1. **Quote.** "Success is the ability to go from one failure to another with no loss of enthusiasm."

   (Winston Churchill (1874-1965))

2. **Review of Linear Systems.**

   (a) A linear system is a set of equations:

$$E_1 : a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1$$
$$E_2 : a_{21}x_1 + a_{22}x_2 + ... + a_{1n}x_n = b_2$$

$$E_n : a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n$$

   (b) The a_{ij}'s and b_i's are constants and the x_i's are unknowns.

   (c) The goal is, given a_{ij}'s and b_i's, to solve for $x_1, x_2, ..., x_n$.

3. **Overview of the Section.**

   (a) First we use Direct Methods.

   (b) We analyze how well they work.

   (c) We determine how to make improvements.

4. **Matrix and Vector Notation.**

   (a) An nxm matrix is a rectangular array of numbers with n rows and m columns.

   (b) An n-dimensional column vector is an nx1 matrix.

   (c) An n-dimensional row vector is an 1xn matrix.

5. **Example.** Matrices and Vectors are the building blocks of Matlab. How do we use them in Matlab?

6. Writing a Linear System in Matrix-Vector Form

7. Writing a Linear System in Matrix-Vector Form - Augmented Form

8. **Solving Linear Systems in Matlab**

   (a) The standard routine is backslash.

   (b) Backslash implements a version of Gaussian elimination (coming soon!).

   (c) Backslash is optimized for **accuracy** but not **efficiency**.

   (d) Matlab also has a command to find the inverse of A, written A^{-1}. Using A^{-1} to solve for x **should not** be used in practice.

9. **MATLAB demo - LSRandom.m** (Canvas lecture notes page).Use Gaussian elimination on a random matrix and observe the error.

```
N=10; % Matrix size
Ntr=100; % Number of trials

errs=zeros(Ntr,1); % Vector of errors
x=ones(N,1); % exact solution vector

for i=1:Ntr

    A=randn(N,N); % Construct a random NxN matrix (normally distributed)
    b=A*x; % Compute the right-hand side vector
    z=A\b; % Solve the linear system

    errs(i)=max(abs(z-x)); % Compute the error
end

% Compute the mean and standard deviation of the error
mean_err=mean(errs)
sdev_err=sqrt(var(errs))
```

10. **Gaussian Elimination.** Given a linear system $Ax = b$, we want to find the unknown x. Gaussian elimination (GE) is an algorithm for doing this.

    Fact - the following row operations do not change the linear system:

    (a) Multiplying an equation, E_i by a non-zero scalar λ.

    (b) Multiplying an equation, E_j by λ and adding to E_i.
    This is denoted by $E_i + \lambda E_j \to E_i$

    (c) Exchanging equations E_i and E_j
    This is denoted by $E_i \leftrightarrow E_j$

11. **Note:** All these operations can easily be done on a computer with floating point arithmetic and memory access.

12. **The Idea:** To use the above operations to reduce the system to a triangular system.

13. **A Triangular System.** An nxn matrix $A = (a_{ij})$ is **upper triangular** if $a_{ij} = 0$ for $i > j$.

    An nxn matrix $A = (a_{ij})$ is **lower triangular** if $a_{ij} = 0$ for $i < j$.

14. **Example**

15. **Solving a triangular system by forward or back substiution.**

    Back substitution for an upper triangular matrix (more typical):

    E_n gives $x_n = b_n/a_n$

    E_{n-1} gives $x_{n-1} = \dfrac{b_n - a_{n-1,n}x_n}{a_{n-1,n}}$

    and, in general,

    $x_i = \dfrac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}$

    Forward substitution solves E_1 for x_1 first and then E_2 for x_2 second etc.

