Una matrice ortogonale sono partciolari matrici quadrate, la matrice Q quadrata è ortogonale se ha righe e colonne ortogonali (prod scal. di una col per se stessa è = a 1 lo stesso vale per riga) e hanno lunghezza geometrica 1, cioè si dice che sono ortonormali.

Orthogonal matrices

a **square** matrix Q is orthogonal if its columns (and rows) are orthogonal and of unit length (**orthonormal** columns and rows)

Il prodtto tra due righe o due colonne è 0 e il prod di una riga o colonna è uno, per esprimerlo possiamo scrivere queste due formule.
I è la matrice di identità.

$$Q^T Q = I$$

qui si riferisce alle colonne

$$QQ^T = I$$

qui alle righe.

Qui ci spiega che la matrice inversa di una matr ortogonale coincide con la trasposta.

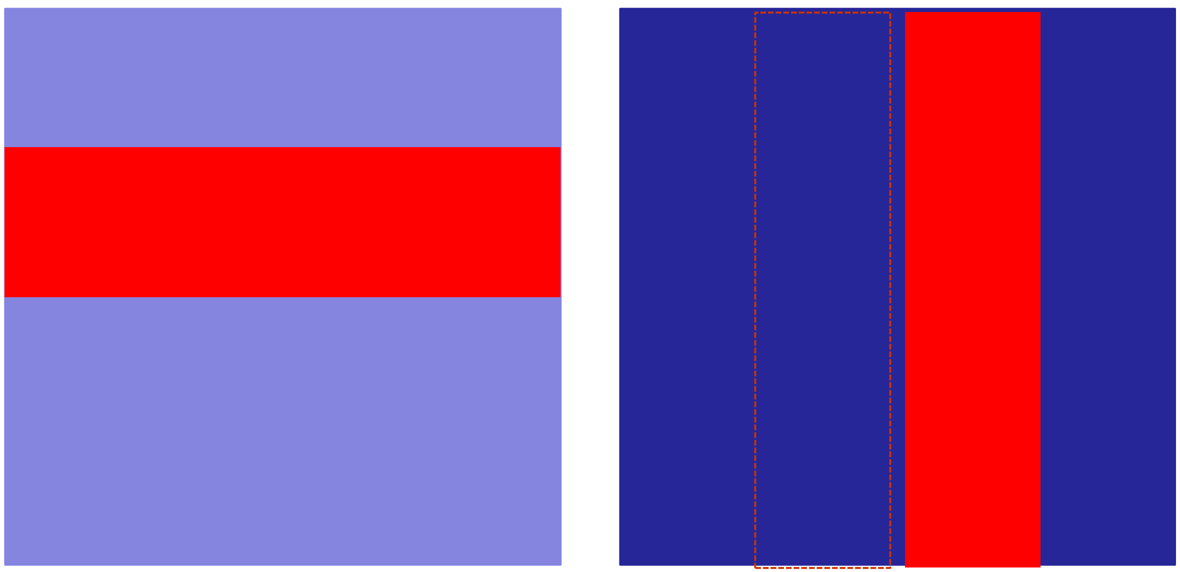
$$Q^T = Q^{-1}$$

Qui mostra appunto quello che succede. Se consideriamo la stessa riga di Qt e la stessa col di Q otteniamo come prod scalare 1. Se invece consideriamo posizioni diverse otteniamo 0. Quindi alla fine otteniamo la matr identità.

