→ The advantage of the method is that, at worst case scenario, convergence is exact for $k=n$. We now prove this.

(6) Krylov subspaces and algorithm convergence

Definition: A Krylov set is defined as the set of vectors defined from an initial vector b as $\{b, Ab, A^2 b, A^3 b, \dots\}$

Successive Krylov subspaces are spaces spanned by successively larger groups of these vectors

$\langle b \rangle$ ← space spanned by b

$\langle b, Ab \rangle$ " b and Ab

$\langle b, Ab, A^2 b \rangle$ "

$\vdots$

Claims: ① The vector x_k constructed belongs to the Krylov subspace $\langle b, Ab, \dots, A^{k-1} b \rangle$

② This subspace is equal to the subspace $\langle p_0, p_1, \dots p_{k-1} \rangle$

Proof: By induction

for $k=1$: $\underline{x_1} = \alpha_0 \underline{p_0} = \alpha_0 b$ ✓

Assume it's true for k: $x_k \in \langle p_0, \dots p_{k-1} \rangle = \langle b, Ab, \dots A^{k-1} b \rangle$

$\underline{x_{k+1}} = \underline{x_k} + \alpha_k \underline{p_k}$; so $x_{k+1} \in \langle p_0, \dots, p_k \rangle$

$\underline{p_k} = \underline{r_k} + \beta_k \underline{p_{k-1}} \Rightarrow \underline{r_k} \in \langle p_0, \dots p_k \rangle$

$= \underline{r}_{k-1} - \alpha_k A p_{k-1} + \beta p_{k-1}$

So $\underline{p_k} \in \langle p_0, \dots p_{k-1} \rangle \oplus \langle A p_{k-1} \rangle$

$\in \langle b, \dots A^{k-1} b \rangle \oplus A \langle b, \dots, A^{k-1} b \rangle$

$\in \langle b, \dots A^k b \rangle$

So

$x_{k+1} \in \langle b, \dots A^k b \rangle$ too

so $\langle p_0 \dots, p_k \rangle = \langle b, \dots, A^k b \rangle$.

<u>Claim</u> (3) The vector x_k minimizes $f(\underline{x})$ within the <u>whole subspace</u> $\langle p_0, \dots p_{k-1} \rangle = \langle b, \dots, A^{k-1} b \rangle$

See proof in Atkinson p 565.
or in Trefethen & Bau p 296.

So this construction progressively enlarges the dimension of the space containing the guess, making sure that at every step the new guess is the global minimum within the new subspace
→ this means that addition of new dimensions to the original subspace will not change the projection of the solution on the subspaces already studied.

As a very important corollary, the last note implies that the error between x_k and the real solution x_* necessarily <u>decreases</u> monotonically. More precisely, it can be shown that

$\| e_k \|_A \leq \| e_{k-1} \|_A$ where the norm

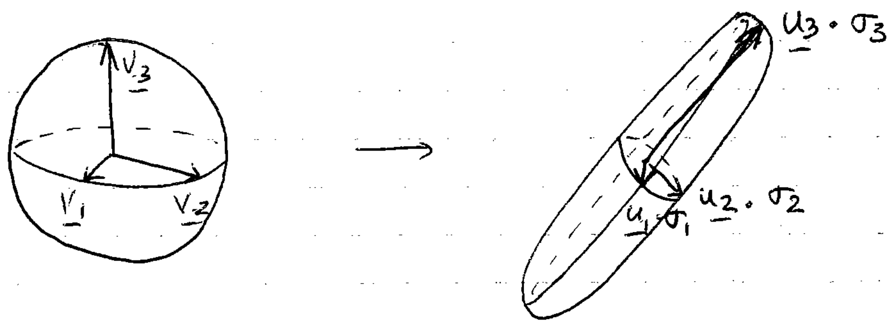
$$\| e_k \|_A = \sqrt{(x_* - x_k)^T A (x_* - x_k)}$$

(7) Condition number and stability issues

This section is quite general, but will also be used to discuss / introduce the idea of preconditioning.

