Lecture 15 Outline

Sunday, October 20, 2019 9:49 PM

QR Factorization

- Classical Gram-Schmidt
- Modified Gram-Schmidt
- Householder Triangularization

Eigensystems

QR Factorization

Sunday, October 20, 2019 9:54 PM

How to compute QR factorization?

$\underset{m\times n}{\underline{A}} = \underset{m\times n}{\underline{\hat{Q}}}\,\underset{n\times n}{\underline{\hat{R}}} \leftrightarrow$ Partial QR factorization

$\underset{m\times n}{\underline{A}} = \underset{m\times m}{\underline{Q}}\,\underset{m\times n}{\underline{R}} \leftrightarrow$ Full QR factorization

$$\underline{\hat{Q}}^T \underline{\hat{Q}} = \underline{\hat{Q}}\,\underline{\hat{Q}}^T = \underline{I}$$

$$\underline{Q}^T \underline{Q} = \underline{Q}\,\underline{Q}^T = \underline{I} \Rightarrow \underline{Q}^T = \underline{Q}^{-1}$$

Classical Gram-Schmidt Algorithm $\rightarrow$

projection based, not stable numerically (round off error)

Modified Gram-Schmidt

Recall that projection can be written as a matrix-vector product $\Rightarrow$

$$\underline{q}_1 = \frac{\underline{P}_1 \underline{a}_1}{\|\underline{P}_1 \underline{a}_1\|}, \quad \underline{q}_2 = \frac{\underline{P}_2 \underline{a}_2}{\|\underline{P}_2 \underline{a}_2\|}, \text{ etc.}$$

$\underline{q}_1 = \frac{\quad}{\|\underline{P}_1 \underline{a}_1\|}$, $\underline{q}_2 = \frac{\quad}{\|\underline{P}_2 \underline{q}_2\|}$, etc.

for some $\underline{P}_j$

Let $\hat{\underline{Q}}_{j-1}$ be the $m \times (j-1)$ matrix of the first $j-1$ columns of $\hat{\underline{Q}}$

where

$$\hat{\underline{Q}} = [\underline{q}_1 \ \underline{q}_2 \ \cdots \ \underline{q}_n]$$

$$\hat{\underline{Q}}_{j-1} = [\underline{q}_1 \ \underline{q}_2 \ \cdots \ \underline{q}_{j-1}]$$

Then

$$\underline{P}_j = \underline{I} - \hat{\underline{Q}}_{j-1} \hat{\underline{Q}}_{j-1}^T$$ $\rightarrow$ matrices of the form $\underline{I} - \underline{v}\,\underline{v}^T$ project onto the perpendicular space of $\underline{v}$

Thus, $\underset{\sim}{P}_j$ is nothing but the repeated perpendicular projections of each prior vector in $\underset{\sim}{\hat{Q}}$ or

$$\underset{\sim}{P}_j = \underset{\sim}{P}_{\perp q_{j-1}} \underset{\sim}{P}_{\perp q_{j-2}} \cdots \underset{\sim}{P}_{\perp q_2} \underset{\sim}{P}_{\perp \underset{\sim}{q}_1}$$

with $\underset{\sim}{P}_1 = \underset{\sim}{I}$

Each $\underset{\sim}{P}_{\perp q_j}$ projects onto the space perpendicular to $\underset{\sim}{q}_j$

Modified Gram-Schmidt uses these ideas to reverse the order of operations, such that

