Introduction to Numerical Methods in Linear Algebra

Larry Caretto
Mechanical Engineering 501A

Seminar in Engineering Analysis

September 20, 2017

California State University Northridge

Outline

- Review last class
- Introduction to numerical methods
- Finite representation of numbers
- Error propagation and round-off error
- Practical considerations in solutions
- Use of pivoting to improve accuracy in solutions of simultaneous linear algebraic equations

California State University Northridge 2

Review Eigens

- Eigenvalues and eigenvectors: $\mathbf{Ax} = \lambda\mathbf{x}$
- Computations using Det[$\mathbf{A} - \mathbf{I}\lambda$] = 0
- Eigenvectors determined by [$\mathbf{A} - \mathbf{I}\lambda$]$\mathbf{x}$ = $\mathbf{0}$ only to a multiplicative constant
- Define $\mathbf{X}$ as matrix where each column is an eigenvector
- Transformations with a matrix of eigenvectors: $\Lambda = \mathbf{X}^{-1}\mathbf{AX}$

California State University Northridge 3

Review Other Matrices

- Orthogonal matrices had orthonormal rows and orthonormal columns
- The inverse of an orthogonal matrix is its transpose
- The eigenvalues of a Hermitian matrix ($\mathbf{A}^* = \mathbf{A}^T$) are real
- An n x n Hermitian matrix has n linearly independent, orthogonal eigenvalues that can form a unitary matrix

California State University Northridge 4

Computer Representations

- Computer is binary machine
  - Numbers represented as series of zeros and one
  - Basic forms are integers and floating point
  - Integers numbers have small range, but exact representation, used for counting
  - Floating point numbers have wide range, but inexact representation
  - Accuracy and range depend on word size

California State University Northridge 5

Representing Integers

- Represented as binary number with offset for negative numbers
- Typical computer uses 32 bits (4 bytes) for integer giving range of 0 to $2^{32} - 1$
- Offsets give range from -2^{31} to $2^{31} - 1$
- Adding one to maximum integer gives minimum integer: $(2^{31} - 1) + 1 = -2^{31}$
- Different computers/compilers have different sizes and signed/unsigned

California State University Northridge 6

Floating Point Numbers

- Typical sizes are 4 bytes for single precision and 8 for double precision
- Number has sign bit, exponent (characteristic) and mantissa
- IEEE 754 (1985) standard for double
  - 11 bits for exponent
  - 1 sign bit
  - 53 effective bits for mantissa because leading one is not stored

California State University Northridge 7

Machine Epsilon

- Smallest ε value such that $1 + \varepsilon \neq 1$
- Depends on mantissa bits
- Usually 1.19×10^{-7} for single and 2.22×10^{-16} for double
  - Single almost seven significant figures
  - Double has fifteen-plus significant figures
- Is 10 times 0.1 = 1? Maybe
  - $0.1_{10} = 0.0001100110011001100110011\ldots_2$
  - $0.1_{10} = 1.10011001100110011001_2 \times 10_2^{-100}{}_2$

California State University Northridge 8

Rounding Error

- Due to inexact representation
- Store floating point representation $\tilde{x}$ of actual number x with error ε_x
- Errors propagate in multiple calculations

$$x = \tilde{x} + \varepsilon_x \qquad y = \tilde{y} + \varepsilon_y$$

$$x \pm y = \tilde{x} + \varepsilon_x \pm (\tilde{y} + \varepsilon_y) = \tilde{x} \pm \tilde{y} + (\varepsilon_x + \varepsilon_y)$$

$$\varepsilon_{rel} \equiv \frac{(x \pm y) - (\tilde{x} \pm \tilde{y})}{x \pm y} = \frac{\varepsilon_x + \varepsilon_y}{x \pm y} \approx \frac{\varepsilon_x + \varepsilon_y}{\tilde{x} \pm \tilde{y}}$$

California State University Northridge 9

Multiplication/Division Error

- Have same result for each although definitions and derivations are different
  - Definitions

$$\varepsilon_{rel} = \frac{xy - \tilde{x}\tilde{y}}{xy} \qquad \varepsilon_{rel} = \frac{\frac{x}{y} - \frac{\tilde{x}}{\tilde{y}}}{\frac{x}{y}}$$

