4.1 Introduction

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

Matrix form:

$$\mathbf{Ax} = \mathbf{b}\,, \qquad \mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Only *square* systems will be considered: $m = n$.

- Two kinds of system/matrix **A**:

  Dense **A** — few 0's, store all elements: $n \lesssim 1000$.

  Sparse **A** — many 0's, store/generate only non-zero elements: $n \lesssim 100000$.

- Two types of solution method:

  Direct — fixed numbers of $\times$, $\div$, $+$, $-$.

  Iterative — numbers of $\times$, $\div$, $+$, $-$ depend on solution accuracy.

4.2.1 Gaussian Elimination with Back Substitution

Example 4.1 Solve

$$\begin{array}{rcrcrcrl} 2x_1 & + & 3x_2 & - & x_3 & = & 5 & (1) \\ 4x_1 & + & 4x_2 & - & 3x_3 & = & 3 & (2) \\ 2x_1 & - & 3x_2 & + & x_3 & = & -1 & (3) \end{array}$$

Solution :

1. Eliminate x_1 from (2) & (3) using (1),

$$\begin{array}{rcrcrcrcrl} & & 2x_1 & + & 3x_2 & - & x_3 & = & 5 & (1) \\ (2) - 2 \times (1) & : & & - & 2x_2 & - & x_3 & = & -7 & (4) \\ (3) - (1) & : & & - & 6x_2 & + & 2x_3 & = & -6 & (5) \end{array}$$

2. Eliminate x_2 from (5) using (4),

$$\begin{array}{rcrcrcrcrl} & & 2x_1 & + & 3x_2 & - & x_3 & = & 5 & (1) \\ & & & - & 2x_2 & - & x_3 & = & -7 & (4) \\ (5) - 3 \times (4) & : & & & & & 5x_3 & = & 15 & (6) \end{array}$$

3. *Back-substitution*: Solve (6), (4), (1) for x_3, x_2, x_1.

$$\begin{array}{rcrcrcl} (6) & \Rightarrow & x_3 & = & 15/5 & = & 3 \\ (4) & \Rightarrow & x_2 & = & (-7 + x_3)/(-2) & = & 2 \\ (1) & \Rightarrow & x_1 & = & (5 - 3x_2 + x_3)/2 & = & 1 \end{array}$$

4.2.2 Matrix Version of Gaussian Elimination

Elementary Row (Gauss) Operations

Vector $\mathbf{r}_i$ denotes row i:

1. interchange $\mathbf{r}_i$ & $\mathbf{r}_j$;
2. replace $\mathbf{r}_i$ by $a\mathbf{r}_i$, $a \neq 0$;
3. replace $\mathbf{r}_i$ by $\mathbf{r}_i + a\mathbf{r}_j$.

Example 4.2 Gaussian elimination with back substitution.

Solution :

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ 2 & -3 & 1 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 5 \\ 3 \\ -1 \end{pmatrix}.$$

1. Form augmented matrix:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ 2 & -3 & 1 & -1 \end{array}\right).$$

2. *Pivot column* — column being eliminated.

   Pivot row — first row with all 0's left of pivot column.

   Pivot — element in pivot column & pivot row.

   If pivot $= 0$, interchange pivot row with lower row.

If no interchange gives pivot $\neq 0$, system is singular: no solution or ∞-many.

Eliminate elements in pivot column *below* pivot.

Pivot successively from left to right.

$$\begin{array}{rcll} \mathbf{r}_2 - 2\mathbf{r}_1 & \to & \mathbf{r}_2 & : \\ \mathbf{r}_3 - \mathbf{r}_1 & \to & \mathbf{r}_3 & : \end{array} \quad \begin{pmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & -6 & 2 & -16 \end{pmatrix}$$

$$\begin{array}{rcll} \mathbf{r}_3 - 3\mathbf{r}_2 & \to & \mathbf{r}_3 & : \end{array} \quad \begin{pmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & 5 & 15 \end{pmatrix}$$

3. Back-substitute to find x_1, x_2, x_3.

Gauss-Doolittle elimination — usual form implemented on computers:

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} a_{1,n+1} \\ \vdots \\ a_{n,n+1} \end{pmatrix},$$