Algorithm: Modified G-S

for $i = 1:n$

$\quad \underset{\sim}{v}_i = \underset{\sim}{q}_i$

end

end

for $i = 1:n$

$r_{ii} = \|\underline{v}_i\|$

$\underline{q}_i = \underline{v}_i / r_{ii}$

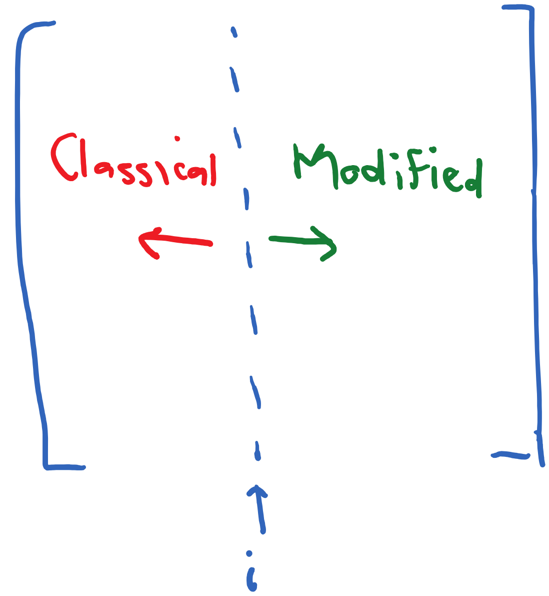
for $j = i+1:n$

$r_{ij} = \underline{q}_i^T \underline{v}_j$

$\underline{v}_j = \underline{v}_j - r_{ij}\underline{q}_i$

end

end



Reduces effects of roundoff error

Operation count for Modified G-S is identical to Classical G-S : $O(2mn^2)$

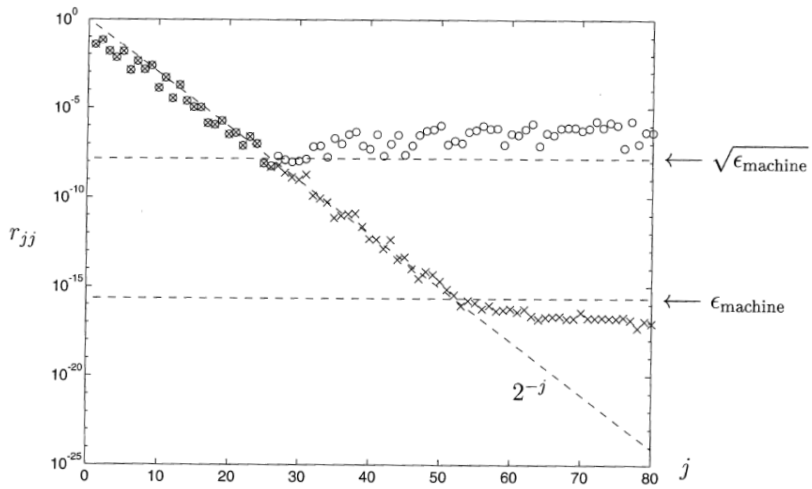


Figure 9.1. *Computed r_{jj} versus j for the QR factorization of a matrix with exponentially graded singular values. On this computer with about 16 digits of relative accuracy, the classical Gram–Schmidt algorithm produces the numbers represented by circles and the modified Gram–Schmidt algorithm produces the numbers represented by crosses.*

Trefethen & Bau (1997)

Householder Triangularization

Look at G-S again

In G-S each operation to compute a column of $\underline{\hat{Q}}$ is an upper triangular matrix multiplication

$$\underline{A}\,\underline{R}_1\underline{R}_2 \cdots \underline{R}_n = \hat{Q} \Rightarrow \underline{A} = \hat{Q}\hat{R}$$

$$\underbrace{\qquad\qquad}_{\hat{\underline{R}}^{-1}}$$

This is called Triangular Orthogonalization:

R gives Q

One can do the reverse: repeated applications of $\underline{Q}$ give $\underline{R}$

$$\underbrace{\underline{Q}_n \underline{Q}_{n-1} \cdots \underline{Q}_2 \underline{Q}_1}_{\hat{\underline{Q}}^T} \underline{A} = \hat{\underline{R}} \Rightarrow \underline{A} = \hat{\underline{Q}} \hat{\underline{R}}$$

This is called Orthogonal Triangularization

Q gives R

For this, we need to find the $\underline{Q}_k$

The main idea is to find a matrix $\underline{Q}_k$ that zeros out the values below a diagonal while preserving all prior zeros

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{\underline{Q}_1} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \xrightarrow{\underline{Q}_2} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \xrightarrow{\underline{Q}_3} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{A} \qquad \underline{Q}_1\underline{A} \qquad \underline{Q}_2\underline{Q}_1\underline{A} \qquad \underline{Q}_3\underline{Q}_2\underline{Q}_1\underline{A}$$

One more requirement: Each $\underline{Q}_k$ must be unitary $\quad \underline{Q}_k^T\underline{Q}_k = \underline{Q}_k\underline{Q}_k^T = \underline{I}$

Choose the following block matrix

$$\underline{Q}_k = \begin{bmatrix} \underline{I} & \underline{0} \\ \underline{0} & \underline{F} \end{bmatrix}$$

