Lecture 18 Review

Sampling theory and aliasing.

Nyquist critical frequency

FFT fun (theory and exercises)

(pick N = power of 2)

Linear Algebra Highlights

- Physics often requires solution of simultaneous linear equations.

  e.g., coupled oscillators, electrical circuits, …

  Set of equations of the form: $Ax = b$ Solve for x, A & b given.

- Physics often requires solution of eiegnevectors & eigenvalues.

  e.g., normal modes, eigenfrequencies, bound energy states, …

  Equations of the form: $Ax = \lambda x$ Solve for x, λ & A given.

  - These are 2 different classes of problems to solve.
  - Techniques are sophisticated. We will use canned software.

Solution of Linear Simultaneous Equations

Gaussian elimination. Easiest to understand.

$$\begin{array}{rcrcrcr} 2u & + & v & +w & = & 1 \\ 4u & + & v & & = & -2 \\ -2u & + & 2v & +w & = & 7 \end{array}$$

$$\begin{array}{rcrcrcr} 2u & + & v & +w & = & 1 \\ & - & 1v & -2w & = & -4 \\ & & 3v & +2w & = & 8 \end{array}$$

pivots

$$\begin{array}{rcrcrcr} 2u & + & v & +w & = & 1 \\ & - & 1v & -2w & = & -4 \\ & & & -4w & = & -4 \end{array}$$

Count number of operations

- "Forward elimination."
- "Back substitution."

Where might this technique break down?

LU Decomposition

unique

Write matrix A = LU i.e., factorize A, always OK if A has non-zero pivots

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_{10} & 1 & 0 & 0 \\ \alpha_{20} & \alpha_{21} & 1 & 0 \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & 1 \end{bmatrix} \begin{bmatrix} \beta_{00} & \beta_{01} & \beta_{02} & \beta_{03} \\ 0 & \beta_{11} & \beta_{12} & \beta_{13} \\ 0 & 0 & \beta_{22} & \beta_{23} \\ 0 & 0 & 0 & \beta_{33} \end{bmatrix}$$

pivots

$$\left.\begin{array}{l} Ax = b = (LU)x = L(Ux) = b \\ Ly = b \\ \underline{U}x = y \end{array}\right\} \text{a way to proceed.}$$

example: $\begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 8 \\ 11 \\ 3 \end{bmatrix}$. $\begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 4 \end{bmatrix}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 11 \\ 3 \end{bmatrix} \quad \text{or} \quad c = \begin{bmatrix} 8 \\ -5 \\ -4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -5 \\ -4 \end{bmatrix} \quad \text{or} \quad x = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

LU Decomposition (2)

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_{10} & 1 & 0 & 0 \\ \alpha_{20} & \alpha_{21} & 1 & 0 \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & 1 \end{bmatrix} \begin{bmatrix} \beta_{00} & \beta_{01} & \beta_{02} & \beta_{03} \\ 0 & \beta_{11} & \beta_{12} & \beta_{13} \\ 0 & 0 & \beta_{22} & \beta_{23} \\ 0 & 0 & 0 & \beta_{33} \end{bmatrix}$$

Ax = b =(LU)x = L(Ux) = b
Ly = b
<u>Ux</u> = y

a way to proceed.

$$y_0 = \frac{b_0}{\alpha_{00}}$$

"forward substitution"

$$y_i = \frac{1}{\alpha_{ii}}\left[b_i - \sum_{j=0}^{i-1} \alpha_{ij} y_j\right] \qquad i = 1, 2, \ldots, N-1$$

$$x_{N-1} = \frac{y_{N-1}}{\beta_{N-1,N-1}}$$

"back substitution"

$$x_i = \frac{1}{\beta_{ii}}\left[y_i - \sum_{j=i+1}^{N-1} \beta_{ij} x_j\right] \qquad i = N-2, N-3, \ldots, 0$$

L & U computed <u>once</u> per A

N^3 steps to solve for x.

LU Decomposition (3)

What to do if A has zero pivots?

If A has an inverse (i.e., is “non-singular”),

reorder rows of A beforehand to prevent zero pivots A $\rightarrow$ PA

PA = LU

P = “permutation matrix” (reorders rows of A)

$$P_{24} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

e.g., swaps row 2 & 4 of A

PAx = Pb has same solution x as Ax = b. (Formally, $x = A^{-1}b$)

Proof: $x = (PA)^{-1}Pb = A^{-1}P^{-1}Pb = A^{-1}b$

Q: If Ax = b, why not just compute $x = A^{-1}b$?

A: Computing A-1 is more “expensive” than computing LU.

