}{8}, -\frac{3}{8}, -\frac{1}{4}, x\right]}{x^{1/8}} \right. \right. \]

\[ (-1)^{1/4} e^{\frac{1}{4} (-2 \text{Log}[1-x] + \text{Log}[x])} x^{9/8} C[2] \text{Hypergeometric2F1}[\frac{3}{8}, 7 \frac{9}{8}, 4 \frac{x}{4}] \}} \]

This verifies the solution using numerical values.

\[ \text{In[3]} := \text{64 x^2 (x - 1)^2 y''[x] + 32 x (x - 1) (3 x - 1) y'[x] + (5 x - 21) y[x] /. sol[[1]] /.}\]

\[ \{x \to \text{RandomComplex[]}\} // \text{Simplify} // \text{Chop} \]

\[ \text{Out[3]} = 0 \]
Solutions to this equation are returned in terms of \( \text{HypergeometricU} \) (the confluent hypergeometric function) and \( \text{LaguerreL} \). This example appears on (equation 2.16, page 403 of [K59]).

\[
\text{In}[4]:= \text{sol} = \text{Simplify} \left[ \text{PowerExpand} \left[ \text{y''\[y[x]\]} + (a x^2 + b x^2 (c - 1)) y[x] == 0, y, x] \right] \right]
\]

\[
\text{Out}[4]= 2 c^2 + 2 c - a x + c - i + c M 2 C[1] \text{HypergeometricU} \left[ \frac{1 b}{\sqrt{a} - c}, \frac{c}{2 + c}, \frac{2 \sqrt{a} x^{1 + c}}{1 + c, \sqrt{-(1 + c)^2}} \right] + C[2] \text{LaguerreL} \left[ \frac{1 b}{\sqrt{a} - c}, - \frac{1}{1 + c}, \frac{2 \sqrt{a} x^{1 + c}}{2 + c, \sqrt{-(1 + c)^2}} \right]
\]

The ODEs for special functions have been studied since the eighteenth century. During the last thirty years, powerful algorithms have been developed for systematically solving ODEs with rational coefficients. An important algorithm of this type is Kovacic’s algorithm, a decision procedure that either generates a solution for the given ODE in terms of Liouvillian functions or proves that the given ODE does not have a Liouvillian solution.

This equation is solved using Kovacic’s algorithm.

\[
\text{In}[5]:= \text{DSolve} \left[ x y''[x] + (10 x^3 - 1) y'[x] + 5 x^2 (5 x^3 + 1) y[x] == 0, y[x], x] \right]
\]

\[
\text{Out}[5]= \left\{ \left\{ y[x] \rightarrow e^{\frac{\sqrt{5} x^2}{2}} C[1] + \frac{1}{2} e^{\frac{\sqrt{5} x^2}{2}} x^2 C[2] \right\} \right\}
\]

The solution returned from Kovacic’s algorithm may occasionally include functions such as \( \text{ExpIntegralEi} \) or an unevaluated integral of elementary functions because, while it is easy to find a second solution for a second-order linear ODE once one solution is known, the integral involved in finding the second solution may be hard to evaluate explicitly.

The solution to this equation is obtained using Kovacic’s algorithm. It includes \( \text{ExpIntegralEi} \).

\[
\text{In}[6]:= \text{DSolve} \left[ 4 x y''[x] + (7 x + 12) y'[x] + 21 y[x] == 0, y[x], x] \right]
\]

\[
\text{Out}[6]= \left\{ \left\{ y[x] \rightarrow e^{-7 x/4} C[2] \left[ 16 e^{7 x/4} + 28 e^{7 x/4} x - 49 x^2 \text{ExpIntegralEi} \left[ \frac{7 x}{4} \right] \right] \right\} \right\}
\]

In general, the solutions for linear ODEs with rational coefficients and order greater than one can be given in terms of \text{DifferentialRoot} objects. This is similar to the representation for solutions of polynomial equations in terms of \text{Root}.
The solution to this equation is given in terms of DifferentialRoot.

\[ \text{In}[7]:= \text{sol} = \text{DSolve}[y''[x] - x^2 y'[x] - y[x] - 1 == 0 \&\& y[0] == 0 \&\& y'[0] == -1, y, x] \]

\[ \text{Out}[7]= \{\{y \rightarrow \text{Function}[x, \text{DifferentialRoot}[\text{Function}[\{y, x\}, \{-1 - y[x] - x^2 y'[x] + y'[x] == 0, y[0] == 0, y'[0] == -1\}]][x]]\} \]

The solution may be evaluated and plotted in the usual way.

\[ \text{In}[8]:= \text{Table}[y[x] /. \text{sol}[[1]], \{x, -1, 1, 0.4\}] \]

\[ \text{Out}[8]= \{1.57269, 0.806671, 0.221252, -0.181388, -0.459045, -0.685918\} \]

\[ \text{In}[9]:= \text{Plot}[y[x] /. \text{sol}[[1]], \{x, -1, 1\}] \]

\[ \text{Out}[9]= \]

**Equations with Non-Rational Coefficients**

The ODEs that arise in practical applications often have non-rational coefficients. In such cases, \texttt{DSolve} attempts to convert the equation into one with rational coefficients using a suitable coordinate transformation.

Here is an equation that has \texttt{Exp} \[ x \] as a coefficient. It is solved by transforming it to Bessel’s equation.

\[ \text{In}[1]:= \text{DSolve}[y''[x] - \text{Exp}[5 x] \ast y[x] == 0, y, x] \]

\[ \text{Out}[1]= \{\{y \rightarrow \text{Function}[x, \frac{2 \sqrt{e^{5 x}}}{5} C[1] + 2 \text{BesselK}\left[0, \frac{2 \sqrt{e^{5 x}}}{5}\right] C[2]]\}\} \]

This equation (equation 2.437, page 507 of \texttt{[K59]}) has trigonometric coefficients. The solution is given in terms of elementary functions.

\[ \text{In}[2]:= \text{DSolve}[y''[x] \ast \text{Sin}[x] \ast \text{Cos}[x] \ast 2 - y'[x] \ast (3 \ast \text{Sin}[x] \ast 2 + 1) \ast \text{Cos}[x] - y[x] \ast \text{Sin}[x] \ast 3 == 0, y, x] \]

\[ \text{Out}[2]= \{\{y \rightarrow \text{Function}[x, C[2] \text{Cos}[x] \frac{\sqrt{13}}{2} + C[1] \text{Cos}[x] \frac{3 \sqrt{13}}{2}]\}\} \]
Here is an equation with a hyperbolic function in the coefficient of \(y'[x]\). The solution is given in terms of Legendre functions.

\[
\text{In}[3] := \text{DSolve}[y''[x] + (k^2 + 2 \cdot \text{Sech}[x]^2) \ y[x] == 0, \ y[x], \ x]
\]

\[
\text{Out}[3] = \{\{y[x] \to C[1] \text{LegendreP}[1, i \ k, \ \text{Tanh}[x]] + C[2] \text{LegendreQ}[1, i \ k, \ \text{Tanh}[x]]\}\}
\]

The solution to this equation is given in terms of \(\text{HypergeometricU}\) and \(\text{LaguerreL}\).

\[
\text{In}[4] := \text{expr} = y''[x] + (-d + d (1 - \text{Exp}[-b \ x])^2) \ y[x] - l \ y[x];
\]

\[
\text{In}[5] := \text{sol} = \text{DSolve}[\text{expr} == 0, \ y, \ x]
\]

\[
\text{Out}[5] = \{\{y \to \text{Function}\{x, \frac{e^{\frac{1}{2} \sqrt{d + \sqrt{1 - \text{Log}[x]^2}}}}{x} \ \text{HypergeometricU}\left[-\frac{b + 2 i \sqrt{d} - 2 \sqrt{1 - \text{Log}[x]^2}}{2 b}, 1 + \frac{2 \sqrt{1 - \text{Log}[x]^2}}{b}, \frac{2 i \sqrt{d} e^{-b x}}{b} \right] + \frac{e^{\frac{1}{2} \sqrt{d + \sqrt{1 - \text{Log}[x]^2}}}}{x} \ \text{LaguerreL}\left[-\frac{b + 2 i \sqrt{d} - 2 \sqrt{1 - \text{Log}[x]^2}}{2 b}, 1 + \frac{2 \sqrt{1 - \text{Log}[x]^2}}{b}, \frac{2 i \sqrt{d} e^{-b x}}{b} \right]\}\}\}
\]

This verifies the solution using random values of \(x, b, d,\) and \(l\).

\[
\text{In}[6] := \text{expr} /. \text{sol}[[1]] /. \{x \to \text{RandomComplex[]}\}, \ b \to \text{RandomComplex[]}, \ d \to \text{RandomComplex[]}, \ l \to \text{RandomComplex[]}\} /\!\!/\text{Simplify} /\!\!/\text{Chop}
\]

\[
\text{Out}[6] = 0
\]

**Inhomogeneous Linear Second-Order Equations**

If the given second-order ODE is inhomogeneous, \text{DSolve} applies the method of variation of parameters to return a solution for the problem.

This solves an inhomogeneous linear second-order ODE. The solution is composed of two parts: the first part is the general solution to the homogeneous equation, and the second part is a particular solution to the inhomogeneous equation.

\[
\text{In}[1] := \text{sol} = \text{DSolve}[x^2 \ y''[x] + y[x] == x^2, \ y[x], \ x]
\]

\[
\text{Out}[1] = \{\{y[x] \to \sqrt{x} \ C[1] \ \text{Cos}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right] + \sqrt{x} \ C[2] \ \text{Sin}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right] + \frac{1}{3} \left(x^2 \ \text{Cos}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right]^2 + x^2 \ \text{Sin}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right]^2\right)\}\}\}
\]

This solves the homogeneous equation, which is an Euler equation.

\[
\text{In}[2] := \text{DSolve}[x^2 \ y''[x] + y[x] == 0, \ y[x], \ x]
\]

\[
\text{Out}[2] = \{\{y[x] \to \sqrt{x} \ C[1] \ \text{Cos}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right] + \sqrt{x} \ C[2] \ \text{Sin}\left[\frac{1}{2} \sqrt{3} \ \text{Log}[x]\right]\}\}\}
\]
Different particular solutions can be obtained by varying the constants $C[1]$ and $C[2]$ in the solution.

\[
\text{In[3]:=} \quad \text{particularsolutions} = \text{Flatten[Table[y[x] /. \text{sol} /. \{C[1] \to i, C[2] \to j\}, \{i, 0, 5, 3\}, \{j, 1, 3\}]]}
\]

\[
\text{Out[3]}=\{\sqrt{x} \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right], \\
2 \sqrt{x} \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right], \\
3 \sqrt{x} \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right], \\
3 \sqrt{x} \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right], \\
3 \sqrt{x} \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right], \\
3 \sqrt{x} \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right] + \frac{1}{3} \left(x^2 \cos \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2 + x^2 \sin \left[\frac{1}{2} \sqrt{3} \log |x| \right]^2\right)
\}
\]

\[
\text{In[4]:=} \quad \text{Plot[Evaluate[particularsolutions], \{x, 0.1, 2\} ]}
\]

\[
\text{Out[4]}=\]

**Nonlinear Second-Order ODEs**

The general form of a nonlinear second-order ODE is

\[
F(x, y, y'(x), y''(x)) = 0.
\]

For simplicity, assume that the equation can be solved for the highest-order derivative $y''(x)$ to give

\[
y''(x) = f(x, y(x), y'(x)).
\]
There are a few classes of nonlinear second-order ODEs for which solutions can be easily found. The first class consists of equations that do not explicitly depend on $y(x)$; that is, equations of the form $y''(x) = f(x, y'(x))$. Such equations can be regarded as first-order ODEs in $u(x) = y'(x)$.

Here is an example of this type.

```
In[1]:= eqn = y''[x] == 5 x y'[x] + y'[x]^2; sol = DSolve[eqn, y, x]
```

```
```

As in the case of linear second-order ODEs, the solution depends on two arbitrary parameters $C[1]$ and $C[2]$.

Here is a plot of the solution for a specific choice of parameters.

```
In[2]:= Plot[ Evaluate[y[x] /. sol /. {C[1] -> -1/2, C[2] -> -1/8}], {x, -0.4, 1}]
```

```
Out[2]=
```

This verifies the solution.

```
In[3]:= eqn /. sol // Simplify
```

```
Out[3]= {True}
```

The second class of easily solvable nonlinear second-order equations consists of equations that do not depend explicitly on $x$ or $y'(x)$; that is, equations of the form $y''(x) = f(y(x))$. These equations can be reduced to first-order ODEs with independent variable $y$. Inverse functions are needed to give the final solution for $y(x)$. 


Here is an example of this type.

\[ In[4] := \text{DSolve}[y''[x] == \text{Exp}[3 \cdot y[x]], y[x], x] \]

\[ Solve::ifun : \text{Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.} \]

\[ Out[4] := \left\{ \left\{ y[x] \to \log \left[ -\frac{3}{2} \sqrt{\frac{1}{3}} \left( -C[1] + C[1] \text{Tanh} \left( \frac{3}{2} \sqrt{\frac{1}{3}} \right) \right) \right] \right\}, \left\{ y[x] \to \log \left[ -\frac{3}{2} \sqrt{\frac{1}{3}} \left( -C[1] + C[1] \text{Tanh} \left( \frac{3}{2} \sqrt{\frac{1}{3}} \right) \right) \right] \right\}, \left\{ y[x] \to \log \left[ -\frac{3}{2} \sqrt{\frac{1}{3}} \left( -C[1] + C[1] \text{Tanh} \left( \frac{3}{2} \sqrt{\frac{1}{3}} \right) \right) \right] \right\} \right\} \]

The third class consists of equations that do not depend explicitly on \( x \); that is, equations of the form \( y''(x) = f(y(x), y'(x)) \). Once again, these equations can be reduced to first-order ODEs with independent variable \( y \).

This example is based on (equation 6.40, page 550 of [K59]). The underlying first-order ODE is an Abel equation. The hyperbolic functions in the solution result from the automatic simplification of Bessel functions.

\[ In[5] := \text{sol} = \text{Solve} \left[ y'[x] \cdot \text{DSolve}[y''[x] == 3 \cdot y[x] \cdot y'[x] + (3 \cdot y[x] \cdot 2 + 4 \cdot y[x] + 1), y[x], x][[1]] \right] // \text{Simplify} \]

\[ Out[5] := \left\{ \text{Log} \left[ \begin{array}{c} -\frac{3}{2} \sqrt{\frac{1}{3}} \left( 6 \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \right) + 3 \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \right] + 6 \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} + 3 \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \cdot \sqrt{\frac{1}{3}} \end{array} \right] \right\} \]

The fourth class consists of equations that are homogeneous in some or all of the variables \( x, y(x), \) and \( y''(x) \). There are several possibilities in this case, but here only the following simple example is considered.

In this equation, each term has a total degree of 2 in the variables \( y[x], y'[x], \) and \( y''[x] \). This equation can be solved by transforming it to a first-order ODE.

\[ In[6] := \text{DSolve}[7 \cdot y[x] \cdot y'[x] - 11 \cdot y'[x] \cdot 2 ^ 2 = 0, y[x], x] \]

\[ Out[6] := \left\{ \left\{ y[x] \to \frac{C[2]}{\left(4 \cdot x + 7 \cdot C[1]\right)^{1/4}} \right\} \right\} \]
The fifth and final class of easily solvable nonlinear second-order ODEs consists of equations that are exact or can be made to be exact using an integrating factor.

Here is an example of this type, based on (equation 6.51, page 554 of [K59]).

\[
\text{In[7]} := \text{eqn} = y'' @ x D + y @ x D * y' @ x D * 2 - x^2 * y' @ x D = 0;
\]

\[
\text{In[8]} := \text{sol} = \text{DSolve}[\text{eqn}, y[x], x]
\]

\[
\text{Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.} \]

\[
\text{Out[8]} = \left\{ y[x] \rightarrow -i \sqrt{2} \text{InverseErf}\left( \frac{i \left( 3 \sqrt{2} \pi C[2] - \frac{3^{1/3} \sqrt{2} \pi (-x)^{1/3} C[1] \text{Gamma}\left[ \frac{1}{3}, -x^{3/2} \right]}{x^{3/2}} \right)}{3 \pi} \right) \right\}
\]

It is important to note that the solutions to fairly simple-looking nonlinear ODEs can be complicated. Verifying and applying the solutions in such cases is a difficult problem.

**Higher-Order ODEs**

**Overview of Higher-Order ODEs**

The general form of an ODE with order \( n \) is

\[
F(x, y(x), y'(x), y''(x), \ldots, y^{(n)}(x)) = 0.
\]

As in the case of second-order ODEs, such an ODE can be classified as linear or nonlinear. The general form of a linear ODE of order \( n \) is

\[
a_0(x) y^{(n)}(x) + a_1(x) y^{(n-1)}(x) + \ldots + a_n(x) y(x) = b(x).
\]

If \( b(x) \) is the zero function, the equation is said to be \textit{homogeneous}. This discussion is primarily restricted to that case.

Many methods for solving linear second-order ODEs can be generalized to linear ODEs of order \( n \), where \( n \) is greater than 2. If the order of the ODE is not important, it is simply called a linear ODE.
Linear Higher-Order Equations with Constant Coefficients

A linear ODE with *constant coefficients* can be easily solved once the roots of the auxiliary equation (or characteristic equation) are known. Some examples of this type follow.

The characteristic equation of this ODE has real and distinct roots: 4, 1, and 7. Hence the solution is composed entirely of exponential functions.

\[
\text{Out[1]} = \text{DSolve}[y''''[x] - 4 \times y'''[x] - 25 \times y''[x] + 28 \times y'[x] = 0, y[x], x]
\]

\[
\text{Out[1]} = \{\{y[x] \to e^{-4x} C[1] + e^{x} C[2] + e^{7x} C[3]\}\}
\]

The characteristic equation of this ODE has two pairs of equal roots: -3 and -5. The repeated roots give rise to the *basis* of the solutions, \(\{e^{3x}, xe^{3x}, e^{5x}, xe^{5x}\}\).

\[
\text{Out[2]} = \text{DSolve}[y''''[x] - 16 \times y'''[x] + 94 \times y''[x] - 240 \times y'[x] + 225 \times y[x] = 0, y[x], x]
\]

\[
\text{Out[2]} = \{\{y[x] \to e^{3x} C[1] + e^{3x} x C[2] + e^{5x} C[3] + e^{5x} x C[4]\}\}
\]

The characteristic equation of this ODE has two pairs of roots with nonzero imaginary parts: 3 + 4i, 3 - 4i, 2 + i, and 2 - i. Hence the solution basis can be expressed with trigonometric and exponential functions.

\[
\text{Out[3]} = \text{DSolve}[y''''[x] - 10 \times y'''[x] + 54 \times y''[x] - 130 \times y'[x] + 125 \times y[x] = 0, y[x], x]
\]

\[
\]

Finally, here is an example that combines all the previous kinds of solutions.

\[
\text{Out[4]} = \text{DSolve}[y''''''[x] - 17 \times y''''[x] + 108 \times y'''[x] - 330 \times y''[x] + 488 \times y'[x] - 280 \times y[x] = 0, y[x], x]
\]

\[
\]

Higher-Order Euler and Legendre Equations

An *Euler equation* is an ODE of the form

\[
x^n y^{(n)}(x) + a_1 x^{n-1} y^{(n-1)}(x) + a_2 x^{n-2} y^{(n-2)}(x) + \ldots + a_n y(x) = 0.
\]

The following is an example of an Euler equation.

\[
\text{In[1]} = \text{DSolve}[x^4 \times y''''[x] - 2 \times x^3 \times y'''[x] - x^2 \times y''[x] + 5 \times x \times y'[x] + y[x] = 0, y[x], x]
\]

\[
\text{Out[1]} = \{\{y[x] \to x^{Root[-1.4 \times 1.16 \times 11.8 \times 11.8 \times 11.8, 2]} C[1] + x^{Root[-1.4 \times 1.16 \times 11.8 \times 11.8, 2]} C[2] +
\]
\[
x^{Root[-1.4 \times 1.16 \times 11.8 \times 11.8, 3]} C[3] + x^{Root[-1.4 \times 1.16 \times 11.8 \times 11.8, 4]} C[4]\}\}
\]
The Legendre linear equation is a generalization of the Euler equation. It has the form

\[(c x + d)^n y^{(n)}(x) + a_1 (c x + d)^{n-1} y^{(n-1)}(x) + a_2 (c x + d)^{n-2} y^{(n-2)}(x) \ldots + a_n y(x) = 0.\]

This is a Legendre linear equation.

\[
\text{Out}[2]= \left\{ \left\{ y(x) \to (5 + 3 x) \text{Root}\left[1.570 \pm 1.1044 \pm 1.540 \pm 1.81 \pm 1.14, 1\right] C[1] + (5 + 3 x) \text{Root}\left[1.570 \pm 1.1044 \pm 1.540 \pm 1.81 \pm 1.14, 2\right] C[2] + (5 + 3 x) \text{Root}\left[1.570 \pm 1.1044 \pm 1.540 \pm 1.81 \pm 1.14, 3\right] C[3] + (5 + 3 x) \text{Root}\left[1.570 \pm 1.1044 \pm 1.540 \pm 1.81 \pm 1.14, 4\right] C[4] \right\} \right\}
\]

**Exact Higher-Order Equations**

A linear ordinary differential equation of order \(n\)

\[a_0(x) y^{(n)}(x) + a_1(x) y^{(n-1)}(x) + \ldots + a_{n-1}(x) y'(x) + a_n(x) y(x) = 0\]

is said to be exact if

\[(-1)^n a_0^{(n)}(x) + (-1)^{(n-1)} a_1^{(n-1)}(x) + \ldots - a_{n-1}'(x) + a_n(x) = 0.\]

The condition of exactness can be used to reduce the problem to that of solving an equation of order \(n-1\).

