

Part III. First Order Ordinary Differential Equations

Section 1. Entering, Solving, Plotting

Here begins a more formal introduction to *Mathematica* and differential equations. If you have only skimmed over Parts I and II, please review the discussion of differential equations found on the last three pages of Part II, Section 3. In this section three first order equations are used to illustrate some of the features of the DSolve function, *Mathematica's* primary tool for generating solutions to ordinary differential equations.

Occasional references to "Ledder" refer to the textbook entitled

Differential Equations: A Modeling Approach

by Glenn Ledder, McGraw-Hill, 2005.

Solve this: DSolve

It is good practice to assign a name to a differential equation before beginning an analysis. Names should also be assigned to solutions. A differential equation for y as a function of t is entered using unevaluated derivatives. All appearances of the function y must be in the form $y[t]$. The equation must be entered as an identity (using two equal signs). See the following example.

$$DE = y'[t] == y[t]/(t^2 + 1)$$

$$y'[t] == \frac{y[t]}{1 + t^2}$$

There are two ways to generate a solution.

Method 1. Enter DSolve[DE, y[t], t]

Method 2. Enter DSolve[DE, y, t]

The output for the first entry is named soln1. The output for the second is called soln2.

```
soln1 = DSolve[ DE, y[t], t ]
```

```
{ {y[t] -> eArcTan[t] C[1] } }
```

```
soln2 = DSolve[ DE, y, t ]
```

```
{ {y -> Function[ {t}, eArcTan[t] C[1] ] } }
```

Observe that both formulas for the general solution display the integration constant as C[1]. This choice of notation is intentional to avoid conflicts with variables that may already be assigned in the worksheet.

Note also that both solutions are in the form of a substitution rule. The first solution is a substitution for $y[t]$ and the second is a substitution for y . The second form is called a "pure function" and it is the one we will use most often. This is because it is easier to manipulate. For example it can be used to quickly check the validity of the solution.

Checking the solution

The solution can be checked by substitution of soln2 into DE.

DE/.soln2

{True}

If soln2 is substituted *Mathematica* does not evaluate the derivative in the identity.

DE/.soln1

$$\{y'[t] == \frac{e^{\text{ArcTan}[t]} C[1]}{1 + t^2}\}$$

Plotting solution curves

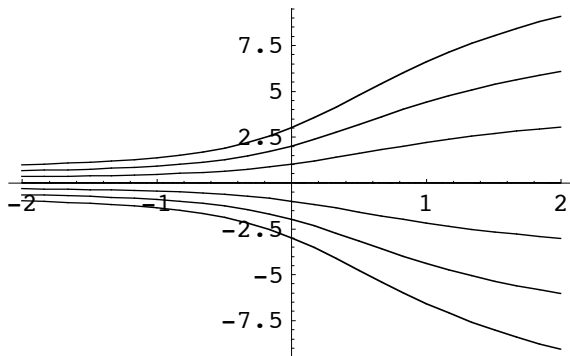
Given the general solution, the Table function can be used to plot several solution curves. It is first necessary to replace C[1] with some other symbol, like c, because C[1] is protected. (We work with soln2.)

soln2[[1,1]]/.C[1]->c

y → Function[{t}, e^{ArcTan[t]} c]

The following plot shows the solutions to DE when c = -3, -2, -1, 0, 1, 2, 3. The Evaluate function is applied to Table to force evaluation of the functions so points can be plotted. The plot is named P1 so it can be shown again later.

**P1 = Plot[Evaluate[Table[y[t]/.%, {c,-3,3}],
{t,-2,2}]**



Solving an initial value problem

If the differential equation is accompanied by an initial condition of the form

$$y(t_0) = y_0$$

we have an initial value problem (IVP for short). Look for a solution by applying DSolve to a set containing DE and the initial condition (as an identity):

DSolve[{DE, y[t0]==y0}, y, t]

For example, the solution to DE satisfying the initial condition $y(1) = 3$ is obtained as follows.

