LINEAR SYSTEMS

- **Topic:** How to efficiently (and accurately) solve a systems of linear equations
  - Problem of independent interest
  - The solution of linear system is often an essential intermediate step in more complex procedures
  - The mathematical tools that we shall now introduce will be extensively used in the following

Preliminaries

- Consider a generic system of linear equations:

$$Ax=b$$

where:

  - x and b are real $n \times 1$ vectors
  - A is a real $n \times n$ matrix known as the coefficient matrix.

- Hence, any system of the form:

$$\boxed{\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, n}$$

- **Theorem:** The system $Ax = b$ has a unique solution for any b if and only if A is nonsingular.
- The obvious way (but not the best one, as we will see) to numerically solve a linear system is to compute the inverse of A and multiply both sides by A^{-1}:

$$x = A^{-1}b$$

- In principle, this procedure works as long as A is nonsingular.
- However, if A is **nearly singular**, the small round-off errors that inevitably arise during computations on real-world computers may propagate explosively and generate large errors in the solution.

- Hence, a linear system characterized by a nearly singular coefficient matrix is unstable: small variations in b lead to large variations in the solution.
- Unfortunately, a small determinant is not a direct sign of near singularity:
  - For instance, the matrix εI_n, where ε is an arbitrarily small number, has independent rows and columns, being therefore clearly nonsingular, but presents an arbitrarily small determinant, since $|\varepsilon I_n| = \varepsilon^n$.
- Hence, alternative indicators of near singularity have to be used (the **condition number**).

- Even if the coefficient matrix is invertible, to obtain the inverse is computationally costly, and should be **avoided**.
- Fortunately, we don't need to explicitly compute the inverse of A in order to solve $Ax = b$:
    - **Direct methods** compute the solution in one step with the highest accuracy, but can be costly if the system is large.
    - **Iterative methods** compute the solution in more steps by successive approximation, and can be more efficient in solving large (and sparse) system, even if convergence is not guaranteed.

The condition number

Definition *Let X and Y be two normed vector spaces, and* $T: X \rightarrow Y$ *a linear operator. We define the induced norm of T as:*

$$\|T\| \equiv \sup_{\{x \in X : \|x\| = 1\}} \|T(x)\|$$

Note that $\|T\|$ *is specific to the norms on X and Y.*

Definition *Let A be a real square matrix. The induced norm of the linear operator* $T \equiv Ax : R^n \rightarrow R^n$ *is called the induced matrix norm of A, and is denoted* $\|A\|$.

Definition *Let X and Y be two normed vector spaces. Furthermore, let $T : X \to Y$ be a bounded linear operator, and $T^{-1} : X \to Y$ its bounded inverse. The* ***condition number*** *of T is defined as:*

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

Remark *If A is a real square matrix, then $\kappa(A) = \|A\| \, \|A^{-1}\|$ is the condition number of the linear operator $T \equiv Ax$. Note that the definition of $\kappa(A)$ makes sense only if A is nonsingular; by convention, the condition number of a singular matrix is ∞.*

We can formally prove that:

1. $\kappa(T) = \|T\| \|T^{-1}\| \geq \|TT^{-1}\| = \|I_n\| = 1$; note that $\kappa(I_n) = 1$, and therefore the "degree" of singularity increases with the condition number.

2. We know that λ is an eigenvalue of A only if λ^{-1} is an eigenvalue of A^{-1}: therefore, $\|A^{-1}\| \geq |\lambda_{\min}|^{-1}$. This implies that $\kappa(A) \geq \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$.

3. The condition number can be interpreted as the elasticity of the solution to $Ax = b$ with respect to b. More precisely, we can show that:
$$\kappa(A) = \frac{\|\tilde{x} - x\|}{\|x\|} \div \frac{\|\delta\|}{\|b\|}$$
where $\tilde{x} = A^{-1}(b + \delta)$ is the solution to a slightly perturbed version of the system.

In practical applications, the condition number depends clearly on the norm on R^n for which it is defined. The most commonly used norms on R^n are:

1. the l_∞ norm, for which:

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$$

where $\|A\|_\infty \equiv \max_j\left(\sum_i |a_{ij}|\right)$;

2. the Euclidean norm, or l_2, for which:

$$\kappa_2(A) = \frac{|\mu_{\max}|}{|\mu_{\min}|}$$

where $\mu_{\max}$ and $\mu_{\min}$ are respectively the largest and smallest *singular values* of A, i.e. the square roots of the largest and smallest eigenvalues of A^*A (A^* is the adjoint of A).