This is an example of an exact ODE.

\[
\text{Out}[5]= 5 y[x] + 5 x y'[x] - y''[x] + y^{(3)}[x]
\]

This verifies the condition for exactness.

\[
\text{Out}[6]= 0
\]
This solves the equation.

\[ \text{In[7]} := \text{sol} = \text{DSolve[ExactODE == 0, y, x]} \]

\[ \text{Out[7]} = \left\{ \begin{array}{l}
y \to \text{Function}\{x\}, \ e^{x/2} \text{AiryAi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 x\right)}{5^{2/3}}\right] C[2] + e^{x/2} \text{AiryBi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 x\right)}{5^{2/3}}\right] C[3] + \\
\left( e^{x/2} \text{AiryBi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 x\right)}{5^{2/3}}\right] \right)^{3/2} \int_{1}^{x} \left(-1\right)^{2/3} e^{\frac{k[1]}{2}} \text{AiryAi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 K[2]\right)}{5^{2/3}}\right] C[1] \right) dK[2] + \\
\left( e^{x/2} \text{AiryBi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 x\right)}{5^{2/3}}\right] \right)^{3/2} \int_{1}^{x} \left(-1\right)^{2/3} e^{\frac{k[1]}{2}} \text{AiryBi}\left[\frac{(-1)^{1/3} \left(\frac{1}{4} - 5 K[1]\right)}{5^{2/3}}\right] C[1] \right) dK[1] \end{array} \right\} \]

This verifies the solution.

\[ \text{In[8]} := \text{ExactODE} /. \text{sol[1]} /. \{x \to \text{RandomReal[]}, C[1] \to \text{RandomReal[]}, C[2] \to \text{RandomReal[]}\} /\! / \text{N} /\! / \text{Simplify} /\! / \text{Chop} \]

\[ \text{Out[8]} = 0 \]

\[ \text{In[9]} := \text{Clear[a0, a1, a2, a3]} \]

### Further Examples of Exactly Solvable Higher-Order Equations

The solutions to many second-order ODEs can be expressed in terms of special functions. Solutions to certain higher-order ODEs can also be expressed using \text{AiryAi}, \text{BesselJ}, and other special functions.

The solution to this third-order ODE is given by products of Airy functions.

\[ \text{In[1]} := \text{sol1 = DSolve[y'''}[x] - 4 \ast (x + 2) \ast y''[x] - 2 \ast y[x] == 0, y, x} \]

\[ \text{Out[1]} = \left\{ \begin{array}{l}
y \to \text{Function}\{x\}, \text{AiryAi}[2 + x]^2 C[1] + \text{AiryAi}[2 + x] \text{AiryBi}[2 + x] C[2] + \text{AiryBi}[2 + x]^2 C[3] \end{array} \right\} \]

The solution to this third-order ODE is given by Bessel functions.

\[ \text{In[2]} := \text{sol2 = DSolve} \left[ x^3 \ast y'''[x] + 3 \ast x^2 \ast y''[x] + (4 \ast x^3 - 11 \ast x) \ast y'[x] + 4 \ast x^2 \ast y[x] == 0, y, x \right] \]

\[ \text{Out[2]} = \left\{ \begin{array}{l}
y \to \text{Function}\{x\}, \\
\text{BesselJ}\left[\sqrt{3}, x\right]^2 C[1] + \text{BesselJ}\left[\sqrt{3}, x\right] \text{BesselY}\left[\sqrt{3}, x\right] C[2] + \text{BesselY}\left[\sqrt{3}, x\right]^2 C[3] \end{array} \right\} \]
This plot shows the oscillatory behavior of the solutions on different parts of the real line.

\[\text{Show}[[\text{Plot}[y[x] \text{ /. sol1 }[[1], {x, -10, 1}]], \text{PlotStyle} \rightarrow \{\text{Red}\}]] + \text{Plot}[[y[x] \text{ /. sol2 }[[1], {x, 2, 10}], \text{PlotStyle} \rightarrow \{\text{Dashing}[\{0.02\}], \text{Blue}\}]]\]

The solution to this fourth-order linear ODE is expressed in terms of HypergeometricPFQ.

\[\text{In}[4]:= \text{HypergeometricPFQTypeEquation} = 30 \times x^3 \times y^{'''}[x] + 193 \times x^2 \times y^{''}[x] + (219 \times x - 30 \times x^2) \times y^{'}[x] + (21 - 90 \times x) \times y'[x] - 30 \times y[x];\]

\[\text{In}[5]:= \text{sol3} = \text{DSolve}[\text{HypergeometricPFQTypeEquation} \equiv 0, y, x]\]

\[\text{Out}[5]= \left\{\left[y \rightarrow \text{Function}\left[\{x\}, \frac{(-1)^{2/3} \times C[1] \times \text{HypergeometricPFQ}\left[\left.\frac{2}{3}, \frac{1}{3}, \frac{5}{6}, \frac{11}{15}\right], x\right]}{x^{4/3}} + \frac{(-1)^{2/5} \times x^{2/5} \times C[3] \times \text{HypergeometricPFQ}\left[\left.\frac{7}{5}, \frac{9}{10}, \frac{41}{15}\right], x\right]}{i \times \sqrt{x} \times C[4]} + \text{HypergeometricPFQ}\left[\left.\frac{3}{2}, \frac{11}{10}, \frac{17}{6}\right], x\right] + C[2] \times \text{HypergeometricPFQ}\left[\left.1, 1, \frac{1}{2}, \frac{3}{5}, \frac{7}{3}\right], x\right]\right]\right\}\]

This verifies that the solution is correct using numerical values.

\[\text{In}[6]:= \text{HypergeometricPFQTypeEquation} \equiv . \text{sol3}[[1]] \equiv . \{x \rightarrow \text{RandomComplex}[]\} // \text{Simplify} // \text{Chop}\]

\[\text{Out}[6]= 0\]

As for second-order linear ODEs, there are modern algorithms for solving higher-order ODEs with rational coefficients. These algorithms give "rational-exponential" solutions, which are combinations of rational functions and exponentials of the integrals of rational functions. These algorithms are combined with techniques such as reduction of order to produce a complete solution for the given ODE.
The general solution to this equation has a rational term and terms that depend on Airy functions. The Airy functions come from reducing the order of the equation to 2.

\[
\text{DSolve}[(6 - 24 \times x + 4 \times x^2 - 10 \times x^3 + 9 \times x^4 - 4 \times x^5 + x^6) \times y[x]) / ((-1 + x)^4 \times x^2 + ((-8 + 30 \times x - 10 \times x^2 + 10 \times x^3 - 3 \times x^4 - 2 \times x^5 + x^6) \times \text{Derivative}[1][y][x]) / ((-1 + x)^3 \times x^2) + ((4 - 16 \times x + 5 \times x^2 - 2 \times x^3) \times \text{Derivative}[2][y][x]) / ((-1 + x)^2 \times x^2) + ((4 - x^2) \times \text{Derivative}[3][y][x]) / ((-1 + x) \times x) \text{ == 0, y, x}
\]

\[\text{Out}[7]= \{\{y \to \text{Function}[\{x\}, \frac{(-1 + x) \times C[1]}{-4 + x^2} + \frac{(-1 + x) \times \text{AiryAi}[x] \times C[2]}{-4 + x^2} + \frac{(-1 + x) \times \text{AiryBi}[x] \times C[3]}{-4 + x^2}\}\}\]

The equations considered so far have been homogeneous; that is, with no term free of \(y(x)\) or its derivatives. If the given ODE is inhomogeneous, \text{DSolve} applies the method of variation of parameters to obtain the solution.

Here is an example of this type. The exponential terms in the solution come from the general solution to the homogeneous equation, and the remaining term is a particular solution (or particular integral) to the problem.

\[
\text{sol = DSolve}[y'''[x] - 13 \times y''[x] + 19 \times y'[x] + 33 \times y[x] \text{ == Cos}[2 \times x], y[x], x]
\]

\[\text{Out}[8]= \{\{y[x] \to e^{-x} \times C[1] + e^{3 \times x} \times C[2] + e^{11 \times x} \times C[3] + \frac{17 \times \text{Cos}[2 \times x] + 6 \times \text{Sin}[2 \times x]}{1625}\}\}\]

This is the general solution to the homogeneous equation.

\[
\text{generalsolution = y[x] /. DSolve}[y'''[x] - 13 \times y''[x] + 19 \times y'[x] + 33 \times y[x] \text{ == 0, y[x], x}[[1]]]
\]

\[\text{Out}[9]= e^{-x} \times C[1] + e^{3 \times x} \times C[2] + e^{11 \times x} \times C[3]\]

This particular solution is part of the general solution to the inhomogeneous equation.

\[
\text{particularsolution} = (y[x] \text{ /. sol[[1]]}) - \text{generalsolution}
\]

\[\text{Out}[10]= \frac{17 \times \text{Cos}[2 \times x] + 6 \times \text{Sin}[2 \times x]}{1625}\]

Thus, the general solution for the inhomogeneous equation is the sum of the general solution to the homogeneous equation and a particular integral of the ODE.

The solution methods for nonlinear ODEs of higher order rely to a great extent on reducing the problem to one of lower order.
Here is a nonlinear third-order ODE with no explicit dependence on \( x \) or \( y[x] \). It is solved by reducing the order to 2 using a simple integration.

\[
\text{In[11]} := \text{DSolve}\left[7 \cdot y'[x] \cdot y''[x] - 11 \cdot y'[x] \cdot 2 = 0, y[x], x\right]
\]

\[
\text{Out[11]} = \left\{\left[y[x] \rightarrow -\frac{C[2]}{3 \left(4x + 7 C[1]\right)^{3/4}} + C[3]\right]\right\}
\]

**Systems of ODEs**

**Introduction to Systems of ODEs**

Systems of ODEs are important in various fields of science, such as the study of electricity and population biology. Like single ODEs, systems of ODEs can be classified as **linear** or **nonlinear**.

A system of linear first-order ODEs can be represented in the form

\[
X'(t) = A(t) X(t) + B(t).
\]

Here \( X(t) \) is a vector of unknown functions, \( A(t) \) is the matrix of the coefficients of the unknown functions, and \( B(t) \) is a vector representing the inhomogeneous part of the system.

In the two-dimensional case, the system can be written more concretely as

\[
x'(t) = p(t) x(t) + q(t) y(t) + u(t)
\]

\[
y'(t) = r(t) x(t) + s(t) y(t) + v(t).
\]

If all the entries of the matrix \( A(t) \) are constants, then the system is said to be **linear with constant coefficients**. If \( B(t) \) is the zero vector, then the system is said to be **homogeneous**.

The important global features of the solutions to linear systems can be clarified by considering homogeneous systems of ODEs with constant coefficients.

**Linear Systems of ODEs**

Here is a system of two ODEs whose coefficient matrix has real and distinct eigenvalues.

\[
\text{In[1]} := A = \{(4, \ -6), \ \{1, \ -1\}\};
\]

\[
\text{In[2]} := \text{Eigenvalues}[A]
\]

\[
\text{Out[2]} = \{2, 1\}
\]
This solves the system. Note that the general solution depends on two arbitrary constants $C_1$ and $C_2$.

This solves the system. Note that the general solution depends on two arbitrary constants $C_1$ and $C_2$.

Here is a plot of some particular solutions obtained by giving specific values to $C_1$ and $C_2$. In this case, the origin is a called a node.

In this system the eigenvalues of the coefficient matrix are complex conjugates of each other.
This solves the system.

```
In[12]:= sol = DSolve[system, {x, y}, t]
```

```
Out[12]= 
y -> Function[{t}, e^t C[2] (Cos[2 t] - 3 Sin[2 t]) + \frac{5}{2} e^t C[1] Sin[2 t]]}}
```

This plots the solution for various values of the arbitrary parameters. The *spiraling* behavior is typical for systems with complex eigenvalues.

```
In[13]:= particularsols = Partition[Flatten[Table[{x[t], y[t]} /. sol /. {C[1] -> 1/i, C[2] -> 1/j}, {i, -10, 10, 8}, {j, -10, 10, 8}]], 2];
```

```
In[14]:= ParametricPlot[Evaluate[particularsols], {t, -35, 35}, PlotRange -> All, PlotPoints -> 70, Method -> {Compiled -> False}]
```

Solving homogeneous systems of ODEs with constant coefficients and of arbitrary order is a straightforward matter. They are solved by converting them to a system of first-order ODEs.

This solves a homogeneous system of ODEs of order 3, with constant coefficients.

```
In[15]:= system = {x'''[t] + y[t], y'''[t] - 64 x[t]};
```
This verifies the solution.


Out[17]= {0, 0}

In general, systems of linear ODEs with non-constant coefficients can only be solved in cases where the coefficient matrix has a simple structure, as illustrated in the following examples.

This first-order system has a diagonal coefficient matrix. The system is uncoupled because the first equation involves only \( x[t] \) and the second equation depends only on \( y[t] \). Thus, each equation in the system can be integrated independently of the other.

In[18]:= DSolve[\{x'[t] == Sin[t] \* x[t], y'[t] == t^2 \* y[t]\}, \{x, y\}, t]

Out[18]= \{\{x -> Function[{t}, \* Cos[t] C[1]], y -> Function[{t}, t^2 C[2]]\}\}

The rows of the coefficient matrix for this system form an orthogonal set of vectors.

In[19]:= A = {{E^t, Tan[t]}, {-Tan[t], E^t}};

In[20]:= A.Transpose[A] / Det[A]

Out[20]= \{\{1, 0\}, \{0, 1\}\}
As for single ODEs, there are sophisticated modern algorithms for solving systems of ODEs with rational coefficients.

In[21]:= \[ In[22]:= x[t_] = {x[t], y[t]}; \\
Out[22]= \{x'[t] = e^t x[t] + Tan[t] y[t], y'[t] = -Tan[t] x[t] + e^t y[t] \}

In[23]:= sol = DSolve[system, \{x, y\}, t] \\

In[24]:= system /. sol[[1]] // Simplify \\
Out[24]= \{True, True\}

Here is a system of three first-order ODEs. The coefficient matrix is upper triangular.

In[25]:= A = {{e^t, 2, 3}, \{0, 2, -1\}, \{0, 0, 1\}}; \\
In[26]:= MatrixForm[A] \\
Out[26]= \begin{pmatrix} e^t & 2 & 3 \\ 0 & 2 & -1 \\ 0 & 0 & 1 \end{pmatrix}

In[27]:= x[t_] = {x[t], y[t], z[t]}; \\
In[28]:= system = MapThread[H[1] \[H[2] &, \{x'[t], A x[t]\}] \\
Out[28]= \{x'[t] = e^t x[t] + 2 y[t] + 3 z[t], y'[t] = 2 y[t] - z[t], z'[t] = z[t] \}

In[29]:= sol = DSolve[system, \{x, y, z\}, t] \\

As for single ODEs, there are sophisticated modern algorithms for solving systems of ODEs with rational coefficients.

This solves a system of two first-order ODEs with rational coefficients. Note that the solution is composed entirely of rational functions.

In[30]:= \[ D[Function[x, (5 + x) w[x] + (6 + 2 x - 3 x^3 - x^5) y[x]] - (3 - 2 x + x^2) (1 + x^3) + x (3 - 2 x + x^2) (-1 + x^3), y[x] = \[ 1 + 20 x^2 + 3 x^3 \] w[x] - 4 x (3 - 2 x + x^2) y[x] \] \\
Out[30]= \{y \rightarrow Function[x, \frac{x C[1]}{-3 - 2 x + x^2} + \frac{x^2 C[2]}{-3 - 2 x + x^2}], w \rightarrow Function[x, \frac{C[1]}{5 + x} + \frac{x^2 C[2]}{5 + x}]\}
In the following example, the algorithm finds one rational solution for \(x[t]\) and \(y[t]\). (The equation for \(z[t]\) is uncoupled from the rest of the system.) Using the rational solution, DSolve is able to find the remaining exponential solution for \(x[t]\) and \(y[t]\).

\[
\begin{align*}
\text{In[31]:=} & \quad \text{system} = \{x'[t] &= (2 \times ((-9 + t + t^2) \times x[t] + (-6 + t^2) \times y[t])) / ((-2 + t) \times t), \\
& \quad y'[t] = (3 \times ((-12 + 2 \times t + t^2) \times x[t] + (24 - 2 \times t - 3 \times t^2) \times y[t])) / ((-2 + t) \times t), \\
& \quad z'[t] = t \times z[t])
\end{align*}
\]

\[
\begin{align*}
\text{In[32]:=} & \quad \text{sol} = \text{DSolve}[\text{system}, \{x, y, z\}, t] \\
\text{Out[32]=} & \quad \begin{cases}
\{x \rightarrow \text{Function}\{t\}, \frac{(2 - t) \times C[1]}{t^3} - 2 \times e^t \times C[2]\}, \\
\{y \rightarrow \text{Function}\{t\}, \frac{(4 + t) \times C[1]}{t^3} + 3 \times e^t \times C[2]\}, \\
\{z \rightarrow \text{Function}\{t\}, e^t \times C[3]\}\}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{In[33]:=} & \quad \text{system} /. \text{sol} // \text{Simplify} \\
\text{Out[33]=} & \quad \{(\text{True}, \text{True}, \text{True})\}
\end{align*}
\]

The systems considered so far have all been homogeneous. If the system is inhomogeneous (that is, if there are terms free from any dependent variables and their derivatives), DSolve applies either the method of variation of parameters or the method of undetermined coefficients to find the general solution.

This solves an inhomogeneous system.

\[
\begin{align*}
\text{In[34]:=} & \quad A = \{(7, -8), \{5, -5\}\}; \\
\text{In[35]:=} & \quad B = \{e^t \times (t / 10), t\}; \\
\text{In[36]:=} & \quad X[t_] = \{x[t], y[t]\}; \\
\text{In[37]:=} & \quad \text{system} = \text{MapThread}[\#1 == \#2 &\{X'[t], A \times X[t] + B\}] \\
\text{Out[37]=} & \quad \{x'[t] = e^{t10} + 7 \times x[t] - 8 \times y[t], y'[t] = t + 5 \times x[t] - 5 \times y[t]\}
\end{align*}
\]
In[38]:= \[sol = DSolve[system, \{x, y\}, t] \]

Out[38]= \{\{x \to Function[t],
   1 \frac{e^t (C[2] \sin[2 t] + e^t C[1] (\cos[2 t] + 3 \sin[2 t])) - 4 e^t \sin[2 t]}{481} e^{-9 t/10} \cos[2 t] - \\
   4 e^t \sin[2 t] \frac{500}{481} e^{-9 t/10} \cos[2 t] - \\
   4 e^t (4 + 10 t) \cos[2 t] + \\
   500 e^{-9 t/10} \sin[2 t] - \\
   500 \frac{2 e^{-9 t/10} \sin[2 t]}{481} - \\
   4 e^t \sin[2 t] \frac{510}{481} e^{-9 t/10} \cos[2 t] - \\
   \frac{2 e^{-9 t/10} \sin[2 t]}{25} - \\
   4 e^t (4 + 10 t) \cos[2 t] + \frac{470}{481} e^{-9 t/10} \sin[2 t] - \\
   \frac{470 e^{-9 t/10} \sin[2 t]}{481} - \\
   \frac{2 e^{-9 t/10} \sin[2 t]}{25} - \\
   4 e^t (-6 + 10 t) \sin[2 t])\},

y \to Function[t],
   \frac{500 e^{-9 t/10} \cos[2 t]}{481} e^t (C[2] \cos[2 t] - 3 \sin[2 t]) + \frac{1}{2} e^t C[1] \sin[2 t] + \\
   \frac{500 e^{-9 t/10} \cos[2 t]}{481} e^t (18 + 70 t) \cos[2 t] + \\
   \frac{500 e^{-9 t/10} \sin[2 t]}{481} e^t (-26 + 10 t) \sin[2 t] + \\
   \frac{500 e^{-9 t/10} \sin[2 t]}{481} e^t (4 + 10 t) \cos[2 t] + \frac{470}{481} e^{-9 t/10} \sin[2 t] - \\
   \frac{470 e^{-9 t/10} \sin[2 t]}{481} - \\
   \frac{2 e^{-9 t/10} \sin[2 t]}{25} - \\
   4 e^t (-6 + 10 t) \sin[2 t])\} \}

Particular solutions to the system can be obtained by assigning values to the constants C[1] and C[2].

