ACM40290: Numerical Algorithms

Numerical Linear Algebra III

Dr Barry Wardell
School of Mathematics and Statistics
University College Dublin

Linear Algebra in MATLAB

Linear Algebra in MATLAB

Reminder: LU factorisation with partial pivoting

$U = A,\ L = \mathrm{I},\ P = \mathrm{I}$
for $j = 1 : n-1$ **do**
    Select $i (\geq j)$ that maximizes $|u_{ij}|$
    Interchange rows of U: $u_{(j,j:n)} \leftrightarrow u_{(i,j:n)}$
    Interchange rows of L: $\ell_{(j,1:j-1)} \leftrightarrow \ell_{(i,1:j-1)}$
    Interchange rows of P: $p_{(j,:)} \leftrightarrow p_{(i,:)}$
    for $i = j+1 : n$ **do**
        $\ell_{ij} = u_{ij} / u_{jj}$
        for $k = j : n$ **do**
            $u_{ik} = u_{ik} - \ell_{ij} u_{jk}$
        end for
    end for
end for

Linear Algebra in MATLAB

- The MATLAB call `[L, U, P] = lu(A)` returns the permutation matrix but the call `[L̂, U] = lu(A)` permutes the lower triangular factor directly.
- In MATLAB, the implicit linear solve **backslash operator**

  ```
  x = A\b
  ```

  is **equivalent** to performing an LU factorisation and doing two triangular solves.

Linear Algebra in MATLAB

```
>> A=[1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16];
>> [L, U, P] = lu(A)

L =
    1.0000         0         0         0
    0.0769    1.0000         0         0
    0.6923    0.3333    1.0000         0
    0.3846    0.6667   -0.6400    1.0000
U =
   13.0000   14.0000   15.0000   16.0000
         0    0.9231    1.8462    2.7692
         0         0   -0.0000         0
         0         0         0    0.0000
P =
     0     0     0     1
     1     0     0     0
     0     0     1     0
     0     1     0     0
```

Linear Algebra in MATLAB

```
>> [L, U] = lu(A)

L =

    0.0769    1.0000         0         0
    0.3846    0.6667   -0.6400    1.0000
    0.6923    0.3333    1.0000         0
    1.0000         0         0         0

U =

   13.0000   14.0000   15.0000   16.0000
         0    0.9231    1.8462    2.7692
         0         0   -0.0000         0
         0         0         0    0.0000
```

Hermitian Matrices

- The MATLAB built in function

  `R=chol(A)`

  gives the Cholesky factorisation and is a good way to **test for positive-definiteness**.

- For Hermitian/symmetric matrices with positive diagonals MATLAB tries a Cholesky factorisation first, before resorting to LU factorisation with pivoting.

- The cost of a Cholesky factorisation is about half the cost of Gaussian Elimination, $n^3/3$ FLOPS.

Hermitian Matrices

```
>> % generate a random matrix and right-hand side
>> n = 2000;
>> Z = randn(n, n);
>> b = randn(n, 1);

>> % solve using LU factorisation
>> tic; x = Z\b; toc
Elapsed time is 0.088781 seconds.

>> % generate a symmetric positive definite matrix A
>> A = Z'*Z;
>> tic; x = A\b; toc
Elapsed time is 0.044225 seconds.
```

Errors and Stability

Vector Norms

- Norms are the abstraction for the notion of a length or **magnitude**.
- For a vector $\boldsymbol{x} \in \mathbb{R}^n$, the p-norm is

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- Special cases of interest are:
  - The 1-norm (L^1 norm, **Manhattan distance**), $||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$
  - The 2-norm (L^2 norm, **Euclidean distance**), $||\mathbf{x}||_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{\sum_{i=1}^{n} |x_i|^2}$
  - The ∞-norm (L^∞ norm, **maximum**), $||\mathbf{x}||_\infty = \max_{1 \le i \le n} |x_i|$
- Note that all of these norms are inter-related in a finite-dimensional setting.

