Matrix exponential in Matlab

This note explains how to form the matrix exponential in Maltab, and how to use it to carry out various computations for linear dynamical systems. For background, see Lectures 9, 10, and 13.

The Matlab function `expm(A)` computes the matrix exponential e^A. In contrast, `exp(A)` computes the elementwise exponential of the matrix A, which is usually not what you want.

Time-invariant autonomous linear dynamical system

Suppose $\dot{x} = Ax$, with $A \in \mathbf{R}^{n\times n}$ (and constant).

To compute $x(t) = e^{tA}$, we can use the code

```
x_t = expm(t*A)*x_0;
```

To compute $x(hk)$, for $k = 0, \ldots, K$ (*i.e.*, to sample the trajectory with time interval $h > 0$), and plot the resulting trajectory, we can use the code

```
exphA = expm(h*A);  % propagates time forward h seconds
x = x_0;
xs = [];  % its columns will contain x(h0),..., x(hK)
for k = 1:K
    x = exphA*x;
    xs = [xs x];
end
plot(xs') % plot each state component versus time
```

Piecewise constant autonomous linear dynamical system

We can compute the state for a time-varying linear dynamical system, $\dot{x}(t) = A(t)x(t)$, when A is piecewise constant. For example, suppose that $A(t) = A_0$ for $0 \le t \le 3.3$, and $A(t) = A_1$ for $t > 3.3$. To find $x(5.2)$ given $x(0.9)$, we can use the code

```
x33 = expm((3.3-0.9)*A0)*x09; % compute x(3.3)
x52 = expm((5.2-3.3)*A1)*x33; % compute x(5.2)
```

Time-invariant linear dynamical system with constant input

Suppose $\dot{x} = Ax + b$, where A and b are constant. The trajectory is, for $t \geq 0$,

$$x(t) = e^{tA}x(0) + \int_0^t e^{(t-\tau)A} b \, d\tau = e^{tA}x(0) + \left(\int_0^t e^{\tau A} \, d\tau \right) b.$$

We can compute the matrix on the right hand side, which also comes up in the first-order-hold discretization, several ways. When A is invertible, we have the formula

$$\int_0^t e^{\tau A} \, d\tau = A^{-1} \left(e^{tA} - I \right).$$

Another method, which works even when A is not invertible, is as follows. We first compute the matrix

$$M = \exp \left(t \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \right).$$

Then we have

$$\int_0^t e^{\tau A} \, d\tau = M_{12}, \tag{1}$$

the $n \times n$ $1,2$ block of M.

To show this, consider the autonomous linear dynamical system

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix},$$

with $x(0) = 0$, $z(0) = b$. Since $\dot{z} = 0$, we have $z(t) = b$, so the system is the affine dynamical system $\dot{x} = Ax + b$, and we have

$$x(t) = \left(\int_0^t e^{\tau A} \, d\tau \right) b.$$

But we also have

$$\begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \exp \left(t \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} 0 \\ b \end{bmatrix}$$

We can find $x(t)$ using

$$\begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \exp \left(t \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} M_{12} b \\ b \end{bmatrix}.$$

Since this holds for any b, we've shown (1).

The Matlab code for computing $x(t)$ given $x(0)$ would then be:

```
xx = expm(t*[A eye(n); zeros(n) zeros(n)])*[x_0; b];
x_t = xx(1:n);
```

Linear dynamical system with piecewise constant input

The ideas above can be combined to compute the state of a linear dynamical system driven by a piecewise constant input, indeed, even when the dynamics matrix is also time-varying, but piecewise constant.