Here is a plot of the solution for one choice of parameters.

In[39]:= \[particularsol = \{x[t], y[t]\} /. sol[[1]] /. \{C[1] \to -1, C[2] \to 2\}; \]

In[40]:= \[Plot[Evaluate[particularsol], \{t, 0, Pi\}] \]

Out[40]=

Nonlinear Systems of ODEs

Following are two examples of nonlinear systems of ODEs that can be solved symbolically using DSolve.

The first three equations in this system of four nonlinear ODEs can be solved independently because none of their right-hand sides depend on p, q, r, or s.

In[1]:= \[system = \{p'[x] == 1, q'[x] == x, r'[x] == 0, s'[x] == r[x] / (p[x] + 4 q[x] r[x])\}; \]
In[2]:= \text{sol} = \text{DSolve}\left[\text{system}, \{p, q, r, s\}, x\right]

Out[2]= \left\{\begin{array}{l}
P \rightarrow \text{Function}\left[x, \frac{x^2}{2} + C[2]\right], \\
q \rightarrow \text{Function}\left[x, \frac{1}{8} \frac{4 C[1]^2 + 4 C[1]}{\left[-324 x + \sqrt{23328 C[1]^3 + (-324 x - 81 C[2])^2 - 81 C[2]}\right]^{1/3}} + \frac{6 \cdot 2^{1/3} C[1]}{3 \cdot 2^{1/3}} \right] \\

\text{r} \rightarrow \text{Function}\left[x, C[3]\right], \\
s \rightarrow \text{Function}\left[x, \frac{1}{8} \frac{4 C[1]^2 + 4 C[1]}{\left[-324 x + \sqrt{23328 C[1]^3 + (-324 x - 81 C[2])^2 - 81 C[2]}\right]^{1/3}} + C[3]\right]
\end{array}\right\}

In[3]:= \text{system} /. \text{sol} // \text{Simplify}

Out[3]= \{\text{True, True, True, True}\}

This system of two nonlinear ODEs is \textit{autonomous}, in the sense that the right-hand sides of the equations do not depend on \(x\).

In[4]:= \text{system} = \{u'[x] = 1/\text{Sqrt}[v[x]], \ v'[x] = u[x]\};

In[5]:= \text{sol} = \text{DSolve}\left[\text{system}, \{u, v\}, x\right][[1]]

Out[5]= \left\{\begin{array}{l}
u \rightarrow \text{Function}\left[x, \frac{1}{16} \frac{4 C[1]^2 + 4 C[1]}{\left[-324 x + \sqrt{23328 C[1]^3 + (-324 x - 81 C[2])^2 - 81 C[2]}\right]^{1/3}} + \frac{6 \cdot 2^{1/3} C[1]}{3 \cdot 2^{1/3}} \\

\text{u} \rightarrow \text{Function}\left[x, \frac{6 \cdot 2^{1/3} C[1]}{3 \cdot 2^{1/3}} \right]
\end{array}\right\}

In[6]:= \text{system} /. \text{sol} // \text{Simplify} // \text{PowerExpand} // \text{Simplify}

Out[6]= \{\text{True, True}\}
The previous two examples demonstrate that the solutions to fairly simple systems are usually complicated expressions of the independent variable. In fact, the solution is often available only in implicit form and may thus contain InverseFunction objects or unevaluated Solve objects.

### Lie Symmetry Methods for Solving Nonlinear ODEs

Around 1870, Marius Sophus Lie realized that many of the methods for solving differential equations could be unified using group theory. Lie symmetry methods are central to the modern approach for studying nonlinear ODEs. They use the notion of symmetry to generate solutions in a systematic manner. Here is a brief introduction to Lie’s approach that provides some examples that are solved in this way by DSolve.

A key notion in Lie’s method is that of an *infinitesimal generator* for a symmetry group. This concept is illustrated in the following example.

Here is the well-known transformation for rotations in the \(x-y\) plane. This is a one-parameter group of transformations with parameter \(t\).

\[
\begin{align*}
\text{In}[1]:= & \quad m = x \cdot \cos[t] + y \cdot \sin[t]; \\
\text{In}[2]:= & \quad n = -x \cdot \sin[t] + y \cdot \cos[t];
\end{align*}
\]

For a fixed value of \(t\), the point \((m, n)\) (in blue) can be obtained by rotating the line joining \((x, y)\) (in red) to the origin through an angle of \(t\) in the counterclockwise direction.
A rotation through $\theta$ can be represented by the matrix

$$M_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$ 

This shows that the set of all rotations in the plane satisfies the properties for forming a group.

The Lie symmetry method requires calculating a first-order approximation to the expressions for the group. This approximation is called an \textit{infinitesimal generator}. This expands the expressions for $m$ and $n$ in a series with respect to $\theta$ and around the origin 0 to obtain linear approximations.

The rotation group arises in the study of symmetries of geometrical objects; it is an example of a symmetry group. The infinitesimal generator, a differential operator, is a convenient local representation for this symmetric group, which is a set of matrices.
The rotation group arises in the study of symmetries of geometrical objects; it is an example of a *symmetry group*. The infinitesimal generator, a differential operator, is a convenient local representation for this symmetric group, which is a set of matrices.

An expression that reduces to 0 under the action of the infinitesimal generator is called an *invariant* of the group.

Here is an invariant for this group.

```
In[12]:= invariant = x^2 + y^2;
```

This states that the distance from the origin to \((x, y) = \sqrt{x^2 + y^2}\), is preserved under rotation.

```
In[13]:= v[invariant]
Out[13]= 0
```

In the following examples, these ideas are applied to differential equations.

This is an example of a Riccati equation, from page 103 of [I99].

```
In[14]:= Riccatiequation = y'[x] + y[x]^2 - 1/x^2 == 0;
```

The equation is invariant under the following *scaling* transformation.

```
In[15]:= m = x*E^t;
In[16]:= n = y*E^(-t);
```

The infinitesimal generator for this one-parameter group of transformations is found as before.

```
In[17]:= Series[m, {t, 0, 1}] // Normal
Out[17]= x + tx

In[18]:= Series[n, {t, 0, 1}] // Normal
Out[18]= y - ty

In[19]:= v = (x*D[#1, x] - y*D[#1, y]) &;
```

Now, the Riccati equation depends on three variables: \(x, y = y[x]\), and \(p = y'[x]\). Hence, the infinitesimal generator \(v\) must be *prolonged* to act on all three variables in this first-order equation.

```
It turns out that the required *prolongation* is as follows.
In[20]:= prolongedv = (x*D[#1, x] - y*D[#1, y] - 2 p*D[#1, p]) &;
```
This shows that the expression for the Riccati equation in the \((x, y, p)\) coordinates is indeed invariant under prolongedv.

\[
\text{In}[21]:= \text{Riccatiexpression} = p + y^2 - (1 / x^2); \\
\text{In}[22]:= \text{prolongedv[Riccatiexpression]} /. \{p \to (1 / x^2) - y^2\} // \text{Together} \\
\text{Out}[22]= 0
\]

Depending on the order of the given equation, the knowledge of a symmetry (in the form of an infinitesimal generator) can be used in three ways.

- If the order of the equation is 1, it gives an integrating factor for the ODE that makes the equation exact and hence solvable.
- It gives a set of canonical coordinates in which the equation has a simple (integrable) form.
- It reduces the problem of solving an ODE of order \(n\) to that of solving an ODE of order \(n - 1\), which is typically a simpler problem.

The \text{DSolve} function checks for certain standard types of symmetries in the given ODE and uses them to return a solution. Following are three examples of ODEs for which \text{DSolve} uses such a symmetry method.

Here is a nonlinear first-order ODE (equation 1.120, page 315 of [K59]).

\[
\text{In}[23]:= \text{FirstOrderODE} = x * y'[x] == y[x] * (x * \text{Log}[x^2 / y[x]] + 2); \\
\text{This ODE has a symmetry with the following infinitesimal generator.} \\
\text{In}[24]:= v = (-2 * \text{Exp}[-x] * y) * \text{D}[#, y] &; \\
\text{The presence of this symmetry allows \text{DSolve} to calculate an integrating factor and return the solution.} \\
\text{In}[25]:= \text{sol} = \text{DSolve}[\text{FirstOrderODE}, y, x] \\
\text{Out}[25]= \{\{y \to \text{Function}[\{x\}, e^{-2 e^x c[1]} x^2]\}\}
\]

This verifies the solution.

\[
\text{In}[26]:= \text{FirstOrderODE} /. \text{sol[[1]]} /. \{x \to \text{RandomReal[]}, c[1] \to \text{RandomReal[]}\} // \text{Simplify} // \text{Chop} \\
\text{Out}[26]= \text{True}
\]

Here is a second-order nonlinear ODE, based on equation 6.93 on page 213 of [K59].

\[
\text{In}[27]:= \text{SecondOrderODE} = x^3 * y''[x] == 6 * (x * y'[x] - y[x])^2;
\]
This equation is invariant under the following scaling transformation.

\[ m = x \cdot e^t; \]

\[ n = y \cdot e^t; \]

The presence of this scaling symmetry allows DSolve to find new coordinates in which the independent variable is not explicitly present. Hence the problem is solved easily.

\[ \text{In}\[30\] := \text{sol} = \text{DSolve}[\text{SecondOrderODE}, y, x] \]

\[ \text{Out}\[30\] := \text{Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.} \]

\[ \text{Out}\[30\] \approx \{y \rightarrow \text{Function}[[x], \frac{1}{6} x \log \left(6 \left(-\frac{C[1]}{x} - C[2]\right)\right)]\} \]

This verifies the solution.

\[ \text{In}\[31\] := \text{SecondOrderODE} \/. \text{sol} \/. \text{Simplify} \]

\[ \text{Out}\[31\] = \{\text{True}\} \]

Finally, here is a system of two nonlinear first-order ODEs that can be solved by using a shift: \( u[x] \rightarrow u[x] - x \). After the shift, the system becomes autonomous (it does not depend explicitly on \( x \)) and hence it can be solved by reduction to a first-order ODE for \( v \) as a function of \( u \). The Solve::ifun message can be ignored; it is generated while inverting the expression for \( \exp[v] \) to give an expression in terms of \( \log \).

\[ \text{In}\[32\] := \text{Clear}[u, v] \]

\[ \text{In}\[33\] := \text{NonlinearSystem} = \{u'[x] = \exp[v[x]] + 1, v'[x] = u[x] - x\}; \]

\[ \text{In}\[34\] := \text{sol} = \text{DSolve}[\text{NonlinearSystem}, \{u, v\}, x] \]

\[ \text{Out}\[34\] \approx \{v \rightarrow \text{Function}[[x], \log[C[1] + C[1] \tan\left(\frac{1}{2} \left(\sqrt{2 x \sqrt{C[1]} + 2 \sqrt{2 \sqrt{C[1]} C[2]}\right)\right)^2]], \]

\[ u \rightarrow \text{Function}[[x], x + \sqrt{2 \sqrt{C[1]} \tan\left(\frac{1}{2} \left(\sqrt{2 x \sqrt{C[1]} + 2 \sqrt{2 \sqrt{C[1]} C[2]}\right)\right)]} \}

\[ \text{In}\[35\] := \text{NonlinearSystem} \/. \text{sol} \/. \text{Simplify} \]

\[ \text{Out}\[35\] = \{\{\text{True, True}\}\} \]

\[ \text{In}\[36\] := \text{Clear}[m, n, u, v] \]

This concludes the discussion of ordinary differential equations.
Partial Differential Equations (PDEs)

Introduction to Partial Differential Equations (PDEs)

A partial differential equation (PDE) is a relationship between an unknown function \( u(x_1, x_2, \ldots, x_n) \) and its derivatives with respect to the variables \( x_1, x_2, \ldots, x_n \).

Here is an example of a PDE.

\[
\text{In[1]:= } \quad \text{equation1} = \frac{\partial u(x, y)}{\partial x} + x \frac{\partial u(x, y)}{\partial y} = \sin(x);
\]

PDEs occur naturally in applications; they model the rate of change of a physical quantity with respect to both space variables and time variables. At this stage of development, \texttt{DSolve} typically only works with PDEs having two independent variables.

The order of a PDE is the order of the highest derivative that occurs in it. The previous equation is a first-order PDE.

A function \( u(x, y) \) is a \textit{solution} to a given PDE if \( u \) and its derivatives satisfy the equation.

Here is one solution to the previous equation.

\[
\text{In[2]:= } \quad \text{sol} = u / . \text{DSolve[equation1, u, \{x, y\}][[1]]} / . \text{C[1][t_] \rightarrow t}
\]

\[
\text{Out[2]= } \quad \text{Function[\{x, y\}, -\text{Cos}[x] + \frac{1}{2} (-x^2 + 2 y)]}
\]

This verifies the solution.

\[
\text{In[3]:= } \quad \text{equation1} / (. u \rightarrow \text{sol})
\]

\[
\text{Out[3]= } \quad \text{True}
\]

Here are some well-known examples of PDEs (clicking a link in the table will bring up the relevant examples). \texttt{DSolve} gives symbolic solutions to equations of all these types, with certain restrictions, particularly for second-order PDEs.
### Table: Equations and their General Forms and Classifications

<table>
<thead>
<tr>
<th>name of equation</th>
<th>general form</th>
<th>classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport equation</td>
<td>[ \frac{\partial u}{\partial x} + c \frac{\partial u}{\partial y} = 0 \text{ with } c \text{ constant} ]</td>
<td>linear first-order PDE</td>
</tr>
<tr>
<td>Burgers’ equation</td>
<td>[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 ]</td>
<td>quasilinear first-order PDE</td>
</tr>
<tr>
<td>eikonal equation</td>
<td>[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 = 1 ]</td>
<td>nonlinear first-order PDE</td>
</tr>
<tr>
<td>Laplace’s equation</td>
<td>[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 ]</td>
<td>elliptic linear second-order PDE</td>
</tr>
<tr>
<td>wave equation</td>
<td>[ \frac{\partial^2 u}{\partial x^2} = c^2 \frac{\partial^2 u}{\partial t^2} ] \text{ where } c \text{ is the speed of light}</td>
<td>hyperbolic linear second-order PDE</td>
</tr>
<tr>
<td>heat equation</td>
<td>[ \frac{\partial^2 u}{\partial x^2} = k \frac{\partial u}{\partial t} ] \text{ where } k \text{ is the thermal diffusivity}</td>
<td>parabolic linear second-order PDE</td>
</tr>
</tbody>
</table>

Recall that the general solutions to PDEs involve arbitrary functions rather than arbitrary constants. The reason for this can be seen from the following example.

The partial derivative with respect to \( y \) does not appear in this example, so an arbitrary function \( C[1][y] \) can be added to the solution, since the partial derivative of \( C[1][y] \) with respect to \( x \) is 0.

```
In[4] := DSolve[D[u[x, y], x] == 1, u[x, y], {x, y}]
```

```
Out[4] = \{\{u[x, y] -> x + C[1][y]\}\}
```

If there are several arbitrary functions in the solution, they are labeled as \( C[1], C[2], \) and so on.

## First-Order PDEs

### Linear and Quasi-Linear PDEs

First-order PDEs are usually classified as linear, quasi-linear, or nonlinear. The first two types are discussed in this tutorial.

A first-order PDE for an unknown function \( u(x, y) \) is said to be linear if it can be expressed in the form

\[
a(x, y) \frac{\partial u(x, y)}{\partial x} + b(x, y) \frac{\partial u(x, y)}{\partial y} + c(x, y) u(x, y) = d(x, y).
\]
The PDE is said to be *quasilinear* if it can be expressed in the form

\[
a(x, y, u(x, y)) \frac{\partial u(x, y)}{\partial x} + b(x, y, u(x, y)) \frac{\partial u(x, y)}{\partial y} = c(x, y, u(x, y)).
\]

A PDE which is neither linear nor quasi-linear is said to be *nonlinear*.

For convenience, the symbols \( z, p, \) and \( q \) are used throughout this tutorial to denote the unknown function and its partial derivatives.

\[
z = u(x, y); \quad p = \frac{\partial u(x, y)}{\partial x}; \quad q = \frac{\partial u(x, y)}{\partial y}
\]

Here is a linear homogeneous first-order PDE with constant coefficients.

\[
In[1]:= \text{z := u[x, y]}
\]

\[
In[2]:= \text{p := D[u[x, y], x]}
\]

\[
In[3]:= \text{q := D[u[x, y], y]}
\]

\[
In[4]:= \text{eqn = 2*p + 3*q + z == 0;}
\]

The equation is linear because the left-hand side is a linear polynomial in \( z, p, \) and \( q \). Since there is no term free of \( z, p, \) or \( q \), the PDE is also homogeneous.

As mentioned earlier, the general solution contains an arbitrary function \( C[1] \) of the argument \( \frac{1}{2}(2 y - 3 x) \).

\[
In[5]:= \text{sol = DSolve[eqn, u, \{x, y\]}
\]

\[
Out[5]= \{\{u \rightarrow \text{Function}[\{x, y\}, e^{-x/2} C[1] \left(\frac{1}{2} (-3 x + 2 y)\right)]\}\}
\]

This verifies that the solution is correct.

\[
In[6]:= \text{eqn /. sol[[1]] // Simplify}
\]

\[
Out[6]= \text{True}
\]

Particular solutions of the homogeneous PDE are obtained by specifying the function \( C[1] \).

\[
In[7]:= \text{particularsolution = u[x, y] /. sol[[1]] /. C[1][a_] \rightarrow \text{Sin}[a]}
\]

\[
Out[7]= e^{-x/2} \text{Sin}\left[\frac{1}{2} (-3 x + 2 y)\right]
\]
Here is a plot of the surface for this particular solution.

In[8]:= Plot3D[particularsolution, {x, -2, 2}, {y, -7, 7}, PlotPoints -> 30]

Out[8]=

The *transport equation* is a good example of a linear first-order homogeneous PDE with constant coefficients.

In this transport equation, \( c = 1 \) for convenience.

In[9]:= DSolve[D[u[x, y], x] + D[u[x, y], y] = 0, u[x, y], {x, y}]
Out[9]= \[\{u[x, y] \rightarrow C[1][-x + y]\}\]

Note that the solution to the transport equation is constant on any straight line of the form \( y = x + \alpha \) in the plane. These straight lines are called the *base characteristic curves*. The equation \( y = x + \alpha \) defines a plane in three dimensions. The intersections of these planes with the solution surface are called *characteristic curves*. Since the characteristic curves are solutions to a system of ODEs, the problem of solving the PDE is reduced to that of solving a system of ODEs for \( x(t), y(t), \) and \( u(t) \), where \( t \) is a parameter along the characteristic curves. These ODEs are called characteristic ODEs.

The solution to an inhomogeneous PDE has two components: the general solution to the homogeneous PDE and a particular solution to the inhomogeneous PDE.

This is a linear inhomogeneous PDE of the first order.

In[10]:= eqn = 7 * p + 3 * q + z = x + y;
The first part of the solution, \(-10 + x + y\), is the particular solution to the inhomogeneous PDE. The rest of the solution is the general solution to the homogeneous equation.

\[
\begin{align*}
\text{In[11]:=} & \quad \text{sol = u[x, y] /. DSolve[eqn, u[x, y], \{x, y\}][[1]] // Expand} \\
\text{Out[11]} & \quad -10 + x + y + e^{-x/7} c[1] \left[ \frac{1}{7} (-3 x + 7 y) \right]
\end{align*}
\]

Here is a linear homogeneous PDE with variable coefficients.

\[
\begin{align*}
\text{In[12]:=} & \quad \text{eqn = Sin}[x] \cdot p + e^{x} \cdot q = 0; \\
\text{In[13]:=} & \quad \text{sol = DSolve[eqn, u, \{x, y\}]} \\
\text{Out[13]} & \quad \{\{u \rightarrow \text{Function}[\{x, y\}, c[1] \left[ y + (1 + i) e^{(1+i) x} \text{Hypergeometric2F1}[\frac{1}{2} - \frac{i}{2}, 1, \frac{3}{2} - \frac{i}{2}, e^{2 i x}] \}]]\}
\end{align*}
\]

This verifies the solution.

\[
\begin{align*}
\text{In[14]:=} & \quad \text{eqn /. sol[[1]] // Simplify} \\
\text{Out[14]} & \quad \text{True}
\end{align*}
\]

Here is a linear inhomogeneous PDE with variable coefficients.

\[
\begin{align*}
\text{In[15]:=} & \quad \text{eqn = p} + x \cdot q = \text{Cos}[x]; \\
\text{In[16]:=} & \quad \text{sol = DSolve[eqn, u, \{x, y\}]} \\
\text{Out[16]} & \quad \{\{u \rightarrow \text{Function}[\{x, y\}, \text{Sin}[x] + c[1] \left[ \frac{1}{2} (-x^2 + 2 y) \}]]\}
\end{align*}
\]

Now consider some examples of first-order quasi-linear PDEs.

