Chapter 5: Linear Systems: Direct Methods

Uri M. Ascher and Chen Greif
Department of Computer Science
The University of British Columbia
{ascher,greif}@cs.ubc.ca

Slides for the book
A First Course in Numerical Methods (published by SIAM, 2011)
`http://bookstore.siam.org/cs07/`

Some slides are from the notes of Dr. Peter Arbenz, ETH.

Goals of this chapter

- To learn practical methods to handle the most common problem in numerical computation;
- to get familiar (again) with the ancient method of Gaussian elimination in its modern form of LU decomposition, and develop pivoting methods for its stable computation;

Outline:

Gaussian elimination and Backward substitution

LU Decomposition

Pivoting strategies

Linear systems: Problem statement

We consider linear systems of equation of the form

$$\sum_{k=1}^{n} a_{ik} x_k = b_i, \qquad i = 1, \ldots, n,$$

or

$$A\,\mathbf{x} = \mathbf{b}.$$

The matrix elements a_{ik} and the right-hand side elements b_i are given. We are looking for the unknowns x_k.

In general

we consider the problem of finding $\mathbf{x}$ which solves

$$A\mathbf{x} = \mathbf{b},$$

where A is a given, real, nonsingular, $n \times n$ matrix, and $\mathbf{b}$ is a given, real vector.

- *Such problems are ubiquitous in applications*!
- Two solution approaches:
  - *Direct methods*: yield exact solution in absence of roundoff error.
    - Variations of **Gaussian elimination**.
    - Considered in this chapter
  - *Iterative methods*: iterate in a similar fashion to what we do for nonlinear problems.
    - Use only when direct methods are ineffective.
    - Considered in Chapter 7 We won't cover Chapter 7.

Backward substitution

- Special case: A is an **upper triangular** matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix},$$

  i.e., all elements below the main diagonal are zero: $a_{ij} = 0, \forall i > j$.

- The algorithm:

$$\begin{aligned} &\text{for } k = n : -1 : 1 \\ &\quad x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}} \\ &\text{end} \end{aligned}$$

Example

$$\begin{aligned} x_1 - 4x_2 + 3x_3 &= -2 \\ 5x_2 - 3x_3 &= 7 \\ -2x_3 &= -2 \end{aligned}$$

In matrix form:

$$\begin{pmatrix} 1 & -4 & 3 \\ 0 & 5 & -3 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -2 \\ 7 \\ -2 \end{pmatrix}.$$

Backward substitution: $x_3 = \frac{-2}{-2} = 1$, then $x_2 = \frac{1}{5}(7 + 3 \cdot 1) = 2$, then $x_1 = -2 + 4 \cdot 2 - 3 \cdot 1 = 3$.

Forward substitution

- Special case: A is a **lower triangular** matrix

$$A = \begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix},$$

where all elements above the main diagonal are zero: $a_{ij} = 0, \forall i < j$.

- The algorithm:

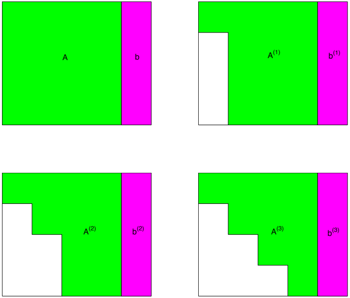
$$\begin{aligned} &\text{for } k = 1 : n \\ &\quad x_k = \frac{b_k - \sum_{j=1}^{k-1} a_{kj} x_j}{a_{kk}} \\ &\text{end} \end{aligned}$$

Gaussian elimination

- Can multiply a row of $A\mathbf{x} = \mathbf{b}$ by a scalar and add to another row: **elementary transformation**.
- Use this to transform A to upper triangular form:

$$MA\mathbf{x} = M\mathbf{b}, \quad U = MA.$$

- Apply backward substitution to solve $U\mathbf{x} = M\mathbf{b}$.



Gaussian elimination (basic)

for $k = 1 : n-1$

  for $i = k+1 : n$

    $l_{ik} = \dfrac{a_{ik}}{a_{kk}}$

    for $j = k+1 : n$

      $a_{ij} = a_{ij} - l_{ik} a_{kj}$

    end

    $b_i = b_i - l_{ik} b_k$

  end

end

Then apply backward substitution.

Note: upper part of A is overwritten by U, lower part no longer of interest.