1. For $j = 1, \ldots, n-1$ do Steps 2–3.
2. Let k be the smallest integer $j \le k \le n$ such that $a_{kj} \neq 0$. If no such k exists, then no unique solution exists. If $k \neq j$ interchange $\mathbf{r}_k$ and $\mathbf{r}_j$.
3. For $i = j+1, \ldots, n$ do Steps 4–5.
4. $m_{ij} = a_{ij}/a_{jj}$
5. $\mathbf{r}_i - m_{ij}\mathbf{r}_j \to \mathbf{r}_i$
6. If $a_{nn} = 0$, then no unique solution exists.
7. For $i = n, \ldots, 1$,
$$x_i = \frac{1}{a_{ii}} \left(a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j \right).$$

Storage: one $n \times (n+1)$ array, if only most recent a_{ij} stored.
Store multipliers m_{ij} $(i = 1, \ldots, n-1; j = i+1, \ldots, n)$ in place of zeroed a_{ij}.

4.2.4 Algebraic Work Measures

M = multiplications & divisions; A = additions & subtractions.

- Gauss-Doolittle elimination (with back-substitution or backward elimination):

$$\left(\frac{n^3}{3}+n^2-\frac{n}{2}\right)M+\left(\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}\right)A\approx\frac{1}{3}n^3M+\frac{1}{3}n^3A\,,\quad n\gg 1\,.$$

  Gaussian-Jordan operation count:

$$\left(\frac{n^3}{2}+n^2-\frac{n}{2}\right)M+\left(\frac{n^3}{2}-\frac{n}{2}\right)A\approx\frac{1}{2}n^3M+\frac{1}{2}n^3A\,,\quad n\gg 1\,.$$

  Operation counts for $n=100$:

  - Gaussian elimination: $343,000M+338,250A$;
  - Gauss-Jordan: $509,950M+499,950A$.

- Multiplication-division counts provide a valid comparison of different methods.

  Column operations are often performed significantly faster than row operations.

  Performance measure: *Flop* = FLoating-point OPerationS — $s:=s+a_{ik}b_{kj}$.

4.2.5 LU Decomposition

If Gauss-Doolittle works on $\mathbf{A}$ without row interchanges, then

$$\mathbf{A} = \mathbf{LU}\,.$$

$$\begin{aligned}
\mathbf{L} &= \text{unit-lower-triangular matrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & 1 \end{pmatrix}. \\
\mathbf{U} &= \text{upper-triangular matrix} = \text{Gauss-reduced form.}
\end{aligned}$$

To solve $\mathbf{Ax} = \mathbf{b}$ given LU-decomposition:

- solve $\mathbf{Ly} = \mathbf{b}$ by forward substitution;
- solve $\mathbf{Ux} = \mathbf{y}$ by back-substitution.

Extremely useful for systems with same $\mathbf{A}$ but different $\mathbf{b}$.

Example 4.4 LU-decomposition of **A** in Example 4.2.

Solution : Gauss-Doolittle elimination gives

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & 5 \end{pmatrix}$$

Forward substitution:

$$y_1 = 5\,, \quad y_2 = 3 - 2y_1 = -7\,, \quad y_3 = -1 - y_1 - 3y_2 = 15\,.$$

Back-substitution as in Example 4.2.

Operation count for computing LU-decomposition:

$$\frac{1}{3}(n^3 - n)M + (\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n)A\,.$$

Operation count for forward & backward substitution:

$$(n^2 - n)M + n^2 A\,.$$

If Gaussian elimination requires row interchanges, then

$$\mathbf{PA = LU}.$$

$\mathbf{P}$ = permutation matrix obtained from Gauss-Doolittle row interchanges on identity matrix.

Example 4.6 Gauss-Doolittle elimination with row interchanges.