  - Common result $\varepsilon_{rel} = \frac{\varepsilon_x}{\tilde{x}} + \frac{\varepsilon_y}{\tilde{y}}$

California State University Northridge 10

Avoiding Round-off Error

- Use higher precision data types
- Beyond single and double there is extended precision with some compilers
- Systems of equations with large number of equations or nearly singular systems can require double precision
- Algorithms can be designed to reduce round-off error

California State University Northridge 11

Quadratic Real Roots

$$x_1 x_2 = \frac{c}{a}$$

```
disc = b * b – 4 * a * c;
if ( disc >= 0 )
{  // real solution here
  if ( b > 0 )
     x1 = (-b – sqrt(disc) ) / ( 2 * a );
  else
     x1 = (-b + sqrt(disc) ) / ( 2 * a );

  x2 = c / ( a * x1 );
}
```

California State University Northridge 12

Gauss Elimination

- Covered previously while discussing solution of systems of equations
- Analytical tool for determining linear dependence or independence
- Basic idea is to manipulate the equations (or data) to make them easier to solve without changing the results
- Systematically create zeros in lower left part of the equations (or data)

California State University Northridge 13

Upper Triangular Form

- Convert original set of equations to

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & \cdots & \alpha_{1n-1} & \alpha_{1n} \\ 0 & \alpha_{22} & \alpha_{23} & \cdots & \cdots & \alpha_{2n-1} & \alpha_{2n} \\ 0 & 0 & \alpha_{33} & \cdots & \cdots & \alpha_{3n-1} & \alpha_{3n} \\ \vdots & \vdots & \vdots & \ddots & & & \\ \vdots & \vdots & \vdots & & \ddots & & \\ 0 & 0 & 0 & \cdots & \cdots & \alpha_{n-1n-1} & \alpha_{n-1n} \\ 0 & 0 & 0 & \cdots & \cdots & 0 & \alpha_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix}$$

California State University Northridge 14

Back Substitution

- Upper triangular form on previous slide is easily solved by back substitution
- $x_n = \beta_n/\alpha_{nn}$
- $x_{n-1} = (\beta_{n-1} - \alpha_{n-1\,n} x_n)/\alpha_{n-1\,n-1}$, *et cetera*
- General equation for back substitution

$$x_i = \frac{\beta_i - \sum_{j=i+1}^{n} \alpha_{ij} x_j}{\alpha_{ii}} \qquad i = n-1, n-2, \ldots, 1$$

California State University Northridge 15

Gauss Elimination Algorithm

- Work on ***augmented matrix***
- Can handle several **b** vectors at one time

$$[\mathbf{A}, \mathbf{b}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & \cdots & a_{1m} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & \cdots & a_{2m} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & \cdots & a_{3m} & b_3 \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & \cdots & a_{nm} & b_n \end{bmatrix}$$

California State University Northridge 16

General Gauss Elimination

- Use each row from row 1 to row n-1 as the "pivot" row
  - Work on each row below the pivot row
    - Multiply pivot row by $a_{row,pivot}/a_{pivot,pivot}$
    - Subtract result from row "row" to make $a_{row,pivot} = 0$
    - Operation requires subtraction for each column of **A** right of pivot column and for **b**
  - Repeat for each row below pivot (except last)
- Use back substitution for **x** values
- Worry about round-off error that depends on the "condition" of the **A** matrix

California State University Northridge 17

Problem Condition

- The condition of a problem is defined as the relative change in result divided by a relative change in input
- A large condition number indicates a problem that may give rise to numerical difficulty
- In solution of one equation in one unknown f(x) = 0 a small value of df/dx near the root can cause problems

California State University Northridge 18

Solving Nonlinear Equations

- Have a system of N simultaneous nonlinear equations written as **f**(**x**) = **0**
- Multivariable Newton's method

$$\mathbf{J}^{(n)} = \left.\frac{\partial f_k}{\partial x_m}\right|^{(n)}$$

$$\mathbf{f}\left(\mathbf{x}^{(n+1)}\right) = \mathbf{f}\left(\mathbf{x}^{(n)}\right) + \mathbf{J}^{(n)}\left(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}\right) + \cdots$$

