Fall 2003 Math 308/501–502
9 Matrix Methods for Linear Systems
9.B Matrix Laplace Transform Method
Mon, 08/Dec

Summary

The Last Hurrah

This is it campers: the Grand Finale, where we marry Laplace transform techniques with matrix methods for linear systems! It's a little ditty we call the **Matrix Laplace Transform Method**.

The Setting

With t as an independent scalar variable, let $\mathbf{A}$ be an $n \times n$ constant matrix, $\mathbf{f}$ an $n \times 1$ vector function of t, $\mathbf{x}(t)$ a function of t, and $\mathbf{x}_0$ an $n \times 1$ constant vector. Also, let s be a scalar variable.

Some Definitions

Denote the Laplace transform of $\mathbf{x}(t)$ by $\hat{\mathbf{x}}(s)$. It is a column vector whose elements are the respective Laplace transforms of the components $x_1(t), \ldots, x_n(t)$ of $\mathbf{x}(t)$. In a similar manner, let $\mathcal{L}\{\mathbf{f}(t)\} = \hat{\mathbf{f}}(s)$. More briefly, we write $\mathcal{L}\{\mathbf{x}\} = \hat{\mathbf{x}}$ and $\mathcal{L}\{\mathbf{f}\} = \hat{\mathbf{f}}$.

Analogously, for a matrix $\mathbf{M} = \mathbf{M}(t)$, we define $\mathcal{L}\{\mathbf{M}\} = \hat{\mathbf{M}}$ to be the matrix whose elements are the respective Laplace transforms of the components $m_{ij}(t)$ of $\mathbf{M}(t)$. Inverse Laplace transforms of vectors or matrices are similarly defined. Moreover, the familiar properties of Laplace transforms carry over as well.

Derivation of the "Grail"

Given the initial value problem $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{f},\ \mathbf{x}(0) = \mathbf{x}_0$, let's employ our usual 4-step procedure from Chapter 7 to obtain its unique solution $\mathbf{x}(t)$.

1. Take the Laplace transform of each side of the ODE.

$$s\hat{\mathbf{x}} - \mathbf{x}(0) = \mathbf{A}\hat{\mathbf{x}} + \hat{\mathbf{f}}$$

2. Substitute for the initial condition.

$$s\hat{\mathbf{x}} - \mathbf{x}_0 = \mathbf{A}\hat{\mathbf{x}} + \hat{\mathbf{f}}$$

3. Solve for the unknown transform $\hat{\mathbf{x}}$.

$$\begin{aligned} s\mathbf{I}\hat{\mathbf{x}} - \mathbf{A}\hat{\mathbf{x}} &= \hat{\mathbf{f}} + \mathbf{x}_0 \\ (s\mathbf{I} - \mathbf{A})\hat{\mathbf{x}} &= \hat{\mathbf{f}} + \mathbf{x}_0 \\ \hat{\mathbf{x}} &= (s\mathbf{I} - \mathbf{A})^{-1}(\hat{\mathbf{f}} + \mathbf{x}_0) \end{aligned}$$

4. Take the inverse Laplace transform of each side to obtain the solution $\mathbf{x} = \mathbf{x}(t)$. Voilà: a one-line solution: the Holy Grail!

$$\mathbf{x} = \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}(\hat{\mathbf{f}} + \mathbf{x}_0)\right\}$$

A Hidden Bonus!

Note that when $\mathbf{f} = \mathbf{0}$, we have

$$\mathbf{x} = \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}(\hat{\mathbf{f}} + \mathbf{x}_0)\right\} = \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}\right\}\mathbf{x}_0,$$

from which it follows that $e^{t\mathbf{A}} = \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}\right\}$, since $\mathbf{x} = e^{t\mathbf{A}}\mathbf{x}_0$ is the unique solution of the IVP. So this gives another (elegant) way of computing the matrix exponential function! Indeed, my calculator implementations of the matrix exponential use this technique. It's one that we'll replicate in the MATLAB examples below.

MATLAB Examples

573/B1

Solve the IVP $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{f},\ \mathbf{x}(0) = \mathbf{x}_0$, where $\mathbf{A} = \begin{bmatrix} 0 & 2 \\ -1 & 3 \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and $\mathbf{x}_0 = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$.