Solution :

$$\begin{array}{rcl} \mathbf{r}_1 \leftrightarrow \mathbf{r}_2 & : & \begin{pmatrix} 2 & -3 & 1 & -1 \\ 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \end{pmatrix} \\ \mathbf{r}_1 \leftrightarrow \mathbf{r}_3 & & \end{array}$$

$$\begin{array}{rcl} \mathbf{r}_2 - \mathbf{r}_1 \rightarrow \mathbf{r}_2 & : & \begin{pmatrix} 2 & -3 & 1 & -1 \\ 0 & 6 & -2 & 6 \\ 0 & 10 & -5 & 5 \end{pmatrix} \\ \mathbf{r}_3 - 2\mathbf{r}_1 \rightarrow \mathbf{r}_3 & & \end{array}$$

$$\mathbf{r}_2 \leftrightarrow \mathbf{r}_3 \quad : \quad \begin{pmatrix} 2 & -3 & 1 & -1 \\ 0 & 10 & -5 & 5 \\ 0 & 6 & -2 & 6 \end{pmatrix}$$

$$\mathbf{r}_3 - \tfrac{3}{5}\mathbf{r}_2 \rightarrow \mathbf{r}_3 \quad : \quad \begin{pmatrix} 2 & -3 & 1 & -1 \\ 0 & 10 & -5 & 5 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

By inspection,

$$\mathbf{U} = \begin{pmatrix} 2 & -3 & 1 \\ 0 & 10 & -5 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & \frac{3}{5} & 1 \end{pmatrix}.$$

NOT

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & \frac{3}{5} & 1 \end{pmatrix},$$

since rows 2 & 3 interchanged before column 2 eliminated.

Apply interchanges: rows 1 & 2, rows 1 & 3, rows 2 & 3; to 3×3 identity matrix:

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

4.2.7 Pivoting and Scaling

Example 4.8 Gauss-Doolittle elimination & 4-digit arithmetic with rounding:

$$\begin{array}{rcrcl} 0.002000x_1 & + & 40.00x_2 & = & 40.02 \\ 5.000x_1 & - & 10.00x_2 & = & 40.00 \end{array}$$

Solution : Exact solution: $x_1 = 10.00, x_2 = 1.000$

$$\text{Augmented matrix} = \begin{pmatrix} 0.002000 & 40.00 & 40.02 \\ 5.000 & -10.00 & 40.00 \end{pmatrix}$$

Pivot $= 0.002000$; $m_{21} = 5.000/0.002000 = 2500$. Thus $\mathbf{r}_2 - m_{21}\mathbf{r}_1 \to \mathbf{r}_2$:

$$(2,2)\text{-element} = -10.00 - 2500 \times 40.00 = -10.00 - 100000 = -100010 = -100000$$

$$(2,3)\text{-element} = 40.00 - 2500 \times 40.02 = 40.00 - 100050 = 40.00 - 100100 = -100060 = -100100\,.$$

Thus

$$\begin{pmatrix} 0.002000 & 40.00 & 40.02 \\ 0 & -100000 & -100100 \end{pmatrix}$$

Back-substitution:

$$x_2 = \frac{-100100}{-100000} = 1.001, \qquad x_1 = \frac{40.02 - 40.00 \times 1.001}{0.002000} = -10.00.$$

Error in computed $x_1 = 200\%$!

- Gaussian elimination is unstable due to round-off.

- Gaussian elimination with complete pivoting is stable.

  Complete pivoting: each pivot is chosen using row and column interchanges as absolute largest element in or to right of pivot column & in or below pivot row.

  Complete pivoting substantially increases computational effort so only used for extreme cases of round-off error.

- Partial pivoting used in practice.

  Partial pivoting: each pivot is chosen using row interchanges as absolute largest element in pivot column in or below pivot row.

  Partial pivoting is easily incorporated into forward elimination.

- Partial- and complete-pivoting with appropriate scaling reduce round-off error in finite precision arithmetic.

Example 4.9 Gauss-Doolittle elimination with partial pivoting:

Solution $|5.000| > |0.002000| \Rightarrow$ interchange rows $\Rightarrow$ pivot $= 5.000$.

$$\begin{pmatrix} 5.000 & -10.00 & 40.00 \\ 0.002000 & 40.00 & 40.02 \end{pmatrix}$$

$m_{21} = 0.002000/5.000 = 4.000 \times 10^{-4}$; $\mathbf{r}_2 - m_{21}\mathbf{r}_1 \rightarrow \mathbf{r}_2$:

$$\begin{pmatrix} 5.000 & -10.00 & 40.00 \\ 0 & 40.00 & 40.00 \end{pmatrix}$$

Back-substitution: $x_2 = 1.000$, $x_1 = 10.00$.