Cost (flop count)

- For the elimination:

O(n^3)

$$\approx 2\sum_{k=1}^{n-1}(n-k)^2 = 2((n-1)^2 + (n-2)^2 + \cdots + 1^2) = \frac{2}{3}n^3 + \mathcal{O}(n^2).$$

- For the backward substitution:

$$\approx 2\sum_{k=1}^{n-1}(n-k) = 2\frac{(n-1)n}{2} \approx n^2.$$ O(n^2)

Example

- Solve $A\mathbf{x} = \mathbf{b}$ for

$$A = \begin{pmatrix} 1 & -4 & 3 \\ 1 & 1 & 0 \\ 3 & -2 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} -2 \\ 5 \\ 6 \end{pmatrix}.$$

- Gaussian elimination: $(A \mid \mathbf{b}) \Rightarrow$

$$\left(\begin{array}{ccc|c} 1 & -4 & 3 & -2 \\ 0 & 5 & -3 & 7 \\ 0 & 10 & -8 & 12 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & -4 & 3 & -2 \\ 0 & 5 & -3 & 7 \\ 0 & 0 & -2 & -2 \end{array}\right).$$

- Backward substitution: $x_3 = \frac{-2}{-2} = 1$, then $x_2 = \frac{1}{5}(7 + 3 \cdot 1) = 2$, then $x_1 = -2 + 4 \cdot 2 - 3 \cdot 1 = 3$.

Outline

- Gaussian elimination and backward substitution
- LU decomposition
- Pivoting strategies
- Efficient implementation
- Cholesky decomposition
- Sparse matrices
- Permutations and ordering strategies
- Estimating error and the condition number

LU decomposition

- What if we have many right hand side vectors, or we don't know $\mathbf{b}$ right away?
- Note that determining transformation M such that $MA = U$ does not depend on $\mathbf{b}$.
- $M = M^{(n-1)} \dots M^{(2)} M^{(1)}$, where $M^{(k)}$ is the transformation of the kth outer loop step. These are elementary lower triangular matrices, e.g.,

$$M^{(2)} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & -l_{32} & \ddots & & \\ & \vdots & & \ddots & \\ & -l_{n2} & & & 1 \end{pmatrix}.$$

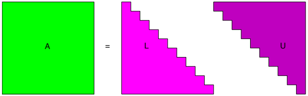
LU decomposition (cont.)

- The matrix M is unit lower triangular.
- The matrix $L = M^{-1}$ is also unit lower triangular:

$$A = LU, \quad L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix}.$$

M unit triangular matrix means: it has ones on the diagonal

LU decomposition (cont.)



So, Gaussian elimination is equivalent to:

1. decompose $A = LU$.
   Now for a given $\mathbf{b}$ *we have to solve* $L(U\mathbf{x}) = \mathbf{b}$:
2. use forward substitution to solve $L\mathbf{y} = \mathbf{b}$;
3. use backward substitution to solve $U\mathbf{x} = \mathbf{y}$.

Example

$$A = \begin{pmatrix} 1 & -4 & 3 \\ 1 & 1 & 0 \\ 3 & -2 & 1 \end{pmatrix}.$$

Obtain

1. $l_{21} = \frac{1}{1} = 1, \; l_{31} = \frac{3}{1} = 3$, so

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}, \quad A^{(1)} = M^{(1)}A = \begin{pmatrix} 1 & -4 & 3 \\ 0 & 5 & -3 \\ 0 & 10 & -8 \end{pmatrix}.$$

2. $l_{32} = \frac{10}{5} = 2$, so

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}, \quad A^{(2)} = M^{(2)}A^{(1)} = \begin{pmatrix} 1 & -4 & 3 \\ 0 & 5 & -3 \\ 0 & 0 & -2 \end{pmatrix}.$$

Example (cont.)

- We thus obtain

$$U = A^{(2)} = M^{(2)} A^{(1)} = \begin{pmatrix} 1 & -4 & 3 \\ 0 & 5 & -3 \\ 0 & 0 & -2 \end{pmatrix},$$

and collect the multipliers l_{21}, l_{31} and l_{32} into the unit lower triangular matrix

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}.$$

- Indeed, $A = LU$:

$$\begin{pmatrix} 1 & -4 & 3 \\ 1 & 1 & 0 \\ 3 & -2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -4 & 3 \\ 0 & 5 & -3 \\ 0 & 0 & -2 \end{pmatrix}.$$

Examples where the LU decomposition is useful

- When we have multiple right-hand sides, form *once* the LU decomposition (which costs $\mathcal{O}(n^3)$ flops); then for each right-hand side only apply forward/backward substitutions (which are computationally cheap at $\mathcal{O}(n^2)$ flops each).
- Can compute A^{-1} by decomposing $A = LU$ once, and then solving $LU\mathbf{x} = \mathbf{e}_k$ for each column $\mathbf{e}_k$ of the unit matrix. These are n right hand sides, so the cost is approximately $\frac{2}{3}n^3 + n \cdot 2n^2 = \frac{8}{3}n^3$ flops.
  (However, typically we try to avoid computing the inverse A^{-1}; the need to compute it *explicitly* is rare.)
- Compute determinant of A by

$$\det(A) = \det(L)\det(U) = \Pi_{k=1}^{n} u_{kk}.$$

Outline

- Gaussian elimination and backward substitution
- LU decomposition
- Pivoting strategies

Example: need for pivoting

- First step of Gaussion elimination:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 3 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{array}\right).$$

- Second step: Now $a_{22}^{(1)} = 0$ and we're stuck.
- Simple remedy: exchange rows 2 and 3:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 3 \\ 1 & 1 & 2 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{array}\right).$$

Here the decomposition has been completed without difficulty.

