Lecture 11

Gaussian Elimination: Error Analysis

Owen L. Lewis

Department of Mathematics and Statistics
University of New Mexico

Sept. 24, 2024

Goals for today. . .

- When can a problem be “nearly” unsolvable?
- Formalize measurements of error (norms).
- Condition number
- Stability.
- How does MATLAB solve a matrix equation?
- Tri-diagonal systems.

Geometric Interpretation of Singularity

Consider a 2×2 system describing two lines that intersect

$$
\begin{aligned}
y &= -2x + 6 \\
y &= \frac{1}{2}x + 1
\end{aligned}
$$

The matrix form of this equation is

$$
\begin{bmatrix} 2 & 1 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}
$$

The equations for two **parallel** but **not intersecting** lines are

$$
\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}
$$

Here the coefficient matrix is singular ($\mathrm{rank}(A) = 1$), and the system is inconsistent

Geometric Interpretation of Singularity

The equations for two **parallel** and **coincident** lines are

$$\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$$

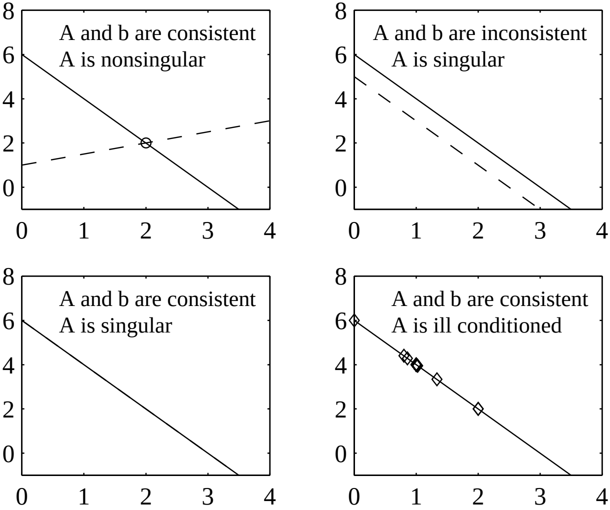
The equations for two **nearly parallel** lines are

$$\begin{bmatrix} 2 & 1 \\ 2+\delta & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 6+\delta \end{bmatrix}$$

Aside:
$Ax = b$ can be solved if $b \cdot y = 0$ for every $y \in null(A^T)$.

Notes

Geometric Interpretation of Singularity



Effect of Perturbations to b

Consider the solution of a 2×2 system where

$$b = \begin{bmatrix} 1 \\ 2/3 \end{bmatrix}$$

One expects that the *exact* solutions to

$$Ax = \begin{bmatrix} 1 \\ 2/3 \end{bmatrix} \qquad \text{and} \qquad Ax = \begin{bmatrix} 1 \\ 0.6667 \end{bmatrix}$$

will be different. Should these solutions be a **lot different** or a **little different**?

Norms

Vectors:

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + \ldots + |x_n|^p\right)^{1/p}$$

$$\|x\|_2 = \left(|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2\right)^{1/2} \quad \text{(2-norm or Euclidian Norm)}$$

$$\|x\|_1 = |x_1| + |x_2| + \ldots + |x_n| = \sum_{i=1}^{n} |x_i| \quad \text{(1-Norm)}$$

$$\|x\|_\infty = \max\left(|x_1|, |x_2|, \ldots, |x_n|\right) = \max_i\left(|x_i|\right)$$

Induces norms on matrices:

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

or equivalently ...

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$

Notes

Norms

For certain norms, no need to calculate these via definition:

$$\|A\|_1 = \max_{1 \leqslant j \leqslant n} \sum_{i=1}^{m} |a_{ij}| \quad \text{(Maximum absolute column sum)}$$

$$\|A\|_\infty = \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{n} |a_{ij}| \quad \text{(Maximum absolute row sum)}$$

$$\|A\|_2 = \sqrt{\max \lambda(A^T A)} = \max \sigma(A) \quad (\sigma(A)\text{: singular value of A})$$

$$\|A\|_F = \sqrt{\sum_{i=1n}^{n} \sum_{j=1}^{n} a_{ij}^2} \quad \text{(Frobenius norm NOT AN INDUCED NORM)}$$