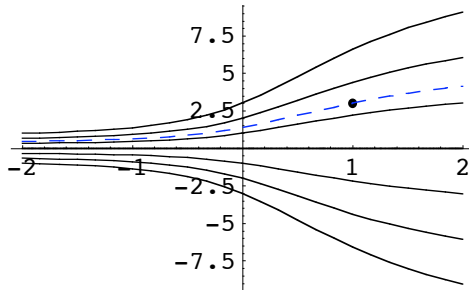
```
soln = DSolve[ {DE, y[1]==3}, y, t]
{{y -> Function[{t}, 3 e- $\frac{t}{4}$  + ArcTan[t] ]}}
```

The initial value can be checked by two substitutions.

```
y[t]/.soln/.t->1
{3}
```

The following picture shows the seven previous solutions, the IVP solution (blue and dashed), and the initial point.

```
IP = ListPlot[ {{1,3}}, PlotStyle->PointSize[0.02]
IVP = Plot[ y[t]/.soln, {t,-2,2},
PlotStyle->{Dashing[{0.03,0.03}], RGBColor[0,0,1]}]
Show[ P1, IP, IVP ]
```



Getting information out of the solution formula: Name the solution function

Suppose, for example, we want to know the value of t for which the IVP solution has the value 1. FindRoot should be able to find it. We first turn the solution into a named function f (see the last page in Part 2, Section 3). We also enter $f[t]$ to check that the correct solution formula has been used.

```
f := soln[[1,1,2]]
f[t]
3 e- $\frac{t}{4}$  + ArcTan[t]
```

Now find the solution to $f(t) = 1$. (The output is named a .)

```
a = FindRoot[ f[t] == 1, {t,0.5} ]
{t -> -0.323875}
```

And check that it is correct.

We wish to apply the function f to the value obtained above. There are two ways to do this. First note that the number 0.323875 can be obtained as $a[[1, 2]]$. This is how we get the first evaluation. The second one uses a substitution (a into $f[t]$).

```
f[ a[[1,2]] ]
1.
```

```
f[t]/.a
1.
```

A solution formula in terms of the initial condition

We look for a solution formula in terms of a generic initial condition, say $y(t_0) = y_0$, as follows.

```
IVPsoln = DSolve[ {DE, y[t0]==y0}, y, t ]

DSolve::bvfail : For some branches of the
  general solution, unable to solve the conditions. More...
{}
```

This method fails. However, another method can be used. First obtain the general solution.

```
gensoln = DSolve[ DE, y , t ]
{{y -> Function[{t}, e^ArcTan[t] C[1]]}}
```

Use the output to make a function with C[1] replaced by c.

```
g := gensoln[[1,1,2]]/.C[1]->c
```

Let's check the g is indeed the function we want.

```
g[t]
g[1]
c e^ArcTan[t]
c e^pi/4
```

Now use the function g to obtain the solution satisfying $g(t_0) = y_0$. First define this as an identity.

```
eqn = g[t0] == y0
c e^ArcTan[t0] == y0
```

Now use Solve to solve for c, substitute into the formula for g[t], and use it to define the function phi. This is carried out below, one step at a time.

```
Solve[ eqn, c ]
{{c -> e^-ArcTan[t0] y0}}

g[t]/.%[[1,1]]
e^ArcTan[t]-ArcTan[t0] y0
```

The next step fails when the evaluate function is not applied.

```
phi[t_,t0_,y0_] := Evaluate[%]
```

Here we check that phi is defined correctly.

phi[u,v,w]

$$e^{\text{ArcTan}[u]-\text{ArcTan}[v]} w$$

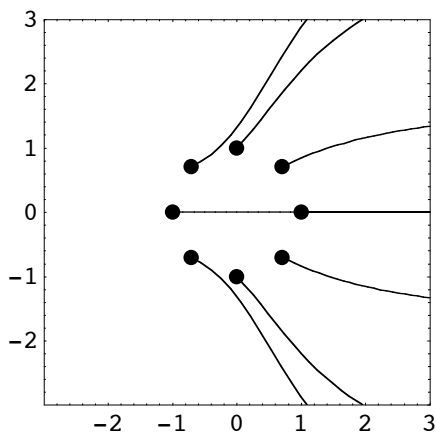
And here we check that phi, evaluated at t0,t0,y0 has the value y0.

phi[t0,t0,y0]

y0

We now use the phi function to create pictures of solutions satisfying specific initial conditions.

```
startpoints = ListPlot[
  Table[ {Cos[n*Pi/4],Sin[n*Pi/4]}, {n,0,7}],
  PlotStyle->PointSize[0.04],
  DisplayFunction->Identity ];
plots = Table[ Plot[ phi[t,Cos[n*Pi/4],Sin[n*Pi/4]], {t,Cos[n*Pi/4],3},
  DisplayFunction->Identity ], {n,0,7} ];
Show[ startpoints, plots, DisplayFunction->$DisplayFunction, AspectRatio->1,
  PlotRange->{{-3,3},{-3,3}}, Frame->True, Axes->None ]
```