Matrix Norms

- Matrix norm **induced** by a given vector norm:

$$||\mathbf{A}|| = \sup_{\mathbf{x}\neq 0} \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} \quad \Longrightarrow \quad ||\mathbf{A}\mathbf{x}|| \leq ||\mathbf{A}||||\mathbf{x}||$$

- Special cases of interest are:
  - The 1-norm or **column sum norm**, $||\mathbf{A}||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$
  - The ∞-norm or **row sum norm**, $||\mathbf{A}||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$
  - The Euclidean of **Frobenius norm**, $||\mathbf{A}||_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$ (note that this is not an induced norm)
  - The 2-norm or **spectral norm**, $||\mathbf{A}||_2 = \sigma_1$ (largest singular value)

$$|\rho(\mathbf{A}^T\mathbf{A})|^{1/2}$$ ρ - spectral radius = max eigenvalue of $\mathbf{A}^T\mathbf{A}$

Condition Number

- A measure of how sensitive a matrix is to perturbations in data is given by the **condition number**

$$\text{cond}(a) = \|\mathbf{A}\|_p \, \|\mathbf{A}^{-1}\|_p$$

- In MATLAB, use command `cond(A,p)`
- Basic properties:
  - `cond(I)` $= 1$
  - `cond(A)` ≥ 1
  - `cond(O)` $= 1$ if **O** is orthogonal
  - `cond(A)` $= \infty$ if **A** is singular

Condition Number

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\|_p \, \|\mathbf{A}^{-1}\|_p$$

- Expensive to compute cond($\mathbf{A}$) as it in principle involves $\mathbf{A}^{-1}$
- Condition number bounded below by

$$1 \leq \frac{|\lambda_{max}|}{|\lambda_{min}|} \leq \text{cond}(A)$$

- MATLAB provides two condition number estimates: `condest` and `rcond`

Condition Number

How does condition number arise?

- Want to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$
- In any real problem, $\mathbf{b}$ ($\mathbf{A}$ as well) is not known exactly.
- Condition number says how errors in $\mathbf{b}$ ($\mathbf{A}$) feed into errors in $\mathbf{x}$. A matrix with a large condition number is said to be **ill-conditioned**.

Stability analysis: rhs perturbations

Perturbations on right hand side (rhs) only:

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b} \implies \mathbf{b} + \mathbf{A}\delta\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$$

$$\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b} \implies ||\delta\mathbf{x}|| \leq ||\mathbf{A}^{-1}||\,||\delta\mathbf{b}||$$

Using the bounds

$$||\mathbf{b}|| \leq ||\mathbf{A}||\,||\mathbf{x}|| \qquad \implies \qquad ||\mathbf{x}|| \geq ||\mathbf{b}||/||\mathbf{A}||$$

the relative error in the solution can be bounded by

$$\frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leq \frac{||\mathbf{A}^{-1}||\,||\delta\mathbf{b}||}{||\mathbf{x}||} \leq \frac{||\mathbf{A}^{-1}||\,||\delta\mathbf{b}||}{||\mathbf{b}||/||\mathbf{A}||} = \mathrm{cond}(\mathbf{A})\frac{||\delta\mathbf{b}||}{||\mathbf{b}||}$$

where the **condition number** cond(A) depends on the matrix norm used:

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

Stability analysis: matrix perturbations

Perturbations of the matrix only:

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} \;\Longrightarrow\; \delta\mathbf{x} = -\mathbf{A}^{-1}(\delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x})$$

$$\frac{||\delta\mathbf{x}||}{||\mathbf{x} + \delta\mathbf{x}||} \leq ||\mathbf{A}^{-1}||\,||\delta\mathbf{A}|| = \mathrm{cond}(\mathbf{A})\frac{||\delta\mathbf{A}||}{||\mathbf{A}||}$$

Conclusion: the conditioning of the linear system is determined by

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

No numerical method can cure an ill-conditioned system.

