Introduction to MATLAB

Violeta Ivanova, Ph.D.
Office for Educational Innovation & Technology

violeta@mit.edu
http://web.mit.edu/violeta/www

Topics

- MATLAB Interface and Basics
- Calculus, Linear Algebra, ODEs
- Graphics and Visualization
- Basic Programming
- Programming Practice
- Statistics and Data Analysis

Resources

- Class materials

  http://web.mit.edu/acmath/matlab/IAP2007

  - Previous session: InterfaceBasics <.zip, .tar>
  - This session: LinsysCalcODE <.zip, .tar>
- Mathematical Tools at MIT web site

  http://web.mit.edu/ist/topics/math

MATLAB Toolboxes & Help

- MATLAB
  + Mathematics
    + Matrices and Linear Algebra
      + Solving Linear Systems of Equations
      + Inverses and Determinants
      Eigenvalues
    + Polynomials and Interpolation
      + Convolution and Deconvolution
    + Differential Equations
      + Initial Value Problems for ODEs and DAEs

MATLAB Calculus & ODEs

Integration & Differentiation

Differential Equations

ODE Solvers

Integration

- Polynomial integration

$$\int p_1 x^n + \ldots + p_n x + p_{n+1} dx = P_1 x^{n+1} + \ldots + P_{n+1} x + C$$

- Analytical solution
  - Built-in function `polyint`

```
>> p = [p1 p2 p3 …];
>> P = polyint(p);  assumes C=0
>> P = polyint(p, k);  assumes C=k
```

More Polynomial Integration

- Area under a curve

$$\int_a^b p_1 x^n + p_2 x^{n-1} \ldots + p_{n-1} x^2 + p_n x + p_{n+1} dx$$

- Built-in function `polyval`

```
>> p = [p1 p2 p3 …];
>> P = polyint(p)
>> area = polyval(P, b) - …
                  polyval(P, a)
```

Differentiation

- Polynomial differentiation

$$\frac{d}{dx}(P_1 x^n + \ldots + P_n x + C) = p_1 x^{n-1} + \ldots + p_n$$

- Built-in function `polyder`

```
>> P = [P1 P2 … Pn C]
>> p = polyder(P)
```

Differential Equations

- Ordinary Differential Equations (ODE)

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

- Differential-Algebraic Expressions (DAE)

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

- MATLAB solvers for ODEs and DAEs

```
>> ode45; ode23; ode113; ode23s …
```

ODE and DAE Solvers

- Syntax

```
>> [T,Y] = solver(odefun,tspan,Y0)
```

`solver`: `ode45`, `ode23`, etc.

`odefun`: function handle

`tspan`: interval of integration vector

`Y0`: vector of initial conditions

`[T, Y]`: numerical solution in two vectors

ODE Example: Mars Lander

- Entry, descent, landing (EDL) force equilibrium

$$\vec{F} = m\vec{a}$$

Lander velocity: v

Drag coefficient: k = 1.2

Lander mass: m = 150 kg

Mars gravity: g = 3.688 m/s^2

$$m\frac{dv(t)}{dt} = mg - kv^2(t)$$

- EDL velocity ODE

$$\frac{dv(t)}{dt} = g - \frac{k}{m}v^2(t)$$

Velocity at t_0=0: v_0=67.056 m/s

Theoretical example courtesy of Ms. Shannon Cheng, MIT Department of Aeronautics & Astronautics.