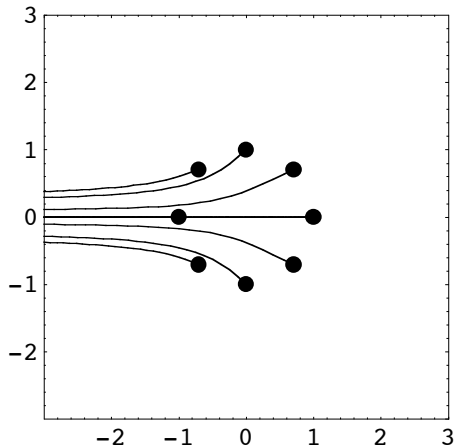


Now run time backwards from the same initial points.

```

plots = Table[ Plot[ phi[t,Cos[n*Pi/4],Sin[n*Pi/4]], {t,-3,Cos[n*Pi/4]},
  DisplayFunction->Identity ], {n,0,7} ];
Show[ startpoints, plots, DisplayFunction->$DisplayFunction, AspectRatio->1,
  PlotRange->{{-3,3},{-3,3}}, Frame->True, Axes->None ]

```



Look before you leap

Before solving any first order differential equation it is a very good idea to put it into normal form

$$y' = f(t, y)$$

and think a bit about the function on the right hand side: $f(t, y)$. That way unusual *Mathematica* output can be anticipated and pitfalls can be avoided. The essential facts to keep in mind are given in the following theorem.

Unique solution theorem Unique IVP solutions are guaranteed to exist inside of t, y rectangles where $f(t, y)$ and the y partial $\partial_y f(t, y)$ are both continuous. (Ledder, Section 2.4, Theorem 2.4.1)

Here is an example (very similar to DE above).

$$\text{DE2} = \mathbf{y}'[\mathbf{t}] == \mathbf{y}[\mathbf{t}]/(\mathbf{t}^2 - 1)$$

$$y' [t] == \frac{y[t]}{-1 + t^2}$$

The differential equation is already in normal form. Note that the function on the right side

$$f(t, y) = \frac{y}{t^2 - 1}$$

is continuous and has a continuous y partial derivative in any rectangle that does not contain any points on the vertical line $t = 1$ or the vertical line $t = -1$.

Let's see *Mathematica's* general solution.

```
gensoln = DSolve[ DE2, y, t ]
```

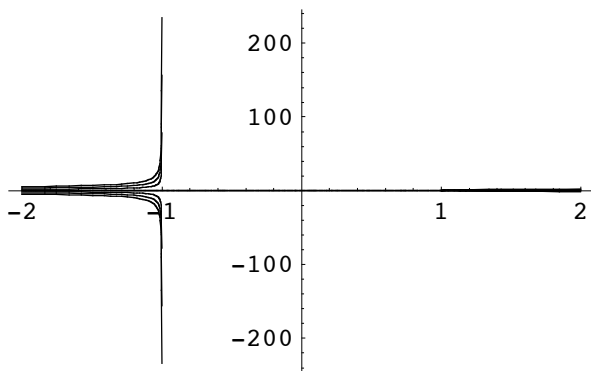
```
{ {y -> Function[{t},  $\frac{\sqrt{-1+t} C[1]}{\sqrt{1+t}}$  ] } }
```

Examination of the solution formula reveals that when C[1] is not 0, it is only valid for t greater than 1 or t less than -1, unless the constant is allowed to take on imaginary values. Note that when t is less than -1, both the numerator and the denominator are pure imaginary so the quotient is real.

Several solution curves are plotted below using an input similar to the one used to generate the first plot in this section.

```
gensoln = gensoln/.C[1]->c;
```

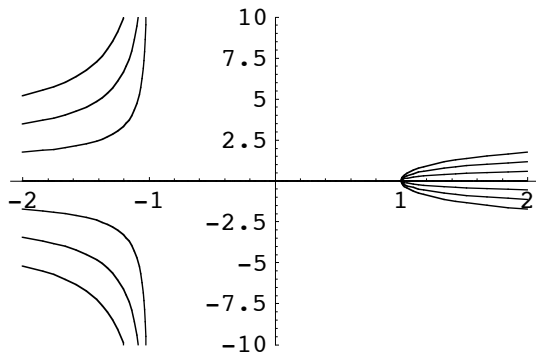
```
Plot[ Evaluate[Table[y[t]/.gensoln, {c,-3,3}]], {t,-2,2}]
```



Several error messages were generated warning that some formulas do not evaluate to real numbers. They were deleted.

It appears that solutions have a vertical asymptote at $t = -1$ and they disappear at $t = 1$. The next plot gets closer by restricting the vertical range to -10..10.

```
Plot[ Evaluate[Table[y[t]/.%, {c,-3,3}], {t,-2,2},
PlotRange->{-10,10}]
```



From this picture we conjecture that all solutions to DE2 that are valid outside the closed interval $[-1,1]$ have a vertical asymptote as t approaches -1 from the left and will approach 0 as t approaches the 1 from the right.

