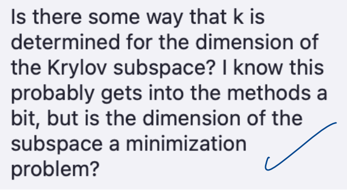
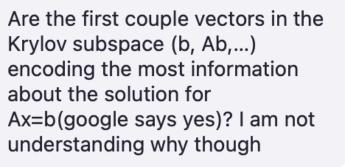
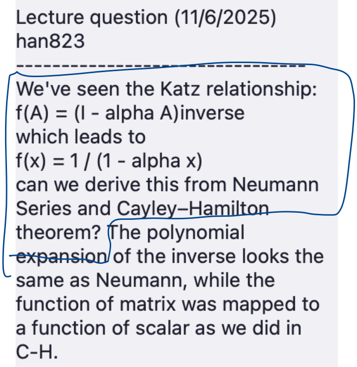
Lecture 20


Is there some way that k is determined for the dimension of the Krylov subspace? I know this probably gets into the methods a bit, but is the dimension of the subspace a minimization problem?

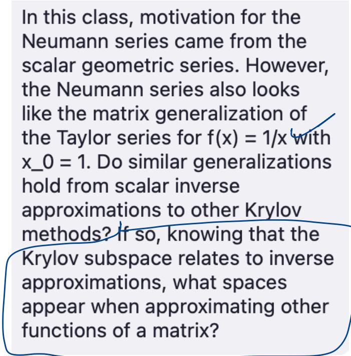

Are the first couple vectors in the Krylov subspace (b, Ab,...) encoding the most information about the solution for Ax=b(google says yes)? I am not understanding why though

Yes, BUT....
There is loads of info in others,

Often 10n steps.


Lecture question (11/6/2025)
han823

We've seen the Katz relationship:
f(A) = (I - alpha A)inverse
which leads to
f(x) = 1 / (1 - alpha x)
can we derive this from Neumann Series and Cayley–Hamilton theorem? The polynomial ~~expansion~~ of the inverse looks the same as Neumann, while the function of matrix was mapped to a function of scalar as we did in C-H.


In this class, motivation for the Neumann series came from the scalar geometric series. However, the Neumann series also looks like the matrix generalization of the Taylor series for f(x) = 1/x with x_0 = 1. Do similar generalizations hold from scalar inverse approximations to other Krylov methods? If so, knowing that the Krylov subspace relates to inverse approximations, what spaces appear when approximating other functions of a matrix?

f(x)

The lecture notes say that Krylov methods generally outperform methods like Richardson and Steepest Descent. Are there situations in practice where the non-Krylov methods are still preferred?

Yes! If A is well-conditioned, then ~~Jacobi / etc. are~~ Steepest descent. are fine.

Tchebyshev's method is a hybrid.

What's a good indicator that we want to switch subspaces?

$b, Ab, A^2b, \ldots$

Numerically, what is the general way to obtain every eigenvalues of A? (A: any kind of matrix)

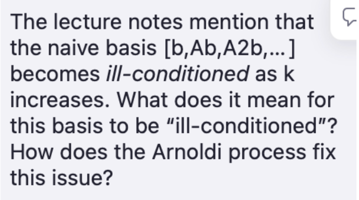
Use QR only!

```
$X^{(1)} = I$
for $i = 1$ to ___
    $y = AX^{(i)}$
    $X^{(i+1)} = QR(y).Q$
$\rightarrow$ X will converge to eigenvectors
```

Subspace iteration

```
$A^{(0)} = A$
for $i = 1$ to ___
    $Q, R = qr(A^{(i-1)})$
    $A^{(i)} = RQ$.
```

$$A^{(\infty)} \rightarrow \text{diag}(\lambda_i\text{'s})$$



The lecture notes mention that the naive basis [b,Ab,A2b,...] becomes *ill-conditioned* as k increases. What does it mean for this basis to be "ill-conditioned"? How does the Arnoldi process fix this issue?

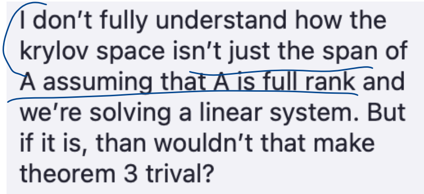
$b, Ab, A^2 b, \ldots$

if X is a matrix, X is ill-conditioned if $\kappa(X)$ is large!

