1.2: Matrices, Vectors, & Gauss-Jordan Elimination

1.2
1/2

The matrix
- entries
- rows
- columns.
- double subscripts (a_{ij})

Types of matrices
- equal matrices.
- square
- diagonal
- upper/lower triangular
- zero
- coefficient matrix
- augmented matrix

Algorithm: a finite procedure written in a fixed symbolic vocabulary, governed by precise instructions, moving in discrete steps 1, 2, 3, ... whose execution requires no insight, cleverness, intuition, intelligence, or perspicuity, and that sooner or later comes to an end.

Berlinski, 2000.

Their's not to reason why
Their's but to do and die.
Tennyson

Vectors (a matrix w/ 1 row or 1 col).

We use "vector" to refer to column vectors.

$\mathbb{R}^n$ is the vector space of all vectors w/ dim. n.

Graphical representations of vectors

$$\vec{v} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \vec{u} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

examples of Gauss-Jordan Elimination

$$\left|\begin{array}{l} x - 2y + 3z = 9 \\ -x + 3y = -4 \\ 2x - 5y + 5z = 17 \end{array}\right| \Rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 3 & 9 \\ -1 & 3 & 0 & -4 \\ 2 & -5 & 5 & 17 \end{array}\right]$$

w/ unique sol. $(1, -1, 2)$

$$\left[\begin{array}{cccc|c} 0 & 1 & 1 & 2 & -3 \\ 1 & 2 & -1 & 0 & 2 \\ 2 & 4 & 1 & -3 & -2 \\ 2 & 5 & 2 & -5 & -5 \end{array}\right] \Rightarrow \left[\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

solution $(2-t, -1+t, -2+t, t)$

$$\left[\begin{array}{cccc|c} 1 & -1 & -1 & 2 & 1 \\ 2 & -2 & -1 & 3 & 3 \\ -1 & 1 & -1 & 0 & -3 \end{array}\right] \Rightarrow \left[\begin{array}{cccc|c} 1 & -1 & 0 & 1 & 2 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

solution $(2+s-t, s, 1+t, t)$

or $$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} + s\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t\begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

(see supplemental algorithms)

GAUSS - JORDAN
ELIMINATION
ALGORITHMS

Solving a system of linear equations
We proceed from equation to equation, from top to bottom.

Suppose we get to the ith equation. Let x_j be the leading variable of the system consisting of the ith and all the subsequent equations. (If no variables are left in this system, then the process comes to an end.)

- If x_j does not appear in the ith equation, swap the ith equation with the first equation below that does contain x_j.
- Suppose the coefficient of x_j in the ith equation is c; thus this equation is of the form $cx_j + \cdots = \cdots$. Divide the ith equation by c.
- Eliminate x_j from all the other equations, above and below the ith, by subtracting suitable multiples of the ith equation from the others.

Now proceed to the next equation.

If an equation *zero* = *nonzero* emerges in this process, then the system fails to have solutions; the system is *inconsistent*.

When you are through without encountering an inconsistency, solve each equation for its leading variable. You may choose the nonleading variables freely; the leading variables are then determined by these choices.

Bretscher, 15

ALGORITHM 6.1

Gaussian Elimination with Backward Substitution

To solve the $n \times n$ linear system

$$\begin{array}{lccccccccc}
E_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & a_{1,n+1} \\
E_2: & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & a_{2,n+1} \\
\vdots & \vdots & & \vdots & & & & \vdots & & \vdots \\
E_n: & a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & a_{n,n+1}
\end{array}$$

INPUT number of unknowns and equations n; augmented matrix $A = (a_{ij})$, where $1 \le i \le n$ and $1 \le j \le n+1$.

OUTPUT solution $x_1, x_2, \ldots, x_n$ or message that the linear system has no unique solution.

Step 1 For $i = 1, \ldots, n-1$ do Steps 2–4. (*Elimination process.*)

Step 2 Let p be the smallest integer with $i \le p \le n$ and $a_{pi} \ne 0$.
If no integer p can be found
then OUTPUT ('no unique solution exists');
STOP.

