II.1 Numerical Linear Algebra

- Gram-Schmidt
  - Standard way
  - Column pivoting (when really close to the same direction)
  - Krylov-Arnoldi (any matrix), Krylov-Lanczos (symmetric matrix)

$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}$$

$$\mathbf{A}_2 = \left\{ \begin{matrix} \mathbf{a}_2 - \left(\mathbf{a}_2^T \mathbf{q}_1\right)\mathbf{q}_1 \\ \vdots \\ \mathbf{a}_n - \left(\mathbf{a}_n^T \mathbf{q}_1\right)\mathbf{q}_1 \end{matrix} \right\} \xrightarrow{\text{pick biggest}} \mathbf{q}_2 = \frac{\mathbf{A}_2}{\|\mathbf{A}_2\|}$$

$$\mathbf{Ax} = \mathbf{b} \rightarrow \begin{cases} \mathbf{A}: \text{ large sparse} \\ \mathbf{b},\ \mathbf{Ab},\ \mathbf{A}(\mathbf{Ab}), \ldots,\ \mathbf{A}^{j-1}\mathbf{b} \ \begin{cases} \text{not good basis} \xrightarrow{\text{orthogonalize}} \text{Gram-Schmidt} \\ \text{maybe nearly dependent} \end{cases} \\ \text{combinations give Krylov space } K_j \\ \mathbf{x}_j: \text{ best/closest solution/vector in Krylov space} \end{cases}$$

Computing Eigenvalues and Singular Values

$\mathbf{A}_0 = \mathbf{Q}_0\mathbf{R}_0 \xrightarrow{?} \mathbf{R}_0\mathbf{Q}_0 = \mathbf{A}_1$: eigenvalues do not change!

$\mathbf{A}_1 = \mathbf{R}_0\mathbf{Q}_0 = \mathbf{R}_0\mathbf{A}_0\mathbf{R}_0^{-1}$: similar matrices $\rightarrow$ same eigenvalues

$$\mathbf{A}_0 = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \rightarrow \mathbf{A}_1 = \begin{bmatrix} & & \\ & & \\ \text{smaller} & & \end{bmatrix} \rightarrow \cdots \rightarrow \mathbf{A}_n = \begin{bmatrix} \text{close to } \lambda's & & \\ & & * \\ & \varepsilon_s & * \end{bmatrix}$$

$$\mathbf{A}_0 = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & 0 \end{bmatrix} = \mathbf{Q}_0\mathbf{R}_0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} 1 & \sin\theta\cos\theta \\ 0 & -\sin^2\theta \end{bmatrix} \rightarrow \mathbf{A}_1 = \mathbf{R}_0\mathbf{Q}_0 = \begin{bmatrix} \cos\theta\left(1+\sin^2\theta\right) & -\sin^3\theta \\ -\sin^3\theta & -\sin^2\theta\cos\theta \end{bmatrix}$$

$$\xrightarrow{\text{improve}} \text{better, faster idea: introduce shifts} \left(\mathbf{A}_0 \rightarrow \mathbf{A}_0 - s\mathbf{I}\right) \rightarrow \begin{cases} \text{shift in every eigenvalues} \\ \text{no change in eigenvectors} \end{cases}$$

$$\mathbf{A}_0 - s\mathbf{I} = \mathbf{R}_0\mathbf{Q}_0 = \mathbf{R}_0\mathbf{Q}_0 + s\mathbf{I} = \mathbf{A}_1$$

$$\mathbf{A}_1 = \mathbf{R}_0\mathbf{Q}_0 + s\mathbf{I} = \mathbf{R}_0\left(\mathbf{A}_0 - s\mathbf{I}\right)\mathbf{R}_0^{-1} + s\mathbf{I} = \mathbf{R}_0\mathbf{A}_0\mathbf{R}_0^{-1}$$

$$\xrightarrow{\text{improve}} \left\{ \begin{array}{l} \text{Hessenberg matrix } (\text{zeros stay zeros in QR}) \\ \text{upper triangular matrix + one extra diagonal} \\ \text{how to get? Arnoldi iteration } (\text{Krylov} \rightarrow \text{Gram-Schmidt}) \end{array} \right\} \mathbf{A}_0 = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}$$