This PDE is quasi-linear because of the term \(z^2\) on the right-hand side.

\[
\begin{align*}
\text{In[17]:=} & \quad \text{eqn = p} + x \cdot q = z^2 + 5; \\
\text{In[18]:=} & \quad \text{sol = DSolve[eqn, u, \{x, y\}]} \\
\text{Out[18]} & \quad \{\{u \rightarrow \text{Function}[\{x, y\}, \sqrt{5} \text{ Tan}[\sqrt{5} x + \sqrt{5} c[1] \left[ \frac{1}{2} (-x^2 + 2 y) \}]]\}
\end{align*}
\]

This verifies the solution.

\[
\begin{align*}
\text{In[19]:=} & \quad \text{eqn /. sol[[1]] // Simplify} \\
\text{Out[19]} & \quad \text{True}
\end{align*}
\]
Burgers’ equation is an important example of a quasi-linear PDE.

\[ \frac{\partial u(x, y)}{\partial x} + u(x, y) \frac{\partial u(x, y)}{\partial y} = 0 \]

It can be written using the notation introduced earlier.

\texttt{In[20]} := \texttt{BurgersEquation = p + q \cdot u \cdot 0;}

The term \( z \cdot q \) makes this equation quasi-linear.

This solves the equation.

\texttt{In[21]} := \texttt{sol = DSolve[BurgersEquation, u, \{x, y\}];}
\texttt{Out[21]} = \texttt{Solve[C[1][u[x, y], y - x u[x, y]] = 0, u[x, y]]}

This verifies the solution to Burgers’ equation.

\texttt{In[22]} := \texttt{pl = p \cdot . \texttt{Solve[D[sol[[1]], x], p][[1]];}}
\texttt{In[23]} := \texttt{ql = q \cdot . \texttt{Solve[D[sol[[1]], y], q][[1]];}}
\texttt{In[24]} := \texttt{pl + z \cdot ql}
\texttt{Out[24]} = 0

A practical consequence of quasi-linearity is the appearance of shocks and steepening and breaking of solutions. Thus, although the procedures for finding general solutions to linear and quasi-linear PDEs are quite similar, there are sharp differences in the nature of the solutions.

\textbf{Nonlinear PDEs}

The general first-order nonlinear PDE for an unknown function \( u(x, y) \) is given by

\[ F(u, p, q) = 0. \]

Here \( F \) is a function of \( u = u(x, y), p = \frac{\partial u(x, y)}{\partial x}, \) and \( q = \frac{\partial u(x, y)}{\partial y}. \)

The term “nonlinear” refers to the fact that \( F \) is a nonlinear function of \( p \) and \( q. \) For instance, the eikonal equation involves a quadratic expression in \( p \) and \( q. \)
The general solution to a first-order linear or quasi-linear PDE involves an arbitrary function. If the PDE is nonlinear, a very useful solution is given by the complete integral. This is a function of \( u(x, y, C[1], C[2]) \), where \( C[1] \) and \( C[2] \) are independent parameters and \( u \) satisfies the PDE for all values of \( (C[1], C[2]) \) in an open subset of the plane. The complete integral can be used to find a general solution for the PDE as well as to solve initial value problems for it.

Here is a simple nonlinear PDE.

\[
\begin{align*}
In[1]:= & \quad z := u[x, y] \\
In[2]:= & \quad p := D[u[x, y], x] \\
In[3]:= & \quad q := D[u[x, y], y] \\
In[4]:= & \quad eqn = p \cdot q \equiv 1;
\end{align*}
\]

The complete integral depends on the parameters \( C[1] \) and \( C[2] \). Since DSolve returns a general solution for linear and quasi-linear PDEs, a warning message appears before a complete integral is returned.

\[
\begin{align*}
In[5]:= & \quad sol = DSolve[eqn, u, \{x, y\}] \\
DSolve::nlpde: & \quad \text{Solution requested to nonlinear partial differential equation. Trying to build a special solution.} \\
Out[5]= & \quad \{u \to \text{Function}[[x, y], C[1] + \frac{x}{C[2]} + y C[2]]\}
\end{align*}
\]

This verifies the solution.

\[
\begin{align*}
In[6]:= & \quad eqn /. sol \\
Out[6]= & \quad \{\text{True}\}
\end{align*}
\]

If the values of \( C[1] \) and \( C[2] \) are fixed, the previous solution represents a plane in three dimensions. Thus, the complete integral for this PDE is a two-parameter family of planes, each of which is a solution surface for the equation.

Next, the envelope of a one-parameter family of surfaces is a surface that touches each member of the family. If the complete integral is restricted to a one-parameter family of planes, for example by setting \( C[2] = 5 C[1] \), the envelope of this family is also a solution to the PDE called a general integral.
This finds the envelope of the one-parameter family given by setting \( C[2] = 5 C[1] \) in the complete integral for the preceding PDE \( p \times q = 1 \).

\[
In[7]:= \text{oneparametersol} = u[x, y] = \frac{x}{5 C[1]} + C[1] + 5 y C[1]
\]

\[
Out[7]= u[x, y] = \frac{x}{5 C[1]} + C[1] + 5 y C[1]
\]

\[
In[8]:= \text{oneparameterenvelope} = \text{Eliminate}[\{\text{oneparametersol}, D[\text{oneparametersol}, C[1]], \{C[1]\}]]
\]

\[
Out[8]= 5 u[x, y]^2 = x (4 + 20 y)
\]

This verifies that the envelope surface is a solution to the PDE.

\[
In[9]:= pl = D[u[x, y], x] /. \text{Solve}[D[\text{oneparameterenvelope}, x], D[u[x, y], x]][[1]]
\]

\[
Out[9]= \frac{2 (1 + 5 y)}{5 u[x, y]}
\]

\[
In[10]:= ql = D[u[x, y], y] /. \text{Solve}[D[\text{oneparameterenvelope}, y], D[u[x, y], y]][[1]]
\]

\[
Out[10]= \frac{2 x}{u[x, y]}
\]

\[
In[11]:= \text{FullSimplify}[pl \times ql, \{\text{oneparameterenvelope}\}]
\]

\[
\]

Like nonlinear ODEs, some nonlinear PDEs also have a singular solution (or singular integral) that is obtained by constructing the envelope of the entire two-parameter family of surfaces represented by the complete integral.

Here is an example of such a construction, (equation 6.4.13, page 429 of [K00]).

\[
In[12]:= \text{sol} = \text{DSolve}[4 z + p^2 + q^2 = 4, u, \{x, y\}]
\]

\[
\text{DSolve::nlpde: Solution requested to nonlinear partial differential equation. Trying to build a special solution.} \quad \star \quad
\]

\[
\]

\[
In[13]:= \text{twoparameterfamily} = u[x, y] = (u[x, y] /. \text{sol}[[1]])
\]

\[
\]

\[
In[14]:= \text{envelopeoftwoparameterfamily} = \text{Eliminate}[\{\text{twoparameterfamily}, D[\text{twoparameterfamily}, C[1]], D[\text{twoparameterfamily}, C[2]], \{C[1], C[2]\}]
\]

\[
Out[14]= u[x, y] = 1
\]
Thus, the singular integral for this PDE is a plane parallel to the $x$-$y$ plane.

To summarize, the complete integral for a nonlinear PDE includes a rich variety of solutions.

- Every member of the two-parameter family gives a particular solution to the PDE.
- The envelope of any one-parameter family is a solution called a general integral of the PDE.
- The envelope of the entire two-parameter family is a solution called the singular integral of the PDE.
- The complete integral is not unique, but any other complete integral for the PDE can be obtained from it by the process of envelope formation.

These remarkable properties account for the usefulness of the complete integral in geometrical optics, dynamics, and other areas of application. Following are various examples of nonlinear PDEs that show different kinds of complete integrals.

Here is the complete integral for the eikonal equation.

```
In[15]:= Eikonal = D[u[x, y], x]^2 + D[u[x, y], y]^2 == 1;

In[16]:= sol = DSolve[Eikonal, u, {x, y}]
```

```
DSolve::nlpde: Solution requested to nonlinear partial differential equation. Trying to build a special solution. >>
```

```

```

This complete integral is a two-parameter family of planes. This type of solution arises whenever the PDE depends explicitly only on $p$ and $q$, but not on $u[x, y]$, $x$, or $y$. For a fixed value of $u[x, y]$, it is a line in the plane at a distance of $C[1]$ units from the origin that makes an angle of $\text{ArcCos}[C[2]]$ with the $x$ axis. This is the familiar picture of wave-front propagation from geometrical optics.

```
This verifies the solution for the eikonal equation.

In[17]:= Eikonal /. sol
```

```
Out[17]= {True, True}
```

This is an example of a Clairaut equation ($z = px + qy + f(p, q)$).

```
In[18]:= Clairaut = z == x * p + y * q + 2 * p * q * \text{Sqrt}[1 - p^2];
```
Once again, the complete integral is a family of planes.

\[ \text{In}[19]:= \text{sol = DSolve}[	ext{Clairaut, u, \{x, y\}}] \]

\[ \text{DSolve::nlpde: } \text{Solution requested to nonlinear partial differential equation. Trying to build a special solution.} \]

\[ \text{Out}[19]= \{u \rightarrow \text{Function}[\{x, y\}, \frac{3 y^2}{2 a} + \frac{2}{3} (x - C[1])^{3/2} + \frac{y C[1]}{a} + C[2]]\} \]

This verifies the solution.

\[ \text{In}[20]:= \text{Clairaut} /\!\!/ \text{sol} \]

\[ \text{Out}[20]= \{\text{True}\} \]

In the following equation, the variables can be separated; that is, the PDE can be written in the form \( f(x, p) = g(y, q) \). Hence, the equation can be integrated easily.

\[ \text{In}[21]:= \text{Separable} = p^2 + a * q = x + 3 y; \]

\[ \text{In}[22]:= \text{sol = DSolve}[	ext{Separable, u, \{x, y\}}] \]

\[ \text{DSolve::nlpde: } \text{Solution requested to nonlinear partial differential equation. Trying to build a special solution.} \]

\[ \text{Out}[22]= \{u \rightarrow \text{Function}[\{x, y\}, \frac{3 y^2}{2 a} + \frac{2}{3} (x - C[1])^{3/2} + \frac{y C[1]}{a} + C[2]]\}, \]

\[ \{u \rightarrow \text{Function}[\{x, y\}, \frac{3 y^2}{2 a} + \frac{2}{3} (x - C[1])^{3/2} + \frac{y C[1]}{a} + C[2]]\} \]

This verifies the solution.

\[ \text{In}[23]:= \text{Separable} /\!\!/ \text{sol} /\!\!/ \text{Simplify} \]

\[ \text{Out}[23]= \{\text{True, True}\} \]

In this example (equation 6.49, page 202 of [K74]), the independent variables \( x \) and \( y \) are not explicitly present.

\[ \text{In}[24]:= \text{MissingIndependentVariables} = a * p^2 + b * p * q = c * z^2; \]

\[ \text{In}[25]:= \text{sol = DSolve}[	ext{MissingIndependentVariables, u, \{x, y\}}] \]

\[ \text{DSolve::nlpde: } \text{Solution requested to nonlinear partial differential equation. Trying to build a special solution.} \]

\[ \text{Out}[25]= \{u \rightarrow \text{Function}[\{x, y\}, e^{\frac{i \sqrt{-y}}{\sqrt{a C[1]} \cdot a C[1]}} \cdot \frac{1}{\sqrt{a C[1] \cdot a C[1]}}] \}

\[ \{u \rightarrow \text{Function}[\{x, y\}, e^{\frac{i \sqrt{-y}}{\sqrt{a C[1] \cdot a C[1]}} \cdot \frac{1}{\sqrt{a C[1] \cdot a C[1]}}}] \} \]
This verifies the solution.

\[\text{In[26]} := \text{MissingIndependentVariables} /. \text{sol} // \text{Simplify}\]

\[\text{Out[26]} = \text{True}\]

Often a coordinate transformation can be used to cast a given PDE into one of the previous types. The expression for the complete integral will then have the same form as for the standard types. Here are some examples of nonlinear PDEs for which \text{DSolve} applies a coordinate transformation to find the complete integral.

This PDE (equation 6.47, page 201 of [K74]) can be reduced to the form \(f(p, q) = 0\) using the transformation \(X = \log(x)\) and \(Y = \log(y)\).

\[\text{In[27]} := \text{DSolve}[x \ast y \ast p \ast q = 1, u[x, y], \{x, y\}]\]

\text{DSolve::nlpde: Solution requested to nonlinear partial differential equation. Trying to build a special solution.}

\[\text{Out[27]} = \{u[x, y] \rightarrow C[2] + C[1] \log[x] + \frac{\log[y]}{C[1]}\}\]

This PDE (equation 6.93, page 213 of [K74]) can be solved easily in a polar coordinate system, in which the variables are separable.

\[\text{In[28]} := \text{DSolve[(y*p - x*q)}^2 + a \ast (x*p + y*q) = b, z, \{x, y\}]\]

\text{DSolve::nlpde: Solution requested to nonlinear partial differential equation. Trying to build a special solution.}

\[\text{Out[28]} = \{u[x, y] \rightarrow \text{ArcTan}\left(\frac{y}{x}\right) \sqrt{C[1]} + C[2] + \frac{b \log\left[\sqrt{x^2 + y^2}\right]}{a} - \frac{C[1] \log\left[\sqrt{x^2 + y^2}\right]}{a}\}\]

This equation (equation 6.36, page 196 of [K74]) can be transformed into a linear PDE using a Legendre transformation.

\[\text{In[29]} := \text{LegendreTransformable} = y \ast p \ast q - z \ast p + a \ast q = 0;\]

\[\text{In[30]} := \text{sol} = \text{DSolve}[\text{LegendreTransformable}, u, \{x, y\}]\]

\text{DSolve::nlpde: Solution requested to nonlinear partial differential equation. Trying to build a special solution.}

\[\text{Out[30]} = \{u \rightarrow \text{Function}\left[\{x, y\}, \frac{1}{4 \ C[1]} \left(-y \ (-2 \ x + 2 \ C[2]) - \sqrt{y^2 \ (-2 \ x + 2 \ C[2])^2 + 8 \ a \ C[1] \ (x^2 - 2 \ x \ C[2] + C[2]^2)}\right]\}, \{u \rightarrow \text{Function}\left[\{x, y\}, \frac{1}{4 \ C[1]} \left(-y \ (-2 \ x + 2 \ C[2]) + \sqrt{y^2 \ (-2 \ x + 2 \ C[2])^2 + 8 \ a \ C[1] \ (x^2 - 2 \ x \ C[2] + C[2]^2)}\right]\}\}

\]
This verifies the solution.

\[ \text{In[31]} := \text{LegendreTransformable / sol / \text{Simplify}} \]

\[ \text{Out[31]} = \{\text{True}, \text{True}\} \]

It should be noted that there is no general practical algorithm for finding complete integrals, and that the answers are often available only in implicit form.

The solution to this example (problem 2, page 66 of [S57]) is in implicit form.

\[ \text{In[32]} := \text{sol} = \text{DSolve}[\{1 + q^2 \cdot z \cdot p \cdot x, u, \{x, y\}\}] \]

\[ \text{Out[32]} = \text{Solve}\left[\frac{1}{2} \left(\text{C[1]} \cdot \text{Log}\left[\text{C[1]} + \sqrt{\text{C[1]}^2 - 4 \cdot u[x, y]^2}\right] + \sqrt{\text{C[1]}^2 - 4 \cdot u[x, y]^2}\right) = y + \text{C[2]} + \text{C[1]} \cdot \text{Log}[x], u[x, y]\right] \]

The solution can be verified as follows.

\[ \text{In[33]} := \text{p} = \text{D[u[x, y], x]} / \text{. Solve[D[sol[[1]], x], D[u[x, y], x]][[1]]} \]

\[ \text{Out[33]} = \frac{\text{C[1]} \cdot \left(\text{C[1]} - \sqrt{\text{C[1]}^2 - 4 \cdot u[x, y]^2}\right)}{2 \cdot u[x, y]} \]

\[ \text{In[34]} := \text{q} = \text{D[u[x, y], y]} / \text{. Solve[D[sol[[1]], y], D[u[x, y], y]][[1]]} \]

\[ \text{Out[34]} = \frac{\text{C[1]} - \sqrt{\text{C[1]}^2 - 4 \cdot u[x, y]^2}}{2 \cdot u[x, y]} \]

\[ \text{In[35]} := \{1 + q^2 \cdot z \cdot p \cdot x \} / \text{Simplify} \]

\[ \text{Out[35]} = 0 \]

Second-Order PDEs

The general form of a linear second-order PDE is

\[
\frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f u = g.
\]
Here \( u = u(x,y) \), and \( a, b, c, d, e, f, \) and \( g \) are functions of \( x \) and \( y \) only—they do not depend on \( u \). If \( g = 0 \), the equation is said to be homogeneous.

The first three terms containing the second derivatives are called the principal part of the PDE. They determine the nature of the general solution to the equation. In fact, the coefficients of the principal part can be used to classify the PDE as follows.

The PDE is said to be **elliptic** if \( b^2 - 4ac < 0 \). The Laplace equation has \( a = 1, b = 0, \) and \( c = 1 \) and is therefore an elliptic PDE.

The PDE is said to be **hyperbolic** if \( b^2 - 4ac > 0 \). The wave equation has \( a = 1, b = 0, \) and \( c = -1 \) and is therefore a hyperbolic PDE.

The PDE is said to be **parabolic** if \( b^2 - 4ac = 0 \). The heat equation has \( a = 1, b = 0, \) and \( c = 0 \) and is therefore a parabolic PDE.

**DSolve** can find the general solution for a restricted type of homogeneous linear second-order PDEs; namely, equations of the form

\[
a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} = 0.
\]

Here \( a, b, \) and \( c \) are constants. Thus, **DSolve** assumes that the equation has constant coefficients and a vanishing non-principal part.

Following are some examples of the three basic types (elliptic, hyperbolic, and parabolic) and an explanation of their significance.

Here is the general solution for Laplace’s equation, an elliptic PDE.

\[
In[1]:= \text{LaplaceEquation} = \text{D}[u[x, y], \{x, 2\}] + \text{D}[u[x, y], \{y, 2\}] = 0;
\]

\[
In[2]:= \text{DSolve}[\text{LaplaceEquation}, u[x, y], \{x, y\}]
\]

\[
Out[2]= \{\{u[x, y] \to \text{C}[1][\text{i} x + y] + \text{C}[2][\text{-i} x + y]\}\}
\]

This general solution contains two arbitrary functions, \( \text{C}[1] \) and \( \text{C}[2] \). The arguments of these functions, \( y + \text{i}x \) and \( y - \text{i}x \), indicate that the solution is constant along the imaginary straight line \( y = \text{-i}x + \alpha \) when \( \text{C}[2] = 0 \) and along \( y = \text{i}x + \alpha \) when \( \text{C}[1] = 0 \). These straight lines are called characteristic curves of the PDE. In general, elliptic PDEs have imaginary characteristic curves.
Here is another elliptic PDE.

\[ a = 3; b = 1; c = 5; b^2 - 4 \cdot a \cdot c \]

\[ -59 \]

\[ u \rightarrow \text{Function}[[x, y], \frac{1}{6} \left( -1 + i \sqrt{59} \right) x + y] + \frac{1}{6} \left( -1 - i \sqrt{59} \right) x + y] \]

The solution is verified as follows.

\[ \text{DSolve}[[\text{eqn} /. \text{sol} // \text{Simplify}]] \]

This finds the general solution of the wave equation, a hyperbolic PDE. The constant \( c \) in the wave equation represents the speed of light and is set to 1 here for convenience.

The characteristic lines for the wave equation are \( x = k + t \) and \( x = k - t \) where \( k \) is an arbitrary constant. Hence the wave equation (or any hyperbolic PDE) has two families of real characteristic curves. If initial conditions are specified for the wave equation, the solution propagates along the characteristic lines. Also, any fixed pair of characteristic lines determine the null cone of an observer sitting at their intersection.

