Direct Methods for linear systems $Ax = b$

basic point: easy to solve triangular systems

$$\begin{bmatrix} \ddots & & & \\ & \times & \times & \times \\ & 0 & \times & \times \\ & 0 & 0 & \times \end{bmatrix}$$

etc.

$a_{n-1,n-1}x_{n-1} = b_{n-1} - a_{n-1,n}x_n$ $\nwarrow$

$\leftarrow$ solve $a_{n,n}x_n = b_n$ then $\nwarrow$

back substitution: takes $\sim n^2$ operations. Need $a_{ii} \neq 0$.

Similar lower triangular (1^{st} equation, then 2^{nd} etc):

forward substitution.

So could solve $Ax = b$ by

$A = QR$ and $\begin{cases} Qy = b & \Rightarrow y = Q^T b \\ Rx = y & \text{back subs. as} R \text{ upper triangular} \end{cases}$

But $\frac{1}{2}$ the number of operations (and other advantages e.g. for sparse) to perform LU factorisation: based on Gauss elimination (successively create zeros below diagonal by following algorithm)

Gauss Elimination:

for columns $j = 1, \ldots, n-1$
    for rows $i = j+1, \ldots, n$
        calculate multiplier $l_{ij} = (a_{ij}/a_{jj})$,   (a_{jj} is the pivot)
        row $i \longleftarrow$ row $i - l_{ij} *$ row j $(\star)$
    end i
end j

$(\star)$ for $k = j+1, \ldots, n$
        $a_{ik} \longleftarrow a_{ik} - l_{ij} a_{jk}$
    end k
    $b_i \longleftarrow b_i - l_{ij} b_j$

reduces to upper triangular matrix U without changing solution in $\sim \frac{2}{3} n^3$ operations.

Back substitution $\Rightarrow$ solution

If store multiplier l_{ij} used to zero a_{ij} as i, j entry of a unit lower triangular matrix L then

$$A = LU \quad \text{with} \quad \begin{cases} Ly = b & \text{forward subs.} \\ Ux = y & \text{back subs.} \end{cases}$$

solves $Ax = b$.

Note: For many b's need only 1 LU factorization.

Recall $a_{ii} \neq 0$ necessary for Gauss Elimination so fails on e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ which is non-singular.

Pivoting:

Row interchanges: often expressed as $PA = LU$, P permutation.

Partial pivoting: when zeroing subdiagonal of p^{th} column

find max $|a_{ip}| = |m|$, $i = p, p+1, \ldots, n$;

m becomes pivot

swap row p with row which gives this max.

Fails if and only if A singular as
$a_{pp} = 0, m = 0 \Rightarrow$ det $A = 0$

Special forms

- A Symmetric positive definite: $A = LL^T$, L lower triangular, Cholesky factorisation.
- A Symmetric Indefinite: $A = LDL^T$, L lower triangular, D block diagonal, 1×1 and 2×2 blocks: Bunch - Parlett, Bunch - Kaufmann factorizations.
- A Banded: eliminate only in band, $\sim \frac{1}{3}nb^2$ operations for LU
  (NB pivoting generally destroys bandedness)
- A Sparse: good software e.g. HSL or $\backslash$ for sparse in matlab.

Ill-conditioning

Proposition: If $\boldsymbol{Ax} = \boldsymbol{b} \quad (1)$ and $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{b} + \boldsymbol{\delta b} \quad (2)$ then

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \leq \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|\frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}$$

Proof: $\boldsymbol{A}^{-1}((2) - (1)) \Rightarrow \boldsymbol{\delta x} = \boldsymbol{A}^{-1}\boldsymbol{\delta b}$

$$\begin{aligned} \text{so } \|\boldsymbol{\delta x}\| = \|\boldsymbol{A}^{-1}\boldsymbol{\delta b}\| &\leq \|\boldsymbol{A}^{-1}\|\|\boldsymbol{\delta b}\| \\ \text{also } \|\boldsymbol{b}\| = \|\boldsymbol{Ax}\| &\leq \|\boldsymbol{A}\|\|\boldsymbol{x}\| \\ \text{or } \frac{1}{\|\boldsymbol{x}\|} &\leq \frac{\|\boldsymbol{A}\|}{\|\boldsymbol{b}\|} \\ \text{so } \frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} &\leq \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|\frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|} \end{aligned}$$

$\uparrow$ relative change in solution — $\uparrow$ condition number — $\uparrow$ relative perturbation of rhs

Also if $\boldsymbol{A}$ is perturbated to $\boldsymbol{A} + \boldsymbol{\delta A}$ then

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x} + \boldsymbol{\delta x}\|} \leq \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|\frac{\|\boldsymbol{\delta A}\|}{\|\boldsymbol{A}\|}$$

(Exercise: Show this)
These results identify $\boldsymbol{\kappa} = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$ (the ‘condition number’ for solution of linear systems) as a measure of ill-conditioning.

Usually necessary if large $\boldsymbol{\kappa}$ to reformulate problem because:

Gauss elimination finds $\tilde{\boldsymbol{x}}$ such that $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}$ is small (not exactly $\boldsymbol{x}$ s.t. $\boldsymbol{Ax} = \boldsymbol{b}$) on a computer.

For many $\boldsymbol{A}$, $\boldsymbol{r}$ small $\Rightarrow \boldsymbol{e} = \boldsymbol{x} - \tilde{\boldsymbol{x}}$ is small but not when $\boldsymbol{\kappa}$ is large as indicated by the above results.

Example: Interpolation: Given N and data $f(x_i)$ at distinct points $x_i, i = 0, 1, \ldots, N$, find polynomial $p(x) = \sum_{k=0}^{n} a_k x^k \quad \in \Pi_n$ such that

$$p(x_i) = f(x_i).$$

This can be written as: solve

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}$$

For $x_k = k + 1$,

		expected accuracy
$n = 4$	$\kappa = 2 \cdot 6 \times 10^4$	12 decimal places
$n = 8$	$\kappa = 4 \cdot 2 \times 10^{10}$	6 decimal places
$n = 12$	$\kappa = 4 \cdot 2 \times 10^{17}$	0 decimal places
$n = 16$	$\kappa = 1 \cdot 9 \times 10^{25}$	no hope of accurate solution

but can reformulate the interpolation problem in many ways
e.g. use a better basis for Π_N than $\{1, x, x^2, \ldots, x^N\}$.
In fact for this problem there are reliable and faster ($O(N^2)$) methods (GVL p183 Vandermonde)

Iterative solution methods for $\boldsymbol{Ax = b}$

idea: split $\boldsymbol{A = M - N}$, so easy to solve systems with $\boldsymbol{M}$, then iterate:

Guess $\boldsymbol{x^{(0)}}$

solve $\boldsymbol{Mx^{(k)} = Nx^{(k-1)} + b}$ for $\boldsymbol{k = 1, 2, \ldots}$

basic point: if $\boldsymbol{\{x^{(k)}\}}$ converges (to $\boldsymbol{x}$, say) then

$$\boldsymbol{Mx = Nx + b,} \quad \text{ie. } \boldsymbol{Ax = b}$$

ie. it converges to the solution.