ODE Example (continued)

- Problem: $\frac{dv(t)}{dt} = g - \frac{k}{m}v^2(t)$
- Solution: `odeLV.m, MarsLander.m`

```
tspan = [0 : 0.05 : 6];
V0 = 67.056;
[t, V] = ode45(@odeLV, tspan, V0)
-------------------------
function DV = odeLV(t, V)
   K = 1.2; M = 150; G = 3.688;
   DV = G - K/M * V^2
```

PDE Solver

- Partial Differential Equations (PDEs)
  - Elliptic and parabolic

    e.g. Laplace's equation: $\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$
- Numerical PDE solver: `pdepe`
  - Syntax:

```
>> solution = pdepe(m, pdefun,
icfun, bcfun, xmesh, tspan)
```

  - Function handles: `pdefun, icfun, bcfun`

MATLAB Linear Systems

Linear Equations & Systems

Eigenvalues & Eigenvectors

Linear Dynamic Networks

Useful Functions

- Matrices & vectors

```
>> [n, m]= size(A)
>> n = length(X)
>> M1 = ones(n, m)
>> M0 = zeros(n, m)
>> En = eye(n)
>> N1 = diag(En)
>> [evecs, evals] = eig(A)
```

Systems of Linear Equations

- Definition:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m &= b_2 \\ \ldots & \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m &= b_m \end{aligned}$$

- Matrix form: Ax = b

$$\begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \ldots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \ldots \\ b_m \end{bmatrix}$$

Solving Linear Equations

- Define matrix `A`

```
>> A = [a11 a12 … a1m
        a21 a22 … a2m
        …
        an1 an2 … anm]
```

- Define vector `b`

```
>> b = [b1; b2; … bm]
```

- Compute vector x

```
>> x = A \ b
```

State Equation

- Ordinary Differential Equations

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

- Linear systems -> State Equation

$$\begin{bmatrix} x_1' \\ \dots \\ x_n' \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} \Rightarrow \boxed{\dot{\bar{x}} = A\bar{x}}$$

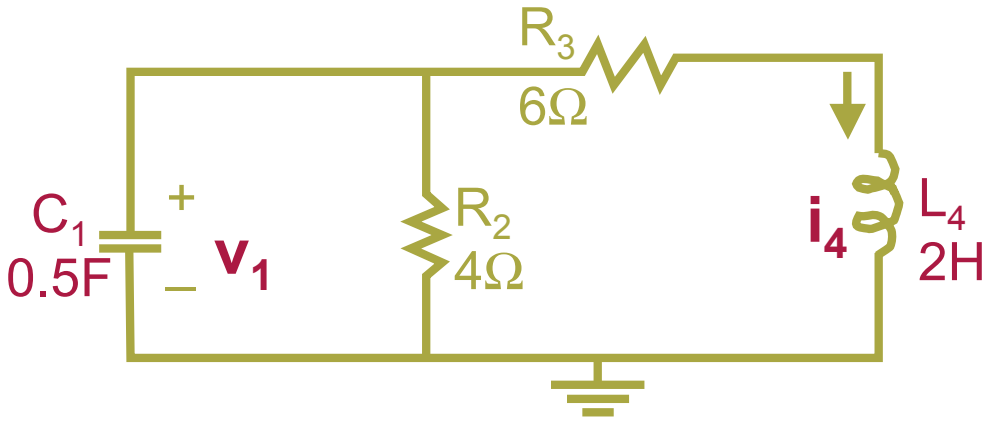
- Eigenvalues, λ_i, and eigenvectors, v_i, of a square matrix ***A***

$$\boxed{A\, v_i = \lambda_i\, v_i}$$

Linear Dynamic Networks

- Solution using **eigenvalue method**
  - Identify the network's states ..... $\bar{x}$
  - Identify initial conditions ..... $\bar{x}(0)$
  - Find the state equation ..... $\dot{\bar{x}} = A\bar{x}$
  - Find eigenvalues & eigenvectors ..... λ_i, v_i
  - The general solution is ..... $\bar{x}(t) = \sum_i a_i \bar{v}_i e^{\lambda_i t}$
  - Solve for a_i ..... $\bar{a} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 & \dots & \bar{v}_n \end{bmatrix}^{-1} \bar{x}(0)$