$$\begin{bmatrix} f_1(\mathbf{x}^{(n+1)}) \\ f_2(\mathbf{x}^{(n+1)}) \\ f_3(\mathbf{x}^{(n+1)}) \\ \vdots \\ \vdots \\ f_N(\mathbf{x}^{(n+1)}) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}^{(n)}) \\ f_2(\mathbf{x}^{(n)}) \\ f_3(\mathbf{x}^{(n)}) \\ \vdots \\ \vdots \\ f_N(\mathbf{x}^{(n)}) \end{bmatrix} + \begin{bmatrix} J_{11} & J_{12} & J_{13} & \cdots & \cdots & J_{1N} \\ J_{21} & J_{22} & J_{23} & \cdots & \cdots & J_{2N} \\ J_{31} & J_{32} & J_{33} & \cdots & \cdots & J_{3N} \\ \vdots & \vdots & \vdots & \ddots & & \\ \vdots & \vdots & \vdots & & \ddots & \\ J_{N1} & J_{N2} & J_{N3} & \cdots & \cdots & J_{NN} \end{bmatrix} \begin{bmatrix} x_1^{(n+1)} - x_1^{(n)} \\ x_2^{(n+1)} - x_2^{(n)} \\ x_3^{(n+1)} - x_3^{(n)} \\ \vdots \\ \vdots \\ x_N^{(n+1)} - x_N^{(n)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

California State University Northridge 19

Solving Nonlinear Equations II

- Have to solve N simultaneous linear equations at each iteration with **f**(**x**$^{(n+1)}$) = 0

$$f_k(\mathbf{x}^{(n+1)}) = f_k(\mathbf{x}^{(n)}) + \sum_{m=1}^{N} J_{km}\left(x_m^{(n+1)} - x_m^{(n)}\right) = f_k(\mathbf{x}^{(n)}) + \sum_{m=1}^{N} \left.\frac{\partial f_k}{\partial x_m}\right|^{(n)} \left(x_m^{(n+1)} - x_m^{(n)}\right)$$

$$\sum_{m=1}^{N} J_{km}\left(x_m^{(n+1)} - x_m^{(n)}\right) = \sum_{m=1}^{N} \left.\frac{\partial f_k}{\partial x_m}\right|^{(n)} \left(x_m^{(n+1)} - x_m^{(n)}\right) = -f_k(\mathbf{x}^{(n)})$$

- Iterations are affected by physical system through partial derivatives

California State University Northridge 20

Solving Nonlinear Equations III

- Thermodynamic property equations are functions of temperature and density
- Want to define state by other properties (e. g., pressure and entropy)
- Newton's method iteration equations

$$\frac{\partial s}{\partial T}\left(T^{(n+1)} - T^{(n)}\right) + \frac{\partial s}{\partial v}\left(v^{(n+1)} - v^{(n)}\right) = s_0 - s\left(T^{(n)}, v^{(n)}\right)$$

$$\frac{\partial P}{\partial T}\left(T^{(n+1)} - T^{(n)}\right) + \frac{\partial P}{\partial v}\left(v^{(n+1)} - v^{(n)}\right) = P_0 - P\left(T^{(n)}, v^{(n)}\right)$$

- Solve by Cramer's rule

California State University Northridge 21

Solving Nonlinear Equations IV

$$\left(T^{(n+1)} - T^{(n)}\right) = \frac{\frac{\partial P}{\partial v}\left[s_0 - s\left(T^{(n)}, v^{(n)}\right)\right] - \frac{\partial s}{\partial v}\left[P_0 - P\left(T^{(n)}, v^{(n)}\right)\right]}{\frac{\partial s}{\partial T}\frac{\partial P}{\partial v} - \frac{\partial s}{\partial v}\frac{\partial P}{\partial T}}$$

$$\left(v^{(n+1)} - v^{(n)}\right) = \frac{\frac{\partial s}{\partial T}\left[P_0 - P\left(T^{(n)}, v^{(n)}\right)\right] - \frac{\partial P}{\partial T}\left[s_0 - s\left(T^{(n)}, v^{(n)}\right)\right]}{\frac{\partial s}{\partial T}\frac{\partial P}{\partial v} - \frac{\partial s}{\partial v}\frac{\partial P}{\partial T}}$$

- Large partial derivative makes solution ill conditioned $\frac{\partial P}{\partial v}$

California State University Northridge 22

Condition of a Matrix I

$$Condition = \left| \frac{\dfrac{Change\ in\ result}{result}}{\dfrac{Change\ in\ input}{input}} \right|$$