Some Important Properties of Norms

$$
\begin{aligned}
\|\alpha x\| &= |\alpha|\|x\| \\
\|Ax\| &\leqslant \|A\|\|x\| \\
\|x + y\| &\leqslant \|x\| + \|y\|
\end{aligned}
$$

Notes

Effect of Perturbations to b

Perturb b with δb such that

$$\frac{\|\delta b\|}{\|b\|} \ll 1,$$

The resulting perturbed system is

$$A(x + \delta x_b) = b + \delta b$$

The perturbations satisfy

$$A \delta x_b = \delta b$$

Analysis shows (see next two slides for proof) that

$$\frac{\|\delta x_b\|}{\|x\|} \leqslant \|A\| \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}$$

Thus, the effect of the perturbation is small *if* $\|A\| \|A^{-1}\|$ is small.

$$\text{if } \|A\| \|A^{-1}\| \sim 1 \quad \text{then} \quad \frac{\|\delta x_b\|}{\|x\|} \ll 1$$

Effect of Perturbations to b (Proof)

Let $x + \delta x_b$ be the *exact* solution to the perturbed system

$$A(x + \delta x_b) = b + \delta b \tag{1}$$

Expand

$$Ax + A\delta x_b = b + \delta b$$

Subtract Ax from left side and b from right side since $Ax = b$

$$A\delta x_b = \delta b$$

Left multiply by A^{-1}

$$\delta x_b = A^{-1}\delta b \tag{2}$$

Effect of Perturbations to b (Proof, p. 2)

Take norm of equation (2)

$$\|\delta x_b\| = \|A^{-1}\,\delta b\|$$

Applying consistency requirement of matrix norms

$$\|\delta x_b\| \leqslant \|A^{-1}\|\|\delta b\| \tag{3}$$

Similarly, $Ax = b$ gives $\|b\| = \|Ax\|$, and

$$\|b\| \leqslant \|A\|\|x\| \tag{4}$$

Rearrangement of equation (4) yields

$$\frac{1}{\|x\|} \leqslant \frac{\|A\|}{\|b\|} \tag{5}$$

Effect of Perturbations to b (Proof)

Multiply Equation (5) by Equation (3) to get

$$\frac{\|\delta x_b\|}{\|x\|} \leqslant \|A\|\|A^{-1}\|\frac{\|\delta b\|}{\|b\|} \tag{6}$$

Summary:

If $x + \delta x_b$ is the *exact* solution to the perturbed system

$$A(x + \delta x_b) = b + \delta b$$

then

$$\frac{\|\delta x_b\|}{\|x\|} \leqslant \|A\|\|A^{-1}\|\frac{\|\delta b\|}{\|b\|}$$

Effect of Perturbations to A

Perturb A with δA such that

$$\frac{\|\delta A\|}{\|A\|} \ll 1,$$

The resulting perturbed system is

$$(A + \delta A)(x + \delta x_A) = b$$

Analysis shows that

$$\frac{\|\delta x_A\|}{\|x + \delta x_A\|} \leqslant \|A\|\|A^{-1}\| \frac{\|\delta A\|}{\|A\|}$$

Thus, the effect of the perturbation is small *if* $\|A\|\|A^{-1}\|$ is small.

$$\text{if} \|A\|\|A^{-1}\| \sim 1 \quad \text{then} \quad \frac{\|\delta x_A\|}{\|x + \delta x_A\|} \ll 1$$

Effect of Perturbations to both A and b

Perturb both A with δA and b with δb such that

$$\frac{\|\delta A\|}{\|A\|} \ll 1 \quad \text{and} \quad \frac{\|\delta b\|}{\|b\|} \ll 1$$

The resulting perturbed system satisfies

$$(A + \delta A)(x + \delta x) = b + \delta b$$

Analysis shows that

$$\frac{\|\delta x\|}{\|x\|} \leqslant \|A\|\|A^{-1}\| \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]$$

Thus, the effect of the perturbation is small *if* $\|A\|\|A^{-1}\|$ is small.

$$\text{if} \quad \|A\|\|A^{-1}\| \sim 1 \quad \text{then} \quad \frac{\|\delta x\|}{\|x + \delta x\|} \ll 1$$

