Solving Linear Systems with the Laplace Transform

Kurt Bryan and SIMIODE

This notebook illustrates using the Laplace transform to solve linear system of ODEs (and also just using Mathematica's ***DSolve*** command).

Consider the linear constant-coefficient nonhomogeneous system

```
In[1]:= de1 = x1'[t] == -5*x1[t] + 6*x2[t] - 2*Sin[t] + 10*Cos[t] - 6*Exp[t]
        de2 = x2'[t] == -3*x1[t] + x2[t] + 6*Cos[t]
```

with initial data x1(0) = 2 and x2(0) = 1.

To solve, Laplace transform both sides of both equations

```
In[3]:= de1lap = LaplaceTransform[de1, t, s]
        de2lap = LaplaceTransform[de2, t, s]
```

Substitute in the initial conditions, and (for convenience) let X1 denote the Laplace transform of x1 and X2 the transform of x2:

```
In[5]:= de1lap2 = de1lap /. {x1[0] → 2, x2[0] → 1,
           LaplaceTransform[x1[t], t, s] → X1, LaplaceTransform[x2[t], t, s] → X2}
        de2lap2 = de2lap /. {x1[0] → 2, x2[0] → 1, LaplaceTransform[x1[t], t, s] → X1,
           LaplaceTransform[x2[t], t, s] → X2}
```

Solve for the transforms X1 and X2

```
In[7]:= Xsols = Solve[{de1lap2, de2lap2}, {X1, X2}]
```

Let X1sol and X2sol denote the transforms

```
In[8]:= X1sol = X1 /. Xsols[[1]]
        X2sol = X2 /. Xsols[[1]]
```

Inverse transform to find the solutions (defined as functions of t)

```
In[12]:= x1sol[t_] = InverseLaplaceTransform[X1sol, s, t]
         x2sol[t_] = InverseLaplaceTransform[X2sol, s, t]
```

A quick check using the ***DSolve*** command:

```
In[14]:= dsols = DSolve[{de1, de2, x1[0] == 2, x2[0] == 1}, {x1, x2}, t]
```

Mathematica's ***DSolve*** output is complicated, but it is the same as the result from the Laplace transform!

In[25]:=
```
xx1 = x1 /. dsols⟦1⟧;
Simplify[xx1[t]]
```

In[27]:=
```
xx2 = x2 /. dsols⟦1⟧;
Simplify[xx2[t]]
```