1. Singular values

Idea: The image of the unit sphere under any $n \times n$ matrix is a hyperellipse



$\underline{v}_1, \underline{v}_2, \underline{v}_3$ = original unit basis

$\underline{u}_1, \underline{u}_2, \underline{u}_3$ = unit vectors along the semi-major axes of the hyperellipse

$\underline{v}_i$ are determined such that

$$\boxed{A v_i = \sigma_i u_i}$$

$\sigma_1, \sigma_2, \sigma_3$ = "stretching factors" = singular values of A

Note:

- If A is not a full-rank matrix, some of the singular values are 0.

  (if A is rank r, exactly r singular values are non-zero).

- By construction $\sigma_i \geq 0$

- By convention, the u_i and v_i are usually ordered such that
  $$\sigma_1 \geq \sigma_2 \geq \sigma_3 \dots \geq \sigma_n \geq 0$$

- This concept is entirely related to the Singular Value Decomposition (see Midterm project).

2 First notion of ill-conditioning

When applying a matrix A to a vector x, imagine first writing $\underline{x}$ on the basis of the vectors $\underline{v}_k$

$$\underline{x} = \sum \alpha_k \underline{v}_k$$

so $$A\underline{x} = \sum_k \alpha_k \sigma_k \underline{u}_k$$

Even if A is non singular, problems may arise if $\sigma_1 \gg \sigma_n$. Indeed, in that case the term $\alpha_n \sigma_n \underline{u}_n$ may be negligible in front of $\alpha_1 \sigma_1 \underline{u}_1$ and roundoff errors may affect it.

Geometrically speaking, it is particularly easy to visualize when $\sigma_n \ll \sigma_{n-1} \sim \sigma_{n-2} \cdots \sigma_1$

(the hyperellipse is very flat in one direction)

or $\sigma_1 \gg \sigma_2 \sim \sigma_3 \cdots \sigma_n$

(the hyperellipse is very elongated in one direction)

$\Rightarrow$ in both cases (and any intermediate case

$\sigma_1 \sim \sigma_2 \sim \cdots \sigma_k \gg \sigma_{k+1} \sim \cdots \sigma_n$)

information is lost if roundoff errors occur.

$\rightarrow$ we expect the sensitivity of a problem to small roundoff errors to be particularly dependent on the value of

$$\kappa = \frac{\sigma_1}{\sigma_n}$$

We now formalize this idea in more mathematical terms.

3. Necessary mathematical tool: the norm of a matrix

We saw that the norm of a vector can be defined in many ways, provided it satisfies

- $\|x\| \geq 0 \quad \forall \underline{x}$ and $\|x\| = 0 \iff \underline{x} = \underline{0}$
- $\|x+y\| \leq \|x\| + \|y\|$
- $\|\alpha x\| = |\alpha| \|x\|$.

A particularly useful norm is the Euclidean norm:

$$\|x\|_2 = \sqrt{x^T x} = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}$$

(see handout for other norms)

We now define the norm of a matrix as the maximum factor by which a matrix A can stretch a vector $\underline{x}$

$$\|A\| = \sup_{\underline{x}} \frac{\|Ax\|}{\|x\|} = \sup_{\substack{\underline{x} \\ \|x\|=1}} \|Ax\|.$$

(note that this holds even when A is not square)

Now by using the Euclidean norm and writing $x = \sum_k \alpha_k v_k$

it is easy to show that $\boxed{\|A\|_2 = \sigma_1}$

In addition, one can also show that

$$\|A^{-1}\|_2 = \frac{1}{\sigma_n}$$

4. Mathematical definition of conditioning

① Idea: Given a mathematical function

$$f: \mathbb{R}^m \to \mathbb{R}^n$$
$$\underline{x} \to f(\underline{x})$$

how do perturbations in the input $\underline{x}$ affect the output $f(\underline{x})$?