$\text{eig}(\mathbf{A})$ in Matlab: (1) reduce $\mathbf{A}$ to Hessenberg (2) QR with shifts

$$\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix}\begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_k \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_k & \mathbf{q}_{k+1} \end{bmatrix}\begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \ddots & \vdots \\ & & h_{kk} \\ & & h_{k+1,k} \end{bmatrix}$$

$\mathbf{AQ}_k = \mathbf{Q}_{k+1}\mathbf{H}_{k+1,k} \rightarrow \mathbf{Q}_k{}^T\mathbf{AQ}_k = \mathbf{Q}_k{}^T\mathbf{Q}_{k+1}\mathbf{H}_{k+1,k} = \mathbf{H}_k \;\left(k = \text{size of } \mathbf{A}\right)$

$\mathbf{H}_k = \mathbf{Q}_k{}^T\mathbf{AQ}_k \rightarrow \mathbf{H} = \mathbf{Q}^{-1}\mathbf{AQ} \rightarrow$ similar to $\mathbf{A}$

if the matrix is symmetric, $\mathbf{A} = \mathbf{S} \rightarrow \mathbf{H}_k = \mathbf{Q}_k{}^T\mathbf{SQ}_k$ is also symmetric $\rightarrow \mathbf{H}_k$ is tridiagonal $\mathbf{T}_k$

Arnoldi $\rightarrow$ Lanczos, Arnoldi iteration needs only one orthogonalization step

$\mathbf{SQ}_k = \mathbf{Q}_{k+1}\mathbf{T}_{k+1,k} \rightarrow \mathbf{T}_k = \mathbf{Q}_k{}^T\mathbf{SQ}_k$

$\mathbf{A} = \mathbf{U\Sigma V}^T \xrightarrow{\mathbf{\Sigma}\text{ invariant}} \mathbf{Q}_1\mathbf{AQ}_2{}^T = \mathbf{Q}_1\left(\mathbf{U\Sigma V}^T\right)\mathbf{Q}_2{}^T = \left(\mathbf{Q}_1\mathbf{U}\right)\mathbf{\Sigma}\left(\mathbf{Q}_2\mathbf{V}\right)^T$

find $\mathbf{Q}_1$, $\mathbf{Q}_2$ so that $\mathbf{Q}_1\mathbf{AQ}_2{}^T$ is bidiagonal where $\left(\mathbf{QU}\right)^{-1} = \mathbf{U}^{-1}\mathbf{Q}^{-1} = \mathbf{U}^T\mathbf{Q}^T = \left(\mathbf{QU}\right)^T$

$$\mathbf{Q}^{-1}\mathbf{SQ} = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & b_2 & \ddots & \\ & & & a_n \end{bmatrix} \text{ for } \lambda's, \quad \mathbf{Q}_1\mathbf{AQ}_2{}^T = \begin{bmatrix} c_1 & d_1 & & \\ 0 & c_2 & d_2 & \\ & 0 & \ddots & \\ & & & c_n \end{bmatrix} \text{ for } \sigma's$$

for bigger matrix $\left(\mathbf{A}:\ \text{million}\right) \rightarrow$ Krylov subspace $\left(k:\ \text{hundred}\right)$

$\mathbf{v} = c_1\mathbf{b} + c_2\mathbf{Ab} + c_3\mathbf{A}^2\mathbf{b} + \cdots + c_k\mathbf{A}^{k-1}\mathbf{b}\left(+error\right)$

III.1 Low Rank Change in A and its Inverse

- Sherman-Morrison-Woodbury formula
  - How small changes in a matrix affect its inverse
  - If A is changed by a rank-one matrix, so is its inverse
  - New data in least squares will produce these changes

$$\text{perturbations of identity: } \Big(\mathbf{I}-\underbrace{\mathbf{u}\mathbf{v}^T}_{\text{rank } 1}\Big)^{-1},\ \Big(\mathbf{I}-\underbrace{\mathbf{U}\mathbf{V}^T}_{\text{rank } k}\Big)^{-1},\ \left(\mathbf{A}-\mathbf{U}\mathbf{V}^T\right)^{-1}$$

$$\left(\mathbf{I}-\mathbf{u}\mathbf{v}^T\right)^{-1}=\mathbf{I}+\frac{\mathbf{u}\mathbf{v}^T}{1-\mathbf{v}^T\mathbf{u}} \rightarrow \text{if a matrix is changed by rank 1, its inverse is changed by rank 1}$$