Partial pivoting

- It is rare to hit precisely a zero pivot, but common to hit a very small one.
- Example:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1+10^{-12} & 2 & 2 \\ 1 & 2 & 2 & 3 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 10^{-12} & 1 & 1 \\ 0 & 1 & 1 & 2 \end{array}\right).$$

- Now we get a multplier $l_{3,2} = 1/10^{-12} = 10^{12}$, so roundoff error in elimination step is magnified by this factor 10^{12}.
- Employ Gaussian elimination with partial pivoting (GEPP) not just to avoid zero pivots but more generally to obtain a *stable* algorithm.

GEPP

- At each stage k choose $q = q(k)$ as the smallest integer for which

$$|a_{qk}^{(k-1)}| = \max_{k \le i \le n} |a_{ik}^{(k-1)}|,$$

  and interchange rows k and q.
- This ensures that pivots are not too small (unless matrix is close to singular) and $|l_{i,k}| \le 1, \ \text{all } i \ge k$.
- $PA = LU$ where P is permutation matrix, e.g.,

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Simple GEPP algorithm

for $k = 1 : n - 1$
  for $i = k + 1 : n$
    $q = \arg \max_{k \le i \le n} |a_{ik}^{(k-1)}|$
    exchange rows k and q
    $l_{ik} = \dfrac{a_{ik}}{a_{kk}}$
    for $j = k + 1 : n$
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
    $b_i = b_i - l_{ik} * b_k$
  end
end

In practice, we keep record of permutations in a 1D array

Forming $PA = LU$

- It's not so obvious, but it's true, that with

$$B = M^{(n-1)} P^{(n-1)} \cdots M^{(2)} P^{(2)} M^{(1)} P^{(1)}, \quad P = P^{(n-1)} \cdots P^{(2)} P^{(1)},$$

we get L lower triangular and

$$B = L^{-1} P.$$

- The matrix L is lower triangular, although not the same as it would be without pivoting. It is obtained by a similar sequence of steps as before, with the addition of permutation steps.
- The permutation matrix P is orthogonal, so

$$A = (P^T L) U.$$

$P^T L$ is "psychologically lower triangular".
In practice, keep record of permutations in a 1D array.

Example revisited (1/3)

Same matrix we worked on a few slides ago, now with pivoting:

$$A = \begin{pmatrix} 1 & -4 & 3 \\ 1 & 1 & 0 \\ 3 & -2 & 1 \end{pmatrix}.$$

Go through first column and find pivot:

$$P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \;;\; P^{(1)}A = \begin{pmatrix} 3 & -2 & 1 \\ 1 & 1 & 0 \\ 1 & -4 & 3 \end{pmatrix}.$$

So, we have

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix}, \; A^{(1)} = M^{(1)}P^{(1)}A = \begin{pmatrix} 3 & -2 & 1 \\ 0 & \frac{5}{3} & -\frac{1}{3} \\ 0 & -\frac{10}{3} & \frac{8}{3} \end{pmatrix}.$$

Example revisited (2/3)

Now, work on $A^{(1)}$:

$$P^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \; ; \;\; P^{(2)}A^{(1)} = \begin{pmatrix} 3 & -2 & 1 \\ 0 & -\frac{10}{3} & \frac{8}{3} \\ 0 & \frac{5}{3} & -\frac{1}{3} \end{pmatrix},$$

and we have

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}, \;\; A^{(2)} = M^{(2)}P^{(2)}M^{(1)}P^{(1)}A = \begin{pmatrix} 3 & -2 & 1 \\ 0 & -\frac{10}{3} & \frac{8}{3} \\ 0 & 0 & 1 \end{pmatrix}.$$

So the upper triangular U is $U = A^{(2)} = M^{(2)}P^{(2)}M^{(1)}P^{(1)}A$.

Example revisited (3/3)

- Let us find L and P. Write

$$U = M^{(2)}P^{(2)}M^{(1)}P^{(1)}A = \underbrace{\left(M^{(2)}\right)}_{\tilde{M}^{(2)}}\underbrace{\left(P^{(2)}M^{(1)}P^{(2)^T}\right)}_{\tilde{M}^{(1)}}\underbrace{\left(P^{(2)}P^{(1)}\right)}_{P}A.$$

- Next, take the elements of L below the diagonal to be those of the $\tilde{M}^{(k)}$ with flipped signs; the permutation matrix P is just the product of the $P^{(k)}$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{3} & -\frac{1}{2} & 1 \end{pmatrix}; \quad U = \begin{pmatrix} 3 & -2 & 1 \\ 0 & -\frac{10}{3} & \frac{8}{3} \\ 0 & 0 & 1 \end{pmatrix}; \quad P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

  Exercise: confirm that indeed, $PA = LU$.

- In MATLAB obtain these matrices by the commands

```
A=[1 -4 3; 1 1 0; 3 -2 1];
[L,U,P]=lu(A);
```

- For more on the general principle illustrated in this example, see pages 107–108 in the book, as well as Exercises 7 and 8 of Chapter 5.