Example: RCL Circuit



$$\begin{aligned}\frac{dv_1}{dt} &= -\frac{1}{2}v_1 - 2i_4 \\ \frac{di_4}{dt} &= \frac{1}{2}v_1 - 3i_4\end{aligned}$$

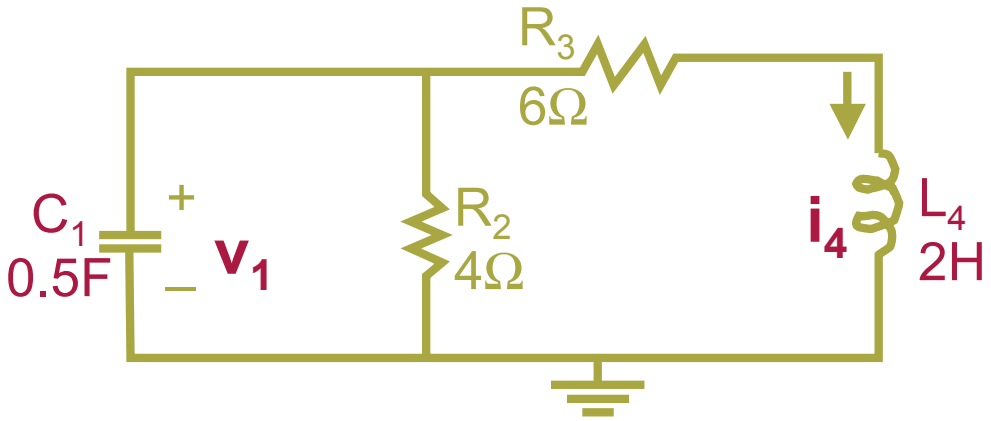
$$\begin{aligned}v_4(0) &= 2 \\ i_4(0) &= 1\end{aligned}$$

- State equation

$$\dot{\bar{x}} = \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \frac{dv_1}{dt} \\ \frac{di_4}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & -2 \\ \frac{1}{2} & -3 \end{bmatrix} \begin{bmatrix} v_1 \\ i_4 \end{bmatrix} = A\bar{x} \qquad \bar{x}(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Theoretical example courtesy of Prof. Stephen Hall, MIT Department of Aeronautics & Astronautics.

Example: RCL Circuit (continued)



$$\dot{\bar{x}} = A\bar{x}$$

$$Av_i = \lambda_i v_i$$

$$\bar{a} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 \end{bmatrix}^{-1} \bar{x}(0)$$

$$\bar{x}(t) = \sum_i a_i \bar{v}_i e^{\lambda_i t}$$

- Analytical solution

$$\bar{x}(t) = a_1\bar{v}_1 e^{\lambda_1 t} + a_2\bar{v}_2 e^{\lambda_2 t} \Rightarrow \begin{vmatrix} v_1(t) = \frac{4}{3}e^{-t} + \frac{2}{3}e^{-2.5t} \\ u_4(t) = \frac{1}{3}e^{-t} + \frac{2}{3}e^{-2.5t} \end{vmatrix}$$

RCL Circuit: MATLAB Solution

$$\dot{\bar{x}} = A\bar{x}$$

$$\bar{x}(0)$$

$$Av_i = \lambda_i v_i$$

$$\bar{a} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 \end{bmatrix}^{-1} \bar{x}(0)$$

$$\bar{x}(t) = \sum_i a_i \bar{v}_i e^{\lambda_i t}$$

```
>> A = [a11 a12; a21 a22]
>> x0 = [v1,0; i4,0]
>> [V, L] = eig(A)
>> λ = diag(L)
>> a = V \ x0
>> v1 = a1*V11*exp(λ1*t)...
        + a2*V12*exp(λ2*t)
   i4 = a1*V21*exp(λ1*t)...
        + a2*V22*exp(λ2*t)
```

Linear Systems Exercises

- Exercise One: `stateeig.m`
  - Matrices & vectors
  - Systems of linear equations
  - Eigenvalues & eigenvectors