Orthogonal matrices

columns orthogonal and of unit length

$$Q^T Q = I$$



$$Q^T \qquad Q$$

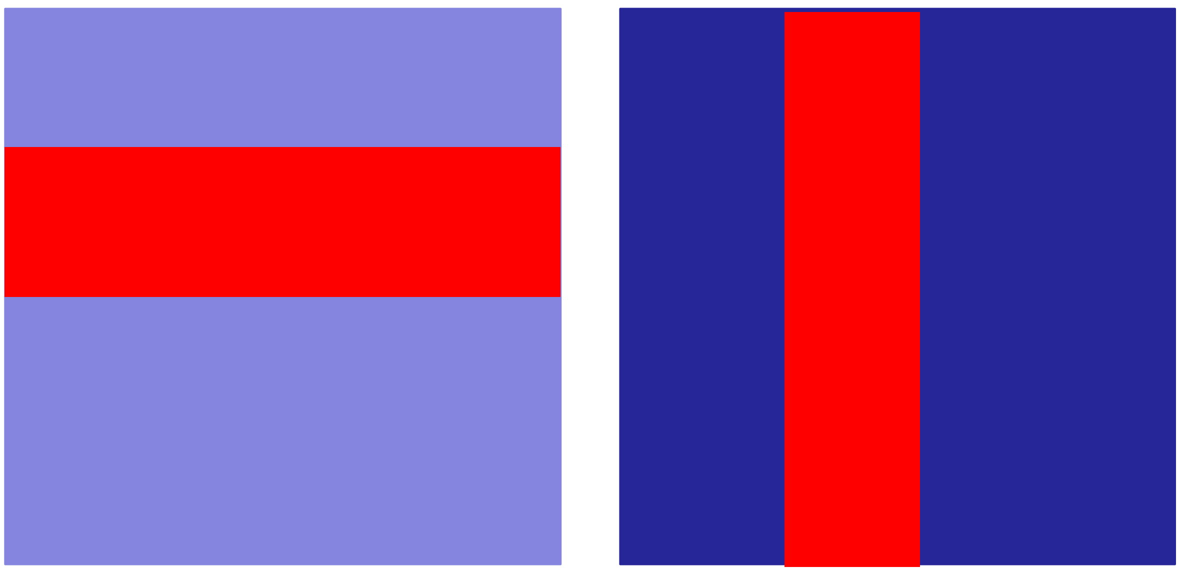
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \boxed{0} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Qui appunto è quello che ho detto prima.

Orthogonal matrices

columns orthogonal and of unit length

$$Q^T Q = I$$



Q^T $\qquad$ Q

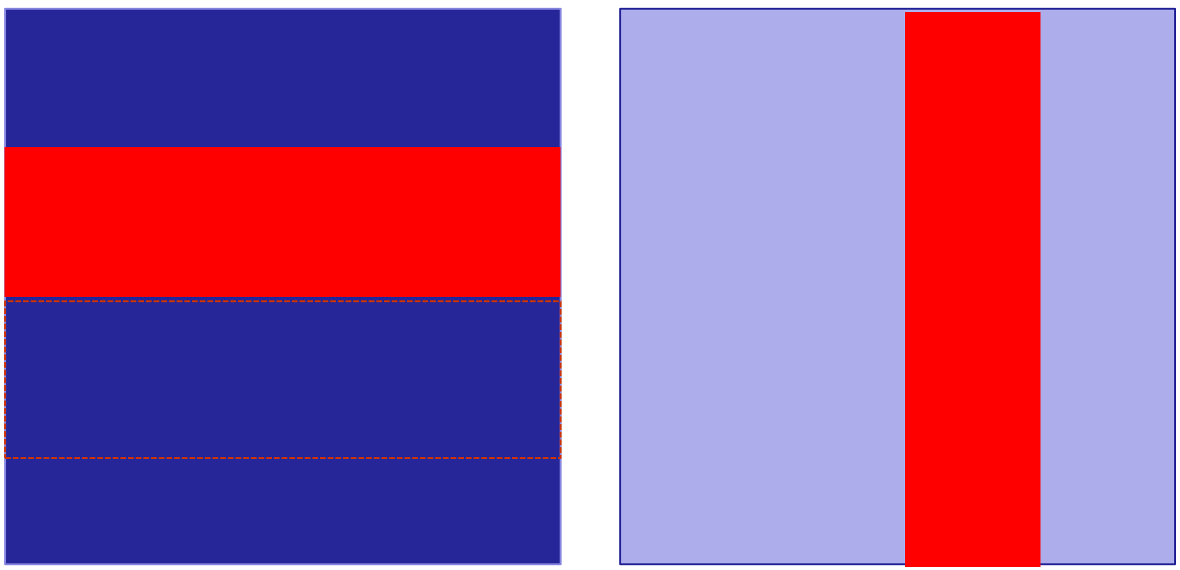
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Qui ovviamente vale la stessa cosa per le righe.