The question is "How does *Mathematica* represent solutions when t is greater than 1 or when t is less than -1 ?" The answer, as illustrated below, is rather unusual (and probably not the one you might guess).

Question: What is the solution to DE2 satisfying the initial condition $y(0) = 1$?

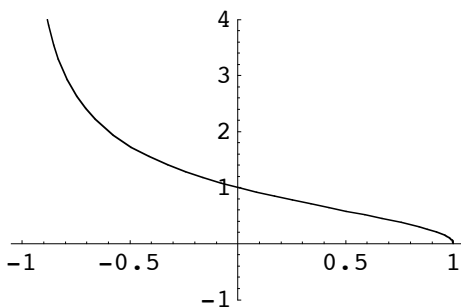
Answer: See *Mathematica's* solution.

```
soln2 = DSolve[ {DE2, y[0]==1}, y, t ]
```

```
{ {y -> Function[ {t}, - $\frac{i \sqrt{-1+t}}{\sqrt{1+t}}$  ] ] }
```

The same formula can be used, but with an imaginary constant: i . See the graph below.

```
Plot[ y[t]/.soln2, {t,-1,1}, PlotRange->{-1,4} ]
```

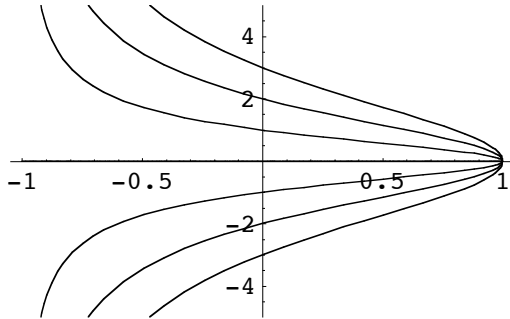


The next picture shows several solutions on the interval $-1 < t < 1$. The values for c are listed first.


```

Table[k*I, {k,-3,3}]
Plot[ Evaluate[Table[y[t]/.gensoln, {c,-3*I,3*I,I}]], {t,-1,1}, PlotRange->{-5,5}]
{-3 i, -2 i, -i, 0, i, 2 i, 3 i}

```



When there is more than one solution formula

As a rule, DSolve attempts to return explicit solution formulas even when there is more than one of them.

```

DE3 = y'[t] == t/y[t]
gensoln3 = DSolve[ DE3, y, t ]
y'[t] == t/y[t]
{{y -> Function[{t}, -sqrt[t^2 + 2 C[1]] ]}, {y -> Function[{t}, sqrt[t^2 + 2 C[1]] ]}}

```

When an initial condition is added, DSolve picks the appropriate formula.

```

soln3 = DSolve[ {DE3, y[0]==1}, y, t ]
Plot[ y[t]/.soln3, {t,-3,3}, PlotRange->{0,4} ]

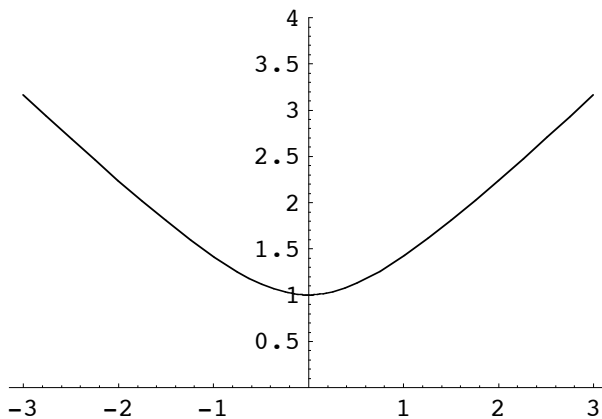
```

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

```

{{y -> Function[{t}, sqrt[1 + t^2 ] ]}}

```



This solution curve appears to be branch of a hyperbola. Several branches can be plotted by making a list of the initial conditions and then using the Table function to make solution formulas.

```

inits = {y[0]==1, y[0]==-1, y[1]==0.1, y[1]==-0.1, y[-1]==0.1, y[-1]==-0.1};
solns = Evaluate[Table[ DSolve[ {DE3, inits[[k]]}, y, t], {k,6}]];
Plot[ Evaluate[Table[ y[t]/.solns[[k]], {k,6}]], {t,-3,3},
      PlotRange->{-3,3}, AspectRatio->1, PlotLabel->"Six Solution Curves" ]

```