Condition number of A

The **condition number**

$$\kappa(A) \equiv \|A\|\|A^{-1}\|$$

indicates the sensitivity of the solution to perturbations in A and b. The condition number can be measured with any p-norm.
The condition number is always in the range

$$1 \leqslant \kappa(A) \leqslant \infty$$

- $\kappa(A)$ *is a mathematical property of* A
- *Any algorithm will produce a solution that is sensitive to perturbations in* A *and* b *if* $\kappa(A)$ *is large.*
- *In exact math a matrix is either singular or non-singular.* $\kappa(A) = \infty$ *for a singular matrix*
- $\kappa(A)$ *indicates how close* A *is to being numerically singular.*
- *A matrix with large* κ *is said to be* ***ill-conditioned***

Computational Stability

In Practice, applying Gaussian elimination with partial pivoting and back substitution to $Ax = b$ gives the **exact solution**, $\hat{x}$, to the **nearby problem**

$$(A + E)\hat{x} = b \quad \text{where} \quad \|E\|_\infty \leqslant \varepsilon_m \|A\|_\infty$$

> *Gaussian elimination with partial pivoting and back substitution "gives exactly the right answer to nearly the right question."*
>
> — *Trefethen and Bau*

Computational Stability

An algorithm that gives the exact answer to a problem that is near to the original problem is said to be **backward stable**. Algorithms that are not backward stable will tend to amplify roundoff errors present in the original data. As a result, the solution produced by an algorithm that is not backward stable will not necessarily be the solution to a problem that is close to the original problem.

Gaussian elimination without partial pivoting is *not* backward stable for arbitrary A.

If A is symmetric and positive definite, then Gaussian elimination without pivoting is backward stable.

The Residual

Let $\hat{x}$ be the numerical solution to $Ax = b$.
$\Rightarrow$ $\hat{x} \neq x$ (x is the exact solution) because of roundoff (or other reasons). The error is easy to define:

$$e = x - \hat{x},$$

but not always easy to evaluate. To compute e we would need to know x. Instead of the **error** we often calculate the **residual**.

The residual measures how close $\hat{x}$ is to satisfying: the original equation

$$r = b - A\hat{x}.$$

The Residual

If the error and residual are defined

$$e = x - \hat{x},$$
$$r = b - A\hat{x},$$

then it follows

$$Ae = r. \qquad \star\star\star$$

It is not hard to show that

$$\frac{\|e\|}{\|\hat{x}\|} \leqslant \kappa(A) \frac{\|r\|}{\|A\| \, \|\hat{x}\|}$$

- Small $\|r\|$ does not guarantee a small $\|e\|$.
- If $\kappa(A)$ is large the $\hat{x}$ returned by Gaussian elimination and back substitution (or any other solution method) is *not* guaranteed to be anywhere near the true solution to $Ax = b$.

Rules of Thumb

- Applying Gaussian elimination with partial pivoting and back substitution to $Ax = b$ yields a numerical solution $\hat{x}$ such that the residual vector $r = b - A\hat{x}$ is small *even if* the $\kappa(A)$ is large.
- If A and b are stored to machine precision ε_m, the numerical solution to $Ax = b$ by any variant of Gaussian elimination is correct to d digits where

$$d = |\log_{10}(\varepsilon_m)| - \log_{10}(\kappa(A))$$

Rules of Thumb

$$d = |\log_{10}(\varepsilon_m)| - \log_{10}(\kappa(A))$$

Example:
MATLAB computations have $\varepsilon_m \approx 2.2 \times 10^{-16}$. For a system with $\kappa(A) \sim 10^{10}$ the elements of the solution vector will have

$$\begin{aligned} d &= |\log_{10}(2.2 \times 10^{-16})| - \log_{10}\left(10^{10}\right) \\ &\approx 15 - 10 \\ &= 5 \end{aligned}$$

correct (decimal) digits

Summary of Limits to Numerical Solution of $Ax = b$

1. $\kappa(A)$ indicates how close A is to being numerically singular
2. If $\kappa(A)$ is “large”, A is **ill-conditioned** and *even the best* numerical algorithms will produce a solution, $\hat{x}$ that cannot be guaranteed to be close to the true solution, x
3. In practice, Gaussian elimination with partial pivoting and back substitution produces a solution with a small residual

$$r = b - A\hat{x}$$

   even if $\kappa(A)$ is large.

