Homogeneous Linear Systems of ODEs in Mathematica

Kurt Bryan and SIMIODE

This notebook illustrates how to analyze homogeneous linear systems of ODEs in Mathematica.

Example System: Consider the homogeneous linear system of ODEs

```
In[1]:= de1 = x1'[t] == x1[t] + 3 * x2[t]
        de2 = x2'[t] == 3 * x1[t] + x2[t]
```

for functions x1(t) and x2(t), with initial data x1(0) = 2, x2(0) = 6.

Solution via DSolve: The solution can be obtained with Mathematica's ***DSolve*** command:

```
In[3]:= DSolve[{de1, de2, x1[0] == 2, x2[0] == 6}, {x1, x2}, t]
```

Solution via Eigen-analysis: The system above is of the form **x' = Ax** where x = <x1,x2> and

```
In[4]:= A = {{1, 3}, {3, 1}};
        A // MatrixForm (* Print A in standard matrix form *)
```

We solve using the eigenvector/value analysis of Section 6.2.2. The eigenvalues and vectors of A can be computed as

```
In[6]:= eigs = Eigensystem[A]
```

The first part of "eigs" (the list {4,-2}) has as its components the eigenvalues of A, that is, the eigenvalues are

```
In[7]:= eigs[[1, 1]]
        eigs[[1, 2]]
```

The second part of "eigs" (the list representing a 2 x 2 matrix) has as its rows (not columns) the corresponding eigenvectors of A. Define a matrix **P**

```
In[9]:= P = Transpose[eigs[[2]]];
        P // MatrixForm
```

as was done in equation (6.18) in the text. Let **x0** denote the vector of initial conditions,

```
In[11]:= x0 = {2, 6}
```

As per the procedure in Section 6.2.2, we can construct the solution to this ODE system by first solving **Pc = x0** for **c**, which in Mathematica takes the form

In[12]:=
```
c = LinearSolve[P, x0]
```

From (6.17) in the text the solution is then

In[13]:=
```
xsol = c⟦1⟧ * Exp[eigs⟦1, 1⟧ * t] * P⟦All, 1⟧ + c⟦2⟧ * Exp[eigs⟦1, 2⟧ * t] * P⟦All, 2⟧
```

The notation "P[[All,j]]" picks off the jth column of the matrix P.

The components of the solution x1(t) and x2(t) are then

In[14]:=
```
xsol⟦1⟧
xsol⟦2⟧
```

Example 2: Consider the system

In[16]:=
```
de1 = x1'[t] == -5 * x1[t] + 6 * x2[t]
de2 = x2'[t] == -3 * x1[t] + x2[t]
```

with initial data x1(0) = 1, x2(0) = 2. This system is governed by the matrix

In[18]:=
```
A = {{-5, 6}, {-3, 1}}
A // MatrixForm
```

Compute the eigenvalues/vectors as

In[20]:=
```
eigs = Eigensystem[A]
```

Following the analysis in the previous example, pick off the eigenvectors and install them in a matrix **P** (as the columns of **P**, not the rows, hence a transpose is needed)

In[22]:=
```
P = Transpose[eigs⟦2⟧]
P // MatrixForm
```

In the general solution (6.17) in the text, the desired initial conditions are obtained by taking the c[k] as the components of the vector c that satisfies Pc = x0 where x0 = <1,2> (equation (6.18)). Compute

In[26]:=
```
c = LinearSolve[P, {1, 2}]
```

The solution (from (6.17)) is then

In[27]:=
```
xsol = c⟦1⟧ * Exp[eigs⟦1, 1⟧ * t] * P⟦All, 1⟧ + c⟦2⟧ * Exp[eigs⟦1, 2⟧ * t] * P⟦All, 2⟧
```

However, a real-valued version is obtained with

In[29]:=
```
xsolreal = ComplexExpand[xsol]
```