Example 4.10 Gauss-Doolittle elimination:

$$\begin{array}{rcrcr} 5.000x_1 & - & 10.00x_2 & = & 40.00 \\ 20.00x_1 & + & 400000x_2 & = & 400200 \end{array}$$

Solution

$$\begin{pmatrix} 5.000 & -10.00 & 40.00 \\ 20.00 & 400000 & 400200 \end{pmatrix}$$

$|20.00| > |5.000| \Rightarrow$ interchange rows $\Rightarrow$ pivot $= 20.00$; $m_{21} = 5.000/20.00 = 0.2500$:

$$\begin{pmatrix} 20.00 & 400000 & 400200 \\ 0 & -100000 & -100100 \end{pmatrix}$$

Back-substitution:

$$x_2 = \frac{-100100}{-100000} = 1.001\,, \qquad x_1 = \frac{400200 - 400000 \times 1.001}{20.00} = -10.00\,.$$

Error in computed $x_1 = 200\%$ even with partial pivoting.

- Equations must be suitably *scaled* before elimination.
  - One method: multiply each row by suitable factor so absolute largest elements in each row are approximately equal (possibly worse than no scaling).
  - Variables/columns can be scaled $\equiv$ change of units.
  - General scaling strategies unreliable.
  - Most problems arising in practice usually well scaled.
  - Singular systems may be difficult to solve accurately due to round-off error: pivots, which should be zero in exact arithmetic, may be small but non-zero in finite-precision arithmetic.

4.3.8 Vector Norms

- Vector norm $\|\mathbf{v}\|$ of $n \times 1$ column vector or $1 \times n$ row vector $\mathbf{v}$ is real number, such that

  1. $\|\mathbf{v}\| > 0$ if $\mathbf{v} \neq 0$ and $\|\mathbf{v}\| = 0$ if $\mathbf{v} = \mathbf{0}$;
  2. $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$ for any scalar a;
  3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ — *triangle inequality.*

- We consider only:

$$\|\mathbf{v}\|_1 = |v_1| + |v_2| + \ldots + |v_n|\,, \quad \|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}\,, \quad \|\mathbf{v}\|_\infty = \max_{1 \leq i \leq n} |v_i|\,.$$

Example 4.11 If $\mathbf{v} = (-1, 3, 5, 7)$ then

$$\begin{aligned} \|\mathbf{v}\|_1 &= |-1| + |3| + |5| + |7| = 16 \\ \|\mathbf{v}\|_2 &= \sqrt{(-1)^2 + (3)^2 + (5)^2 + (7)^2} = \sqrt{84} \\ \|\mathbf{v}\|_\infty &= \max\{1, 3, 5, 7\} = 7\,. \end{aligned}$$

- A sequence of vectors $\{\mathbf{v}_i\}$ converges to $\mathbf{v}$ in the norm $\|\ \|$ iff $\|\mathbf{v}_i - \mathbf{v}\| \to 0$ as $i \to \infty$.

  $\|\ \|_1$, $\|\ \|_2$ & $\|\ \|_\infty$ are equivalent: convergence of vector sequence in one norm implies convergence in any norm.

4.3.9 Matrix Norms

Given a vector norm define the *matrix norm*:

$$\|\mathbf{A}\| = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}\,.$$

Vector norms $\|\ \|_1$, $\|\ \|_2$, $\|\ \|_\infty$ induce matrix norms: $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_2$, $\|\mathbf{A}\|_\infty$.

$$\begin{aligned}\|\mathbf{A}\|_1 &= \max_{1\le j\le n}\left\{\sum_{i=1}^{n}|a_{ij}|\right\} = \text{maximum absolute column sum of } \mathbf{A}.\\ \|\mathbf{A}\|_\infty &= \max_{1\le i\le n}\left\{\sum_{j=1}^{n}|a_{ij}|\right\} = \text{maximum absolute row sum of } \mathbf{A}.\end{aligned}$$

Example 4.12

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ 2 & -3 & 1 \end{pmatrix}, \qquad \|\mathbf{A}\|_1 = 10\,, \qquad \|\mathbf{A}\|_\infty = 11\,.$$

Properties: (of matrix norms induced by a vector norm)

1. $\|\mathbf{AB}\| \le \|\mathbf{A}\|\ \|\mathbf{B}\|$.
2. $\|\mathbf{A}^j\| = \|\mathbf{A}\|^j$, $j = 1, 2, \ldots$.
3. $\|\mathbf{A}+\mathbf{B}\| \le \|\mathbf{A}\| + \|\mathbf{B}\|$.
4. $\|a\mathbf{A}\| \le |a|\ \|\mathbf{A}\|$.