Here is another example of a hyperbolic PDE.

\[ a = 2; b = 7; c = -1; b^2 - 4 \cdot a \cdot c \]

\[ 57 \]

\[ u \rightarrow \text{Function}[[x, y], \frac{1}{4} \left( 7 + \sqrt{57} \right) x + y] + \frac{1}{4} \left( 7 - \sqrt{57} \right) x + y] \]
The solution can be verified as follows.

```
In[12]:= eqn /. sol // Simplify
Out[12]= {True}
```

Finally, here is an example of a parabolic PDE.

```
In[13]:= a = 3; b = 30; c = 75; b^2 - 4 a * c
Out[13]= 0

In[14]:= eqn = a * D[u[x, y], {x, 2}] + b * D[u[x, y], x, y] + c * D[u[x, y], {y, 2}] == 0;
In[15]:= sol = DSolve[eqn, u, {x, y}]
```

The equation has only one family of real characteristics, the lines $y = 5x + a$. In fact, any parabolic PDE has only a single family of real characteristics.

The solution can be verified as follows.

```
In[16]:= eqn /. sol // Simplify
Out[16]= {True}
```

The heat equation is parabolic, but it is not considered here because it has a nonvanishing non-principal part, and the algorithm used by DSolve is not applicable in this case.
Differential-Algebraic Equations (DAEs)

Introduction to Differential-Algebraic Equations (DAEs)

The systems of equations that govern certain phenomena (in electrical circuits, chemical kinetics, etc.) contain a combination of differential equations and algebraic equations. The differential equations are responsible for the dynamical evolution of the system, while the algebraic equations serve to constrain the solutions to certain manifolds. It is therefore of some interest to study the solutions of such differential-algebraic equations (DAEs).

Here is a simple example of a DAE. The first equation is an ODE for the function $x(t)$, while the second equation constrains the functions $x(t)$ and $y(t)$ to lie in a submanifold (a straight line) in the space.

$$
\text{In[1]}:= \text{dae} = \{x'[t] = y[t], x[t] + y[t] = 1\};
$$

These tutorials are restricted to linear DAEs, which are defined as systems of equations of the following type.

$$
A x'(t) + B x(t) = F
$$

Here $A$ and $B$ are matrix functions of the independent variable $t$, $F$ is a vector function of $t$, and $x(t)$ is the vector of unknowns. If the matrix $A$ is nonsingular (that is, invertible) then this is a system of ODEs. Thus, the system is a DAE if the matrix $A$ is singular.

If $F = 0$, then the system is said to be homogeneous. As for ODEs, the general solution to a DAE is composed of the general solution to the corresponding homogeneous problem and a particular solution to the inhomogeneous system.

DSolve can find the solutions to all DAEs in which the entries of the matrices $A$ and $B$ are constants. Such DAEs are said to have constant coefficients. The algorithm used by DSolve is based on decomposing both $A$ and $B$ into a nonsingular and nilpotent part. This decomposition is used to calculate a generalized inverse for $A$ and $B$, which effectively reduces the problem to solving a system of ODEs.
It is important to realize that the initial values for a DAE must be prescribed carefully to guarantee a solution for the problem. This can be seen by considering the following system of equations.

\[
\{x_1(t) + x_2'(t) = 0, x_2(t) = 0, x_1(0) = 1, x_2(0) = 0\}
\]

This gives

\[
x_2(t) = 0 \implies x_2'(t) = 0 \implies x_1(t) = 0.
\]

Hence the only solution is

\[
x_1(t) = 0 \text{ and } x_2(t) = 0.
\]

But this solution is inconsistent with the initial condition \(x_1(0) = 1\).

\texttt{DSolve} can solve DAEs with constant coefficients; see "Examples of DAEs".

\section*{Examples of DAEs}

This is a simple homogeneous DAE with constant coefficients.

\begin{verbatim}
In[1]:= eqns = \{x'[t] - y[t] == 0, x[t] + y[t] == 0\};
\end{verbatim}

This finds the general solution. It has only one arbitrary constant because the second equation in the system specifies the relationship between \(x'[t]\) and \(y[t]\).

\begin{verbatim}
In[2]:= sol = DSolve[eqns, \{x, y\}, t]
\end{verbatim}

\begin{verbatim}
Out[2]= \{(x \to \text{Function}[t, \frac{1}{4} e^{-t} C[1]], y \to \text{Function}[t, -\frac{1}{4} e^{-t} C[1]])\}
\end{verbatim}

This verifies the solution.

\begin{verbatim}
In[3]:= eqns /. sol // Simplify
Out[3]= \{\{True, True\}\}
\end{verbatim}

Here is an inhomogeneous system derived from the previous example.

\begin{verbatim}
In[4]:= eqns = \{x'[t] - y[t] == \text{Sin}[t], x[t] + y[t] == 1\};
\end{verbatim}
The general solution is composed of the general solution to the corresponding homogeneous system and a particular solution to the inhomogeneous equation.

```math
In[5] := sol = \{x[t], y[t]\} /. DSolve[eqns, \{x[t], y[t]\}, t][[1]] // Expand
```

```
Out[5] = \{1 + \frac{1}{4} e^{-t} C[1] - \frac{\cos[t]}{2}, \frac{\sin[t]}{2} - \frac{1}{4} e^{-t} C[1] + \frac{\cos[t]}{2} - \frac{\sin[t]}{2}\}
```

This solves an initial value problem for the previous equation.

```math
In[6] := eqns = \{x'[t] - y[t] == \sin[t], x[t] + y[t] == 1, x[\pi] == 1/2\};
In[7] := sol = DSolve[eqns, \{x, y\}, t]
```

```
Out[7] = \{\{x \rightarrow Function[t, \frac{1}{2} e^{-t} (2 e^t - 2 e^{-t} + e^t \cos[t] - e^t \sin[t])],
           y \rightarrow Function[t, \frac{1}{2} e^{-t} (2 e^t + e^t \cos[t] - e^t \sin[t])]\}\}
```

Here is a plot of the solution and the constraint (algebraic) condition.

```math
In[8] := Plot[\{x[t] /. sol, y[t] /. sol, (x[t] + y[t]) /. sol\}, \{t, 3, 8\}]
```

```
Out[8] = Plot graph of ...
```

In this DAE, the inhomogeneous part is quite general.

```math
In[9] := Clear[x, y, z, f, g, h, t]
In[10] := eqns = \{x[t] + y'[t] == f[t], 2 y[t] + z'[t] == g[t], 5 z[t] == h[t]\};
```

Note that there are no degrees of freedom in the solution (that is, there are no arbitrary constants) because \(z[t]\) is given algebraically, and thus \(x[t]\) and \(y[t]\) can be determined uniquely from \(z[t]\) using differentiation.

```math
In[11] := sol = DSolve[eqns, \{x, y, z\}, t]
```

```
Out[11] = \{\{x \rightarrow Function[t, f[t] - \frac{g'[t]}{2} + \frac{h'[t]}{10}],
           y \rightarrow Function[t, \frac{g[t]}{2} - \frac{h'[t]}{10}],
           z \rightarrow Function[t, \frac{h[t]}{5}\}\}\}
```

```math
In[12] := eqns /. sol // Simplify
Out[12] = \{\{True, True, True\}\}
```
In this example, the algebraic constraint is present only implicitly: all three equations contain derivatives of the unknown functions.

```
In[13]:= Clear[x1, x2, x3, t, eqns]
```

```
In[14]:= eqns = {x2[t] + 2 x3[t] + x1'[t] - 2 x3'[t] == 0,
                 -27 x1[t] - 22 x2[t] - 17 x3[t] - x1'[t] + 2 x3'[t] == 0,
                 18 x1[t] + 14 x2[t] + 10 x3[t] + 2 x1'[t] + 3 x2'[t] + 2 x3'[t] == 0};
```

The Jacobian with respect to the derivatives of the unknown functions is singular, so that it is not possible to solve for them.

```
In[15]:= A = D[eqns[[All, 1]], {{x1'[t], x2'[t], x3'[t]}}]
Out[15]= {{1, 0, -2}, {-1, 0, 2}, {2, 3, 2}}
```

```
In[16]:= Det[A]
Out[16]= 0
```

The differential-algebraic character of this problem is clear from the smaller number of arbitrary constants (two rather than three) in the general solution.

```
In[17]:= sol = DSolve[eqns, {x1, x2, x3}, t]
```

Systems of equations with higher-order derivatives are solved by reducing them to first-order systems.

Here is the general solution to a homogeneous DAE of order two with constant coefficients.

```
In[18]:= eqns = {x''[t] == y[t], x[t] + 4 y[t] == 0};
```

```
In[19]:= sol = DSolve[eqns, {x, y}, t]
```

```
In[20]:= eqns /. sol // Simplify
Out[20]= {{True, True}}
```

This inhomogeneous system of ODEs is based on the previous example.

```
In[21]:= eqns = {x''[t] == y[t], x[t] + 4 y[t] == 6 Sin[t]};
```

Differential Equation Solving with DSolve
In[22]:= sol = DSolve[eqns, {x, y}, t]

Out[22]= 

\[
\begin{align*}
\text{x} & \rightarrow \text{Function}\left[ t \rightarrow \frac{176}{125} \left( \frac{\text{C}[2] \cos\left( \frac{t}{2} \right)}{2} - \frac{\text{C}[1] \sin\left( \frac{t}{2} \right)}{2} + \frac{1}{16} \left( -8 \cos[t] - 22 \sin[t] \right) \right) + \\
& \quad \frac{16}{125} \left( \frac{\text{C}[1] \cos\left( \frac{t}{2} \right)}{2} + 2 \text{C}[2] \sin\left( \frac{t}{2} \right) + \frac{1}{8} \left( 44 \cos[t] - 4 \sin[t] \right) \right) \right], \\
\text{y} & \rightarrow \text{Function}\left[ t \rightarrow \frac{44}{125} \left( \frac{\text{C}[2] \cos\left( \frac{t}{2} \right)}{2} - \frac{\text{C}[1] \sin\left( \frac{t}{2} \right)}{2} + \frac{1}{16} \left( -8 \cos[t] - 22 \sin[t] \right) \right) - \\
& \quad \frac{4}{125} \left( \frac{\text{C}[1] \cos\left( \frac{t}{2} \right)}{2} + 2 \text{C}[2] \sin\left( \frac{t}{2} \right) + \frac{1}{8} \left( 44 \cos[t] - 4 \sin[t] \right) \right) + \frac{3 \sin[t]}{2} \right) \right]
\end{align*}
\] 

In[23]:= eqns /. sol // Simplify

Out[23]= 

\{(True, True)\}

Here is an initial value problem for the previous system of equations.

In[24]:= eqns = {x''[t] == y[t], x[t] + 4 y[t] == Sin[t], x[Pi] == 1, x'[Pi] == 0};

In[25]:= sol = DSolve[eqns, {x, y}, t]

Out[25]= 

\[
\begin{align*}
\text{x} & \rightarrow \text{Function}\left[ t \rightarrow \frac{1}{3} \left( 2 \cos\left( \frac{t}{2} \right) + 3 \sin\left( \frac{t}{2} \right) - \sin[t] \right) \right], \\
\text{y} & \rightarrow \text{Function}\left[ t \rightarrow \frac{1}{12} \left( -2 \cos\left( \frac{t}{2} \right) - 3 \sin\left( \frac{t}{2} \right) + 4 \sin[t] \right) \right]
\end{align*}
\]

Here is a plot of the solution.

In[26]:= Plot[{x[t] /. sol, y[t] /. sol}, {t, -7, 7}]

Out[26]=

Finally, here is a system with a third-order ODE. Since the coefficients are exact quantities, the computation takes some time.

In[27]:= Clear[p, q, r]

The symbolic solution of DAEs that are nonlinear or have non-constant coefficients is a difficult problem. Such systems can often be solved numerically with the Mathematica function NDSolve.
Initial and Boundary Value Problems

Introduction to Initial and Boundary Value Problems

DSolve can be used for finding the general solution to a differential equation or system of differential equations. The general solution gives information about the structure of the complete solution space for the problem. However, in practice, one is often interested only in particular solutions that satisfy some conditions related to the area of application. These conditions are usually of two types.

- The solution \( x(t) \) and/or its derivatives are required to have specific values at a single point, for example, \( x(0) = 1 \) and \( x'(0) = 2 \). Such problems are traditionally called initial value problems (IVPs) because the system is assumed to start evolving from the fixed initial point (in this case, 0).

- The solution \( x(t) \) is required to have specific values at a pair of points, for example, \( x(0) = 3 \) and \( x(1) = 5 \). These problems are known as boundary value problems (BVPs) because the points 0 and 1 are regarded as boundary points (or edges) of the domain of interest in the application.

The symbolic solution of both IVPs and BVPs requires knowledge of the general solution for the problem. The final step, in which the particular solution is obtained using the initial or boundary values, involves mostly algebraic operations, and is similar for IVPs and for BVPs.

IVPs and BVPs for linear differential equations are solved rather easily since the final algebraic step involves the solution of linear equations. However, if the underlying equations are nonlinear, the solution could have several branches, or the arbitrary constants from the general solution could occur in different arguments of transcendental functions. As a result, it is not always possible to complete the final algebraic step for nonlinear problems. Finally, if the underlying equations have piecewise (that is, discontinuous) coefficients, an IVP naturally breaks up into simpler IVPs over the regions in which the coefficients are continuous.
Linear IVPs and BVPs

To begin, consider an initial value problem for a linear first-order ODE.

This is a linear first-order ODE.

\[ \text{In[1]} := \text{linearequation} = y'[t] - 3 t \cdot y[t] = 1; \]

Notice that the general solution is a linear function of the arbitrary constant \( C[1] \).

\[ \text{In[2]} := \text{generalsolution} = \text{DSolve}[\text{linearequation}, y[t], t] \]

\[ \text{Out[2]} = \left\{ \left\{ y[t] \to e^{\frac{t^2}{2}} C[1] + e^{\frac{t^2}{2}} \sqrt{\frac{\pi}{6}} \text{Erf}\left[\sqrt{\frac{3}{2}} t\right] \right\} \right\} \]

This finds a particular solution for the initial condition \( y[0] = 4 \).

\[ \text{In[3]} := \text{particularsolution} = \text{DSolve}\{\text{linearequation}, y[0] = 4\}, y, t] \]

\[ \text{Out[3]} = \left\{ \left\{ y \to \text{Function}\left[\{t\}, \frac{1}{6} e^{\frac{t^2}{2}} \left( 24 + \sqrt{6 \pi} \text{Erf}\left[\sqrt{\frac{3}{2}} t\right] \right) \right] \right\} \right\} \]

This verifies that the solution satisfies both the equation and the initial condition.

\[ \text{In[4]} := \text{linearequation} /. \text{particularsolution}[[1]] \]

\[ \text{Out[4]} = \text{True} \]

\[ \text{In[5]} := y[0] /. \text{particularsolution}[[1]] \]

\[ \text{Out[5]} = 4 \]

Here is the solution to the same problem with the general initial condition \( y[0] = a \).

\[ \text{In[6]} := \text{particularsolution} = \text{DSolve}\{\text{linearequation}, y[0] = a\}, y, t] \]

\[ \text{Out[6]} = \left\{ \left\{ y \to \text{Function}\left[\{t\}, \frac{1}{6} e^{\frac{t^2}{2}} \left( 6 a + \sqrt{6 \pi} \text{Erf}\left[\sqrt{\frac{3}{2}} t\right] \right) \right] \right\} \right\} \]
This plots several integral curves of the equation for different values of \( a \). The plot shows that the solutions have an inflection point if the parameter \( a \) lies between \(-1\) and \(1\), while a global maximum or minimum arises for other values of \( a \).

\[
\text{In[7]:= \hspace{1em} Plot[Evaluate@Table[y[t] /. particularsolution[[1]] /. a \to i, \{i, -2, 2, 0.6\}], \{t, -1.8, 1.8\}, PlotRange \to \{-4, 4\}\].}
\]

Out[7]=

Here is the solution to a linear second-order equation with initial values prescribed for \( x[t] \) and \( x'[t] \) at \( t = 0 \).

\[
\text{In[8]:= \hspace{1em} linearsecondorderODE} = x''[t] + 5 \times x'[t] + 6 \times x[t] = 0;
\]

\[
\text{In[9]:= \hspace{1em} generalsolution} = \text{DSolve[linearsecondorderODE, x, t]}
\]

Out[9]= \[
\{\{x \to \text{Function}[\{t\}, e^{-3} C[1] + e^{-2} C[2]]\}\}\]

\[
\text{In[10]:= \hspace{1em} particularsolution} = \text{DSolve[\{linearsecondorderODE, x[0] = 1, x'[0] = 1\}, x, t]}
\]

Out[10]= \[
\{\{x \to \text{Function}[\{t\}, e^{-3} (-3 + 4 e^{t})]\}\}\]

This verifies that the solution satisfies the equation and the initial conditions.

\[
\text{In[11]:= \hspace{1em} linearsecondorderODE /. particularsolution[[1]] // Simplify}
\]


\[
\text{In[12]:= \hspace{1em} x[0] /. particularsolution[[1]]}
\]

Out[12]= 1

\[
\text{In[13]:= \hspace{1em} x'[0] /. particularsolution[[1]]}
\]

Out[13]= 1
Here is a plot of the solution.

\[ \text{In}[14]:= \text{Plot}[x[t] /. \text{particularsolution}, \{t, -1/3, 2\}] \]

\[ \text{Out}[14]= \]

To get more information about the solutions for the problem, set \( x'[0] = x_0 \).

\[ \text{In}[15]:= \text{particularsolution} = \text{DSolve}[[\text{linearsecondorderODE}, x[0] = 1, x'[0] = x_0], x, t] \]

\[ \text{Out}[15]= \{\{x \rightarrow \text{Function}[\{t\}, e^{-3 t} (-2 + 3 e^t - x_0 + e^t x_0) \}]\} \]

Here is a plot of the solutions for different initial directions. The solution approaches \(-\infty\) or \(\infty\) as \( t \to -\infty \) according to whether the value of \( x_0 \) is less than or greater than \(-2\), which is the largest root of the auxiliary equation for the ODE.

\[ \text{In}[16]:= \text{Show}[\text{GraphicsArray}[\text{Partition[}}\]
\[ \text{Table[Plot[Evaluate[x[t] /. particularsolution[[1]] /. x0 \[Rule] i], \{t, -1, 3\}, PlotStyle \rightarrow \{\text{Red}\}, PlotLabel \rightarrow \text{StringJoin["x0 = ", ToString[i]], Ticks \rightarrow \text{None}, \{i, -4, 1\}], \{3\}], \text{ImageSize \rightarrow 400}]]\]

\[ \text{Out}[16]= \]

Here is a BVP for an inhomogeneous linear second-order equation.

\[ \text{In}[17]:= \text{inhomogeneousequation} = y''[x] + y[x] = e^x; \]

\[ \text{In}[18]:= \text{generalsolution} = \text{DSolve}[\text{inhomogeneousequation}, y, x] \]

\[ \text{Out}[18]= \{\{y \rightarrow \text{Function}[\{x\}, C[1] \cos[x] + C[2] \sin[x] + \frac{1}{2} e^x (\cos[x]^2 + \sin[x]^2) \}]\} \]
In[19]:= DSolve[{inhomogeneousequation, y[0] == 1, y[1] == 1/2}, y, x]

Out[19]= 

It should be noted that, in contrast to initial value problems, there are no general existence or uniqueness theorems when boundary values are prescribed, and there may be no solution in some cases.

This problem has no solution because the term with \( C[2] \) in the general solution vanishes at both \( x = 0 \) and \( x = \pi \). Hence there are two inconsistent conditions for the parameter \( C[1] \) and the solution is an empty set.

In[20]:= DSolve[{inhomogeneousequation, y[0] == 1, y[\[Pi]] == 6}, y, x]

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

Out[20]= {}

The previous discussion of linear equations generalizes to the case of higher-order linear ODEs and linear systems of ODEs.

Here is the solution to an Initial Value Problem (IVP) for a linear ODE of order four.

In[21]:= fourthorderODE = y''''[x] + 2*y''[x] + y[x] == Cos[x];

In[22]:= sol = DSolve[{fourthorderODE, y[0] == 1, y'[0] == 6, y''[0] == 3, y'''[0] == -1}, y, x]

Out[22]= 

This verifies the solution and the initial conditions.

In[23]:= {fourthorderODE, y[0], y'[0], y''[0], y'''[0]} /. sol // Simplify

Out[23]= {{True, 1, 6, 3, -1}}

Since this is a fourth-order ODE, four independent conditions must be specified to find a particular solution for an IVP. If there is an insufficient number of conditions, the solution returned by DSolve may contain some of the arbitrary parameters, as follows.