- Need norms to represent size $\|\mathbf{A}\| = \max_{\|\mathbf{x}\|} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$
- Use matrix norm that is consistent with definition of vector norm

$$\|\mathbf{A}\| \geq \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \Rightarrow \|\mathbf{A}\|\|\mathbf{x}\| \geq \|\mathbf{A}\mathbf{x}\|$$

California State University Northridge 23

Condition of a Matrix II

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\|} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$$

- Both **Ax** and **x** are matrices
- Can use any matrix norm to compute ||**A**||
- Choosing infinity norm as vector norm gives row sum $\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$
- Choosing one norm as vector norm gives column sum $\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$

California State University Northridge 24

Condition of a Matrix III

- Correct and incorrect solutions are **x** and $\tilde{\mathbf{x}}$
- Computable error residual, **r** = **b** - **A** $\tilde{\mathbf{x}}$

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}) = \mathbf{r}$$

$$(\mathbf{x} - \tilde{\mathbf{x}}) = \mathbf{A}^{-1}\mathbf{r} \quad \Rightarrow \quad \|\mathbf{x} - \tilde{\mathbf{x}}\| = \|\mathbf{A}^{-1}\mathbf{r}\| \le \|\mathbf{A}^{-1}\|\|\mathbf{r}\|$$

$$\|\mathbf{b}\| = \|\mathbf{A}\mathbf{x}\| \le \|\mathbf{A}\|\|\mathbf{x}\| \quad \Rightarrow \quad \frac{1}{\|\mathbf{x}\|} \le \frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}$$

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \|\mathbf{A}^{-1}\|\|\mathbf{r}\|\frac{1}{\|\mathbf{x}\|} \le \|\mathbf{A}^{-1}\|\|\mathbf{r}\|\frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}$$

California State University Northridge 25

Condition of a Matrix IV

- Define the relative size of the residual as ||**r**|| / ||**b**|| (which we can calculate)

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} = \kappa(\mathbf{A})\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$$

- Condition number $\kappa(\mathbf{A}) \equiv \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$
- Small is < 10; large is about 100 or more
- Expect large condition numbers to create problems in solutions

California State University Northridge 26

Condition of a Matrix V

- Ill conditioning comes from near linear dependence in matrix rows

$$\mathbf{A} = \begin{bmatrix} 1.00001 & 0.99999 \\ 1 & 1 \end{bmatrix} \quad \mathbf{A}^{-1} = \begin{bmatrix} 50000 & -49999.5 \\ -50000 & 50000.5 \end{bmatrix}$$

$\|\mathbf{A}\|_\infty = 2$ $\|\mathbf{A}\|_1 = 2.00001$
$\|\mathbf{A}^{-1}\|_\infty = 100000.5$ $\|\mathbf{A}^{-1}\|_1 = 100000$

Condition number = 200001 confirming ill conditioning apparent from original **A**

California State University Northridge 27

Condition of a Matrix VI

- Equations [B-5] and [B-12] in notes show that changes or errors in **A** or **b** affect solution through the condition number

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x} + \delta\mathbf{x}\|} \approx \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \kappa(\mathbf{A})\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}$$

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \kappa(\mathbf{A})\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

California State University Northridge 28

Reducing Round-off Error

- Example of equation order for problem with known solution: x = 2/3, y = 1/3

original order ***order reversed***

$$\begin{bmatrix} 0.00003 & 3 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.0002 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 0.00003 & 3 \end{bmatrix}\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 1.0002 \end{bmatrix}$$

- Gauss elimination gives

$$\begin{bmatrix} 0.00003 & 3 \\ 0 & -9999 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.0002 \\ 1 - \frac{1.0002}{0.0003} \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 0 & 2.9997 \end{bmatrix}\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 0.9999 \end{bmatrix}$$

- Look at effect of precision on solutions

California State University Northridge 29

N = Number of Significant Figures

N	Original problem		Equations reversed	
	x	y	x	y
5	.33333	.70000	.33333	.66667
6	.333333	.670000	.333333	.666667
7	.3333333	.6670000	.3333333	.6666667
8	.33333333	.66670000	.33333333	.66666667

California State University Northridge 30

What Happened?