Step 3 If $p \ne i$ then perform $(E_p) \leftrightarrow (E_i)$.

Step 4 For $j = i+1, \ldots, n$ do Steps 5 and 6.

Step 5 Set $m_{ji} = a_{ji}/a_{ii}$.

Step 6 Perform $(E_j - m_{ji}E_i) \rightarrow (E_j)$;

Step 7 If $a_{nn} = 0$ then OUTPUT ('no unique solution exists');
STOP.

Step 8 Set $x_n = a_{n,n+1}/a_{nn}$. (*Start backward substitution.*)

Step 9 For $i = n-1, \ldots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] \Big/ a_{ii}$.

Step 10 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP.

Burden & Faires, 358

ALGORITHM

6.2

Gaussian Elimination with Partial Pivoting

To solve the $n \times n$ linear system

$$\begin{aligned}
E_1 &: \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1} \\
E_2 &: \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1} \\
&\vdots \\
E_n &: \quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_{n,n+1}
\end{aligned}$$

INPUT number of unknowns and equations n; augmented matrix $A = (a_{ij})$ where $1 \le i \le n$ and $1 \le j \le n+1$.

OUTPUT solution $x_1, \ldots, x_n$ or message that the linear system has no unique solution.

Step 1 For $i = 1, \ldots, n$ set $NROW(i) = i$. *(Initialize row pointer.)*

Step 2 For $i = 1, \ldots, n-1$ do Steps 3–6. *(Elimination process.)*

Step 3 Let p be the smallest integer with $i \le p \le n$ and $|a(NROW(p), i)| = \max_{i \le j \le n} |a(NROW(j), i)|$. *(Notation: $a(NROW(i), j) \equiv a_{NROW_i, j}$.)*

Step 4 If $a(NROW(p), i) = 0$ then OUTPUT ('no unique solution exists'); STOP.

Step 5 If $NROW(i) \ne NROW(p)$ then set $NCOPY = NROW(i)$;
$NROW(i) = NROW(p)$;
$NROW(p) = NCOPY$.
(Simulated row interchange.)

Step 6 For $j = i+1, \ldots, n$ do Steps 7 and 8.

Step 7 Set $m(NROW(j), i) = a(NROW(j), i)/a(NROW(i), i)$.

Step 8 Perform $(E_{NROW(j)} - m(NROW(j), i) \cdot E_{NROW(i)}) \to (E_{NROW(j)})$.

Step 9 If $a(NROW(n), n) = 0$ then OUTPUT ('no unique solution exists'); STOP.

Step 10 Set $x_n = a(NROW(n), n+1)/a(NROW(n), n)$. *(Start backward substitution.)*

Step 11 For $i = n-1, \ldots, 1$

$$\text{set } x_i = \frac{a(NROW(i), n+1) - \sum_{j=i+1}^{n} a(NROW(i), j) \cdot x_j}{a(NROW(i), i)}.$$

Step 12 OUTPUT $(x_1, \ldots, x_n)$; *(Procedure completed successfully.)* STOP.

Burden and Faires, 368

ALGORITHM
6.3

Gaussian Elimination with Scaled Partial Pivoting

The only steps in this algorithm that differ from those of Algorithm 6.2 are:

Step 1 For $i = 1, \ldots, n$ set $s_i = \max_{1 \le j \le n} |a_{ij}|$;
if $s_i = 0$ then OUTPUT ('no unique solution exists');
STOP.
set $NROW(i) = i$.

Step 2 For $i = 1, \ldots, n-1$ do Steps 3–6. (*Elimination process.*)

Step 3 Let p be the smallest integer with $i \le p \le n$ and

$$\frac{|a(NROW(p), i)|}{s(NROW(p))} = \max_{i \le j \le n} \frac{|a(NROW(j), i)|}{s(NROW(j))}.$$

Burden & Faires, 371