5. $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = \mathbf{0}$.

6. All norms are equivalent; e.g.

$$\frac{1}{\sqrt{n}}\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_\infty \le \sqrt{n}\|\mathbf{A}\|_2\,,\ \frac{1}{\sqrt{n}}\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_1 \le \sqrt{n}\|\mathbf{A}\|_2\,,\ \frac{1}{n}\|\mathbf{A}\|_1 \le \|\mathbf{A}\|_\infty \le n\|\mathbf{A}\|_1\,.$$

7. $\|\mathbf{Ax}\| \le \|\mathbf{A}\|\ \|\mathbf{x}\|$.

8. *Banach's Lemma.* $\mathbf{P}$ square & $\|\mathbf{P}\| < 1 \Rightarrow \mathbf{I} + \mathbf{P}$ invertible &

$$\frac{1}{1 + \|\mathbf{P}\|} \le \|(\mathbf{I} + \mathbf{P})^{-1}\| \le \frac{1}{1 - \|\mathbf{P}\|}\,. \tag{4.1}$$

Example 4.13

$$\mathbf{A} = \begin{pmatrix} 1.1 & -0.6 \\ 0.8 & 0.9 \end{pmatrix} \qquad \Rightarrow \qquad \mathbf{A} = \mathbf{I} + \mathbf{P}\,, \quad \text{where} \quad \mathbf{P} = \begin{pmatrix} 0.1 & -0.6 \\ 0.8 & -0.1 \end{pmatrix}\,.$$

Since $\|\mathbf{P}\|_\infty = 0.9 < 1$, $\mathbf{A}$ is invertible & $1/1.9 \le \|\mathbf{A}^{-1}\|_\infty \le 1/0.1$.

4.3.10 Condition of a System of Linear Equations

$$\left.\begin{array}{rcrcl} x & + & y & = & 2.00 \\ x & + & 1.01y & = & 2.01 \end{array}\right\}$$ nearly parallel lines; exact solution: $x = 1$, $y = 1$.

$$\left.\begin{array}{rcrcl} x & + & y & = & 2.00 \\ x & + & 0.99y & = & 2.02 \end{array}\right\}$$ exact solution: $x = 4$, $y = -2$.

Problem is ill-conditioned.
Geometrically: nearly parallel lines $\Rightarrow$ small change can greatly shift intersection.

The following result quantifies the ill-conditioning of $\mathbf{Ax} = \mathbf{b}$ for invertible $\mathbf{A}$: Let $\mathbf{A}$ change to $\mathbf{A} + \delta\mathbf{A}$, $\mathbf{b}$ change to $\mathbf{b} + \delta\mathbf{b}$, where $\delta\mathbf{A}$ is small:

$$r = \|(\delta\mathbf{A})\mathbf{A}^{-1}\| < 1\,.$$

Then $\mathbf{x}$ changes to $\mathbf{x} + \delta\mathbf{x}$, where $(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ &

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \frac{1}{1-r}\mathrm{cond}(\mathbf{A})\left\{\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}\right\}, \quad \text{where} \quad \mathrm{cond}(A) := \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|.$$

Condition number $\mathrm{cond}(\mathbf{A})$ measures condition of systems with coefficient matrix $\mathbf{A}$.