Linear Algebra and Octave

Exercise (from chemistry !): $\alpha\, O_2 + \beta\, C_4H_9NH_2 \rightarrow \gamma\, CO_2 + \delta\, H_20 + N_2$

Find correct stoichiometry (i.e., find α, β, γ, δ).

Often need to perform matrix calculations quickly (i.e., w/o writing code)

Use `octave` (freeware)

```
prompt> octave
octave:1>
```

Use as caclulator: `2/83`

Standard set of math functions:

cos	cosine
exp	exponential
log	Natural log
log10	Log base 10
tanh	Hyperbolic tangent
atan	Arc-tangent
round	Round to nearest integer

Colon notation:

```
octave:31>: e= 2:6
octave:31>: f= 2:6:40
```

Semicolon usage:

```
octave:31>: f= 2:6;
```

Octave (2)

Extensive help utility: try `help -i`

Matrix manipulation

```
octave:45> a= [ 1,3;2,7]
octave:46> a'
octave:47> f = [ 1:6]'
```

Built-in functions for large matrices.

```
octave:51>: s = zeros(M,N)
```
w/ M,N = integers

```
octave:52>: r = ones(M,N)
octave:53>: rr = linspace(x1,x2,N)
octave:53>: r = logspace(x1,x2,N)
```

Octave (3)

Plotting: basic command is `plot(x,y)`

uses gnuplot

```
octave:85> angles = [0:pi/3:2*pi];
octave:87> y = sin(angles)
octave:88> plot (angles, y)
```

Functions:

name

output

input

```
octave:151>: function s = dub(x)
           >   s = 2*x;
           > end
octave:152>: dub(35)
```

Octave notes

$Ax = b$

Define A in usual way.

`octave:3> a = [1,3,5; 1, 5, 6; 3, 7, 9]` for example.

`octave:4> b = [2, 5,9]’` for example.

`octave:4> x = a\b`

a\ is octave speak for a $^{-1}$.

octave does NOT compute the inverse of a to solve for x.

Many variants to LU decomposition.

These depend on structure of A: degree of symmetry, sparseness, …

Linear Algebra Highlights

- Physics often requires solution of eiegnevectors & eigenvalues.
  e.g., normal modes, eigenfrequencies, bound energy states, …

Equations of the form: $Ax = \lambda x$ Solve for λ & x, A is given.
Nonlinear equation.

$(A - \lambda I)x = 0$ Eigenvectors x lie in the "nullspace" of A - λI

For λ to be an eigenvalue of A:

1) non-zero x for which Ax = λx
2) A - λI is singular
3) det(A - λI) = 0

Each is necessary and sufficient

#3 implies sum of n eigenvalues of A = sum of diagonal entries of A

$$\begin{Vmatrix} (a_{11}-\lambda) & a_{12} & a_{13} \\ a_{21} & (a_{22}-\lambda) & a_{23} \\ a_{31} & a_{32} & (a_{33}-\lambda) \end{Vmatrix} = (a_{11}-\lambda)[(a_{22}-\lambda)(a_{33}-\lambda)-a_{32}a_{23}] + \cdots = 0$$

Eigenvalues

#3 also implies:

If A is triangular (lower <u>or</u> upper), λ's appear on diagonal of A

$$\begin{Vmatrix} (a_{11}-\lambda) & a_{12} & a_{13} \\ 0 & (a_{22}-\lambda) & a_{23} \\ 0 & 0 & (a_{33}-\lambda) \end{Vmatrix} = (a_{11}-\lambda)(a_{22}-\lambda)(a_{33}-\lambda) = 0$$

Now, suppose the n X n matrix A has n linearly independent eigenvectors

Then if you write them as the column vectors of a matrix S

$S^{-1}AS$ is diagonal w/ the λ's of A along the diagonal:

$$S^{-1}AS = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

Eigenvalues (2)

Here's why $S^{-1}AS = \Lambda$

$$AS = A\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \\ | & | & & | \end{bmatrix}$$

$$\begin{bmatrix} | & | & & | \\ \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

$AS = S\Lambda$ or $S^{-1}AS = \Lambda$

"similarity transformation"

Furthermore, for any non-singular matrix M:
If $B = M^{-1}AM$ then A & B have same λ's w/ same multiplicities.

Here's why:

$$\det(B - \lambda I) = \det(M^{-1}AM - \lambda I) = \det(M^{-1}(A - \lambda I)M)$$
$$= \det M^{-1} \det(A - \lambda I) \det M = \det(A - \lambda I)$$

Strategy for Finding λ and x

Strategy to find λ's and eigenvectors x of A.