Stability analysis: mixed perturbations

Now consider general perturbations of the data:

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$$

$$\Rightarrow \quad \frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leq \frac{\operatorname{cond}(\mathbf{A})}{1 - \operatorname{cond}(\mathbf{A})\frac{||\delta\mathbf{A}||}{||\mathbf{A}||}} \left(\frac{||\delta\mathbf{b}||}{||\mathbf{b}||} + \frac{||\delta\mathbf{A}||}{||\mathbf{A}||} \right)$$

Important practical estimate:
Roundoff error in data, with rounding unit u ($\approx 10^{-16}$ for double precision), produces a relative error

$$\frac{||\delta\mathbf{x}||_\infty}{||\mathbf{x}||_\infty} \lesssim 2u \operatorname{cond}(\mathbf{A})$$

It certainly makes no sense to try to solve systems with $\operatorname{cond}(\mathbf{A}) > 10^{16}$

Example: ill-conditioned matrix

$A = \begin{bmatrix} 1 & .99 \\ .99 & .98 \end{bmatrix}$, $\lambda_1 = 1.98$, $\lambda_2 = -0.00005$, $\text{cond}^*(A) = 39,600$

Matrix is ill-conditioned.

Consider the system

$$\left.\begin{aligned} x_1 + .99\,x_2 &= 1.99 \\ .99\,x_1 + .98\,x_2 &= 1.97 \end{aligned}\right\} \Rightarrow \begin{aligned} x_1 &= 1 \\ x_2 &= 1 \end{aligned}$$

The system

$$\left.\begin{aligned} x_1 + .99\,x_2 &= 1.989903 \\ .99\,x_1 + .98\,x_2 &= 1.970106 \end{aligned}\right\} \Rightarrow \begin{aligned} x_1 &= 3.0 \\ x_2 &= -1.0203 \end{aligned}$$

Popular Misconception

- Often said that a small determinant is an indication of an ill-conditioned matrix. **Not true!**
- Counter example: $A_{nxn} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_i = {}^1/_{10}$
- $\det(\mathbf{A}) = 10^{-n}$. Small, even for moderate n values.
- Solution of $\mathbf{Ax} = \mathbf{b}$ is $x_i = 10\, b_i$

$$\mathrm{cond}(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|} = 1$$

Errors and Residuals

In many applications we want to solve $\mathbf{Ax} = \mathbf{b}$ such that

$$\|\mathbf{Ax} - \mathbf{b}\| = \|\mathbf{r}\|$$

is small.

What is the relationship between $\mathbf{r}$ and errors in $\mathbf{x}$?

Errors and Residuals

Let $\tilde{x}$ be computed solution

$$\underline{\delta x} = \underline{x} - \underline{\tilde{x}}$$

$$\underline{r} = \underline{b} - A\underline{\tilde{x}}$$

$$A\underline{\delta x} = A\underline{x} - A\underline{\tilde{x}} = b - A\tilde{x} = \underline{r}$$

$$\underline{\delta x} = A^{-1}\underline{r}$$

$$\|\underline{\delta x}\| \leq \|A^{-1}\| \, \|\underline{r}\|$$

Errors and Residuals

$$\frac{\|\delta x\|}{\|\underline{x}\|} \le \|A^{-1}\| \frac{\|\underline{r}\|}{\|\underline{x}\|}$$

$$= \|A\| \|A^{-1}\| \frac{\|\underline{r}\|}{\|A\| \|\underline{x}\|}$$

$$\le \|A\| \|A^{-1}\| \frac{\|\underline{r}\|}{\|A\underline{x}\|} = \|A\| \|A^{-1}\| \frac{\|\underline{r}\|}{\|\underline{b}\|}$$

$$\frac{\|\underline{\delta x}\|}{\|\underline{x}\|} \le \mathrm{cond}(A) \frac{\|\underline{r}\|}{\|\underline{b}\|}$$

Errors and Residuals

Conclusion: If A is ill-conditioned then small $\|\mathbf{r}\|$ does not imply small $||\delta\mathbf{x}||/||\mathbf{x}||$

Example

Example: Which solution is better?

$$\begin{bmatrix} .78 & .563 \\ .913 & .659 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} .217 \\ .254 \end{bmatrix}$$

Suppose we obtain two approximate solutions

$$\tilde{x} = \begin{bmatrix} .341 \\ -.087 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} .999 \\ -1.000 \end{bmatrix}$$

Residuals

$$\underline{b} - A\tilde{x} = [0.000001, \ 0]^T$$

$$\underline{b} - A\hat{x} = [0.00078, \ 0.000913]^T$$

Forward Errors

$$x - \tilde{x} = [0.659, \ -0.913]$$

$$x - \hat{x} = [0.001, \ 0]$$