MT308 Numerical Analysis

Girne American University
Faculty of Engineering
Spring 2026
Dr. Kamil Dimililer

1

MATLAB has several built-in constants:

- **pi** returns the constant value for π
- **eps** returns the difference between 1 and the next largest double number,
- **inf** represents ∞,
- **-inf** represents−∞
- **nan** or **NaN** represents a special Not a Number value.

```
>> 0/0
ans = NaN
>> 3/0
ans = Inf
>> -3/0
ans = -Inf
>> eps
ans = 2.2204e-16
>> pi
ans = 3.1416
>> cos(pi/4)
ans = 0.7071
```

2

A list of basic MATLAB matrix construction routines

eye Identity matrix (1 on diagonal, 0 elsewhere)
zeros Matrix filled with 0
ones Matrix filled with 1
rand Uniformly distributed numbers between 0 and 1
randn Normally distributed numbers between 0 and 1

The arguments may …
Single scalar (N) — Generates a N × N matrix
Two scalars (N and M) — Generates an N × M matrix

3

Some useful matrix operations:

- D = diag(v) returns a square diagonal matrix with the elements of vector v on the main diagonal.
- x = diag(A) returns a column vector of the main diagonal elements of A.

```
>> v=[4 5 6 7]
v =

   4   5   6   7

>> A=diag(v)
A =

Diagonal Matrix

   4   0   0   0
   0   5   0   0
   0   0   6   0
   0   0   0   7
```

```
>> B=5*rand(3,4)
B =

   0.9865   3.2215   4.9346   1.2829
   4.5367   3.0771   2.6420   1.7821
   4.4703   4.3196   2.0001   3.7226

>> s=diag(B)
s =

   0.9865
   3.0771
   2.0001
```

4

A norm in numerical methods is a function that assigns a non-negative, real-valued "magnitude" or "length" to a vector or matrix, essential for measuring error, stability, and convergence.

- **Vector Norms ($\|\boldsymbol{x}\|$):** Measure the size of a vector.
  - **1-norm ($\|\boldsymbol{x}\|_1$):** Sum of absolute values: $\sum_{i=1}^{n}|x_i|$.
  - **2-norm ($\|\boldsymbol{x}\|_2$):** Square root of the sum of squares: $\sqrt{\sum_{i=1}^{n}|x_i|^2}$.
  - **Infinity norm ($\|\boldsymbol{x}\|_\infty$):** Maximum absolute value: $\max|x_i|$.
- **Matrix Norms ($\|\boldsymbol{A}\|$):** Measure the size of a matrix and are often induced by vector norms, meaning $\|A\| = \sup_{x\neq 0}\frac{\|Ax\|}{\|x\|}$.
  - **1-norm (Maximum column sum):** $\max_j\sum_{i=1}^{n}|a_{ij}|$.
  - **Infinity norm (Maximum row sum):** $\max_i\sum_{j=1}^{n}|a_{ij}|$.
  - **Frobenius norm:** Square root of sum of absolute squares of all elements: $\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$.

5

Some useful matrix operations:

- n = norm(x) returns the Euclidean norm of vector x
- n = norm(x,p) returns the generalized vector p-norm.
- n = norm(X) returns the 2-norm or maximum singular value of matrix X, which is approximately max(svd(X)).
- n = norm(X,p) returns the p-norm of matrix X, where p is 1, 2, or Inf:
  - If p = 1, then n is the maximum absolute column sum of the matrix.
  - If p = 2, then n is approximately max(svd(X)). This value is equivalent to norm(X).
  - If p = Inf, then n is the maximum absolute row sum of the matrix.

$$|\mathbf{x}|_2 = |\mathbf{x}| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

$$|\mathbf{x}|_p \equiv \left(\sum_i |x_i|^p\right)^{1/p}.$$

```
>> v=[3,4]
v =

   3   4

>> norm(v)
ans = 5
```

```
>> A=[2,4,5;1 2 3]
A =

   2   4   5
   1   2   3

>> norm(A)
ans = 7.6756
```

6

Iterative Methods to Solve Linear Systems

- Iterative methods are useful if the size of matrix is huge and if there are many 0 elements. During the iterations matrix is not changed so sparcity is preserved
- Starts with an initial approximation for the solution vector (x^0)
- At each iteration updates the x vector by using the sytem Ax=b
- Each iteration involves a matrix-vector product
- Write the system Ax=b in an equivalent form **x=Ex+f**
- (like x=g(x) for fixed-point iteration)
- Starting with x^0, generate a sequence of approximations $\{x^k\}$ iteratively by **x^{k+1}=Ex^k+f**

7

7

Jacobi's Method

The Jacobi iterative method is obtained by solving the ith equation in Ax = b for x_i to obtain (provided $a_{ii} \neq 0$)

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-\frac{a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \ldots, n.$$

For each k ≥ 1, generate the components $x_i^{(k)}$ of $x^{(k)}$ from the components of $x^{(k-1)}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij}x_j^{(k-1)} \right) + b_i \right], \qquad \text{for } i = 1, 2, \ldots, n.$$

8

```
function x = jacobi_it(A,b,x0,tol)
n = length(b);
x = x0;
while norm(A*x - b, inf) > tol
    x_new = zeros(n,1);
    for i = 1:n
        s = A(i,1:i-1)*x(1:i-1) + A(i,i+1:n)*x(i+1:n);
        x_new(i) = (b(i) - s)/A(i,i);
    end
    x = x_new;
end
end
```

9

Gauss Seidel Method

It is similar to the Jacobi Method
In the Gauss–Seidel Method, updated values of the unknowns are used at each step, even if the updating occurs in the current step.

Remember the example

$$3u + v = 5, u + 2v = 5$$

$$u = \frac{5 - v}{3}$$
$$v = \frac{5 - u}{2}.$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix}$$
$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-10/9}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix}$$
$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-35/18}{3} \\ \frac{5-55/54}{2} \end{bmatrix} = \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix}.$$

10

Example:

Apply the Gauss–Seidel Method to the system

$$\begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}.$$

The Gauss–Seidel iteration is

$$u_{k+1} = \frac{4 - v_k + w_k}{3}$$
$$v_{k+1} = \frac{1 - 2u_{k+1} - w_k}{4}$$
$$w_{k+1} = \frac{1 + u_{k+1} - 2v_{k+1}}{5}.$$

11

Solution:

Starting with $x_0 = [u_0, v_0, w_0] = [0, 0, 0]$, we calculate

$$\begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} \frac{4-0-0}{3} = \frac{4}{3} \\ \frac{1-8/3-0}{4} = -\frac{5}{12} \\ \frac{1+4/3+5/6}{5} = \frac{19}{30} \end{bmatrix} \approx \begin{bmatrix} 1.3333 \\ -0.4167 \\ 0.6333 \end{bmatrix}$$

and

$$\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{101}{60} \\ -\frac{3}{4} \\ \frac{251}{300} \end{bmatrix} \approx \begin{bmatrix} 1.6833 \\ -0.7500 \\ 0.8367 \end{bmatrix}.$$

The system is strictly diagonally dominant, and therefore the iteration will converge to the solution $[2, -1, 1]$. ◀

12