$$\text{check : } \left(\mathbf{I}-\mathbf{u}\mathbf{v}^T\right)\left(\mathbf{I}+\frac{\mathbf{u}\mathbf{v}^T}{1-\mathbf{v}^T\mathbf{u}}\right)=\mathbf{I}-\mathbf{u}\mathbf{v}^T+\frac{\left(\mathbf{I}-\mathbf{u}\mathbf{v}^T\right)\mathbf{u}\mathbf{v}^T}{1-\mathbf{v}^T\mathbf{u}}=\mathbf{I}\leftarrow\left[\left(\mathbf{I}-\mathbf{u}\mathbf{v}^T\right)\mathbf{u}\mathbf{v}^T=\mathbf{u}\mathbf{v}^T-\mathbf{u}\left(\mathbf{v}^T\mathbf{u}\right)\mathbf{v}^T=\left(1-\mathbf{v}^T\mathbf{u}\right)\mathbf{u}\mathbf{v}^T\right]$$

$$(n\times n)\ \text{matrix to invert: } \Big(\mathbf{I}_n-\underbrace{\mathbf{U}\mathbf{V}^T}_{(n\times k)(k\times n)}\Big)^{-1}=\mathbf{I}_n+\mathbf{U}\underbrace{\left(\mathbf{I}_k-\mathbf{V}^T\mathbf{U}\right)^{-1}}_{(k\times k)}\mathbf{V}^T$$

$$\text{check : } \left(\mathbf{I}_n-\mathbf{U}\mathbf{V}^T\right)\left(\mathbf{I}_n+\mathbf{U}\left(\mathbf{I}_k-\mathbf{V}^T\mathbf{U}\right)^{-1}\mathbf{V}^T\right)=\mathbf{I}_n-\mathbf{U}\mathbf{V}^T+\underbrace{\left(\mathbf{I}_n-\mathbf{U}\mathbf{V}^T\right)\mathbf{U}}_{\mathbf{U}-\mathbf{U}\mathbf{V}^T\mathbf{U}=\mathbf{U}\left(\mathbf{I}-\mathbf{V}^T\mathbf{U}\right)}\left(\mathbf{I}_k-\mathbf{V}^T\mathbf{U}\right)^{-1}\mathbf{V}^T$$

$$\left(\mathbf{A}-\mathbf{U}\mathbf{V}^T\right)^{-1}=\mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I}_k-\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}$$

Use 1: suppose $\mathbf{Aw}=\mathbf{b}$ is solved for $\mathbf{w}$ → now, solve $\left(\mathbf{A}-\mathbf{uv}^T\right)\mathbf{x}=\mathbf{b}$ quickly

$$\left.\begin{array}{l}\mathbf{Aw}=\mathbf{b}\\ \mathbf{Az}=\mathbf{u}\end{array}\right\}\rightarrow D=1-\mathbf{v}^T\mathbf{z}\rightarrow \mathbf{x}=\mathbf{w}+\frac{\mathbf{v}^T\mathbf{w}}{D}\mathbf{z}$$

Use 2: new measuremnet/data comes in, changes things but leaves a big part unchanged and you finf that new $\hat{\mathbf{x}}$

$$(\text{old})\;\mathbf{Ax}=\mathbf{b}\xrightarrow{\text{normal equation}}\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}}=\mathbf{A}^T\mathbf{b}$$

$$(\text{new})\;\begin{bmatrix}\mathbf{A}\\ \mathbf{v}^T\end{bmatrix}\mathbf{x}=\begin{bmatrix}\mathbf{b}\\ \mathbf{b}_{new}\end{bmatrix}\xrightarrow{\text{normal equation}}\begin{bmatrix}\mathbf{A}^T & \mathbf{v}\end{bmatrix}\begin{bmatrix}\mathbf{A}\\ \mathbf{v}^T\end{bmatrix}\hat{\mathbf{x}}_{new}=\begin{bmatrix}\mathbf{A}^T & \mathbf{v}\end{bmatrix}\begin{bmatrix}\mathbf{b}\\ \mathbf{b}_{new}\end{bmatrix}$$

