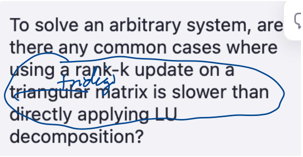
Lecture 20



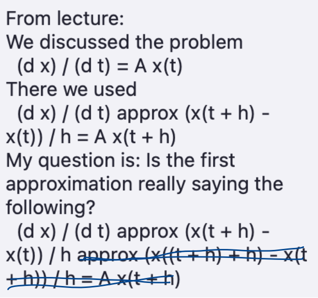
To solve an arbitrary system, are there any common cases where using a rank-k update on a ~~triangular~~ tridiag matrix is slower than directly applying LU decomposition?

If k is big Enough—

k > n/2? n/10?

Tridiag... rank k

You can Exploit more Structure here!



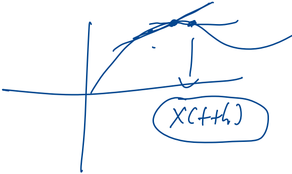
From lecture:
We discussed the problem
(d x) / (d t) = A x(t)
There we used
(d x) / (d t) approx (x(t + h) - x(t)) / h = A x(t + h)
My question is: Is the first approximation really saying the following?
(d x) / (d t) approx (x(t + h) - x(t)) / h ~~approx (x((t + h) + h) - x(t + h)) / h = A x(t + h)~~

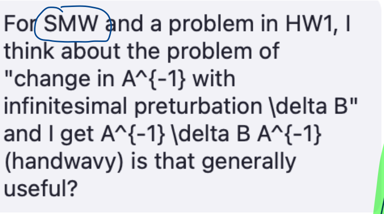
$$\boxed{\frac{dx}{dt} \approx \frac{x(t+h)-x(t}{h}}$$

$$= A\,x(t+$$

OR

$$= A\,x(t)$$





For SMW and a problem in HW1, I think about the problem of "change in A^{-1} with infinitesimal preturbation \delta B" and I get A^{-1} \delta B A^{-1} (handwavy) is that generally useful?

$A^{-1} + \delta B$ or

$(A + \delta B)^{-1}$

you want

$\partial A^{-1} / \partial A$

h) c/using t+h, t to approx

c/using t, t+h to approx.

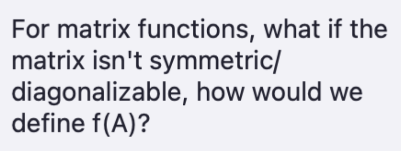
$O(n^2 \text{ by } n^2)$

object...

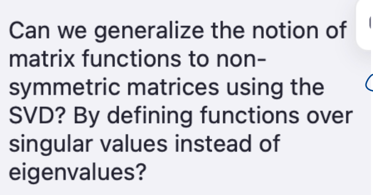
[illegible]

$\underline{x_i(A)} = A^{-1} \underline{e_i} \Rightarrow A x_i(A) = e_i$

∂ ✓



For matrix functions, what if the matrix isn't symmetric/ diagonalizable, how would we define f(A)?



Can we generalize the notion of matrix functions to non-symmetric matrices using the SVD? By defining functions over singular values instead of eigenvalues?

No ☹

1. If $f(x)$ has a Convergent power series $f(x) = \sum \alpha_k x^k$

or —

Then $f(A) = \sum \alpha_k A$

2. $A = XJX^{-1}$ is Jordan

$f(A) = X f(J) X^{-1}$

Applicatio
eigenvalu
up?

ns where generalized
e decomposition show ✓

When perturbing A with a rank-1 update
(A+uv')y=c,
how do we know what c is?

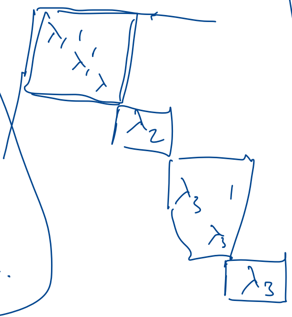
It's given in the problem.

Why the rank-1 update drastically reduces computation compared to re-factoring the entire matrix? Is there a scenario where applying a rank-1 update is especially beneficial?

If k is close to n, do we get any computational advantages for rank-k update or is it more or less the same as just solving the system again?

$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ vs. $A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \to$ No degenerate λ-vals

What is $f(J)$?



K

form.

$\boxed{f(J_i)}$ involves derivatives of f if they exist.

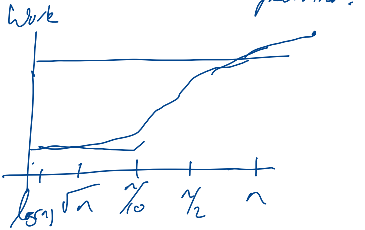
Something like

Probably wrong, but close!

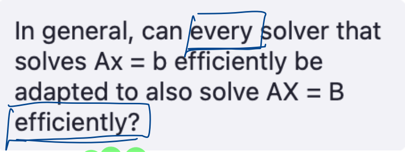
$$\begin{bmatrix} f(\lambda_1) & f'(\lambda_1) & f''(\lambda_1)/2 \\ & f(\lambda_1) & f'(\lambda_1) \\ & & f(\lambda_1) \end{bmatrix}$$

→ $O(n^2)$ work to update vs.

$O(n^3)$ to Refactorize.



$A = I$, $L = I$, $U = I$.

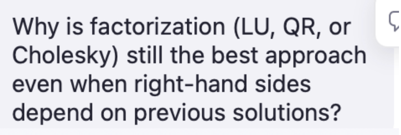

In general, can every solver that solves $Ax = b$ efficiently be adapted to also solve $AX = B$ efficiently?

Yes!

↳ SIMD work
(i.e. a group of cols at a time.)

↳ Do it in parallel

↳


Why is factorization (LU, QR, or Cholesky) still the best approach even when right-hand sides depend on previous solutions?

↳ $O(n^2)$ to solve after factorization vs. $O(n^3)$ to Recompute.

But, we don't need to Recompute to change RHS.

Apps of gen. λ-vecs: ✓

- Spectral Cluster

  $$A\underline{x} = \lambda D x.$$

- Finite Elem

  $$Ax = \lambda M x$$

  M - mass matrix

- Stats - See Elements of stats. learning.

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