The Backslash Operator

Consider the scalar equation

$$5x = 20 \qquad \implies \qquad x = (5)^{-1}20$$

The extension to a system of equations is, of course

$$Ax = b \qquad \implies \qquad x = A^{-1}b$$

where $A^{-1}b$ is the formal solution to $Ax = b$
In MATLAB notation the system is solved with

```
x = A\b
```

The Backslash Operator

Given an $n \times n$ matrix A, and an $n \times 1$ vector b the \ operator performs a sequence of tests on the A matrix. MATLAB attempts to solve the system with the method that gives the least roundoff and the fewest operations.
When A is an $n \times n$ matrix:

1. MATLAB examines A to see if it is a permutation of a triangular system

   If so, the appropriate triangular solve is used.

2. MATLAB examines A to see if it *appears* to be symmetric and positive definite.

   If so, MATLAB attempts a Cholesky factorization
   and two triangular solves.

3. If the Cholesky factorization fails, or if A does not appear to be symmetric,

   MATLAB attempts an LU factorization
   and two triangular solves.

More Algorithms for Special Systems

- tridiagonal systems
- banded systems

Tridiagonal

A tridiagonal matrix A

$$\begin{bmatrix} d_1 & c_1 & & & & & & \\ a_1 & d_2 & c_2 & & & & & \\ & a_2 & d_3 & c_3 & & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & a_{i-1} & d_i & c_i & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & & a_{n-1} & d_n \end{bmatrix}$$

- storage is saved by not saving zeros
- only $n + 2(n-1) = 3n - 2$ places are needed to store the matrix (i.e., $O(n)$ storage) versus n^2 storage for dense system
- can operations be saved? yes!

Tridiagonal

$$\begin{bmatrix} d_1 & c_1 & & & & & & \\ a_1 & d_2 & c_2 & & & & & \\ & a_2 & d_3 & c_3 & & & & \\ & & \dots & \dots & \dots & & & \\ & & & a_{i-1} & d_i & c_i & & \\ & & & & \dots & \dots & \dots & \\ & & & & \dots & \dots & \dots & \\ & & & & & & a_{n-1} & d_n \end{bmatrix}$$

Start forward elimination (without any special pivoting)

1. subtract a_1/d_1 times row 1 from row 2
2. this eliminates a_1, changes d_2 and does not touch c_2
3. continuing:

$$d_i = d_i - \left(\frac{a_{i-1}}{d_{i-1}} c_{i-1}\right)$$

   for $i = 2 \dots n$

Tridiagonal

$$\begin{bmatrix} \tilde{d}_1 & c_1 & & & & & & \\ & \tilde{d}_2 & c_2 & & & & & \\ & & \tilde{d}_3 & c_3 & & & & \\ & & & \cdots & \cdots & & & \\ & & & & \tilde{d}_i & c_i & & \\ & & & & & \cdots & \cdots & \\ & & & & & \cdots & \cdots & \\ & & & & & & & \tilde{d}_n \end{bmatrix}$$

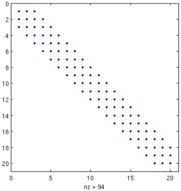
This leaves an upper triangular (2-band). With back substitution:

1. $x_n = \tilde{b}_n / \tilde{d}_n$
2. $x_{n-1} = (1/\tilde{d}_{n-1})(\tilde{b}_{n-1} - c_{n-1}x_n)$
3. $x_i = (1/\tilde{d}_i)(\tilde{b}_i - c_i x_{i+1})$

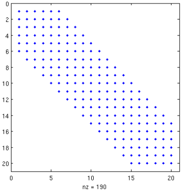
Tridiagonal Algorithm

```
input: n, a, d, c, b
for i = 2 to n
    xmult = a_{i-1}/d_{i-1}
    d_i = d_i - xmult · c_{i-1}
    b_i = b_i - xmult · b_{i-1}
end
x_n = b_n/d_n
for i = n-1 down to 1
    x_i = (b_i - c_i x_{i+1})/d_i
end
```

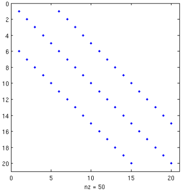
m-band



$m = 5$



$m = 11$



$m = 11$

- the m correspond to the total width of the non-zeros
- after a few passes of GE *fill-in* with occur within the band
- so an empty band costs (about) the same as a non-empty band
- one fix: reordering (e.g. Cuthill-McKee)
- generally GE will cost $\mathcal{O}(m^2 n)$ for m-band systems