In[24]:= DSolve[{fourthorderODE, y[0] == 1, y'[0] == 6}, y, x]

Out[24]= 


Finally, here is the solution of an IVP for a linear system of ODEs.

\[\begin{align*}
\text{In[25]:=} & \quad \text{Clear}[x, y, z, t] \\
\text{In[26]:=} & \quad \text{linearsystem} = \{x'[t] = x[t] - 4 y[t] + 1, y'[t] = 4 x[t] + y[t], z'[t] = z[t]\}; \\
\text{In[27]:=} & \quad \text{initialvalues} = \{x[0] = 2, y[0] = -1, z[0] = 1\}; \\
\text{In[28]:=} & \quad \text{sol} = \text{DSolve}[\text{Join}[\text{linearsystem}, \text{initialvalues}], \{x, y, z\}, t] \\
\text{Out[28]=} & \quad \left\{
\begin{array}{l}
x \rightarrow \text{Function} \left[ t, \frac{1}{17} \left(35 e^t \cos[4 t] - \cos[4 t]^2 + 21 e^t \sin[4 t] - \sin[4 t]^2\right)\right], \\
y \rightarrow \text{Function} \left[ t, \frac{1}{17} \left(-21 e^t \cos[4 t] + 4 \cos[4 t]^2 + 35 e^t \sin[4 t] + 4 \sin[4 t]^2\right)\right], \\
z \rightarrow \text{Function} \left[ t, e^t\right]\end{array}\right\}
\end{align*}\]

This verifies that the solution satisfies the system and the initial conditions.

\[\begin{align*}
\text{In[29]:=} & \quad \{\text{linearsystem}, \text{initialvalues}\} /\!. \text{sol[[1]]} /\!\!/ \text{Simplify} \\
\text{Out[29]=} & \quad \{\{\text{True}, \text{True}, \text{True}\}, \{\text{True}, \text{True}, \text{True}\}\}
\end{align*}\]

The solutions \(x[t]\), \(y[t]\), and \(z[t]\) are parametrized by the variable \(t\) and can be plotted separately in the plane or as a curve in space.

\[\begin{align*}
\text{In[30]:=} & \quad \text{Plot}[\text{Evaluate}[\{x[t], y[t], z[t]\} /\!. \text{sol}], \{t, -2, 1\}] \\
\text{Out[30]=} & \quad \text{Plot}\text{-graph}
\end{align*}\]
Nonlinear IVPs and BVPs

Many real-world applications require the solution of IVPs and BVPs for nonlinear ODEs. For example, consider the *logistic equation*, which occurs in population dynamics.

This is the logistic equation.

\[
\text{LogisticEquation} = y'[t] = r(1 - (y[t] / K)) \cdot y[t];
\]

The right-hand side of the equation can be expanded to a quadratic polynomial in \(y[t]\). Hence, the logistic equation is simply a Riccati equation, and its general solution can be easily found.

\[
\text{DSolve}[\text{LogisticEquation}, y, t]
\]

This sets the intrinsic growth rate \(r\) to 1 \(\div\) 2 and the saturation level \(K\) to 4 and solves the initial value problem. The warning message from \text{Solve} \ is issued while solving for the arbitrary constant \(C[1]\) from the general solution.

\[
\text{DSolve}[\{\text{LogisticEquation} \/. \{r \rightarrow (1/2), K \rightarrow 4\}, y[0] \rightarrow 1\}, y, t]
\]

\[
\text{Solve::ifun}:
\text{Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.} \quad \Rightarrow
\]

\[
\text{DSolve}[\{\text{LogisticEquation} \/. \{r \rightarrow (1/2), K \rightarrow 4\}, y[0] \rightarrow 1\}, y, t]
\]

\[
\{\{y \rightarrow \text{Function}[[t], \frac{4 \cdot e^{t/2}}{3 + e^{t/2}}]\}}\]
This solves the initial value problem for the logistic equation with symbolic parameters \( r \) and \( K \).

\[
In[4]:= \text{sol} = \text{DSolve}[[\text{LogisticEquation}, \text{y}[0] = a \ast K], \text{y}, t]
\]

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

\[
Out[4]= \left\{ \left[ \text{y} \rightarrow \text{Function}[[t], \frac{a e^{r t} K}{1 - a + a e^{r t}}] \right] \right\}
\]

This verifies that the solution satisfies the equation and the initial condition.

\[
In[5]:= \{\text{LogisticEquation}, \text{y}[0]\} \/. \text{sol}[[1]] // \text{Simplify}
\]

\[
Out[5]= \{\text{True}, a \ast K\}
\]

Here is a plot of the solution for different values of \( r \) and \( K \).

\[
In[6]:= \text{Plot}[\text{Evaluate}[\text{Table}[\text{y}[t] \/. \text{sol}[[1]], \{K \rightarrow 4, a \rightarrow i, r \rightarrow (1/3)\}, \{i, 2, 1/10, -1/3\}]], \{t, 0, 8\}, \text{PlotRange} \rightarrow \text{All}]
\]

Here is an example of an IVP for a second-order nonlinear ODE whose general solution can be obtained in explicit form.

\[
In[7]:= \text{eqn} = \text{y}''[x] - (1/2) * (\text{y}'[x]^2 / \text{y}[x]) + 1 / (2 * \text{y}[x]) = 0;
\]

\[
In[8]:= \text{sol} = \text{DSolve}[[\text{eqn}, \text{y}[0] = 1, \text{y}'[0] = 2], \text{y}, x]
\]

\[
Out[8]= \left\{ \left[ \text{y} \rightarrow \text{Function}[[x], \frac{1}{4} \left( 4 + 8 x + 3 x^2 \right) \right] \right\}
\]

This verifies that the solution satisfies the equation and the initial conditions.

\[
In[9]:= \{\text{eqn}, \text{y}[0], \text{y}'[0]\} \/. \text{sol}[[1]] // \text{Simplify}
\]

\[
Out[9]= \{\text{True}, 1, 2\}
Finally, here is a boundary value problem for a nonlinear second-order ODE. The solution is required to satisfy boundary conditions at 0 and infinity. The Solve::ifun message is generated while finding the general solution in terms of \text{JacobiSN}, the inverse of \text{EllipticF}. The DSolve::bvlim messages are given because the limit required for satisfying the condition $y'[\text{Infinity}] = 0$ cannot be calculated for either branch of the general solution. However, the solution for the boundary value problem is found using an alternative method to determine the values of the constants $C[1]$ and $C[2]$ in the general solution.

\begin{verbatim}
In[10]:= generalsolution = DSolve[{y'[x] / 2 == y[x]^3 - y[x]}, y[x], x]

Out[10]= Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. \[ \}

Out[10]= \left\{ \begin{array}{l}
\text{JacobiSN} \left[ \sqrt{x^2 + x^2 \sqrt{1 - C[1]} + 2 \times C[2]} + 2 x \sqrt{1 - C[1]} \ C[2] + C[2]^2 + \sqrt{1 - C[1]} C[2]^2, \right. \\
\frac{1 - \sqrt{1 - C[1]}}{1 + \sqrt{1 - C[1]}} - i \left. \frac{1}{1 - \sqrt{1 - C[1]}} \right] \text{JacobiSN} \left[ \sqrt{x^2 + x^2 \sqrt{1 - C[1]} + 2 \times C[2]} + 2 x \sqrt{1 - C[1]} \ C[2] + C[2]^2 + \sqrt{1 - C[1]} C[2]^2, \right. \\
\frac{1 - \sqrt{1 - C[1]}}{1 + \sqrt{1 - C[1]}} + i \left. \frac{1}{1 - \sqrt{1 - C[1]}} \right] \text{JacobiSN} \left[ \sqrt{x^2 + x^2 \sqrt{1 - C[1]} + 2 \times C[2]} + 2 x \sqrt{1 - C[1]} \ C[2] + C[2]^2 + \sqrt{1 - C[1]} C[2]^2, \right. \\
\frac{1 - \sqrt{1 - C[1]}}{1 + \sqrt{1 - C[1]}} \right\}
\end{array} \right.
\end{verbatim}

\begin{verbatim}
In[11]:= sol = DSolve[{y'[x] / 2 == y[x]^3 - y[x], y[0] == 0, y'[\text{Infinity}] == 0}, y[x], x]

Out[11]= \left\{ \begin{array}{l}
\text{Tanh} \left[ \sqrt{x^2} \right], \text{Tanh} \left[ \sqrt{x^2} \right] \right\}
\end{array} \right.
\end{verbatim}
It may not always be possible to obtain a symbolic solution to an IVP or BVP for a nonlinear equation. Numerical methods may be necessary in such cases.

**IVPs with Piecewise Coefficients**

The differential equations that arise in modern applications often have discontinuous coefficients. **DSolve** can handle a wide variety of such ODEs with piecewise coefficients. Some of the functions used in these equations are **UnitStep**, Max, Min, Sign, and Abs. These functions and combinations of them can be converted into Piecewise objects.

This converts the given expression into a Piecewise expression.

In[1]:= PiecewiseExpand[UnitStep[x] + Max[x, x^2]]

Out[1]=

```
{x^2, x < 0
1 + x, 0 <= x <= 1
1 + x^2, True
```

Here is the general solution to a first-order ODE that contains UnitStep.

In[2]:= DSolve[y'[x] == UnitStep[x], y, x]

Out[2]= 

```
{{y -> Function[{x}, C[1] + x UnitStep[x]]}}
```

Here is the solution to the same ODE with an initial condition.

In[3]:= eqn = {y'[x] == UnitStep[x], y[0] == 1};

In[4]:= sol = DSolve[eqn, y, x]

Out[4]= 

```
{{y -> Function[{x}, 1 + x UnitStep[x]]}}
```
The solution can be plotted in the usual way. Note that the solution is continuous but not differentiable at $x = 0$.

```
In[5]:= Plot[y[x] /. sol, {x, -3, 3}]
```

![Plot of the solution](image1)

This verifies the solution.

```
In[6]:= Simplify[eqn /. sol[[1]], x > 0 || x < 0]
Out[6]= {True, True}
```

Here is a piecewise ODE that has Max in its coefficients.

```
In[7]:= sol = DSolve[{y'[x] + Max[x, 1] y[x] == 0, y[0] == 1}, y[x], x]
```

```
Out[7]= {{{y[x] -> \[Piecewise] \begin{cases} e^{-x} & x \leq 1 \\ e^{x} & \frac{x^2 - 1}{2} < 0 \end{cases}}}}
```

```
In[8]:= Plot[y[x] /. sol, {x, -3, 3}]
```

![Plot of the solution](image2)

A piecewise ODE can be thought of as a collection of ODEs over disjoint intervals such that the expressions for the coefficients and the boundary conditions change from one interval to another. Thus, different intervals have different solutions, and the final solution for the ODE is obtained by patching together the solutions over the different intervals.
For this piecewise ODE, the expression for FinalSol is obtained by patching together SolFromMinusInfinityToTwo and SolFromTwoToInfinity. The boundary condition for the interval \((-\infty, 2]\) is simply \(y[0] = 1\), while the initial condition for the interval \([2, \infty)\) is \(y[2] = e^2\) (given by the final value for the solution over the first interval).

\[
\text{In}[9]:= \text{FinalSol} = \text{DSolve}[\{y'[t] = \text{If}[t \leq 2, y[t], -y[t]/2], y[0] = 1\}, y, t]
\]

\[
\text{Out}[9]= \left\{\left\{y \rightarrow \text{Function}[[t], \begin{cases}
e^t & t \leq 2 \\
\frac{1}{2} e^t & \text{True}
\end{cases}]}\right\}\right.
\]

\[
\text{In}[10]:= \text{SolFromMinusInfinityToTwo} = \text{DSolve}[\{y'[t] = y[t], y[0] = 1\}, y, t]
\]

\[
\text{Out}[10]= \left\{\left\{y \rightarrow \text{Function}[[t], e^t]\right\}\right\}
\]

\[
\text{In}[11]:= \text{SolFromTwoToInfinity} = \text{DSolve}[\{y'[t] = -y[t]/2, y[2] = e^2\}, y, t]
\]

\[
\text{Out}[11]= \left\{\left\{y \rightarrow \text{Function}[[t], e^{\frac{1}{2} t}]\right\}\right\}
\]

If there are a large number of discontinuities in a problem, it is convenient to use \text{Piecewise} directly in the formulation of the problem.

This second-order ODE contains a \text{Piecewise} term.

\[
\text{In}[12]:= \text{eqn} = \{y''[t] + y[t] = \text{Piecewise}[[\{-1, t < 0\}, \{1, t < 1\}, \{\text{Sin}[t], t < 2\}\}], y[0] = 1, y'[0] = 1\};
\]

\[
\text{In}[13]:= \text{sol} = \text{DSolve}[\text{eqn}, y, t]
\]

\[
\text{Out}[13]= \left\{\left\{y \rightarrow \text{Function}[[t], \\
\begin{cases}
-1 + 2 \cos[t] + \sin[t] & t \leq 0 \\
1 + \sin[t] & 0 < t < 1 \\
\end{cases}\right]\right\}
\]

This ODE contains the \text{Clip} function. The solutions are given in terms of Airy functions.

\[
\text{In}[14]:= \text{eqn} = \{y''[x] - \text{Clip}[x] \star y[x] = 0, y[0] = 0, y'[0] = -1\};
\]
In[15]:= DSolve[eqn, y, x]

Out[15]= {y -> Function[{x},

\[\frac{1}{6 \cos[1]^2 \sin[1]^{1/3}}\]
\[\left(3 x 3^{1/3} \text{AiryAi}[-1] \cos[1] \cos[x] \Gamma[\frac{1}{3}] - 35/6 \text{AiryBi}[-1] \cos[1] \cos[x] \Gamma[\frac{1}{3}] +
3 x 3^{1/3} \text{AiryAiPrime}[-1] \cos[x] \Gamma[\frac{1}{3}] \sin[1] - 35/6 \text{AiryBiPrime}[-1] \cos[x] \Gamma[\frac{1}{3}] \sin[1] -
3 x 3^{1/3} \text{AiryBiPrime}[-1] \cos[1] \sin[x] - 3 x 3^{1/3} \text{AiryAi}[-1] \Gamma[\frac{1}{3}] \sin[1] \sin[x] +
3 x 3^{1/3} \text{AiryBi}[-1] \Gamma[\frac{1}{3}] \sin[1] \sin[x] -
\frac{1}{6} \left(3 x 3^{1/3} \text{AiryAi}[x] \Gamma[\frac{1}{3}] - 35/6 \text{AiryBi}[x] \Gamma[\frac{1}{3}]\right)\right) \quad -1 <
-\frac{1}{4 x 3^{1/3}} e^{-1 x} \left[-3 e^{2 x} \text{AiryAi}[1] - 3 e^{2 x} \text{AiryAi}[1] +
3 e^{2 x} \text{AiryAiPrime}[1] - 3 e^{2 x} \text{AiryAiPrime}[1] + \sqrt{3} e^{2 x} \text{AiryBi}[1] +
\sqrt{3} e^{2 x} \text{AiryBi}[1] - \sqrt{3} e^{2 x} \text{AiryBiPrime}[1] + \sqrt{3} e^{2 x} \text{AiryBiPrime}[1] \right) \Gamma[\frac{1}{3}]\right]}

True}
Working with DSolve—A User’s Guide

Introduction to Working with DSolve

The aim of these tutorials is to provide a self-contained working guide for solving different types of problems with DSolve.

The first step in using DSolve is to set up the problem correctly. The next step is to use DSolve to get an expression for the solution. Once the solution has been found, it can be verified using symbolic or numerical techniques, or it can be plotted using a Mathematica function such as Plot, Plot3D, or ContourPlot. Plots often reveal information about the solution that might not be evident from its closed-form expression.

If no boundary conditions are specified for a problem, the output from DSolve is some form of a general solution containing arbitrary parameters. The GeneratedParameters option can be used to label these arbitrary parameters.

In many applications, differential equations contain symbolic parameters, such as the rate of growth in the logistic equation. A differential equation can also contain inexact quantities, such as machine numbers arising from previous calculations. Both symbolic parameters and inexact quantities are allowed by DSolve, but it is good to be aware of their presence and interpret the solution correctly.

When DSolve makes any assumptions or encounters difficulty during a calculation, it issues a warning message outlining the problem. These messages can usually be ignored, but sometimes they point to serious limitations in the answer given for the problem.

It is helpful to analyze the statement of the problem for possible ambiguities—in other words, to make sure that the problem is well posed—so that meaningful answers can be obtained from DSolve.
Setting Up the Problem

The first argument given to \texttt{DSolve} is the differential equation, the second argument is the unknown function, and the last argument identifies the independent variable.

Here is the input for solving for a first-order linear ODE using \texttt{DSolve}. The variable \texttt{sol} identifies the solution for use in further work.

\begin{verbatim}
In[1]:= sol = DSolve[y'[x] + 5 y[x] == 1, y[x], x]

Out[1]= \{(y[x] -> \(\frac{1}{5} + e^{-5x} C[1]\))\}
\end{verbatim}

The output of \texttt{DSolve} is a list of solutions for the differential equation. The extra list is required since some equations have multiple solutions. Here, since the equation is of order 1 and is linear, there is only one solution: \(y[x] \rightarrow \frac{1}{5} + e^{-5x} C[1]\). The solution has an undetermined constant \(C[1]\) because no initial condition was specified. The solution can be extracted from the list of solutions using a part specification.

This extracts the solution.

\begin{verbatim}
In[2]:= m = sol[[1]]

Out[2]= \{(y[x] -> \(\frac{1}{5} + e^{-5x} C[1]\))\}
\end{verbatim}

This form of the solution is useful for finding \(y[x]\) itself, but not for finding derivatives of \(y[x]\) or the value of \(y[x]\) at a point.

This shows the value of \(y[x]\) given by the solution.

\begin{verbatim}
In[3]:= y[x] /. m

Out[3]= \(\frac{1}{5} + e^{-5x} C[1]\)
\end{verbatim}

The solution does not apply to \(y'[x]\) or \(y[1]\) because the solution is a rule for \(y[x]\) only.

\begin{verbatim}
In[4]:= y'[x] /. m

Out[4]= y'[x]

In[5]:= y[1] /. m

\end{verbatim}
If the solution will be used in further work, it is best to specify the unknown function using \( y \) rather than \( y(x) \). This gives the solution using pure functions of the type `Function[x, expr].`

Here, the unknown function is specified as \( y \). The solution is a pure function.

\[
\text{In}[6]:= \quad \text{sol} = \text{DSolve}[y'[x] + 5 y[x] = 1, y, x]
\]

\[
\text{Out}[6]= \quad \{\{y \rightarrow \text{Function}[\{x\}, \frac{1}{5} + \text{e}^{-5x} C[1]\}\}\}
\]

When the solution is in the form of pure functions, expressions can be found for derivatives of \( y \) and for the values of \( y \) at specific points.

This gives expressions for \( y(x) \), \( y'(x) \), and \( y[1] \).

\[
\text{In}[7]:= \quad m = \text{sol}[[1]]
\]

\[
\text{Out}[7]= \quad \{y \rightarrow \text{Function}[\{x\}, \frac{1}{5} + \text{e}^{-5x} C[1]\}\}
\]

\[
\text{In}[8]:= \quad y[x] \/. m
\]

\[
\text{Out}[8]= \quad \frac{1}{5} + \text{e}^{-5x} C[1]
\]

\[
\text{In}[9]:= \quad y'[x] \/. m
\]

\[
\text{Out}[9]= \quad -5 \text{e}^{-5x} C[1]
\]

\[
\text{In}[10]:= \quad y[1] \/. m
\]

\[
\text{Out}[10]= \quad \frac{1}{5} + \frac{C[1]}{\text{e}^5}
\]

When a problem has multiple solutions, you can pick out individual solutions from the solution list or you can work directly with the list.

This solves a nonlinear first-order equation. There are two solutions.

\[
\text{In}[11]:= \quad \text{sol} = \text{DSolve}[y'[x]^2 = x + 11, y, x]
\]

\[
\text{Out}[11]= \quad \{\{y \rightarrow \text{Function}[\{x\}, \frac{2}{3} (11 + x)^{3/2} + C[1]\]\imators, \{y \rightarrow \text{Function}[\{x\}, \frac{2}{3} (11 + x)^{3/2} + C[1]\]\imators\}\}
\]

The solutions can be extracted using part specifications.

\[
\text{In}[12]:= \quad y[x] \/. \text{sol}[[1]]
\]

\[
\text{Out}[12]= \quad \frac{2}{3} (11 + x)^{3/2} + C[1]
\]
DSolve

\[ y[x] \rightarrow \frac{2}{3} (11 + x)^{3/2} + C[1] \]

This returns a list of both expressions.

\[ y[x] \rightarrow \frac{2}{3} (11 + x)^{3/2} + C[1], \frac{2}{3} (11 + x)^{3/2} + C[1] \]

To solve a system of equations, the first argument to DSolve must be a list of the equations and the second argument must be a list of the unknown functions.