Original

$$\begin{bmatrix} 0.00003 & 3 \\ 0 & -9999 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.0002 \\ 1 - \frac{1.0002}{0.0003} \end{bmatrix}$$

Rows Reversed

$$\begin{bmatrix} 1 & 1 \\ 0 & 2.9997 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 0.9999 \end{bmatrix}$$

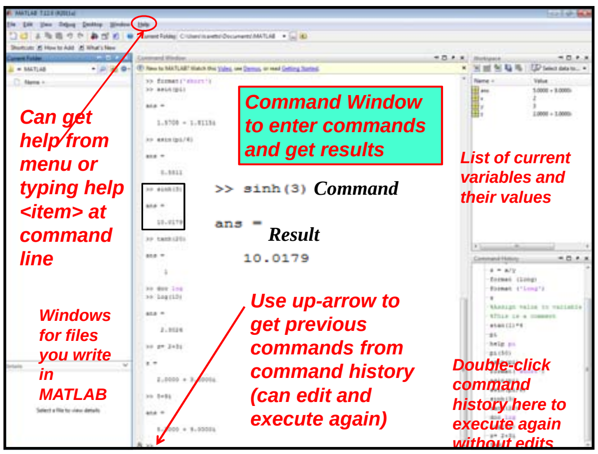
- Problem is in the 1 – 1.0002/0.0003 term
- Division inaccuracy swamps subtraction
- Try to have large elements on pivot row to avoid such divisions
- Pivoting strategy: Use row with maximum (scaled) element as pivot row

California State University Northridge 31

More on Pivoting

- Can exchange rows or columns to get maximum element on pivot
- Exchanging rows only is easier and often effective
  - Commonly used in earlier programs
  - Exchanging columns changes the identity of the unknowns requiring more computer work to keep track of changes
- Newer software now exchanges both

California State University Northridge 32



Entering Arrays

- Enter a row vector by enclosing data in [ ] separated by a space

>> row = [12 -3 5 7 0]
row =
12 -3 5 7 0

- ***Enter a column vector by enclosing data, separated by a semicolon (;) in [ ]***

>> col = [-3; 6; 0; 4]
col =
-3
6
0
4

California State University Northridge 34

Entering a Matrix

- Enter matrix data row by row
- Put spaces between data in the same row
- Put a semicolon to start data on next row
  - MATLAB uses the … as a continuation signal
  - After the … hit Enter and continue input of same command on a new line

```
>> A=[1 2; 3 4]
A =
     1     2
     3     4
>> B = [1 2 3; 4...
        5 6; 7 8...
        9]
B =
     1     2     3
     4     5     6
     7     8     9
```

California State University Northridge 35

Entering a Matrix II

- Pressing enter after each row of data can be used to enter a matrix
- Using semicolons, all data can be placed on one row (see below)
- Continuation is only needed to start new line in the middle of data entry

```
B =      (Result)
     1     2     3
     4     5     6
     7     8     9
>> B = [1 2 3
        4 5 6
        7 8 9]
>> B = [1 2 3; 4...
        5 6; 7 8 9]
>> B = [1 2 3; 4 5 6; 7 8 9]
```

California State University Northridge 36

MATLAB Linear Solver

- For an n x n matrix **A**, and a n x m right-hand side matrix **b** MATLAB produces a n x m solution **x** = $\mathbf{A}^{-1}\mathbf{b}$ by either of the following commands
  - x = A/b
  - x = mldivide(A, b)
- Each column of the **x** result contains the solution of **Ax** = **b** for the corresponding column of the **b** array

California State University Northridge
37

MATLAB Eigenvalues/vectors

- Use function eig for both eigenvalues and eigenvectors
- Basic command is [V,D] = eig(A)
  - A is a square matrix
  - D is a diagonal matrix giving the eigen-values of A on the diagonal (**Λ** matrix)
  - V is a square matrix whose columns are the eigenvectors of A (**X** matrix)

California State University Northridge
38

Excel Data to MATLAB

\>> A = xlsread('Gaussian.xlsm', 'Data', 'B6:CW105'); *File Name* *Worksheet*

\>> b = xlsread('Gaussian.xlsm', 'Data', 'CX6:DI105'); ← *Data Range*

\>> xExact = xlsread('Gaussian.xlsm', 'Answers', 'B3:M102');

\>> x = A\b

\>> RMS = sqrt(mean(x – xExact).^2)

California State University Northridge
39