Follow instructions in m-file …

Complex Eigenvalues

- Example: complex eigenvalues and eigenvectors from state equation

$$\bar{x}(t) == \begin{bmatrix} 0.5+0.5j \\ 0.5-j \end{bmatrix} e^{(-1+3j)t} + \begin{bmatrix} 0.5-0.5j \\ 0.5+j \end{bmatrix} e^{(-1-3j)t}$$

- Euler's Formula

$$e^{\alpha+\beta j} = e^{\alpha}\left(\cos\beta + j\sin\beta\right)$$

- Solution in terms of real variables

$$x_1(t) = \ldots = \left(\cos 3t - \sin 3t\right)e^{-t}$$

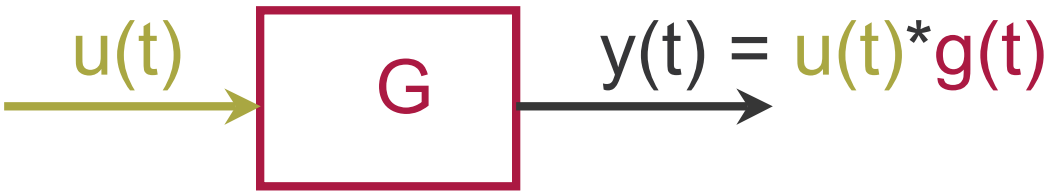
```
>> x1 =(cos(3*t)-sin(3*t))*exp(-t)
```

Convolution

- Definition

$$g(t) \otimes u(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)d\tau$$

- Example: signals & systems



Polynomial Convolution

- Polynomials:

  $x(s) = s^2 + 2s + 5; \quad y(s) = 2s^2 + s + 3$

```
>> x = [1  2  5]; y = [2  1  3]
```

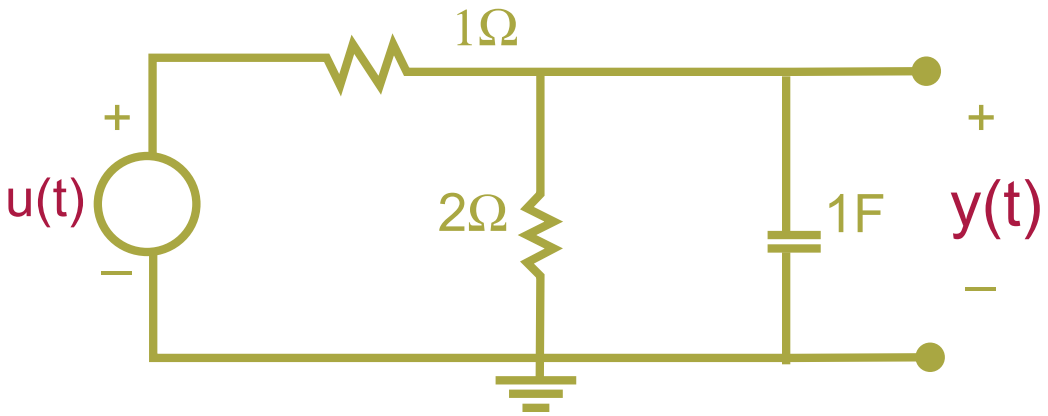
- Convolution and polynomial multiplication

```
>> w = conv(x, y)
>> w =
     2   5   15   11   15
```

