$$\operatorname{sign}(z) = \begin{cases} +1 & z \in \text{right half-plane} \\ -1 & z \in \text{left h. p.} \end{cases}$$

M with no eigenvalues on the imaginary axis

$\operatorname{sign}(M)$

$$\begin{cases} X_0 = M \\ X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-1}\right) \end{cases}$$

If z is large,

$$\frac{1}{2}\left(z + z^{-1}\right) \approx \frac{1}{2} z$$

If $z \ll 1$, $\quad \frac{1}{2}\left(z + z^{-1}\right) \approx z^{-1}$

so convergence is slow when far away from ± 1

Idea to fix it: $\quad \operatorname{sign}(M) = \operatorname{sign}(\alpha M)$ for each $\alpha > 0$.

We would like to take the eigenvalues as close to 1 as possible by choosing α appropriately

If $\alpha\lambda_1, \dots \alpha\lambda_n$ are the eigenvalues of αM, then we want

$$\sqrt[n]{\left|(\alpha\lambda_1)(\alpha\lambda_2) \cdots (\alpha\lambda_n)\right|} = 1$$

$$\Leftrightarrow \quad \alpha \sqrt[n]{|\lambda_1 \lambda_2 \cdots \lambda_n|} = 1$$

$$\Leftrightarrow \quad \alpha = \frac{1}{|\det(M)|^{1/n}} \qquad M \in \mathbb{C}^{n \times n}$$

This is called <u>determinantal scaling</u>

It comes for free, since the same decomposition (e.g. LU) can be used to compute X^{-1} and $\det(X)$

Alternatives: use power methods to approximate $\lambda_1(X_k)$, $\lambda_n(X_k)$ and balance them.

Remarks on stability:

The sign function is sensitive to perturbation:

$$M = \begin{bmatrix} A & C \\ 0 & B \end{bmatrix} \qquad \operatorname{sign}(M) = \left[\begin{array}{c|c} -1 & Z \\ \hline 0 & 1 \end{array}\right]$$

Z is obtained via a Sylvester equation, if $\operatorname{sep}(A,B) \ll 1$, Z is large.

If $\tilde{M} = M+E$ $\quad \|E\| = \varepsilon$ $\qquad \|\tilde{Z} - Z\| \leq \frac{\varepsilon}{\operatorname{sep}(A,B)^2}$

However, if we are only interested in computing invariant subspaces,

$$\operatorname{Ker}\left(\operatorname{sign}(M) + I\right)$$

Perturbing M to $M+E$ changes the invariant subspace only by $\frac{\varepsilon}{\operatorname{sep}(A,B)}$

In practice, this method produces invariant subspaces as accurate as the reordered Schur method.

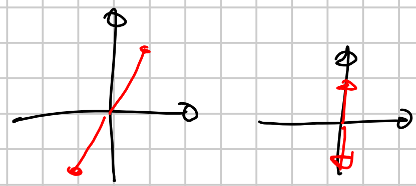
Note that X_0 ill-conditioned $\Leftrightarrow$ problem is ill-conditioned.

Matrix square root

Principal square root:

For $z \in \mathbb{C}$, if $z \notin \{\text{negative reals}\}$,

$f(z) = z^{1/2}$ is the square root in the RHP (right half-plane)



$f(0) = 0^{1/2} = 0$

$f(\text{negative real})$ undefined

(there is no continuous way to do it)

Note that
$$(z^2)^{1/2} = z \cdot \operatorname{sign}(z)$$

so $$\operatorname{sign}(M) = (M^2)^{1/2} \cdot M^{-1}$$

Also, one can prove that
$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}$$

Schur-Parlett for the square root:

<u>Idea</u>:
- $M = QUQ^*$
- compute $S = U^{1/2}$ (triangular factor)
- $M^{1/2} = QU^{1/2}Q^*$