Orthogonal matrices

rows orthogonal and of unit length

$$QQ^T = I$$



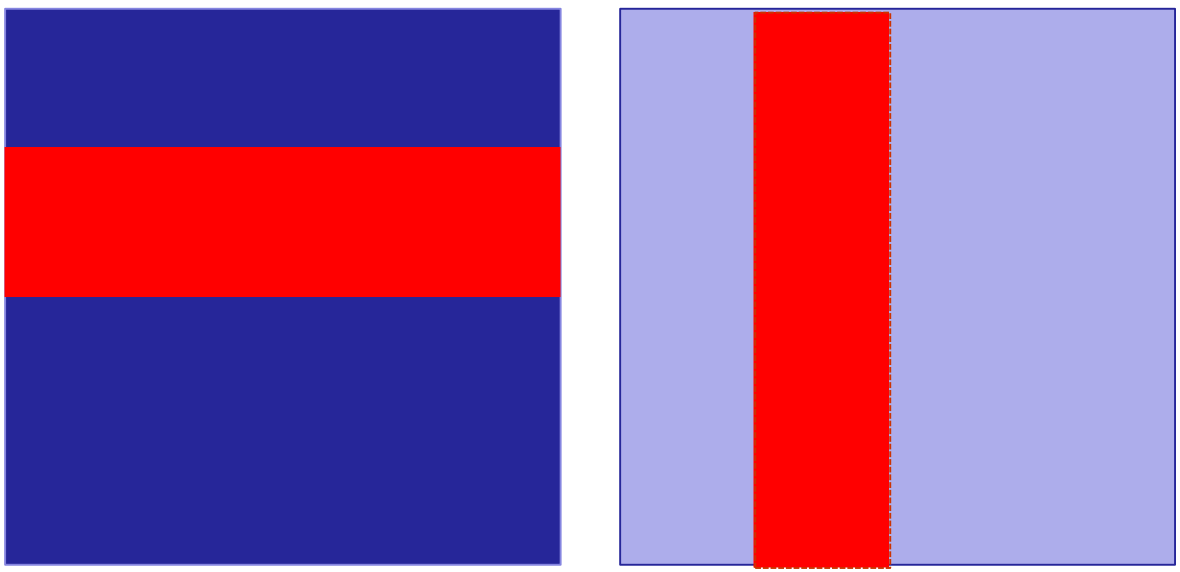
Q $\qquad$ Q^T

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \boxed{0} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Orthogonal matrices

rows orthogonal and of unit length

$$QQ^T = I$$



Q $\quad$ Q^T

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Una matrice non quadrata W con colonne ortogonali e di lung un ma non è una matrice ortogonale, si chiama matrice con colonne ortonormali e si definisce nel rettangolo verde.

a **non square** matrix W with orthogonal and unit length columns (**orthonormal** columns) **is not** an **orthogonal matrix**

W has the property (**orthonormal** columns) :

$$W^T W = I$$

attention:

$$WW^T \neq I$$

Attenzione: in questo caso la ortonormalità della matrice W è testimoniata da quella formula verde, MA ATTENZIONE W*Wt non è uguale all'identita. Questa è una matrice importante perchè è il proiettore ortogonale sul range W

$$WW^T$$

is the orthogonal projector onto the $range(W)$

Un tipico esempio di matrice ortogonale è la matrice di rotazione orario in R quadro. Questa matrice se applicata con la formula in basso cioè * per x ruota x e ottiene y. Y è un vettore che ha la stessa lung di X e ruotato in senso orario di un angolo theta. Se ho y e voglio x devo molt y*Q trasposto, sempre perché l'invera i Q è la trasposta per la definizione di matr ortogonale.

Orthogonal matrices

clockwise rotation matrix in R^2

$$Q = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$y = Qx$$

y has the same length as x and is rotated by the angle θ clockwise

Questo è uno "strano esempio", che però ci da modo di vedere un tencnica che si usa per costruire algoritmi che vedremo più avanti.
Lo ignoro non credo che serve.

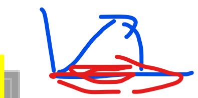
Orthogonal matrices

rotation matrix in R^2

Il problema è che <bbiamo un vettore x che conosicamo allora vogliamo ruotare x e renderlo y la cui però seconda componente è 0 cioè il vettore si trova solo sull'asse delle X.

$$Q = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

$$y = Qx$$



Exercise: choose c and s such that the second component of y is 0

Quindi vogliamo che y2 = 0, y2 si ottiene con il prod scalare tra ultima riga di Q * X. quindi scrivendolo per esteso ottengono questo -sx+cx

$$0 = y_2 = \begin{pmatrix} -s & c \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -sx_1 + cx_2$$

Dopodichè porto il membro a sx e uno a dx e elevo al quadrato e ottengono quello sottolineato in rosso. S e C sono seni e coseni quindi ricordando l'equazione fondamentale della trigonometrica che sin quadro + cos quadro = 1 posso scrivere uno in funzione dell'altro e ottengo quindi quest'ultimo quadrato, quello in blu.

$$sx_1 = cx_2, \; s^2x_1^2 = c^2x_2^2$$

$$s^2 + c^2 = 1$$

$$(1 - c^2)x_1^2 = c^2x_2^2$$

Quindi passanfo al quadrato posso usare l'equazione fondamentale della trigonometria quindi da 2 incognite sono passato a 1

Dopodichè risolve l'equazione facendo un po di calcoli e alla fine ottengono la relazione verde e facendo la radice ottengo il valore di C. Questa operazione che prende il vettore e lo ruota, cioè definisce la matrc di rotazione che lo porta su un solo asse è detta Givens rotation.