- If small "errors" in $\underline{x}$ result only in equivalently small "errors" in $f(\underline{x})$ then the problem is well-conditionned.
- If small "errors" in $\underline{x}$ result in much larger "errors" in $f(\underline{x})$ then the problem is ill-conditionned

let's define the relative condition number as

$$\alpha(\underline{x}) = \lim_{\delta \to 0} \sup_{\|\delta x\| \leq \delta} \frac{\|\delta f\| / \|f\|}{\|\delta x\| / \|x\|} \quad \leftarrow \frac{\text{relative change in } f}{\text{rel. change in } x}$$

where $\delta f = f(x + \delta x) - f(x)$.

$\Rightarrow$ $\alpha(\underline{x})$ large (for some $\underline{x}$) means that (for that $\underline{x}$), a small change $\underline{\delta x}$ results in a large change $\|\delta f\| / \|f\|$

If $f(x)$ is a differentiable function, then

$$\delta f = J(x) \delta x$$

↑ Jacobian matrix.

So $$\sup_{\|\delta x\| \leq \delta} \frac{\|\delta f\|}{\|\delta x\|} = \sup \frac{\|J(x)\delta x\|}{\|\delta x\|} = \|J(x)\|$$ by definition

and $$\alpha(x) = \|J(x)\| \cdot \frac{\|x\|}{\|f(x)\|}$$

(2) Conditionning of the problem : what is the error on Ax when A is fixed ?

Ax is a linear function so $J \equiv A$

$$\Rightarrow \quad \kappa(x) = \|J(x)\| \cdot \frac{\|x\|}{\|Ax\|}$$

$$= \|A\| \cdot \frac{\|x\|}{\|Ax\|}$$

If we use the 2-norm, $\|x\|_2 = \|A^{-1}Ax\|_2$
(by Green's inequality) $\leq \|A^{-1}\| \|Ax\|_2$

so $\kappa_2(x) \leq \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_m}$

and for some $\underline{x}$ the upper bound is actually achieved.

→ this confirms our expectations that $\frac{\sigma_1}{\sigma_m}$ is a good descriptor of the conditionning of a problem.

The quantity $\|A\| \|A^{-1}\| = \kappa(A)$ is called the condition number of A.
If A is singular, by convention $\kappa(A) = \infty$.

(3) Conditionning of a system of equations w.r.t. perturbations in A.

let's now suppose we are designing an algorithm working on A which progressively introduces small roundoff errors. How is $\underline{x} = A^{-1} b$ affected?

→ Similarly, it is possible to show that the condition number is $\kappa(A) = \|A\| \|A^{-1}\|$.

What this implies in practise is that an algorithm working on A introducing relative error $\frac{\|\delta A\|}{\|A\|}$ no larger than $O(\epsilon_{machine})$ in the coefficients result in an error on the outcome

$$\frac{\|\delta x\|}{\|x\|} = O\left(\kappa(A)\, \epsilon_{machine}\right).$$

→ if $\kappa(A)$ is very large, many significant digits in the solution are lost

Note that this assumes roundoff errors themselves are only $O(\epsilon_{machine})$. Some algorithms (cf G.E without pivoting) introduce roundoff errors $\gg \epsilon_{machine}$!

5. In practise, how to evaluate $\kappa(A)$?

- Calculating A^{-1}, then $\|A^{-1}\|$ is too CPU-expensive

- See LAPACK routines **CON (specialized) or DGESVX.f (driver routine) for an estimate of the reciprocal of the condition number (returned in RCOND)

- Note that the Lapack routines return either the reciprocal of $\kappa_1(A) = \|A\|_1 \|A^{-1}\|_1$