Here is an example of a system of first-order linear equations with three unknowns. Because this system is linear, there is only one solution.

\[ \text{sol} = \text{DSolve}[ \]
\[ (x'[t] = y[t] + z[t], y'[t] + z[t] - x[t] = 0, z'[t] + y[t] = x[t]), \{x, y, z\}, t] \]

Each solution to the system is a list of replacement rules for the unknown functions. The expressions for the unknown functions can be extracted as in previous examples.

\[ \{x[t] \rightarrow \text{Function}[t, \frac{1}{3} e^{-2t} (1 + 2 e^{3t}) C[1] + \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[2] + \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[3])], \]
\[ y[t] \rightarrow \text{Function}[t, \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[1] + \frac{1}{3} e^{-2t} (1 + 2 e^{3t}) C[2] - \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[3])], \]
\[ z[t] \rightarrow \text{Function}[t, \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[1] - \frac{1}{3} e^{-2t} (-1 + e^{3t}) C[2] + \frac{1}{3} e^{-2t} (1 + 2 e^{3t}) C[3])] \}

If initial conditions are prescribed for the problem, some or all of the undetermined constants can be eliminated.

Here the value of the unknown function and its derivative are both prescribed at the initial point.

\[ \text{DSolve}[\{y''[x] + y[x] = 5, y[0] = 1, y'[0] = 7\}, y[x], x] \]

\[ \{y[x] \rightarrow 5 - 4 \cos[x] + 7 \sin[x]\} \]
If only the initial value is specified, then the solution still contains an arbitrary constant.

\[
\text{In[18]} := \text{DSolve}[\{y''[x] + y[x] == 5, y[0] == 1\}, y[x], x]
\]

\[
\text{Out[18]} = \{\{y[x] \to 5 - 4 \cos[x] + C[2] \sin[x]\}\}
\]

For a partial differential equation, the third argument to DSolve is a list of the independent variables for the equation.

This solves a PDE with independent variables \(x\) and \(y\). \(C[1]\) represents an arbitrary function of \(y + \cos[y[x]]\).

\[
\text{In[19]} := \text{DSolve}[D[u'[x, y], x] + \sin[x] * D[u[x, y], y] == 8, u, \{x, y\}]
\]

\[
\text{Out[19]} = \{\{u \to \text{Function}[\{x, y\}, 8 x + C[1] y + \cos[x]]\}\}
\]

A differential-algebraic equation is specified in the same way as a system of ordinary differential equations.

Here is an example of a DAE with an initial condition.

\[
\text{In[20]} := \text{DSolve}[\{x'[t] + y[t] == \sin[t], x[t] + y[t] == 1, x[0] == 4\}, \{x, y\}, t]
\]

\[
\text{Out[20]} = \{\{x \to \text{Function}[\{t\}, \frac{1}{2} (2 + 7 e^t - \cos[t] - \sin[t])], y \to \text{Function}[\{t\}, \frac{1}{2} (-7 e^t + \cos[t] + \sin[t])]\}\}
\]

Note that it is not always possible to give the solutions for a problem in explicit form. In this case, the solution is given using an unevaluated Solve object or using InverseFunction.

The solution to this equation is not available explicitly. The output represents an implicit solution.

\[
\text{In[21]} := \text{sol} = \text{DSolve}[y'[x] + y[x]^3 + y[x]^2 == 1, y[x], x]
\]

\[
\text{Solve::tdep}: \text{The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\[
\text{Solve::tdep}: \text{The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\[
\text{Out[21]} = \text{Solve}[\text{RootSum}[-1 + m1^2 + m1^3 \&, \frac{\text{Log}[-n1 + y[x]]}{2 n1 + 3 m1^2}\&] = -x + C[1], y[x]]
\]

The solution can be extracted as usual with a part specification.

\[
\text{In[22]} := \text{sol[[1]]}
\]

\[
\text{Out[22]} = \text{RootSum}[-1 + m1^2 + m1^3 \&, \frac{\text{Log}[-n1 + y[x]]}{2 n1 + 3 m1^2}\&] = -x + C[1]
\]
The solutions to this equation are given as `InverseFunction` objects.

In[23]:= sol = DSolve[Derivative[2][y][x] + y[x] * Derivative[1][y][x]^4 == 0, y, x]

Solve::tdep:
The equations appear to involve the variables to be solved for in an essentially non-algebraic way. 

Solve::tdep:
The equations appear to involve the variables to be solved for in an essentially non-algebraic way. 

Out[23]= 

Each solution can be rewritten as an implicit equation by eliminating the `InverseFunction` object as follows.

In[24]:= soly = y[x] /. sol[[1]]

Out[24]= InverseFunction[C[1] Log[1 + \sqrt{-2 C[1] + \Pi^2}] - \frac{1}{2} \Pi1 \sqrt{-2 C[1] + \Pi^2} &] [x + C[2]]

In[25]:= implicitequation = (Head[soly][[1]][y[x]] == soly[[1]])


**Verification of the Solution**

The solution given by `DSolve` can be verified using various methods. The easiest method involves substituting the solution back into the equation. If the result is `True`, the solution is valid.

In this simple example, the solution is verified by substitution. Note that the first argument to `DSolve` is assigned to `eqn` for convenience in later work.

In[1]:= eqn = y'[x] = x;

In[2]:= sol = DSolve[eqn, y, x]

Out[2]= 

In[3]:= eqn /. sol

Out[3]= {True}
In this example, the equation and an initial condition are verified by substitution.

\[
\text{In}[4]:= \text{eqn} = \{y'[x] = x, y[0] = 1\};
\]

\[
\text{In}[5]:= \text{sol} = \text{DSolve}[\text{eqn}, y, x]
\]

\[
\text{Out}[5]= \{\{y \rightarrow \text{Function}[\{x\}, \frac{1}{2} (2 + x^2)]\}\}
\]

\[
\text{In}[6]:= \text{eqn} /. \text{sol}
\]

\[
\text{Out}[6]= \{(\text{True}, \text{True})\}
\]

Sometimes the result of the substitution is more complicated than \text{True} or \text{False}. Such examples can be verified by using \text{Simplify} to simplify the result of the substitution. If the simplified result is \text{True}, the solution is valid.

Here is the general solution for a second-order inhomogeneous equation.

\[
\text{In}[7]:= \text{eqn} = y''[x] + 5 y'[x] + 6 y[x] = 1;
\]

\[
\text{In}[8]:= \text{sol} = \text{DSolve}[\text{eqn}, y, x]
\]

\[
\text{Out}[8]= \{\{y \rightarrow \text{Function}[\{x\}, \frac{1}{6} + e^{-3 x} C[1] + e^{-2 x} C[2]]\}\}
\]

This substitutes the solution back into the equation.

\[
\text{In}[9]:= \text{eqn} /. \text{sol}
\]

\[
\text{Out}[9]= \left\{9 e^{-3 x} C[1] + 4 e^{-2 x} C[2] + 5 \left(-3 e^{-3 x} C[1] - 2 e^{-2 x} C[2]\right) + 6 \left(\frac{1}{6} + e^{-3 x} C[1] + e^{-2 x} C[2]\right) = 1\right\}
\]

The solution can be verified using \text{Simplify}.

\[
\text{In}[10]:= \text{Simplify}[\text{eqn} /. \text{sol}]
\]

\[
\text{Out}[10]= \{(\text{True})\}
\]

Here is a linear PDE whose solution can be verified using \text{Simplify}.

\[
\text{In}[11]:= \text{PDE} = \text{D}[u[x, y], x] + \text{Sin}[x] \ast \text{D}[u[x, y], y] = x^2;
\]

\[
\text{In}[12]:= \text{sol} = \text{DSolve}[\text{PDE}, u, \{x, y\}]
\]

\[
\text{Out}[12]= \{\{u \rightarrow \text{Function}[\{x, y\}, \frac{1}{3} (x^3 + 3 C[1] [y \cdot \text{Cos}[x]])]\}\}
\]

\[
\text{In}[13]:= \text{Simplify}[\text{PDE} /. \text{sol}]
\]

\[
\text{Out}[13]= \{(\text{True})\}
\]
If the equation involves special functions, it may be necessary to use \texttt{FullSimplify} to verify the solution.

Here is an example of this type involving Bessel's functions.

\begin{verbatim}
In[14]:= eqn = x*y''[x] + y'[x] - y[x] = 1;
In[15]:= sol = DSolve[eqn, y, x]
Out[15]= \{\{y \rightarrow Function[{x}, -1 + BesselI[0, 2 \sqrt{x}] C[1] + 2 BesselK[0, 2 \sqrt{x}] C[2]]\}\}
In[16]:= FullSimplify[eqn /. sol]
Out[16]= \{True\}
\end{verbatim}

If the solution is large or if \texttt{Simplify} and \texttt{FullSimplify} do not succeed in verifying the solution, a numerical check can be made by using \texttt{RandomReal} or \texttt{RandomComplex} to generate values for all the variables and parameters in the problem. It is advisable in such cases to repeat the check with several sets of random values.

Here is an example where numerical verification is useful.

\begin{verbatim}
In[17]:= Clear[a, y, x, r]
In[18]:= eqn = y''[x] - (a*x^6 + x^2)*y[x];
In[19]:= sol = DSolve[eqn := 0, y, x]
Out[19]= \{\{y \rightarrow Function[{x}, \frac{2^{3/8} e^{\sqrt{\frac{a}{x^3}}} \bigl(x^4\bigr)^{3/8} C[1] \text{HypergeometricU}\left[-1.3 \frac{\sqrt{a}}{8 \sqrt{2}}, \frac{3}{4}, -1.2 \frac{\sqrt{a}}{x^4}\right] + \frac{2^{3/8} e^{\sqrt{\frac{a}{x^3}}} \bigl(x^4\bigr)^{3/8} C[2] \text{LaguerreL}\left[-1.3 \frac{\sqrt{a}}{8 \sqrt{2}}, -\frac{1}{4}, -1.2 \frac{\sqrt{a}}{x^4}\right]}{x^{3/2}} \}\}\}
In[20]:= Union[Flatten@Table[Chop[eqn /. sol /. \{x \rightarrow RandomReal[], a \rightarrow RandomReal[], C[1] \rightarrow RandomReal[], C[2] \rightarrow RandomReal[]\}], \{i, 1, 10\}]]
Out[20]= \{0\}
\end{verbatim}

Although numerical checks cannot verify a solution with certainty, more rigorous checks can be made by using higher precision or by allowing the variables to take complex values.

This verifies the previous solution with higher precision.

\begin{verbatim}
In[21]:= Chop[eqn /. sol /. \{x \rightarrow RandomReal[\{1, 2\}, WorkingPrecision \rightarrow 20], a \rightarrow RandomReal[\{1, 2\}, WorkingPrecision \rightarrow 20], C[1] \rightarrow RandomReal[\{1, 2\}, WorkingPrecision \rightarrow 20], C[2] \rightarrow RandomReal[\{1, 2\}, WorkingPrecision \rightarrow 20]\}]
Out[21]= \{0\}
\end{verbatim}
This uses complex random values to verify the previous solution.

\[ r := \text{RandomComplex[]} \]

\[ \text{Chop}[\text{N[eqn /. sol /. \{x \to r, a \to r, C[1] \to r, C[2] \to r\}}]] \]

The previous methods are of use only when the solution is available in explicit form. The final example shows how to verify the solution of a first-order ODE when it is given in implicit form.

This solves a first-order ODE.

\[ \text{eqn} = y'[x] + 2 \times x \times y[x]^2 + y[x]^3; \]

\[ \text{sol} = \text{DSolve}[\text{eqn} == 0, y, x] \]

\[ \text{Solve::tdep} : \text{The equations appear to involve the variables to be solved for in an essentially non-algebraic way.} \]

\[ \text{Out[25]} = \text{Solve}\left[ \frac{-x \text{AiryAi}\left[x^2 - \frac{1}{y[x]}\right] + \text{AiryAiPrime}\left[x^2 - \frac{1}{y[x]}\right]}{-x \text{AiryBi}\left[x^2 - \frac{1}{y[x]}\right] + \text{AiryBiPrime}\left[x^2 - \frac{1}{y[x]}\right]} + C[1] == 0, y[x] \right] \]

\[ \text{Out[26]} = \text{sol[[1]]} \]

\[ \text{Out[26]} = \frac{-x \text{AiryAi}\left[x^2 - \frac{1}{y[x]}\right] + \text{AiryAiPrime}\left[x^2 - \frac{1}{y[x]}\right]}{-x \text{AiryBi}\left[x^2 - \frac{1}{y[x]}\right] + \text{AiryBiPrime}\left[x^2 - \frac{1}{y[x]}\right]} + C[1] == 0 \]

This verifies the solution by simplifying its derivative.

\[ \text{Simplify[\text{D[sol[[1]], x, y'[x]]}]} \]

\[ \text{Out[27]} = \{y'[x] \to -y[x]^2 (2 \times y[x]) \} \]
Plotting the Solution

A plot of the solution given by DSolve can give useful information about the nature of the solution, for instance, whether it is oscillatory in nature. It can also serve as a means of solution verification if the shape of the graph is known from theory or from plotting the vector field associated with the differential equation. A few examples that use different Mathematica graphics functions follow.

Here is the general solution to a linear first-order equation.

\[
\text{In}[1] := \text{sol} = \text{DSolve}[\frac{\text{y}'[x]}{x} + x \cdot \text{y}[x] = \cos[x^2], \text{y}, x]
\]

\[
\text{Out}[1] := \text{\text{\text{y} \rightarrow Function[\{x\}, e^{x^2} \cdot C[1] + \frac{1}{2} \cdot e^{x^2} \cdot \sqrt{\frac{\pi}{10}} \cdot \left(\sqrt{1 + 2i \cdot \text{Erfi}\left[\frac{1}{2} \cdot x\right]} + \sqrt{1 - 2i \cdot \text{Erfi}\left[\frac{1}{2} + i \cdot x\right]\right)\]}}\]

The solution can be plotted for specific values of the constant C[1] using Plot. The use of Evaluate reduces the time taken by Plot and can also help in cases where the solution has discontinuities.

\[
\text{In}[2] := \text{Plot[Evaluate[\text{y}[x] /. \text{sol} /. \{C[1] \rightarrow 1\}], \{x, -7, 7\}, \text{PlotRange} \rightarrow \text{All}]}
\]

Here is the plot for a linear second-order ODE with initial values prescribed at 0.

\[
\text{In}[3] := \text{sol} = \text{DSolve}[\frac{\text{y}''[x]}{\text{y}[x]} = -4 \cdot \exp[-x/4], \text{y}[0] = 1, \text{y}'[0] = 1/2, \text{y}, x]
\]

\[
\text{Out}[3] := \text{\text{\text{y} \rightarrow Function[\{x\}, \left(BesselJ[0, 16 \cdot \sqrt{-x^4}] \cdot \text{BesselY}[0, 16] - BesselJ[0, 16] \cdot \text{BesselY}[0, 16 \cdot \sqrt{-x^4}]\right) + \frac{4 \cdot BesselJ[1, 16] \cdot \text{BesselY}[0, 16 \cdot \sqrt{-x^4}] - 4 \cdot BesselJ[0, 16 \cdot \sqrt{-x^4}] \cdot \text{BesselY}[1, 16]}{4 \cdot (4 \cdot BesselJ[1, 16] \cdot \text{BesselY}[0, 16] - BesselJ[0, 16] \cdot \text{BesselY}[1, 16])}]\]

\[
\text{In}[4] := \text{Plot[Evaluate[\text{y}[x] /. \text{sol}], \{x, 0, 30\}]}
\]

\[
\text{Out}[4] := \text{\text{\text{y} \rightarrow Function[\{x\}, BesselJ[0, 16 \cdot \sqrt{-x^4}] \cdot \text{BesselY}[0, 16] - BesselJ[0, 16] \cdot \text{BesselY}[0, 16 \cdot \sqrt{-x^4}] + \frac{4 \cdot BesselJ[1, 16] \cdot \text{BesselY}[0, 16 \cdot \sqrt{-x^4}] - 4 \cdot BesselJ[0, 16 \cdot \sqrt{-x^4}] \cdot \text{BesselY}[1, 16]}{4 \cdot (4 \cdot BesselJ[1, 16] \cdot \text{BesselY}[0, 16] - BesselJ[0, 16] \cdot \text{BesselY}[1, 16])}]\]
This nonlinear equation has two solutions that can be plotted on the same graph.

\[
\text{In}[5]:= \text{sol} = \text{DSolve}[\{y'[x]^2 = x - x^3, y[0] = 1\}, y, x]
\]

\[
\text{Out}[5]= \left\{\begin{array}{l}
\left\{y \rightarrow \text{Function}[\{x\}, \frac{1}{5 \sqrt{x} \sqrt{1 - x^2}} \left(5 \sqrt{x} \sqrt{1 - x^2} + 2 x^{3/2} \sqrt{1 - x^2} \sqrt{-x (-1 + x^2)} + 4 \sqrt{-x (-1 + x^2)} \text{EllipticE}[\text{ArcSin}\left[\sqrt{x}\right], -1] - 4 \sqrt{-x (-1 + x^2)} \text{EllipticF}[\text{ArcSin}\left[\sqrt{x}\right], -1]\right)\},
\left\{y \rightarrow \text{Function}[\{x\}, \frac{1}{5 \sqrt{x} \sqrt{1 - x^2}} \left(5 \sqrt{x} \sqrt{1 - x^2} - 2 x^{3/2} \sqrt{1 - x^2} \sqrt{-x (-1 + x^2)} - 4 \sqrt{-x (-1 + x^2)} \text{EllipticE}[\text{ArcSin}\left[\sqrt{x}\right], -1] + 4 \sqrt{-x (-1 + x^2)} \text{EllipticF}[\text{ArcSin}\left[\sqrt{x}\right], -1]\right)\}
\end{array}\right.
\]

\[
\text{In}[6]:= \text{Plot}[\text{Evaluate}[y[x] /. \text{sol}], \{x, 1/100, 9/10\}]
\]

The solution to this Abel ODE is given in implicit form.

\[
\text{In}[7]:= \text{sol} = \text{DSolve}[y'[x] = -5 y[x]^2 - \frac{y[x]^3}{x}, y[x], x]
\]

\[
\text{Out}[7]= \text{Solve}\left\{-5 x = \frac{2 e^\frac{1}{2} \left(\frac{5 x, y[x]}{2}ight)^2}{\sqrt{2} \pi \text{Erfi}\left[\frac{-5 x, y[x]}{\sqrt{2}}\right]} \left[2 C[1] + \sqrt{2} \pi \text{Erfi}\left[\frac{-5 x, y[x]}{\sqrt{2}}\right]\right], y[x]\right\}
\]

A contour plot can be used to study the nature of the solution. Each contour line corresponds to a solution to the ODE for a fixed value of \(C[1]\).

\[
\text{In}[8]:= \text{expr} = \text{C}[1] /. \text{Solve}[\text{sol}[1], \text{C}[1]()[1]] /. \{y[x] \rightarrow y\}
\]

\[
\text{Out}[8]= -\frac{2 e^\frac{1}{2} \left(\frac{5 x, y[x]}{2}\right)^2 + 5 \sqrt{2} \pi x \text{Erfi}\left[\frac{-5 x, y[x]}{\sqrt{2}}\right]}{10 x}
\]
Here is the plot of the solutions to a system of two linear ODEs. The `WorkingPrecision` option in `Plot` is required because the solution is fairly complicated.

```math
\text{In}[10]:= \text{sol} = \text{DSolve}\left[\{x'[t] + t \cdot y[t] = 0, 2 \cdot y'[t] - 3 \cdot x[t] = 0, x[0] = 1, y[0] = 3\}, \{x, y\}, t\right]
```

```math
\text{Out}[10]= \left\{\begin{align*}
x & \Rightarrow \text{Function}\left[t, \frac{1}{6} \left(-3 \cdot 3^{1/3} \text{AiryAiPrime}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{1}{3}\right] + 3^{5/6} \text{AiryBiPrime}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{1}{3}\right] + 9 \left(-1\right)^{1/3} 2^{2/3} \\
& \quad \text{AiryAiPrime}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{2}{3}\right] + 3 \left(-1\right)^{1/3} 2^{2/3} \sqrt{3} \text{AiryBiPrime}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{2}{3}\right]\right)\right), \\
y & \Rightarrow \text{Function}\left[t, \frac{1}{4} \left(3 \left(-1\right)^{2/3} 2^{1/3} \text{AiryAi}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{1}{3}\right] - \left(-1\right)^{2/3} 2^{1/3} \sqrt{3} \text{AiryBi}\left[\left(-\frac{3}{2}\right)^{1/3} t\right]\right) \Gamma\left[\frac{1}{3}\right] + 6 \times 3^{2/3} \text{AiryAi}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{2}{3}\right] + 6 \times 3^{1/6} \text{AiryBi}\left[\left(-\frac{3}{2}\right)^{1/3} t\right] \Gamma\left[\frac{2}{3}\right]\right)\right]\right\}
```

```math
\text{In}[11]:= \text{Plot}\left[\text{Evaluate}\left[\{x[t], y[t]\} /. \text{sol}\right], \{t, -1, 10\}, \text{WorkingPrecision} \rightarrow 20\right]
```