If cond($\mathbf{A}$) small, small changes in $\mathbf{b}$ & $\mathbf{A}$ produce only small changes in $\mathbf{x}$.
But if cond($\mathbf{A}$) large, small changes in $\mathbf{b}$ & $\mathbf{A}$ may only produce small changes in $\mathbf{x}$.
E.g.

$$\mathbf{A} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad \mathbf{A}^{-1} = \begin{pmatrix} 1 & -k \\ 0 & 1 \end{pmatrix}.$$

In norms $\| \ \|_1$ & $\| \ \|_\infty$,

$$\|\mathbf{A}\| = \|\mathbf{A}^{-1}\| = 1 + k\,, \quad k \gg 1\,, \qquad \Rightarrow \qquad \text{cond}(\mathbf{A}) = (1+k)^2\,.$$

If $\mathbf{b} = (1, 1)$, then $\mathbf{x} = (1 - k, 1)$.
If $\mathbf{b}$ is perturbed, so that $\mathbf{b} + \delta\mathbf{b} = (1 + \delta_1, 1 + \delta_2)$, then $\delta\mathbf{x} = (\delta_1 - k\delta_2, \delta_2)$.
Thus system is well-conditioned for this $\mathbf{b}$:

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq 2\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}\,,$$

But not for $\mathbf{b} = (1, 0)$.

If cond($\mathbf{A}$) is large, then $\mathbf{Ax} = \mathbf{b}$ will be ill-conditioned for some $\mathbf{b}$.

4.3.11 Stability of Gaussian Elimination with Pivoting

- The solution $\hat{\mathbf{x}}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ computed by Gaussian elimination with complete pivoting exactly solves a nearby system

$$(\mathbf{A} + \mathbf{E})\hat{\mathbf{x}} = \mathbf{b}\,,$$

  where for all practical purposes

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \epsilon_{\text{mach}}\,.$$

  Gaussian elimination with complete pivoting is stable.

- One measure of the accuracy of the computed solution is by how much it fails to satisfy the exact equations, namely the *residual* $\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$.

$$\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\| = \|\mathbf{E}\hat{\mathbf{x}}\| \leq \|\mathbf{E}\|\,\|\hat{\mathbf{x}}\|\,.$$

  Thus,

$$\frac{\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|}{\|\mathbf{A}\|\,\|\hat{\mathbf{x}}\|} \leq \epsilon_{\text{mach}}\,.$$

  The computed solution solves the exact equations as accurately as can be expected given the precision of the computation.

- Another measure of the accuracy of the solution is $\mathbf{x} - \hat{\mathbf{x}}$.
  From $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, $\mathbf{x} - \hat{\mathbf{x}} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\hat{\mathbf{x}})$. Hence

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \|\mathbf{A}^{-1}\|\,\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\| \leq \|\mathbf{A}^{-1}\|\,\|\mathbf{E}\|\,\|\hat{\mathbf{x}}\|\,.$$

Since $\mathrm{cond}(\mathbf{A}) := \|\mathbf{A}^{-1}\|\,\|\mathbf{A}\|$,

$$\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|}{\|\hat{\mathbf{x}}\|} \leq \mathrm{cond}(\mathbf{A})\epsilon_{\mathrm{mach}}\,,$$

Gaussian elimination with complete pivoting will compute the solution with about $-\log_{10}\mathrm{cond}(\mathbf{A})-\log_{10}\epsilon_{\mathrm{mach}}$ *significant figures*.
Thus a system of equations is well-conditioned for a computer with ϵ_{mach} if $\log_{10}\mathrm{cond}(\mathbf{A}) < -\log_{10}\epsilon_{\mathrm{mach}} - 1$ in the sense that at least one significant figure can be computed on that machine using the stable method of Gaussian elimination with complete pivoting.

- The stability analysis of Gaussian elimination with complete pivoting applies to partial pivoting: instability is highly unlikely and partial pivoting is used in preference to complete pivoting.

- `SGEFS` from LINPACK, MATLAB. Subroutines still needed in many situations.

4.5 Iteration Methods

4.5.1 Gauss-Seidel Iteration

- Rearrange system of equations:

$$\begin{aligned} x_1 &= (b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1,n-1}x_{n-1} - a_{1n}x_n)/a_{11} \\ x_2 &= (b_2 - a_{21}x_1 - a_{23}x_3 - \ldots - a_{2,n-1}x_{n-1} - a_{2n}x_n)/a_{22} \\ &\vdots \\ x_n &= (b_n - a_{n1}x_1 - a_{n2}x_2 - a_{n3}x_3 - \ldots - a_{n,n-1}x_{n-1})/a_{nn}\,. \end{aligned}$$

- $\mathbf{x}^{(0)} = \mathbf{0}$.
  Substitute $x_2^{(0)}, x_3^{(0)}, \ldots, x_n^{(0)}$ into RHS of first equation $\Rightarrow x_1^{(1)}$.
  Substitute $x_1^{(1)}, x_3^{(0)}, x_4^{(0)}, \ldots, x_n^{(0)}$ into RHS of second equation $\Rightarrow x_2^{(1)}$.
  Do RHS of equation $\Rightarrow \mathbf{x}^{(1)} = (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots, x_n^{(1)})$.