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

where $\|A\|_1$ is the norm of A based on $\|x\|_1$

$$\left(\|x\|_1 = \sum_i |x_i|\right)$$

and

$\|A\|_\infty$ is the norm of A based on $\|x\|_\infty$

$$\left(\|x\|_\infty = \max_i |a_i|\right)$$

Example

$$A = \begin{pmatrix} 1+10^{-k} & 1 \\ 1 & 1 \end{pmatrix}$$

$$\|A\|_1 = \max_j \sum_i |a_{ij}| = \max\left(2+10^{-k}, 2\right) = 2+10^{-k}$$

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| = 2+10^{-k} \quad \text{(same)}$$

look for A^{-1} :

$$\left(\begin{array}{cc|cc} 1+10^{-k} & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array}\right)$$

$$\to \begin{pmatrix} 1 & \frac{1}{1+10^{-k}} & \frac{1}{1+10^{-k}} & 0 \\ 0 & \frac{10^{-k}}{1+10^{-k}} & -\frac{1}{1+10^{-k}} & 1 \end{pmatrix}$$

$$\to \begin{pmatrix} 1 & 0 & \frac{1+10^{k}}{1+10^{-k}} & -10^{k} \\ 0 & 1 & -\frac{1}{10^{-k}} & \frac{1+10^{-k}}{10^{-k}} \end{pmatrix}$$

So $$A^{-1} = \begin{pmatrix} \frac{1+10^{k}}{1+10^{-k}} & -10^{k} \\ -10^{k} & 1+10^{k} \end{pmatrix}$$

so $\|A^{-1}\|_1 \cong 2+10^{k}$

$\|A^{-1}\|_\infty \simeq 2+10^{k}$ $\Rightarrow$ $\boxed{\mathcal{K}(A) \simeq 2\,10^{k} \text{ for large } k}$

Note that if $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ then

$x = A^{-1}(b) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

if $b = \begin{pmatrix} 1+\epsilon \\ 1 \end{pmatrix}$ then $A^{-1}b \cong \begin{pmatrix} \epsilon 10^{k} \\ 1 \end{pmatrix}$

③ Convergence of Conjugate Gradient algorithm

In fact, it can be shown that

$$\|e_n\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n \|e_0\|_A$$

(see Trefethen & Bau). p 300

where $\kappa = \dfrac{\lambda_{max}}{\lambda_{min}} = \dfrac{\text{largest evalue}}{\text{smallest evalue}}$.

(note that by assumption A is positive definite so all the evalues are > 0)

- For quick convergence, we would hope that κ is "not too large" (ideally we want $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \ll 1$)

in practise typical $\kappa \cong 1 \to 10$ is great!

- For ill-conditionned matrices, the gain of iterative methods vs standard direct methods is negligible

$\Rightarrow$ IDEA OF PRE-CONDITIONNING

(9) Preconditionning ideas

- The convergence rate of an iterative method depends on the condition number $\alpha = \frac{\lambda_{max}}{\lambda_{min}}$ of A.

- For ill-conditionned matrices $\alpha \gg 1$ and the convergence can be v. slow.

- Idea: Instead of solving $Ax=b$, solve the equivalent system

$$KAx = Kb \quad \text{where}$$

K = a non singular matrix
KA = a better conditionned matrix.

- Note: Although the product KA is never actually formed in a preconditionned algorithm, it is vital that the product Kv or K^Tv or $K^{-1}v$ be fast for any vector v (a $< n^2$ process)

  ↳ K should be a simple or v. sparse matrix.

- Finding preconditionners that satisfy both requirements is ongoing research

  – different types of matrices A respond better to different types of preconditionne

  – Preconditionning can be mathematicall based or physically base

Mathematical	Physical
Diagonal	Coarse grid / Multigrid
Incomplete Cholesky (Golub & V. Loan)	low-pass filters

See Appendix on preconditionning for actual implementation

(10) <u>Generalization to non-symmetric matrices</u>

The idea of minimizing $f(\underline{x}_k)$ for $\underline{x}_k \in S_k$
(the Krylov subspace

$$S_k = \langle b, Ab, \dots A^{k-1} b \rangle$$

for successively larger values of k can be
generalized to non-symmetric matrices

The methods are known as GMRES
(Generalized Minimum Residuals)

See Trefethen & Bau Lecture 35.