$\underline{I} \in (k-1)\times(k-1)$ identity matrix

$\underline{F} \in (m-k+1)\times(m-k+1)$

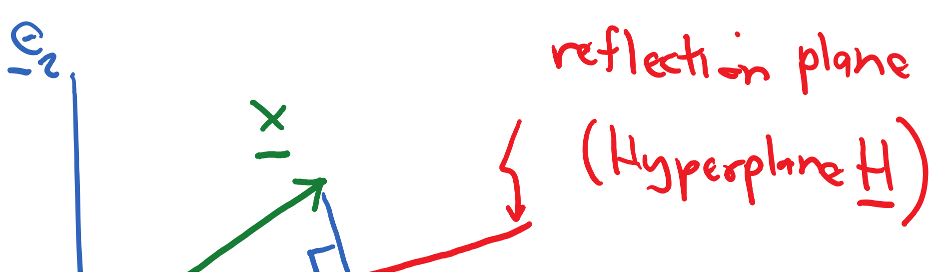
Householder reflector matrix

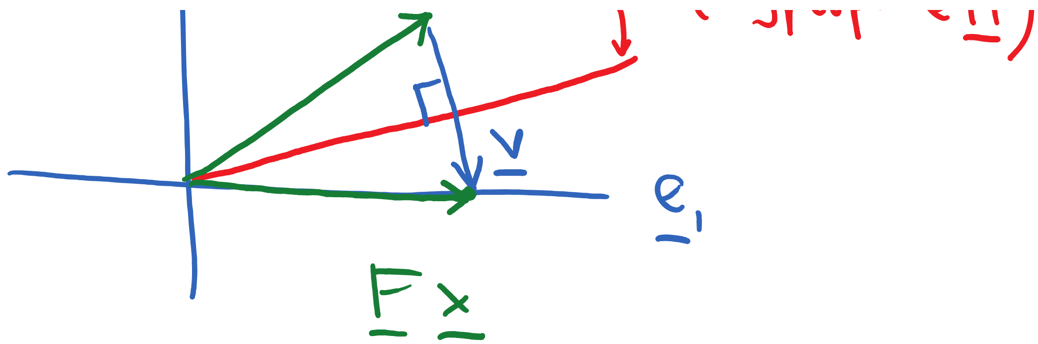
$\underline{F}$ is a specific type of operation, defined as follows:

$$\underline{x} = \begin{bmatrix} a \\ b \\ c \\ \vdots \\ \vdots \end{bmatrix} \qquad \underline{F} \Rightarrow \underline{F}\,\underline{x} = \begin{bmatrix} \|\underline{x}\|_2 \\ 0 \\ 0 \\ \vdots \\ \vdots \end{bmatrix} = \|\underline{x}\|_2\, \underline{e}_1$$

$$\underline{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ \vdots \end{bmatrix}$$

How does this appear in 2d?





Hyperplane: A plane with a dimension one less than the embedding plane

(in 2d, H is 1d; in 3d, H is 2d, ...)

To determine this projection, look at the "error" vector between $\underline{F}\,\underline{x}$ and $\underline{x}$.

Let

$$\underline{v} = \underline{F}\,\underline{x} - \underline{x} = \|\underline{x}\|_2\, \underline{e}_1 - \underline{x}$$

Is defined once $\underline{x}$ is defined

The key is that $\underline{v}$ is perpendicular to the hyperplane H (see diagram above in 2d)

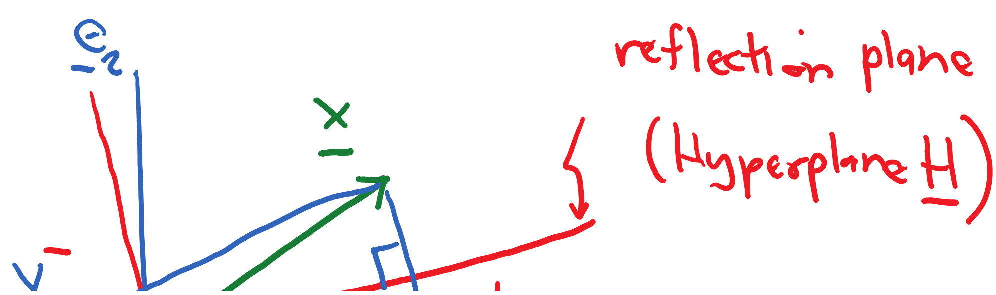
To develop $\underline{F}$, project a vector $\underline{y}$ onto $\underline{H}$:

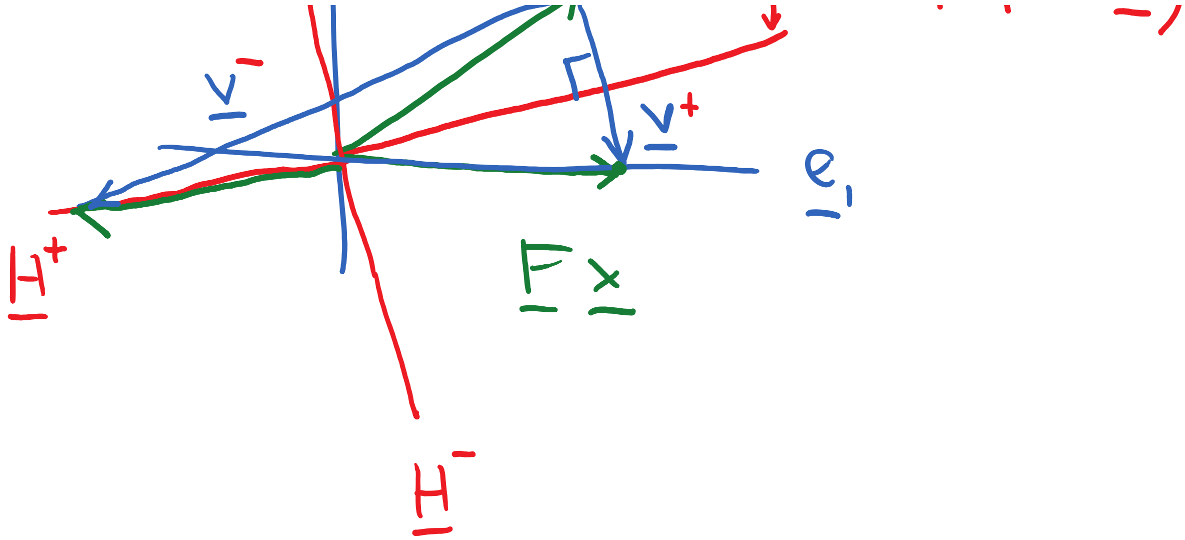
$$\underline{P}\,\underline{y} = \left(\underline{I} - \frac{\underline{v}\,\underline{v}^T}{\underline{v}^T\underline{v}}\right)\underline{y} = \underline{y} - \underline{v}\left(\frac{\underline{v}^T\underline{y}}{\underline{v}^T\underline{v}}\right)$$

However, we actually need to go twice as far (see diagram above)

$$\underline{F}\,\underline{y} = \underbrace{\left(\underline{I} - \frac{2\underline{v}\,\underline{v}^T}{\underline{v}^T\underline{v}}\right)}_{\underline{F}}\underline{y} = \underline{y} - 2\underline{v}\left(\frac{\underline{v}^T\underline{y}}{\underline{v}^T\underline{v}}\right)$$

Note: Householder reflectors are $\underline{\text{not}}$ unique (see $\underline{H}^+$ and $\underline{H}^-$ below)





$$\underline{v}^+ = +\|\underline{x}\|_2\,\underline{e}_1 - \underline{x}$$

$$\underline{v}^- = -\|\underline{x}\|_2\,\underline{e}_1 - \underline{x}$$

Mathematically, choice does <u>not</u> matter

Numerically, choice does matter & we want large $\|\underline{v}\|$

$\Rightarrow$ Set $\underline{v} = -\text{sign}(x_1)\,\|\underline{x}\|_2\,\underline{e}_1 - \underline{x}$

w/ $\text{sign}(x) = 1$ if $x = 0$

and x_1 = first component of $\underline{x}$

After clearing the $\ominus$:

$$\underline{v} = \text{sign}(x_1)\,\|\underline{x}\|_2\,\underline{e}_1 + \underline{x}$$

Algorithm: Householder QR

for k = 1 : n

$\underline{x} = \underline{A}(k:m, k)$

$\underline{v}_k = \text{sign}(x_1)\,\|\underline{x}\|_2\,\underline{e}_1 + \underline{x}$

$\underline{v}_k = \underline{v}_k / \|\underline{v}_k\|_2$

$$\underline{A}(k:m, k:n) = \underline{A}(k:m, k:n) - 2\underline{v}_k\left(\underline{v}_k^T\,\underline{A}(k:m, k:n)\right)$$

end

After these steps, $\underline{A}$ will be upper triangular

Note: Here $\hat{\underline{Q}}$ is never computed

To find $\hat{\underline{Q}}$ do the following, first define the operation of $\hat{\underline{Q}}\,\underline{x}$

for $k = n:-1:1$

$$\underline{x}(k:m) = \underline{x}(k:m) - 2\underline{v}_k\left(\underline{v}_k^T\,\underline{x}(k:m)\right)$$

end

To find $\hat{\underline{Q}}$ apply this operation to the identity matrix

$$\underline{Q}\,\underline{I} = \underline{Q} = \left[\underline{Q}\,\underline{e}_1 \;\; \underline{Q}\,\underline{e}_2 \;\; \cdots\right]$$

with $\underline{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$, $e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$, ...