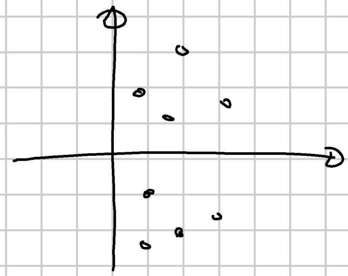
To compute $f(S)$, the standard Schur-Parlett method involves recursions of the form
$$S_{ij} = \frac{(\text{numerator})}{U_{jj} - U_{ii}}.$$
and has trouble if M has close eigenvalues.
For the matrix square root, we can avoid the problem: we compute S not from $SU = US$, but from $S^2 = U$

$$U_{ij} = (S^2)_{ij} = S_{ii}\boxed{S_{ij}} + S_{i,i+1}S_{i+1,j} + \dots + S_{i,j-1}S_{j-1,j} + \boxed{S_{ij}}S_{jj}$$

$$\Rightarrow S_{ij} = \frac{U_{ij} - \sum_{i<k<j} S_{ik}S_{kj}}{S_{ii}+S_{jj}}$$

The denominator does not involve $U_{jj}-U_{ii}$.

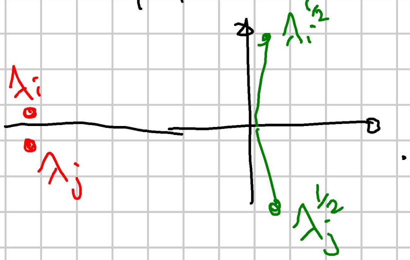
$$S_{ii}+S_{jj} = \lambda_i^{1/2}+\lambda_j^{1/2}$$



$\lambda_{ii}, \lambda_{jj} \in RHP \Rightarrow Re(\lambda_{ii}+\lambda_{jj})>0$

and the denominator cannot vanish.

Actually, the case when $S_{ii}+S_{jj}$ is small is only when $\|L_{sqrt,M}\|$ is large: either λ_i, λ_j are small, or



λ_i, λ_j close to the negative real line.

sqrtm in Matlab uses blockwise Schur-Parlett in a recursive way:

$$M = \begin{bmatrix} A & C \\ 0 & B \end{bmatrix}$$

with A, B square blocks with almost the same size,

then $$M^{1/2} = \begin{bmatrix} A^{1/2} & Z \\ 0 & B^{1/2} \end{bmatrix}$$

$A^{1/2}, B^{1/2}$ computed recursively

Z computed from

$$\begin{bmatrix} A^{1/2} & Z \\ 0 & B^{1/2} \end{bmatrix}\begin{bmatrix} A^{1/2} & Z \\ 0 & B^{1/2} \end{bmatrix} = \begin{bmatrix} A & C \\ 0 & B \end{bmatrix}$$

$$A^{1/2}Z + ZB^{1/2} = C \quad \leftarrow \text{Sylvester equation.}$$

Theorem: Let U be upper triangular, $\tilde{S}$ be the matrix computed with the Schur-Parlett variant in machine arithmetic.

$$\tilde{S}^2 = U + \Delta \qquad |\Delta_{ij}| = (|S|^2)_{ij} \cdot O(nu)$$

(u = machine precision)

Theorem: $\tilde{X}$ computed with the Schur-Parlett variant, for a general $M \in \mathbb{C}^{n\times n}$ (not nec. triangular) satisfies

$$\|\tilde{X}^2 - M\|_F \leq \|X\|_F^2 \cdot O(n^3 u)$$

This is not backward stability, which would be

$$\|\tilde{X}^2 - M\|_F \leq \|M\|_F \cdot O(n^3 u)$$

$$\Leftrightarrow \tilde{X}^2 = M + \Delta \qquad \frac{\|\Delta\|_F}{\|M\|_F} = O(n^3 u)$$

Instead of having Δ small w.r.t. $\|M\|_F$, we only know that it is small w.r.t. $\|\tilde{X}\|_F^2$

$$\|M\| = \|X^2\| \leq \|X\| \cdot \|X\|.$$

$$\tilde{X}^2 = M + \Delta \qquad \frac{\|\tilde{X} - X\|}{\|X\|_F} = O(u) \cdot \text{condition number}$$

$$\frac{\|\Delta\|}{\|M\|} = O(u)$$

$$S_{12} = \frac{U_{12}}{S_{22} + S_{11}} \qquad \tilde{S}_{12} = \frac{U_{12}}{\tilde{S}_{22} + \tilde{S}_{11}}(1+\delta_1)(1+\delta_2)$$

$$\tilde{S}_{11}\tilde{S}_{12} + \tilde{S}_{12}\tilde{S}_{22} = U_{12}(1+\delta_1)(1+\delta_2) = U_{12} + \boxed{U_{12}(\delta_1+\delta_2)}$$

$$\|U\tilde{x} - b\| = O(\text{mach. precision})$$

⚠ Wed May 7 → makeup lecture 9-11

Newton method for the square root:

Multivariate Newton method: $F(X) = X^2 - M$

$X_{k+1} = X_k - E$ $\qquad E = \mathrm{Jac}_{F,X_k}^{-1}[F(X_k)]$

Fréchet derivative, $\mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$

The Fréchet derivative of the map $F(X) = X^2 - M$ in X is the operator $E \mapsto XE + EX$

$$F(X+E) = (X+E)^2 - M = \underbrace{X^2 - M}_{F(X)} + \underbrace{\boxed{EX + XE}}_{L_X[E]} + \underbrace{E^2}_{o(\|E\|)}$$

$$L_{X_k}[E] = F(X) = X_k^2 - M$$

E solves the Sylvester equation $X_k E + E X_k = X_k^2 - M$

Algorithm:

1. Choose X_0
2. For $k = 0, 1, 2, 3, \ldots$
   - solve $\boxed{X_k E + E X_k = X_k^2 - M}$ to find E
   - set $X_{k+1} = X_k - E$

Problem: expensive, one Schur form at each step

However, we can do something better: suppose X_0 and M commute. Then, $E = (2X_0)^{-1}(X_0^2 - M)$ solves the equation:

E is a function of X_0, M and it commutes with X_0, M, and we have

$$X_0(2X_0)^{-1}(X_0^2-M)+(2X_0)^{-1}(X_0^2-M)X_0$$
$$=(2X_0)(2X_0)^{-1}(X_0^2-M)=X_0^2-M \qquad \checkmark$$

Now,
$$X_1=X_0-E=X_0-(2X_0)^{-1}(X_0^2-M)$$
is built with X_0, M, and one can see that it commutes with M: $\quad X_1M=MX_1$

<u>One can prove by induction:</u>

Let $M\in\mathbb{C}^{n\times n}$, X_0 s.t. $X_0M=MX_0$
Then, 1. $E=(2X_k)^{-1}(X_k^2-M)$ solves the Sylvester equation appearing in the Newton method
2. X_k commutes with M.

So we can simplify the algorithm:

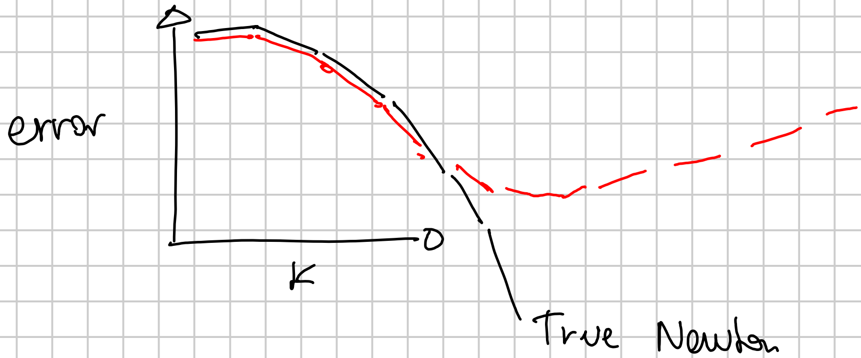
1. Choose X_0 that commutes with M
2. For $k=0,1,2,\ldots$
$$E=(2X_k)^{-1}(X_k^2-M)$$
$$X_{k+1}=X_k-E$$

Or equivalently:

$$X_{k+1}=X_k-E=(2X_k)^{-1}(2X_k^2-(X_k^2-M))=(2X_k)^{-1}(X_k^2+M)$$
$$=\frac{1}{2}(X_k+X_k^{-1}M)$$

<u>Modified Newton iteration:</u>
$$X_{k+1}=\frac{1}{2}(X_k+X_k^{-1}M)$$
with $X_0=\alpha I$ or $X_0=\alpha M$

<u>However:</u>


error
k
0
True Newton