16. **Example** Solve the following linear system

    $E_1 : x_1 - x_2 + 2x_3 - x_4 = -8$

    $E_2 : 2x_1 - 2x_2 + 3x_3 - 3x_4 = -20$

    $E_3 : x_1 + x_2 + x_3 = -2$

    $E_4 : x_1 - x_2 + 4x_3 + 3x_4 = 12$

example continued

17. **Gaussian Elimination** See page 361 − 363 of Burden and Faires for the general Gaussian elimination procedure for linear systems.

18. **Failure of Gaussian Elimination in exact arithmetic**. Gaussian elimination can failure in two case:

    1) The system $Ax = b$ has no solution.

    2) The system $Ax = b$ has infinitely many solutions.

19. **Examples of GE failure**

20. **Gaussian Elimination in floating point arithmetic.** Some linear systems with GE will be subjected to amplification of round-off error.

21. **Example** Solve the following system using four-digit rounding arithmetic (the system has exact solution $x_1 = 10.00$ and $x_2 = 1.000$):

$$0.0030000x_1 + 59.14x_2 = 59.17$$

$$5.291x_1 - 6.130x_2 = 46.78$$

22. **Pivoting** Notice that in the above example, if $a_1 1$ had been zero, we would have interchanged rows. Our idea is, to increase **robustness**, to perform row interchanges when they are not strictly necessary.

    This is called **partial pivoting** and it works as follows:

    (a) Find $j = 1, 2, ...n$ such that $|a_{j1}$ is maximized.
    (b) Perform the interchange $E_i \leftrightarrow E_j$
    (c) Repeat the same strategy at steps $2, 3, 4, ..., n-1$ of GE.

23. **Example** Redo the previous example using partial pivoting.

24. **Notes on using Gaussian elimination:.**

    (a) Gaussian elimination should **always** be implemented with (at least) partial pivoting.

    (b) Using partial pivoting greatly increases the **robustness** of the computation to round-off error.

    (c) More sophisticated strategies of scaled partial pivoting (BF pg. 375) and full pivoting (BF pg. 379) may be needed.

25. **Efficiency (in general)** To determine the computation time of an algorithm operating on data of size, (N), the standard approach is to count the number of floating point operations (flops). This is the number of additions and subtractions and the number of multiplications and divisions required to perform the operation. It is an approximation of the time taken since many things are not taken into account. For example, time required to access memory, parallel computing etc.

26. **Efficiency of Gaussian Elimination and back substitution**

cont'd

27. **Matlab Demo - TicToc.m (in Canvas on the lecture notes page)**

The Matlab command tic starts timing and the command toc records the elapsed time.

The TicToc demo indicates that the mean time to solve $Ax = b$ using the Matlab command backslash is:

For $N = 1000$, the mean time is 0.0305 seconds. For $N = 2000$, the mean time is 0.1760 seconds.

This is an increase of $0.1760/0.0305 = 6$ times.

28. **Robustness of Gaussian Elimination with Pivoting**

Main point: Even with pivoting, round-off error may still accumulate. This is occurs when the matrix A is ill-conditioned.

29. **The condition number of a matrix, written $\kappa(A)$.**

(a) $\kappa(A)$ is a scalar function of A which measures how close to singular A is.

(b) For non-singular A, $1 \leq \kappa(A) < \infty$.

(c) Large $\kappa(A)$ means that A is ill-conditioned.

(d) In Matlab, the command cond(A) computes $\kappa(A)$.

(e) If z is the partial pivoting solution of $Ax = b$, then $x - z \approx \kappa(A)\epsilon$, $\epsilon =$machine epsilon $\approx (2.2)10^{-16}$.

30. **Example**

The Hilbert matrix is ill-conditioned.

Matlab Demo LSHilbertCond.m (in Canvas, lecture notes page).

31. **Moral of the story**

(a) When solving a linear system, $Ax = b$ with GE, the expected loss of **accuracy** is roughly $\kappa(A)\epsilon$.

(b) Loss of accuracy can be estimated by computing $\kappa(A)$ using the cond(A) command in Matlab.

(c) Matlab has built-in warnings to detect inaccurate computations due to ill-conditioning.