QR Factorizations

Triangular Orthonormalization ($\hat{R} \to \hat{Q}$)

Gram-Schmidt (Classical + Modified)

Orthogonal Triangularization ($\hat{Q} \to \hat{R}$)

Householder Reflections

Givens Rotations (easier parallelization)

Eigensystems

Sunday, October 20, 2019 11:34 PM

To motivate, consider the solutions to the following ordinary differential equation (ODE)

$$\frac{dy}{dt} = ay \quad \text{for } a = \text{constant}$$

Aside:

$$\frac{d^2y}{dt^2} = ay + bt$$

Independent variable: t

Dependent variable: $y \equiv y(t)$

Order: 1

Homogeneity: Yes

Linearity: Yes

Solution is $y(t) = C e^{at}$

Check: $\frac{dy}{dt} = \frac{d}{dt}\left(Ce^{at}\right) = aCe^{at} = ay(t)$

What happens if we have a set of two ODEs?

$$\frac{dy_1}{dt} = ay_1, \quad \frac{dy_2}{dt} = by_2$$

Then $y_1(t) = C_1 e^{at}, \quad y_2(t) = C_2 e^{bt}$

uncoupled

However, if instead we have

$$\frac{dy_1}{dt} = a y_2 , \quad \frac{dy_2}{dt} = b y_1$$

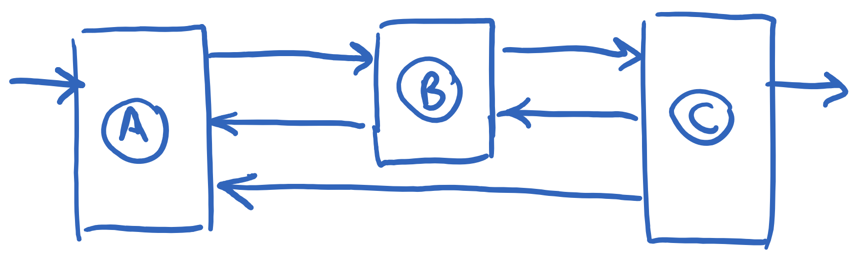
These ODEs are coupled. We need y_1 to solve for y_2 and vice versa

This is quite common in applications!

1. Chemical reactions

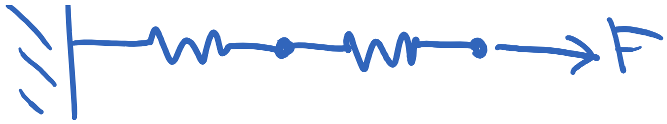
$$CO + H_2O \rightleftharpoons CO_2 + H_2$$

2. Flow between tanks



3. Series of springs

k_1 k_2



Generalize into the following for two dependent variables $y_1(t)$ & $y_2(t)$

$$\frac{dy_1}{dt} = a_{11} y_1 + a_{12} y_2$$

with constants $a_{11}, a_{12}, a_{21}, a_{22}$

$$\frac{dy_2}{dt} = a_{21} y_1 + a_{22} y_2$$

Let

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\Rightarrow \frac{d\underline{y}}{dt} = \begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{bmatrix} \Rightarrow \frac{d\underline{y}}{dt} = \underline{A}\,\underline{y}$$

Assume a solution

$$y_1 = x_1 e^{\lambda t}, \quad y_2 = x_2 e^{\lambda t}$$

$$\frac{dy_1}{dt} = \lambda x_1 e^{\lambda t}, \quad \frac{dy_2}{dt} = \lambda x_2 e^{\lambda t}$$

$$\frac{d\underline{y}}{dt} = \underline{A}\,\underline{y}$$

$$\begin{bmatrix} \lambda x_1 \cancel{e^{\lambda t}} \\ \lambda x_2 \cancel{e^{\lambda t}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \cancel{e^{\lambda t}} \\ x_2 \cancel{e^{\lambda t}} \end{bmatrix}$$

$$\underline{A}\,\underline{x} = \lambda \underline{x}$$ Eigensystem of $\underline{A}$

$$\underline{A}\,\underline{x} - \lambda \underline{x} = \underline{0}$$

Frequently appears in applications

$$(\underline{A} - \lambda \underline{I})\vec{x} = \vec{0}$$

λ : Eigenvalue

$\underline{x}$: Eigenvector