- Repeat until $\mathbf{x}^{(K)}$ is sufficiently close to exact solution:
  terminate if $\|\mathbf{x}^{(K)} - \mathbf{x}^{(K-1)}\| < \epsilon$.

- Matrix form: decompose $\mathbf{A}$ into $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$, where

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & a_{nn} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & a_{n-1,n} \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Gauss-Seidel iteration:

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}\left\{\mathbf{b} - \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)}\right\}$$

or

$$(\mathbf{D} + \mathbf{L})\mathbf{x}^{(k+1)} = -\mathbf{U}\mathbf{x}^{(k)} + \mathbf{b}\,.$$

Example 4.18

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

Solution Exact: $x_1 = 1$, $x_2 = -1$.

Rearrange equations:

$$\begin{aligned} x_1 &= (3 + x_2)/2 \\ x_2 &= (-3 - x_1)/4\,. \end{aligned}$$

Gauss-Seidel iteration scheme:

$$\begin{aligned} x_1^{(k+1)} &= (3 + x_2^{(k)})/2 \\ x_2^{(k+1)} &= (-3 - x_1^{(k+1)})/4\,. \end{aligned}$$

Convergence to $x_1 = 1.000$ and $x_2 = -1.000$.

Example 4.19

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

(same equations in reverse order).

Solution Rearrange:

$$\begin{aligned} x_1 &= -3 - 4x_2 \\ x_2 &= -3 + 2x_1 \,. \end{aligned}$$

Gauss-Seidel iteration scheme:

$$\begin{aligned} x_1^{(k+1)} &= -3 - 4x_2^{(k)} \\ x_2^{(k+1)} &= -3 + 2x_1^{(k+1)} \,. \end{aligned}$$

Clearly diverging.

- Convergence depends on the order of the equations. For best hope of convergence order equations so that diagonal elements $a_{11}, a_{22}, \ldots, a_{nn}$ are as absolutely large as possible.

Example 4.20

$$\begin{aligned} x + 3y - 2z &= 7 \\ x + 2y + 3z &= 10 \\ 2x - y + z &= 5 \end{aligned}$$

Solution Reorder equations to absolutely maximise elements on main diagonal:

$$\begin{aligned} 2x - y + z &= 5 \\ x + 3y - 2z &= 7 \\ x + 2y + 3z &= 10 \end{aligned}$$

Gauss-Seidel iteration scheme:

$$\begin{aligned} x^{(k+1)} &= (5 + y^{(k)} - z^{(k)})/2 \\ y^{(k+1)} &= (7 - x^{(k+1)} + 2z^{(k)})/3 \\ z^{(k+1)} &= (10 - x^{(k+1)} - 2y^{(k+1)})/3\,. \end{aligned}$$

Matrix form:

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}^{-1} \left\{ \begin{pmatrix} 5 \\ 7 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 2 & 0 \end{pmatrix} \mathbf{x}^{(k+1)} - \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{x}^{(k)} \right\}.$$

Iterates have converged to $x = 3$, $y = 2$, $z = 1$ correct to four decimal places.

- Gauss-Seidel iteration is most useful for sparse matrices.

4.5.2 Gauss-Jacobi Iteration

New iterates are calculated for all the variables using only the old iterates. It is more complex than Gauss-Seidel iteration to program and is therefore less commonly used. The matrix form is

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}\{\mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)}\}$$

or

$$\mathbf{D}\mathbf{x}^{(k+1)} = -(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{b}\,.$$

Example 4.21 Solve the equations in Example 4.20 using Gauss-Jacobi iteration.