$\kappa(X) = 10^{13}$ that's big!

$$(X) \geq \frac{1}{\sqrt{u}} \approx 10^8$$

u from FP system



I don't fully understand how the krylov space isn't just the span of A assuming that A is full rank and we're solving a linear system. But if it is, than wouldn't that make theorem 3 trival?

$$\{b, Ab, A^2 b, A^3 b, A^4 b, \ldots\}$$

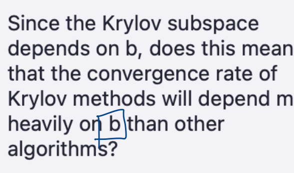
spans the set of all

$$\{ y = Ax \;\boxed{\text{for } x \in \mathbb{R}^n} \}$$

$$A = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}, \quad b = \begin{bmatrix} d \\ 0 \end{bmatrix}$$

Then $\boxed{\text{Span}(A)} \neq K_n(A, b)$

$$A^{-1} b = \begin{bmatrix} B^{-1} d \\ 0 \end{bmatrix}$$



Since the Krylov subspace depends on b, does this mean that the convergence rate of Krylov methods will depend m heavily on b than other algorithms?

No, you can do b specific analysis for all algs.

ore

Lecture question (11/6/2025)
chen3203

Could you give some examples of well-known algorithms that are not backward stable, and explain whether their instability comes from the algorithm itself or from the fact that the underlying problem is ill-conditioned?

✓

I've heard the word energy a lot, x^TAx in A norm is called energy, and in ML many people talk about energy base models. Are the use energy for similar things or completely different things?

Energy ⟺ Some non-neg function

(basin - shaped)

x^TAx is an Energy if A is spd. Seen

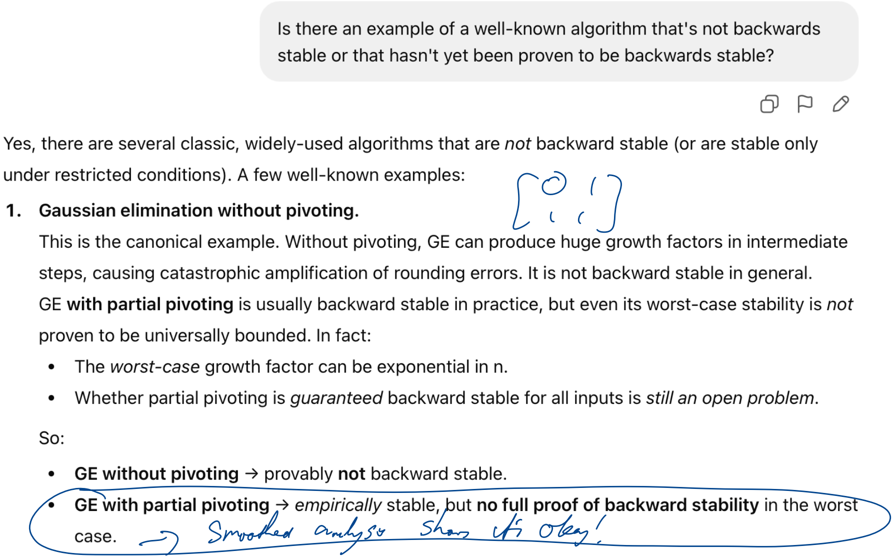
$x^TAx \geq 0$ if A is spsd. $A \succeq 0$.

In the generalized eigenvalue problem, what changes if B is not spd or if it is singular?

It's Complicated!

You get infinite λ evals.

det



Is there an example of a well-known algorithm that's not backwards stable or that hasn't yet been proven to be backwards stable?

Yes, there are several classic, widely-used algorithms that are *not* backward stable (or are stable only under restricted conditions). A few well-known examples:

$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$

1. **Gaussian elimination without pivoting.**
   This is the canonical example. Without pivoting, GE can produce huge growth factors in intermediate steps, causing catastrophic amplification of rounding errors. It is not backward stable in general.
   GE **with partial pivoting** is usually backward stable in practice, but even its worst-case stability is *not* proven to be universally bounded. In fact:
   - The *worst-case* growth factor can be exponential in n.
   - Whether partial pivoting is *guaranteed* backward stable for all inputs is *still an open problem*.

   So:

   - **GE without pivoting** → provably **not** backward stable.
   - **GE with partial pivoting** → *empirically* stable, but **no full proof of backward stability** in the worst case. → Smoothed analysis shows it's okay!

. **Conjugate Gradient (CG) method** for solving SPD linear systems.