The number $\kappa^*(A) \equiv \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$ is called *spectral condition number* of A, and is often used as a norm-independent estimator for the true condition number.

Direct solution methods

- The matrix A **may** be diagonal, lower triangular, or upper triangular:

$$\underbrace{\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}}_{Diagonal}, \underbrace{\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{Lower\ triangular}, \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}}_{Upper\ triangular}$$

1. If the matrix is diagonal, then $x_i = b_i/a_i$ for $\forall i$.

2. If the matrix is lower triangular, we may solve for x by *forward substitution*: $x_1 = b_1/a_{11}, x_2 = (b_2 - a_{21}x_1)/a_{22}, x_i = \dfrac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}}$.

3. If the matrix is upper triangular, we can proceed by *backward substitution*: $x_n = b_n/a_{nn}, x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}$, and so on.

- Note that we solved the linear system without explicitly inverting the coefficient matrix: in other words, we applied a **direct solution method**.
- If A is neither diagonal nor triangular, a general approach is needed.
- **Gaussian elimination** solves linear systems characterized by nonsingular coefficient matrices by transforming them into equivalent upper triangular systems that can be solved via backward substitution.

Consider the following system:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A^{[0]}} x = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{b^{[0]}}$$

and assume that $a_{11} \neq 0$.

Subtract the first row multiplied by $l_{i1} = a_{i1}/a_{11}$ from the remaining $n-1$ rows, where $i = 2, 3, \ldots, n$, to obtain:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{[1]} & a_{23}^{[1]} \\ 0 & a_{32}^{[1]} & a_{33}^{[1]} \end{bmatrix}}_{A^{[1]}} x = \underbrace{\begin{bmatrix} b_1 \\ b_2^{[1]} \\ b_3^{[1]} \end{bmatrix}}_{b^{[1]}}$$

where $a_{ij}^{[1]} \equiv a_{ij} - l_{i1} a_{1j}$ and $b_i \equiv b_i - l_{i1} b_1$.

Assume now that $a_{22}^{[1]} \neq 0$, and subtract the second row of $A^{[1]}$ multiplied by $l_{i2} = a_{i1}^{[1]}/a_{22}^{[1]}$ from the remaining $n-2$ rows of $A^{[1]}$, to obtain:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{[1]} & a_{23}^{[1]} \\ 0 & 0 & a_{33}^{[2]} \end{bmatrix}}_{A^{[2]}} x = \underbrace{\begin{bmatrix} b_1 \\ b_2^{[1]} \\ b_3^{[2]} \end{bmatrix}}_{b^{[2]}}$$

where $a_{ij}^{[2]} \equiv a_{ij}^{[1]} - l_{i2}a_{2j}^{[1]}$ and $b_i^{[2]} \equiv b_i^{[1]} - l_{i2}b_2^{[1]}$.

The resulting upper triangular system $A^{[2]}x = b^{[2]}$ can now be solved by backward substitution. The procedure followed to obtain $A^{[2]}$ is known as *row reduction*.

For a generic $n \times n$ matrix A:

$$\prod_{i=n-1}^{1} L^{[i]} A^{[0]} = A^{[n-1]}$$

where:

$$L^{[i]} \equiv I_n - \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & l_{i+1,i} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & l_{ni} & \cdots & 0 \end{bmatrix}$$

Note that $\prod_{i=n-1}^{1} L^{[i]}$ is invertible by construction, and therefore:

$$A = LU$$

where $A = A^{[0]}$ by definition, $L \equiv \left(\prod_{i=n-1}^{1} L^{[i]}\right)^{-1}$ is a lower triangular matrix with only unit diagonal elements, and $U \equiv A^{[n-1]}$ is an upper diagonal matrix.

- This is called the ***LU* decomposition** (or factorization) of the matrix A.
- Row reduction produces a **unique** LU decomposition for any non singular square matrix.
- Once the LU decomposition of A is available, we can complete the Gaussian elimination procedure and:
  - replace $Ax=b$ with the equivalent system $LUx=b$;
  - solve the lower triangular system $Lz=b$ for z;
  - solve the upper triangular system $Ux=z$ for x.