Orthogonal matrices

rotation matrix in R^2

$$Q = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

$$y = Qx$$

$$1 - c^2 = \frac{c^2 x_2^2}{x_1^2}$$

$$\frac{c^2 x_2^2}{x_1^2} + c^2 = 1$$

$$c^2 \left(\frac{x_2^2}{x_1^2} + 1 \right) = 1$$

$$c^2 = \frac{x_1^2}{x_1^2 + x_2^2}$$

$$c = \frac{|x_1|}{\sqrt{x_1^2 + x_2^2}}$$

Givens rotation

Orthogonal matrices

Le matrc ortogonali hanno alcune proprietà. La prima è che preserva la lunghezza del vettore. Cioè se x ha una certa lung e lo moltiplico per la matrc ortogonale allora ottengo un nuovo vettore con la stessa lunghezza. Si può mostrare che tutte le molt per le matr ortogonali sono o rotazioni o reflessioni rispetto ad esse. In entrambe le situazioni comunque non si cambia la lunghezza (per capire se la molt è reflesisone o rotazione deve calcolare il determinante.). L'altra proprieta è che il prodtto tra due matrici ortognali è ancora una matrice ortogonale

properties

$$\|Qx\|_2 = \|x\|_2$$

preserve length

a multiplication for an orthogonal matrix is either a **rotation** (if $det(Q)=1$) or a **reflexion** (if $det(Q)$=-1)

the **product** of two orthogonal matrices is an orthogonal matrix

Quest'altra proprietà dice che se ho due vettori e ne faccio il loro prodotto scalare otteniamo un numero che è lo stesso che avrei se trasformassi x e y con Q, la trasformazione singolarmente elle due componenti.

Orthogonal matrices

properties

$$\left(Qx\right)^{T} \cdot \left(Qy\right) = x^{T} y$$

invariance of the scalar product for orthogonal transformations

Poichè il prod scalare è legato all'angolo allora l'angolo tra i due vettori non cambia se applichiamo la stessa trasformazione ortogonale.

↓

invariance of the angle between the two vectors

Altre proprietà. Le matr ortognali hanno norma 2 = 1 il prod di una matrc ort per un altra matrice = alla norma due della matr quindi preserva la norma. Lo stesso vale per la Frobenius norm

Orthogonal matrices

properties

$$\|Q\|_2 = 1$$

$$\|QA\|_2 = \|A\|_2$$

preserve the 2-norm

$$\|QA\|_F = \|A\|_F$$

preserve the Frobenius norm

ogni matrice A, quatrata o rettnagola si puo scrivere come il prodotto di Q*R dove Q è ortogonale quindi quadrata e R è una matrice triangolare superiore.

QR Factorization

$$\boxed{A = QR}$$

the QR factorization exists for any matrix

Q is **orthogonal**

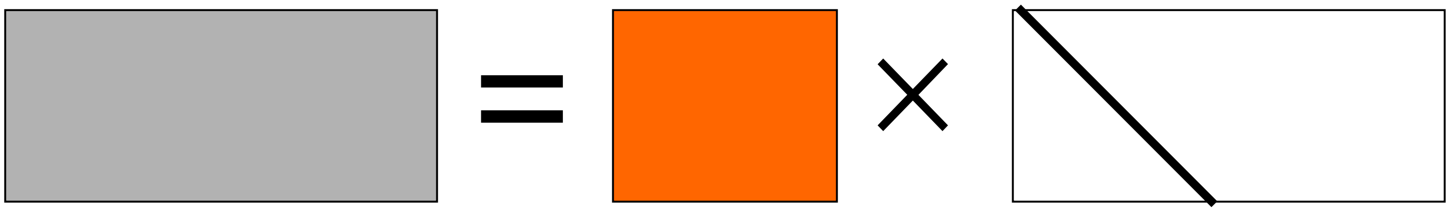
R is **upper triangolar**

```
[Q,R]=qr(A)
```

questo è il comando in matlab.