In exact arithmetic, CG converges in at most n steps and behaves well theoretically.

In finite precision arithmetic, CG loses conjugacy due to rounding, which can lead to:

- Slowdown or stagnation.
- The need for *re-orthogonalization or preconditioning* to regain useful behavior.

Result: CG is **not backward stable as a solver for Ax=b**.

The standard interpretation is weaker: it is *approximately stable* for solving the **normal equations** associated to A, which is a weaker guarantee and not always desirable.

Okay. If $U = \text{Lanczos}(A, n)$ were BW stable, then

Then U must be orthogonal.

Chris Paige shows that Lanczos

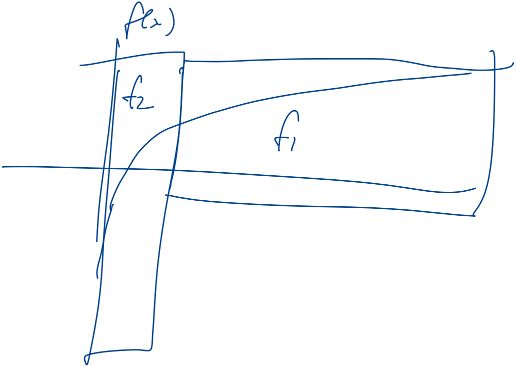
$$U = \text{Lanczos}\left(\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}, n\right)$$

$$\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$$

Got it: you mean *elementary transcendental functions* where we know that the usual approximations are not backward stable for all arguments.

A clean example is **log(x)** for x near 1.
If you compute log(1+x) by just calling log(1+x), you get catastrophic cancellation when x is tiny. That computation is not backward stable in that region, because the nearest representable output does not correspond to any nearby exact input. The fix is the specialized **log1p(x)** routine designed specifically to restore backward stability.

Same style story for **expm1(x)** computing exp(x) - 1 when x is small. Using exp(x) - 1 directly loses the significant digits; expm1(x) is the stable form.



Matrix poly + Rational functions

+ C-H theorem +

Complex analysis.

$$p(A) = \sum_{i=0}^{d} c_i A^i$$ degree d matrix polynomial.

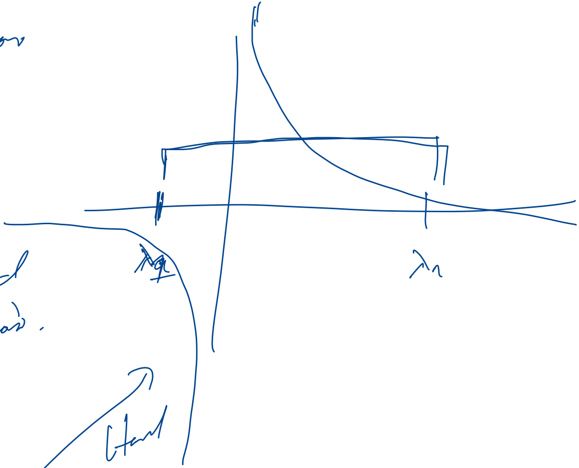
p(A) is also a matrix function.

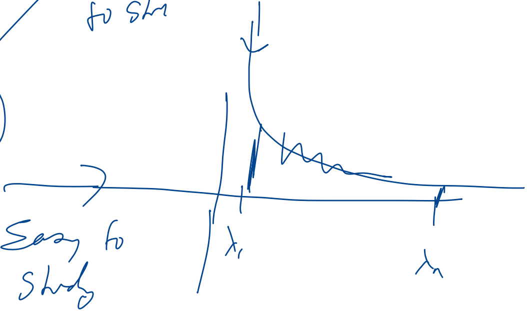
What's the best p(x) to approximate $f(x) = \frac{1}{x}$?

(Over the matrix spectrum.)

$f(x) = \frac{1}{x}$

$(x) = \frac{x}{\alpha x + 5x^2}$





$$\boxed{\begin{aligned} g(x) &= X\cdot(\boxed{\alpha X+5X^2})^{-1} \\ &= \underline{(\alpha X+5X^2)^{-1}}\, X \end{aligned}}$$

If $p(X)$ and $q(X)$ are polys in X, then

$$p(X)\,q(X) = q(X)\,p(X)$$

because polynom in Matrix Commute!

vid F. Gleich - 2025 CS 515 - Purdue University - Computer Science - David F. Gleich - 2025 CS 515 - Purdue University - Computer Science - David F. Gleich - 2025 CS 515 - Purdue