$$\left(\mathbf{A}^T\mathbf{A}+\underbrace{\mathbf{vv}^T}_{\substack{\text{ramk 1}\\ \text{change}}}\right)\hat{\mathbf{x}}=\mathbf{A}^T\mathbf{b}+\mathbf{vb}_{new}:\text{ recursuve least squares}\rightarrow\begin{cases}\left(\mathbf{A}^T\mathbf{A}+\mathbf{vv}^T\right)^{-1}=\left(\mathbf{A}^T\mathbf{A}\right)^{-1}-C\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{vv}^T\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\\ C=\dfrac{1}{1-\underbrace{\mathbf{v}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v}^T}_{\rightarrow\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{y}=\mathbf{v}^T}}\end{cases}$$

$$\text{least squares}\rightarrow\begin{cases}(1)\text{ standard: covariance}=\mathbf{I}\begin{cases}\text{the data is not correlated}\\ \text{it all has the same varaince}\end{cases}\\ (2)\text{ covariance}\neq\mathbf{I}\begin{cases}\text{weighted least squares}\\ \text{correlated least squares}\end{cases}\rightarrow\text{how errors are correlated}\\ \rightarrow\text{state equations (dynamic part) in control theory: position of satellite}\end{cases}$$

Kalman filter for dynamic least squares: significantly improved version of recursive least squares including the covariance matrix

$$\mathbf{M}=\mathbf{I}-\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\rightarrow\mathbf{M}^{-1}=?$$

$$\mathbf{M}=\mathbf{I}-\mathbf{u}\mathbf{v}^T\text{ where }\mathbf{u}=\mathbf{v}=\begin{bmatrix}1\\1\\1\end{bmatrix}\rightarrow\mathbf{M}^{-1}=\mathbf{I}+\frac{\mathbf{u}\mathbf{v}^T}{1-\mathbf{v}^T\mathbf{u}}=\mathbf{I}+\frac{1}{1-3}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$$\mathbf{M}=\mathbf{I}-\begin{bmatrix}0&1&1\\0&0&1\\0&0&0\end{bmatrix}=\mathbf{I}-\mathbf{U}\mathbf{V}^T=\mathbf{I}-\begin{bmatrix}1&0\\0&1\\0&0\end{bmatrix}\begin{bmatrix}0&1&1\\0&0&1\end{bmatrix}\rightarrow\mathbf{M}^{-1}=\mathbf{I}_3+\mathbf{U}\left(\mathbf{I}_2-\mathbf{V}^T\mathbf{U}\right)^{-1}\mathbf{V}^T=\mathbf{I}_3+\mathbf{U}\begin{bmatrix}1&-1\\0&1\end{bmatrix}^{-1}\mathbf{V}^T$$

Kalman Filter: $\mathbf{x}(\mathbf{t})\rightarrow\Delta\mathbf{x}=\mathbf{v}\Delta\mathbf{t}\rightarrow\mathbf{x}_{n+1}$

$$\begin{matrix}\text{original}\\\text{state update}\\\text{measurement update}\end{matrix}\begin{bmatrix}\mathbf{A}&\mathbf{0}\\-\mathbf{I}&\mathbf{I}\\\mathbf{0}&\mathbf{r}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{old}\\\mathbf{x}_{new}\end{bmatrix}=\begin{bmatrix}\mathbf{b}\\\mathbf{v}\Delta\mathbf{t}\\\mathbf{b}_{m+1}\end{bmatrix}\xrightarrow[\text{full normal equations}]{\text{instead of solving}}\begin{cases}\text{prediction }\hat{\mathbf{x}}_{state}\leftarrow\text{state equation}\\\hat{\mathbf{x}}_{new}=\hat{\mathbf{x}}_{state}+\mathbf{K}\left(\mathbf{b}_{m+1}-\mathbf{r}\hat{\mathbf{x}}_{state}\right)\\\mathbf{K}:\text{ gain matrix from }\mathbf{A},\mathbf{r},\mathbf{V}_{state},\mathbf{V}_{\mathbf{b}}\end{cases}$$

state/measurement equations have their own covariance matrices.

variance or covariance $\mathbf{V}$ measures their different reliabilities

$$\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}}=\mathbf{A}^T\mathbf{b}\rightarrow\mathbf{A}^T\mathbf{V}^{-1}\mathbf{A}\hat{\mathbf{x}}=\mathbf{A}^T\mathbf{V}^{-1}\mathbf{b}$$