- Deconvolution and polynomial division

```
>> [y, r] = deconv(w, x)
```

Example: RC Circuit



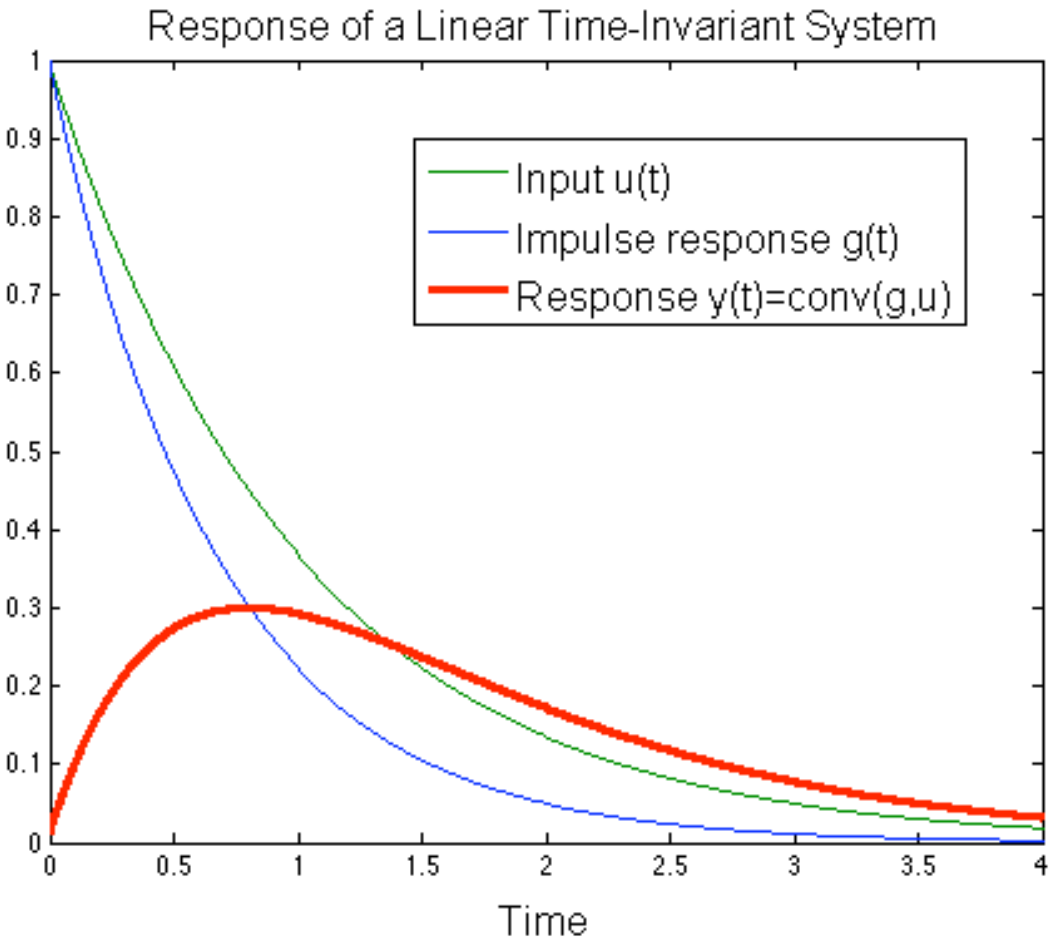
Input:

$$u(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

- Impulse response: $g(t) = \begin{cases} e^{-1.5t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$
- System response: $y(t) = \begin{cases} 2e^{-t} - 2e^{-1.5t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$

Theoretical example courtesy of Prof. Stephen Hall, MIT Department of Aeronautics & Astronautics.

Example: RC Circuit (continued)



RC Circuit: MATLAB Solution

$$0 \le t \le t_{\max}$$

$$g(t) = e^{\alpha t}$$

$$u(t) = e^{\beta t}$$

$$y(t) = g(t) \otimes u(t)$$

```
>> t  = [0 : dt : tmax]
>> nt = length(t)
>> g  = exp(α*t)
>> u  = exp(β*t)
>> y  = conv(g, u)*dt
>> y  = y(1 : nt)
>> plot(t, u, 'b-', …
        t, g, 'g-', …
        t, y, 'r-')
```

Linear Systems Exercises

- Exercise Two: `convolution.m`
  - Matrices & vectors
  - Convolution

Follow instructions in m-file …