\text{Out}[11]= [Graph showing plots of the functions over the time range -1 to 10]
The ParametricPlot function can be used to trace the solution curve \( \{x[t], y[t]\} \) in the plane.

```math
In[12]:= ParametricPlot[Evaluate[{x[t], y[t]} /. sol], {t, -1, 7}, WorkingPrecision \to 20]
```

```math
Out[12]=
```

Here is the plot for the solution to a DAE.

```math
In[13]:= sol = DSolve[{x''[t] + 3*y[t] = UnitStep[t],
x[t] - 5*y[t] = t^2, x[0] = 1, x'[0] = 1}, \{x, y\}, t]
```

```math
Out[13]= \{\{x \to Function\[t], \frac{1}{96} \left( -430 \ t + 177 \ t^2 + 526 \ Cos\left[ \sqrt{\frac{3}{5}} \ t \right] +
62 \sqrt{15} \ Sin\left[ \sqrt{\frac{3}{5}} \ t \ UnitStep[-t] \right] - 160 UnitStep[-t] + 160 Cos\left[ \sqrt{\frac{3}{5}} \ t UnitStep[-t] \right] \right),
y \to Function\[t], \frac{1}{480} \left( -430 \ t + 81 \ t^2 + 526 \ Cos\left[ \sqrt{\frac{3}{5}} \ t \right] + 62 \sqrt{15} \ Sin\left[ \sqrt{\frac{3}{5}} \ t \right] -
160 UnitStep[-t] + 160 Cos\left[ \sqrt{\frac{3}{5}} \ t UnitStep[-t] \right] \right) \}\}
```

```math
In[14]:= Plot[Evaluate[{x[t], y[t], x[t] - 5*y[t]} /. sol], {t, -2, 2}]
```

```math
Out[14]=
```
Here is the general solution to a linear PDE.

\[ \text{In}[15]:= \quad \text{sol} = \text{DSolve}[D[u[x, y], x] + x^2 \cdot D[u[x, y], y] = \exp[x], u, \{x, y\}] \]

\[ \text{Out}[15]= \quad \{\{u \rightarrow \text{Function}\{x, y\}, e^x \cdot C[1]\left[\frac{1}{3} (-x^3 + 3 y)\right]\}\}\]

Here is a plot of the solution surface for a particular choice of the arbitrary function \(C[1]\).

\[ \text{In}[16]:= \quad \text{Plot3D}[\text{Evaluate}[u[x, y] / . \text{sol} / . \{C[1][t_] \rightarrow \sin[3 \cdot t]\}[[1]]], \{x, -2, 2\}, \{y, -2, 2\}, \text{Mesh} \rightarrow \text{False}] \]

\[ \text{Out}[16]= \]

### The `GeneratedParameters` Option

The general solution to a differential equation contains undetermined coefficients that are labeled \(C[1], C[2]\), and so on.

This example has one undetermined parameter, \(C[1]\).

\[ \text{In}[1]:= \quad \text{DSolve}[y'[x] + y[x] = 1, y[x], x] \]

\[ \text{Out}[1]= \quad \{\{y[x] \rightarrow 1 + e^x C[1]\}\}\]

To change the name of the undetermined parameter, use the `GeneratedParameters` option.

This changes the name of the undetermined coefficient to \(P[1]\).

\[ \text{In}[2]:= \quad \text{DSolve}[y'[x] + y[x] = 1, y[x], x, \text{GeneratedParameters} \rightarrow P] \]

\[ \text{Out}[2]= \quad \{\{y[x] \rightarrow 1 + e^x P[1]\}\}\]

The parameter \(C\) should be thought of as a pure function that acts on a set of indices to generate different constants \(C[i]\).
This shows the behavior of \( C \).

\[
\text{In}[3]:= \quad \text{parameter} = C[#] &;
\]

\[
\text{In}[4]:= \quad \text{indexset} = \{1, 2, 3, 4\};
\]

\[
\text{In}[5]:= \quad \text{parameter}[\text{indexset}[\{1\}] = C[1]
\]

\[
\text{In}[6]:= \quad \text{parameter}[\text{indexset}[\{3\}] = C[3]
\]

Internally, the use of a pure function allows \text{DSolve} to increment the argument \( i \) in \( C[i] \) correctly for higher-order ODEs and systems of ODEs.

\[
\text{GeneratedParameters} \text{ can be specified using a pure function.}
\]

\[
\text{In}[7]:= \quad \text{DSolve}[y''[x] + y[x] = 1, y[x], x, \text{GeneratedParameters} \to (\text{const}[#] &)]
\]

\[
\text{Out}[7]= \{\{y[x] \to 1 + \text{const}[1] \text{Cos}[x] + \text{const}[2] \text{Sin}[x]\}\}
\]

Using a pure function is particularly useful if you want to begin indexing the parameters at any value other than 1 (the default).

\[
\text{This uses a pure function to label the parameters in the previous example } \text{const}[2] \text{ and } \text{const}[3].
\]

\[
\text{In}[8]:= \quad \text{DSolve}[y''[x] + y[x] = 1, y[x], x, \text{GeneratedParameters} \to (\text{const}[1 + #] &)]
\]

\[
\text{Out}[8]= \{\{y[x] \to 1 + \text{const}[1] \text{Cos}[x] + \text{const}[2] \text{Sin}[x]\}\}
\]

It is sometimes useful to display the solution using subscripts or other styles for the parameter indices.

\[
\text{Here, the parameters are named using subscripted variables.}
\]

\[
\text{In}[9]:= \quad \text{DSolve}[y''[x] + y[x] = 1, y[x], x, \text{GeneratedParameters} \to (c_{#1} &)]
\]

\[
\text{Out}[9]= \{\{y[x] \to 1 + \text{Cos}[x] c_1 + \text{Sin}[x] c_2\}\}
\]

Finally, with \text{Module} variables, you can get names for the parameters that are unique across different invocations of \text{DSolve}.

\[
\text{Here the same } \text{DSolve} \text{ call generates different parameter names.}
\]

\[
\text{In}[10]:= \quad \text{DSolve}[y''[x] + y[x] = 1, y[x], x, \text{GeneratedParameters} \to \text{Module}[(c, C[#] &)]
\]

\[
\text{Out}[10]= \{\{y[x] \to 1 + \text{Cos}[x] c_{102}[1] + c_{102}[2] \text{Sin}[x]\}\}
\]
Symbolic Parameters and Inexact Quantities

The differential equations that arise in practice are of two types.

- Equations in which the only variables are the independent and dependent variables. Thus, all the variables that appear in the first argument to `DSolve` are also in the second or third arguments.

- Equations in which there are other symbolic quantities, such as mass or the spring constant. The solution in this case depends on the independent variables, the dependent variables, and the additional symbolic parameters.

Here is an example of the first type.

```plaintext
In[1]:= DSolve[y''[x] + 8*x*y[x] == 0, y, x]
```

Here is an example of the second type. This equation has a symbolic parameter \( k \).

```plaintext
In[2]:= sol = DSolve[{y''[x] + k^2*y[x] == x^2, y[0] == 3 k + 1, y'[0] == k}, y, x]
Out[2]= {{y -> Function[{x}, (-2 + k^2 x^2 + 2 Cos[k x] + k^4 Cos[k x] + 3 k^3 Cos[k x] + k^4 Sin[k x])/k^4]}}
```

`DSolve` is equipped to deal with both types of equations. It is extremely useful to have the solution available for all possible values of the parameters in the second type of equation.

Here is a plot of the previous solution for different values of the parameter \( k \).

```plaintext
In[3]:= Plot[Evaluate@Table[y[x] /. sol /. {k -> i}, {i, 1, 5, 2}]], {x, -7, 7}]
Out[3]=
```

It should be noted that the presence of symbolic parameters can lead to fairly complicated output.

Differential Equation Solving with `DSolve`
It should be noted that the presence of symbolic parameters can lead to fairly complicated output.

This is seen in the following example (equation 2.14, page 401 of [K59]).

\[
\text{In[4]} := \text{eqn} = y''(x) - c x^a y(x);
\]

\[
\text{In[5]} := \text{sol} = \text{DSolve}[\text{eqn} == 0, y, x]
\]

\[
\text{Out[5]} = \{y \rightarrow \text{Function}[\{x\}, (-1)^{\frac{1}{2-a}} (2 + a)^{-\frac{1}{2-a}} x^{\frac{1}{2-a}} \text{BesselI}[\frac{1}{2 + a}, \frac{2 \sqrt{c} x^{\frac{2}{2-a}}}{2 + a}] \Gamma[2] \Gamma[1 + \frac{1}{2 + a}] + (2 + a)^{-\frac{1}{2-a}} x^{\frac{1}{2-a}} \text{BesselI}[\frac{1}{2 - a}, \frac{2 \sqrt{c} x^{\frac{2}{2-a}}}{2 + a}] \Gamma[1] \Gamma[1 + \frac{1}{2 + a}] \}\}
\]

However, for some special values of the parameters, the solution might be significantly simpler.

For these values of \(a\) and \(c\), the solution is much more simple.

\[
\text{In[6]} := \text{sol1} = y(x) / . \text{sol}[[1]] / . \{a \rightarrow 0, c \rightarrow 4\}
\]

\[
\]

Occasionally, a solution is valid for most, but not all, values of the parameters.

Since the input in this example is not valid at \(a = 0\), the solution has the same limitation.

\[
\text{In[7]} := \text{sol} = \text{DSolve}[[\{y'[x] == x, y[0] == 1/a\}, y, x]]
\]

\[
\text{Out[7]} = \{\{y \rightarrow \text{Function}[\{x\}, \frac{2 + a x^2}{2 a}]\}\}
\]

Of course, there is a simple remedy in this case: setting \(k = \frac{1}{a}\).

\[
\text{In[8]} := \text{sol} = \text{DSolve}[[\{y'[x] == x, y[0] == k\}, y, x]]
\]

\[
\text{Out[8]} = \{\{y \rightarrow \text{Function}[\{x\}, \frac{1}{2} \left(2 k + x^2\right)]\}\}
\]

In summary, the ability to solve differential equations with symbolic parameters is a powerful and essential feature of any symbolic solver such as DSolve. However, the following points should be noted.

- The solution might be complicated, and such calculations often require significant time and memory.
- The answer might not be valid for certain exceptional values of the parameters.
The solution might be easy to verify symbolically for some special values of the parameters, but in the general case a numerical verification method is preferable.

Numerical quantities in Mathematica can be of three types: infinite precision, machine precision, or arbitrary precision. The first type of number is referred to as “exact”, while the remaining two types represent incomplete information and are therefore called “inexact”.

Here is a simple example showing all three kinds of numbers.

\[\text{In}[9]:= \text{exactpi} = \pi;\]
\[\text{In}[10]:= \text{Precision}[\text{exactpi}]\]
\[\text{Out}[10]= \infty\]

\[\text{In}[11]:= (\text{machinepi} = \text{N[\pi]}) // \text{InputForm}\]
\[\text{Out}[11]//\text{InputForm}= 3.141592653589793\]

\[\text{In}[12]:= \text{Precision}[\text{machinepi}]\]
\[\text{Out}[12]= \text{MachinePrecision}\]

\[\text{In}[13]:= \text{highprecisionpi} = \text{N[\pi, 30]}\]
\[\text{Out}[13]= 3.14159265358979323846264338328\]

\[\text{In}[14]:= \text{Precision}[\text{highprecisionpi}]\]
\[\text{Out}[14]= 30.\]

Since DSolve is a symbolic solver, the algorithms used by it are primarily based on the assumption of exact input. However, equations that contain inexact quantities are handled by DSolve in the usual way.

These equations contain the inexact number 3. and 3.`40, respectively.

\[\text{In}[15]:= \text{DSolve}[x'[t] = 3. \cdot x[t], x, t]\]
\[\text{Out}[15]= \{\{x \rightarrow \text{Function}[\{t\}, e^{3. \cdot t} C[1]]\}\}\]

\[\text{In}[16]:= \text{DSolve}[x'[t] = 3.`40 \cdot x[t], x, t]\]
\[\text{Out}[16]= \{\{x \rightarrow \text{Function}[\{t\}, e^{3.00000000000000000000000000000000000000 \cdot t} C[1]]\}\}\]

Inexact input could arise, for example, when the coefficients in the equations are derived from a previous calculation and are known only approximately. In such cases, it might not be practical to convert the equations to exact form, as this could slow down the calculation significantly.
Here is a system of linear ODEs that all have exact coefficients. Note that even with a fairly small value of \( n \), the calculation takes a long time to finish.

\[
\text{In}[17]:= \quad n = 8;  \\
\quad x_0[t_] := 0;  \\
\quad x_n[t_] := 1;  \\
\quad eqns = Table[\{x_i'[t] - (x_{i+1}[t] - 2 x_i[t] + x_{i-1}[t]), x_i[0] - (1/n)\}, \{i, n-1\}];  \\
\quad vars = Table[x_i, \{i, n-1\}];  \\
\quad sol = DSolve[Map[\# == 0 &, Flatten[eqns]], vars, t]; // Timing  \\
\text{Out}[17]= \{35.703, Null\}
\]

\[
\text{In}[18]:= \quad \text{LeafCount}[sol]  \\
\text{Out}[18]= 212851
\]

This verifies the solution. Since the solution is complicated, a numerical verification method is used.

\[
\text{In}[19]:= \quad eqns /. sol /. \{t \to \text{RandomReal}[\{0, 1\}, \text{WorkingPrecision} \to 200]\} // N // \text{Chop} // \text{Flatten} // \text{Union}  \\
\text{Out}[19]= \{0\}
\]

If a single inexact quantity is introduced (in the function \( x_0[t] \)), the solution is returned more quickly.

\[
\text{In}[20]:= \quad n = 8;  \\
\quad x_0[t_] := 0.;  \\
\quad x_n[t_] := 1;  \\
\quad eqns = Table[\{x_i'[t] - (x_{i+1}[t] - 2 x_i[t] + x_{i-1}[t]), x_i[0] - (1/n)\}, \{i, n-1\}];  \\
\quad vars = Table[x_i, \{i, n-1\}];  \\
\quad sol = DSolve[Map[\# == 0 &, Flatten[eqns]], vars, t]; // Timing  \\
\text{Out}[20]= \{0.75, Null\}
\]

\[
\text{In}[21]:= \quad \text{LeafCount}[sol]  \\
\text{Out}[21]= 1563
\]

\[
\text{In}[22]:= \quad eqns /. sol /. \{t \to \text{RandomReal}[\{0, 1\}]\} // N[\#] & // \text{Chop} // \text{Flatten} // \text{Union}  \\
\text{Out}[22]= \{0\}
\]

Thus, it is often desirable to continue working with inexact quantities even within a symbolic function such as \text{DSolve}. However, it should be noted that the solution obtained in such cases could have a certain amount of numerical error and should be checked carefully. It is therefore recommended that if the problem size is not too large (for instance, if there are fewer than five equations), the input should be converted to exact form using the \text{Rationalize} function.

\[
\text{This equation contains inexact quantities.}  \\
\text{In}[23]:= \quad \text{DSolve}[\{x'[t] = 0.0001 \times x[t], x[0] = 3.07\}, x[t], t]  \\
\text{Out}[23]= \{\{x[t] \to 3.07 e^{0.0001 t}\}\}
\]
Here the equation is converted to exact form before being solved.

\[
\text{In}[24]:= \text{DSolve}[\text{Rationalize}[\{x'[t] == 0.0001 \times x[t], \ x[0] == 3.07, \ t\}, 0], \ x[t], \ t]
\]

\[
\text{Out}[24]= \left\{\left\{x[t] \rightarrow \frac{307 \times e^{0.0001 \times t}}{100}\right\}\right\}
\]

**Is the Problem Well-Posed?**

\text{DSolve} returns a general solution for a problem if no initial or boundary conditions are specified.

The general solution to this equation is returned.

\[
\text{In}[1]:= \text{DSolve}[y'[x] == 1 - y[x], \ y, \ x]
\]

\[
\text{Out}[1]= \left\{\left\{y \rightarrow \text{Function}[[x], 1 + e^{-x} \times c[1]]]\right\}\right\}
\]

However, if initial or boundary conditions are specified, the output from \text{DSolve} must satisfy both the underlying differential equation as well as the given conditions.

Here is an example with a boundary condition.

\[
\text{In}[2]:= \text{eqns} = \{y'[x] == 1 - y[x], \ y[3] == 5\};
\]

\[
\text{In}[3]:= \text{sol} = \text{DSolve}[\text{eqns}, \ y, \ x]
\]

\[
\text{Out}[3]= \left\{\left\{y \rightarrow \text{Function}[[x], e^{-x} \times \left(4 e^3 + e^x\right)]\right\}\right\}
\]

\[
\text{In}[4]:= \text{eqns} /. \text{sol}
\]

\[
\text{Out}[4]= \left\{\{\text{True}, \text{True}\}\right\}
\]

In such cases, it is useful to check whether \text{DSolve} has been asked a reasonable question—in other words, to check whether the problem is well-posed. An initial or boundary value problem is said to be *well-posed* if a solution for it is guaranteed to exist in some well-known class of functions (for example, analytic functions), if the solution is unique, and if the solution depends continuously on the data. Given an ODE of order \(n\) (or a system of \(n\) first-order equations) and \(n\) initial conditions, there are standard existence and uniqueness theorems that show that the problem is well-posed under a specified set of conditions. The right-hand side of the first-order linear ODE in the previous example is a polynomial in \(y(x)\) and hence infinitely differentiable. This is sufficient to apply Picard’s existence and uniqueness theorem, which only requires that the right-hand side be Lipschitz-continuous.
Most problems that arise in practice are well-posed since they are derived from sound theoretical principles. However, as a note of caution, the following are examples where DSolve might have difficulty finding a satisfactory solution to the problem.

Here is the solution to a first-order ODE in which the right-hand side fails to satisfy the Lipschitz condition around 0.

\[ y' = \frac{1}{y}, \quad y(0) = 1 \]

In[5]:= \[ \text{generalsol} = \text{DSolve}[[y'[x] == 1/y[x]], y, x] \]

Out[5]= \[
\left\{ \left\{ y \to \text{Function} \left[ \{x\}, -\sqrt{2 - \sqrt{x + C[1]}} \right] \right\}, \left\{ y \to \text{Function} \left[ \{x\}, \sqrt{2 - \sqrt{x + C[1]}} \right] \right\} \right\}
\]

The general solution has two branches.

In[6]:= Plot[Evaluate[y[x] /. generalsol /. {C[1] \to 1}], {x, -1, 3}]

This initial value problem is well-posed in a region around the initial condition and hence DSolve succeeds in picking out the correct branch for the given initial condition.

In[7]:= \[ \text{DSolve}[[y'[x] == 1/y[x], y[0] == 1], y, x] \]

\text{DSolve::bvnul:}
For some branches of the general solution, the given boundary conditions lead to an empty solution. \[ \]

Out[7]= \[
\left\{ \left\{ y \to \text{Function} \left[ \{x\}, \sqrt{1 + 2 x} \right] \right\} \right\}
\]

Here is a second-order ODE. The boundary conditions do not allow any solution to this problem.

In[8]:= \[ \text{DSolve}[[y''[x] + y[x] == 0, y[0] == 1, y[\text{Pi}] == 3], y[x], x] \]

\text{DSolve::bvnul:}
For some branches of the general solution, the given boundary conditions lead to an empty solution. \[ \]

Out[8]= \[
\{\}\]

In this example, DSolve returns a pair of solutions. As the table shows, the first solution is only valid for values of \( x \) greater than or equal to 2.

In[9]:= eqn = y'[x] = y[x]^ (1/2);
Finally, it is possible that a problem has a solution, but that \texttt{DSolve} fails to find it because the general solution is in implicit form or involves higher transcendental functions.

In this example, a solution is available only after inverting the roles of the dependent and independent variables.

\texttt{In[12]:=} \texttt{DSolve[y'[x] = 1 / (x - y[x]) \\ &
& y[0] = 1, y, x]}

\texttt{Out[12]= \{\{x \to \text{Function[\{y\}, \frac{e - 2 e^y + e y}{e}]\}\}\}}

This concludes the discussion of the basic principles for effectively working with \texttt{DSolve}. See the list of "references" that were found to be useful either during the development of \texttt{DSolve} or during the preparation of this documentation.
References