- Gaussian elimination computes efficiently both the determinant and the inverse of a matrix.
- We know that $|A|=|L||U|$, i.e. that the determinant of a triangular matrix is the product of its diagonal elements, and that L has unit diagonal elements. Therefore:

$$|A| = |U| = \prod_{i=1}^{n} a_{ii}^{[i-1]}$$

- A^{-1} can be efficiently computed by solving n linear systems of the form $Ax_i=e_i$ where x_i corresponds to the i_{th} column of A^{-1} and e_i to the i_{th} column of I_n.

Other decompositions

Theorem *Any real square matrix A can be decomposed as:*

$$A = QR$$

where Q is unitary matrix, i.e. $Q'Q = QQ' = I$, and R is an upper triangular matrix.

The system $Ax = b$ can then be rewritten as:

$$QRx = b$$

and multiplied by Q' to obtain an equivalent system easily solvable via backward substitution:

$$Rx = Q'b$$

Since QR decomposition does not require pivoting, it may seem a more reliable solution method, but unfortunately the currently available algorithms are far more computationally intensive than Gaussian elimination with pivoting.

In the (unlikely) case that the matrix A is symmetric and positive definite, a very efficient alternative to Gaussian elimination is available.

Theorem *Any real square symmetric positive definite matrix A can be decomposed into:*

$$A = CC'$$

where C is a lower triangular matrix with positive diagonal elements.

This is known as *Cholesky decomposition*, and can be easily and efficiently computed.

The solution to $Ax = b$ is then obtained in two steps: the lower triangular system $Cz = b$ is solved for z, and the upper triangular system $C'x = z$ for x.

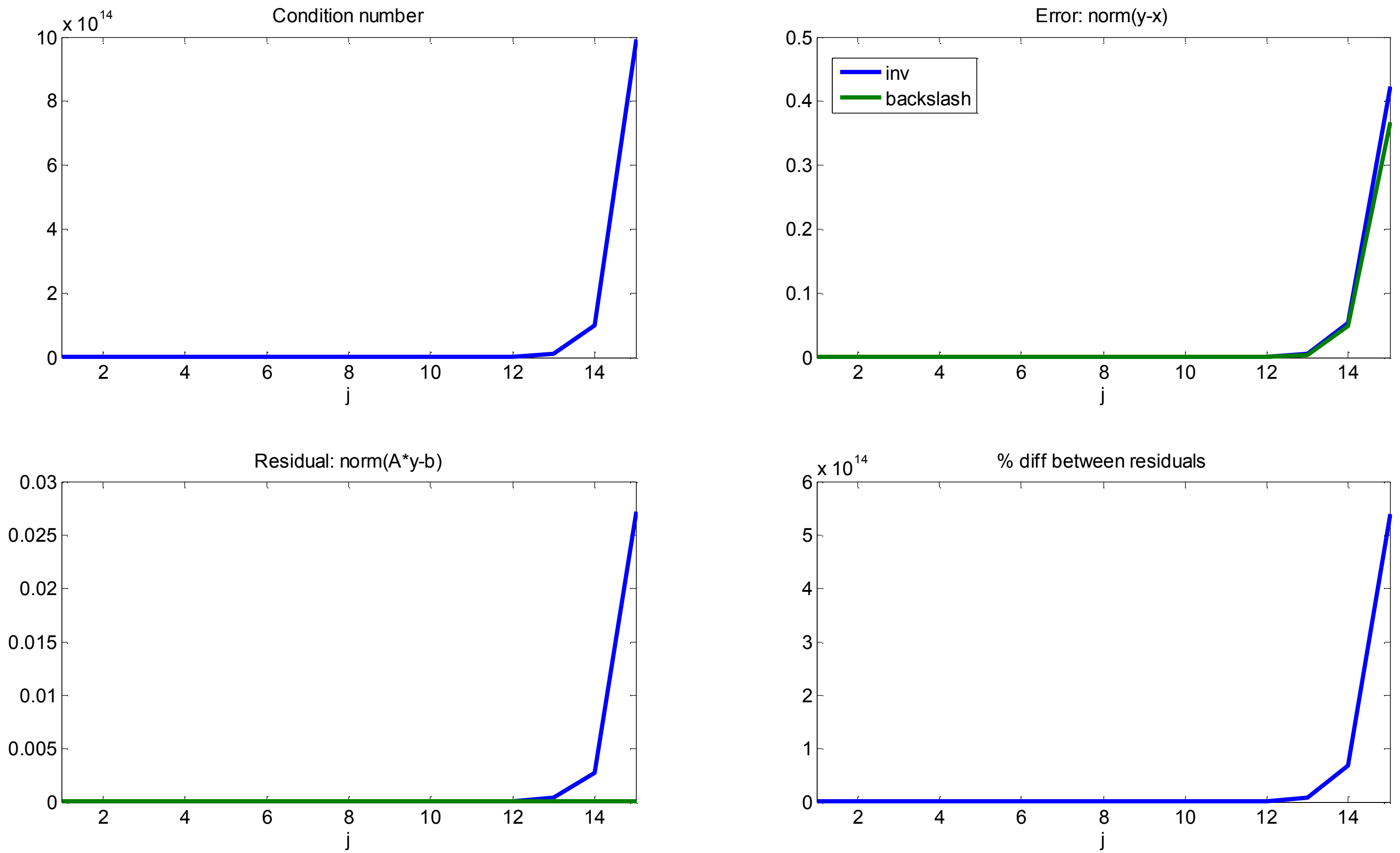
- Let us build a random matrix *A* of order 500 so that its condition number is 10^{10} and its l_2-*norm* is 1.
- By construction, the exact solution *x* is a random vector of length 500, and therefore the right-hand side of the equation is defined as $b=Ax$.
- Hence, the system is badly conditioned but internally consistent.
- Let us solve the system by direct computation of the inverse and by Gaussian elimination, and compare the l_2-norm of the numerical errors.


```
n=1000;
Q=orth(randn(n));
d=logspace(0,-10,n);
A=Q*diag(d)*Q';
x=randn(n,1);
b=A*x;
tic, y=inv(A)*b; toc
err=norm(y-x)
res=norm(A*y-b)
tic, y=A\b; toc
err=norm(y-x)
res=norm(A*y-b)
```