Solution Reorder the equations. Gauss-Jacobi iteration scheme:

$$\begin{aligned} x^{(k+1)} &= (5 + y^{(k)} - z^{(k)})/2 \\ y^{(k+1)} &= (7 - x^{(k)} + 2z^{(k)})/3 \\ z^{(k+1)} &= (10 - x^{(k)} - 2y^{(k)})/3\,. \end{aligned}$$

Matrix form:

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}^{-1} \left\{ \begin{pmatrix} 5 \\ 7 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -2 \\ 1 & 2 & 0 \end{pmatrix} \mathbf{x}^{(k)} \right\}\,.$$

Very slow convergence to $x = 3$, $y = 2$, $z = 1$.

4.5.3 Convergence (A)

- Gauss-Seidel iteration is of the form

$$\mathbf{M}\mathbf{x}^{(k+1)} = -\mathbf{N}\mathbf{x}^{(k)} + \mathbf{b}\,, \qquad \text{where } \mathbf{A} = \mathbf{M} + \mathbf{N},\ \mathbf{M} = \mathbf{D} + \mathbf{L},\ \mathbf{N} = \mathbf{U}.$$

Gauss-Jacobi iteration: A = M + N, M = D, N = U + L

- If a solution $\mathbf{x}$ to the problem exists, then

$$\mathbf{M}\mathbf{x} = -\mathbf{N}\mathbf{x} + \mathbf{b}\,.$$

Let $\delta\mathbf{x}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$. Subtracting, $\mathbf{M}\delta\mathbf{x}^{(k+1)} = -\mathbf{N}\delta\mathbf{x}^{(k)}$ or

k+1 not k

$$\delta\mathbf{x}^{(k+1)} = -\mathbf{M}^{-1}\mathbf{N}\delta\mathbf{x}^{(k)} = \mathbf{H}\delta\mathbf{x}^{(k)} = \mathbf{H}^{k}\delta\mathbf{x}^{(0)}\,.$$

- If $\|\mathbf{H}\| < 1$, then $\mathbf{x}^{(k)} \to \mathbf{x}$ as $k \to \infty$.
  For $\mathbf{H}^k \to \mathbf{0}$, if $\|\mathbf{H}\| < 1$, since $\|\mathbf{H}^k\| \le \|\mathbf{H}\|^k$. But then $\delta\mathbf{x}^{(k)} \to \mathbf{0}$.

- Gauss-Seidel converges if $\mathbf{A}$ is *strictly diagonally-dominant*:

$$|a_{ii}| > \sum_{j=1, j\neq i} |a_{ij}|\,, \qquad i = 1, 2, \ldots, n\,.$$

  i.e. in each row diagonal element is absolutely greater than absolute sum of other elements.

- For special classes of matrices if Gauss-Jacobi iteration converges then Gauss-Seidel iteration converges faster, e.g. tridiagonal matrices. ($\mathbf{A}$ is *tridiagonal* if $a_{ij} = 0$ for $|i - j| > 1$.)

	Example 4.18		Example 4.19	
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0	0	0
1	1.5000	−1.1250	−3	−9
2	0.9375	−0.9844	33	63
3	1.0078	−1.0020	−255	−513
4	0.9990	−0.9998	2049	4095
5	1.0001	−1.0000		
6	1.0000	−1.0000		

Table 4.1: Results for Gauss-Seidel iteration of Examples 4.18 and 4.19.

	Gauss-Seidel Iteration			Gauss-Jacobi Iteration		
k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	0	0	0	0	0	0
1	2.5000	1.5000	1.5000	2.5000	2.3333	3.3333
2	2.5000	2.5000	.8333	2.0000	3.7222	.9444
3	3.3333	1.7778	1.0370	3.8889	2.2963	.1852
4	2.8704	2.0679	.9979	3.5556	1.1605	.5062
5	3.0350	1.9870	.9970	2.8272	1.4856	1.3745
6	2.9950	1.9997	1.0019	2.5556	2.3073	1.4005
7	2.9989	2.0016	.9993	2.9534	2.4152	.9433
8	3.0012	1.9991	1.0002	3.2359	1.9777	.7388
9	2.9995	2.0003	1.0000	3.1195	1.7472	.9362
10	3.0002	1.9999	1.0000	2.9055	1.9176	1.1287
11	3.0000	2.0000	1.0000	2.8945	2.1173	1.0864
12	3.0000	2.0000	1.0000	3.0154	2.0928	.9570

Table 4.2: Results of Examples 4.20 and 4.21.