Solution

```
%
% NSS4-9.B/Example B1
%
syms s t
A = sym([0 2; -1 3]);
f = sym([0; 0]);
x0 = sym([-1; 3]);
%
I = eye(2);
x = ilaplace((s*I-A) \ (laplace(f) + x0))

x =

[ -8*exp(t)+7*exp(2*t)]
[ -4*exp(t)+7*exp(2*t)]

check1 = simple(diff(x,t) - A*x - f)

check1 =

[ 0]
[ 0]

check2 = simple(subs(x, t, sym(0)))

check2 =

[ -1]
[  3]

%
echo off; diary off
```

573/B2

Solve the IVP $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{f},\ \mathbf{x}(0) = \mathbf{x}_0$, where $\mathbf{A} = \begin{bmatrix} 3 & -2 \\ 4 & -1 \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} \sin t \\ -\cos t \end{bmatrix}$, and $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Solution

```
%
% NSS4-9.B/Example B2
%
syms s t
A = sym([3 -2; 4 -1])

A =

[  3, -2]
[  4, -1]

f = sym([sin(t); -cos(t)])

f =

[  sin(t)]
[ -cos(t)]

x0 = sym([0; 0])

x0 =

[ 0]
[ 0]

%
I = eye(2)
I =
     1     0
     0     1
x = ilaplace((s*I-A) \ (laplace(f) + x0));
pretty(x)

        [
        7/10 cos(t) - 1/10 sin(t) - 7/10 exp(t) cos(2 t) + 2/5 exp(t) sin(2 t)
        ]
        [11                     11                                       ]
        [-- cos(t) + 7/10 sin(t) - -- exp(t) cos(2 t) - 3/10 exp(t) sin(2 t)]
        [10                     10                                       ]
check1 = simple(diff(x,t) - A*x - f)

check1 =

[ 0]
[ 0]

check2 = simple(subs(x, t, sym(0)))

check2 =

[ 0]
[ 0]

%
echo off; diary off
```

M-211/87 [from the 3rd Edition of your lab manual]

Compute $e^{t\mathbf{A}}$ for $\mathbf{A} = \begin{bmatrix} -9 & 7 & -27 & 17 \\ 7 & -4 & 18 & -13 \\ 4 & -1 & 8 & -7 \\ -1 & 4 & -9 & 3 \end{bmatrix}$.

Solution

We compute $e^{t\mathbf{A}}$ as $\mathcal{L}^{-1}\left\{(s\mathbf{I}-\mathbf{A})^{-1}\right\}$, then compare it with the output from **expm**. At first it seems as if there is a difference in the third column. Recalling two hyperbolic function definitions, however, this discrepancy is cleared up. Results are equivalent! Here is $e^{t\mathbf{A}}$ as returned by $\mathcal{L}^{-1}\left\{(s\mathbf{I}-\mathbf{A})^{-1}\right\}$.

$$\begin{bmatrix} e^{-t}(4-2t)-3e^{t} & \frac{9}{2}e^{t}-(2t+\frac{9}{2})e^{-t} & -27\sinh t & \frac{15}{2}e^{t}+(2t-\frac{15}{2})e^{-t} \\ 2e^{t}+(3t-2)e^{-t} & (3t+4)e^{-t}-3e^{t} & 18\sinh t & (5-3t)e^{-t}-5e^{t} \\ e^{t}+(2t-1)e^{-t} & (2t+\frac{3}{2})e^{-t}-\frac{3}{2}e^{t} & \cosh t+8\sinh t & (\frac{5}{2}-2t)e^{-t}-\frac{5}{2}e^{t} \\ (t+1)e^{-t}-e^{t} & \frac{3}{2}e^{t}+(t-\frac{3}{2})e^{-t} & -9\sinh t & \frac{5}{2}e^{t}-(t+\frac{3}{2})e^{-t} \end{bmatrix}$$

```
%
% M-211/87 [from the new 3rd edition]
%
syms s t
A = sym([-9 7 -27 17; 7 -4 18 -13; ...

A =

[  -9,   7, -27,  17]
[   7,  -4,  18, -13]
[   4,  -1,   8,  -7]
[  -1,   4,  -9,   3]

        4 -1 8 -7; -1 4 -9 3])
I = eye(4);
etA1 = ilaplace(inv(s*I - A)); % See typset output.
etA2 = simple(expm(t*A));
my_diff = expand(etA1 - etA2);
my_diff = subs(my_diff, {'sinh(t)', 'cosh(t)'}, ...
    {'(exp(t)-exp(-t))/2', '(exp(t)+exp(-t))/2'});
no_diff = simple(my_diff)

no_diff =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]

[ 0, 0, 0, 0]
```

Epilog

Hey, troops, it's been swell. Good luck on your finals!