Perform similarity transformations to diagonalize A

$P^{-1}AP = A'$ ← Diagonal (A and A' have same eigenvalues)

$P \equiv P_1P_2P_3 \ldots$ P could be a product of many transformations.

$$P^{-1}AP = \ldots P_3^{-1}P_2^{-1}(P_1^{-1}AP_1)P_2P_3\ldots$$

Amazingly, we can get eigenvectors from P

Suppose the n eigenvectors u_i of A are linearly indpt and are a basis.

$$v_i \equiv P^{-1}u_i$$

$$A'v_i = (P^{-1}AP)(P^{-1}u_i) = P^{-1}Au_i = \lambda v_i$$

$$v_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Only non-zero term (remember, A' is diagonal)

Now note: $Pv_i = P(P^{-1}u_i) = u_i$

eigenvectors of A = columns of P !

Examples of P-type Transformations

$$P = \begin{bmatrix} 1 & & & & & \\ \ldots & \cos\theta & \ldots & \sin\theta & & \\ & \vdots & 1 & \vdots & & \\ & -\sin\theta & \ldots & \cos\theta & & \\ & & & & & 1 \end{bmatrix}$$

Works on some elements of A

Seek to eliminate off-diagonal terms

Also sometimes useful to factorize A: A = PQ

then note: QP = $(P^{-1} A)P$

similarity transformation

Octave Calisthenics

$$A = \begin{bmatrix} 1 & 5 \\ 2 & 8 \end{bmatrix}$$

Q: What is the sum S of the eigenvalues?

Q: What are the eigenvalues? Solve first by hand, then by `octave`.

FYI: octave:19> `help -i eig` will be helpful !!

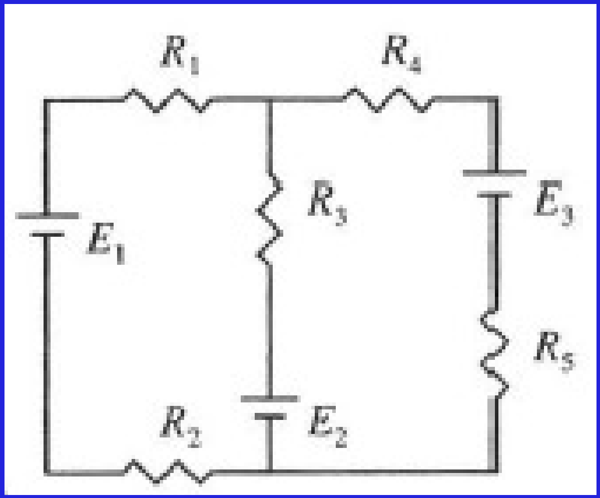
Q: What is the LU decomposition of A?

NB: `octave` computes A = PLU w/ 1's along diagonal of L

Q: What are the eigenvectors of A ?

Q: What is A^{-1} ? Verify this.

Octave Calisthenics (2)



R1 = R2 = 1Ω

R3 = R4 = 2Ω

R5 = 5Ω

E1 = 2V

E3 = 5V

E2 = 10V

Find I in all legs.

Octave 1st ODE

Recall Volterra prey-predator equations

$$\begin{cases} dx/dt = x(a - by) \\ dy/dt = -y(c - dx) \end{cases}$$

a = 1.0, b = 0.5, c = 0.95, d = 0.25

Solved previously w/ GSL routines.

Can be solved w/ `octave`

vectors

```
function xdot = vp(x,t)
xdot = zeros(2,1);
a = 1.0;
b = 0.5;
c = 0.95;
d = 0.25;
xdot(1) = x(1)*(a - b* x(2));
xdot(2) = -x(2)*(c - d*x(1));
endfunction
```

scalar

$$\frac{dx}{dt} = f(x,t)$$

1st ODE Solution via octave (2)

Set initial conditions: `x0 = [5; 5]`

column vector

```
t =  linspace(0, 500, 1000);
y =  lsode("vp", x0, t);
```

column vector is returned

"x" variable

```
plot(t,y)
```

"y" variable

Change ICs. What do you see?

Octave 1st ODE (3)

For convenience, functions can be placed in files.

Example: vp.m (the "m" extension is for compatibility w/ MATLAB)

```
# example of function file
function xdot = vp(x,t)
xdot = zeros(2,1);
    a = 1.0;
    b = 0.5;
    c = 0.95;
    d = 0.25;
    xdot(1) = x(1)*(a - b* x(2));
    xdot(2) = -x(2)*(c - d*x(1));
endfunction
```

Usage: `octave:41> vp`

Note: octave will throw errors (b/c x is not specified) . Ignore them.

Planetary Orbits via `octave`

Solve for Earth's motion around the Sun using `octave`.

Produce a plot showing orbit !!

Summary

Gaussian elimination.

LU decomposition

`octave` intro

Finding eigenvalues and eigenvectors

`octave` calisthenics

Solving ODEs w/ `octave`

Don't suffer in silence. Scream for help!!!