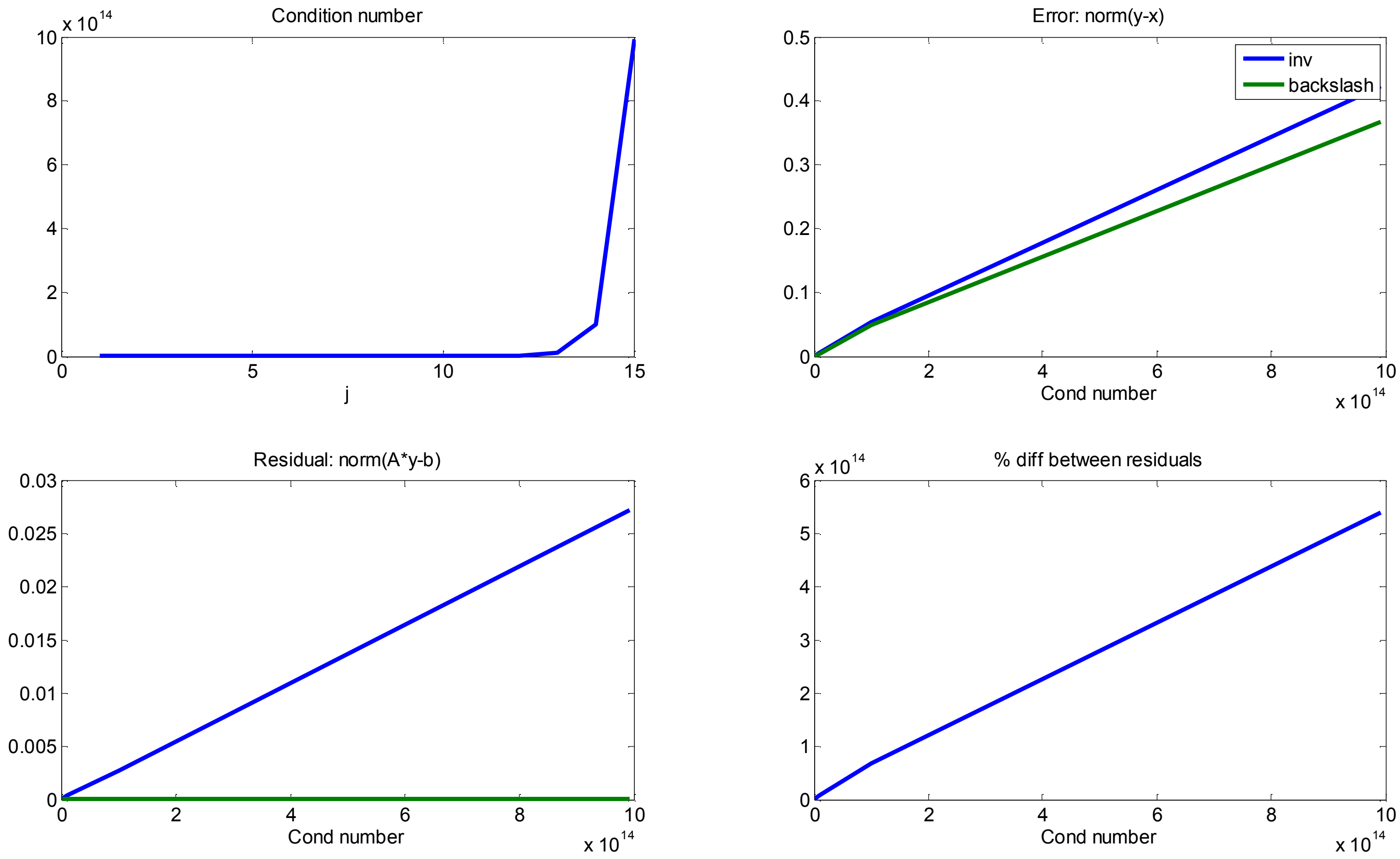
```
Elapsed time is 0.106780 seconds.
err = 9.1007e-006
res = 6.9634e-007
Elapsed time is 0.056587 seconds.
err = 8.3066e-006
res = 6.0796e-015
```

```
n=1000;
Q=orth(randn(n));
x=randn(n,1);

h=15;
err=zeros(h,2);
res=zeros(h,2);
condn=zeros(h,1);

for j=1:h

d=logspace(0,-j,n);
A=Q*diag(d)*Q';
b=A*x;

condn(j)=cond(A);

y1=inv(A)*b;
y2=A\b;

err(j,1)=norm(y1-x);
res(j,1)=norm(A*y1-b);
err(j,2)=norm(y2-x);
res(j,2)=norm(A*y2-b);

end
```

```
subplot(2,2,1), plot(1:h,condn,'LineWidth',3)
title('Condition number')
xlabel('j')
subplot(2,2,2), plot(1:h,err,'LineWidth',3)
title('Error: norm(y-x)')
xlabel('j')
legend('inv','backslash')
subplot(2,2,3), plot(1:h,res,'LineWidth',3)
title('Residual: norm(A*y-b)')
xlabel('j')
subplot(2,2,4), plot(1:h,100*(res(:,1)./res(:,2)-1),'LineWidth',3)
title('% diff between residuals')
xlabel('j')

pause

subplot(2,2,1), plot(1:h,condn,'LineWidth',3)
title('Condition number')
xlabel('j')
subplot(2,2,2), plot(condn,err,'LineWidth',3)
title('Error: norm(y-x)')
xlabel('Cond number')
legend('inv','backslash')
subplot(2,2,3), plot(condn,res,'LineWidth',3)
title('Residual: norm(A*y-b)')
xlabel('Cond number')
subplot(2,2,4), plot(condn,100*(res(:,1)./res(:,2)-1),'LineWidth',3)
title('% diff between residuals')
xlabel('Cond number')
```



Condition number
x 10^14
10
8
6
4
2
0
2
4
6
8
10
12
14
j
Error: norm(y-x)
0.5
0.4
0.3
0.2
0.1
0
inv
backslash
Residual: norm(A*y-b)
0.03
0.025
0.02
0.015
0.01
0.005
0
% diff between residuals
x 10^14
6
5
4
3
2
1
0


Condition number
x 10^14
j
Error: norm(y-x)
inv
backslash
Cond number
x 10^14
Residual: norm(A*y-b)
Cond number
x 10^14
% diff between residuals
x 10^14
Cond number
x 10^14