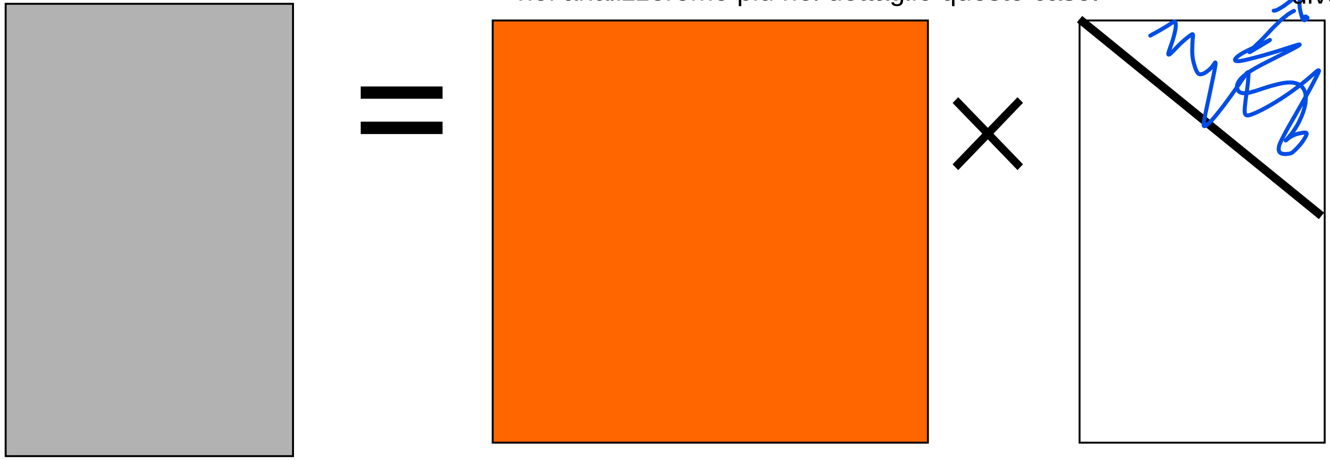
Per quanto riguarda le dimensioni A sarà MxN Q MxM e R è MxN

$$\boxed{A = QR}$$



noi analizzeremo più nel dettaglio questo caso.

qui ci sono gli ele
diversi da 0



Per ottenere R a partire da A dobbiamo moltiplicare a con Q trasposto e otteniamo R

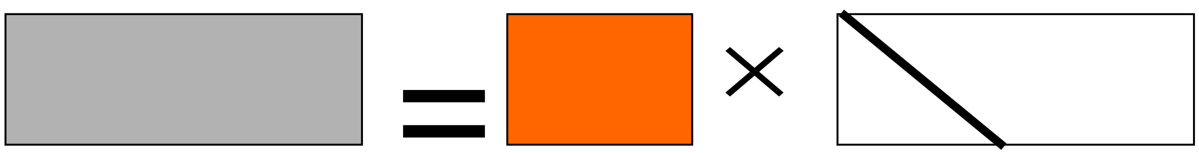
$$\boxed{A \quad m\times n \quad , \quad Q \quad m\times m \quad , \quad R \quad m\times n}$$

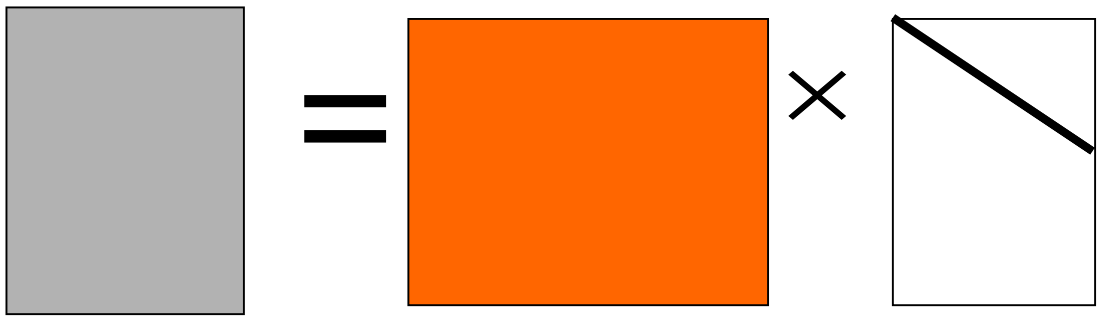
$$A = QR \Rightarrow a_j = Qr_j$$

a_j j-th column of A

r_j j-th column of R

r_j contains the components of a_j respect to the basis formed by the columns of Q

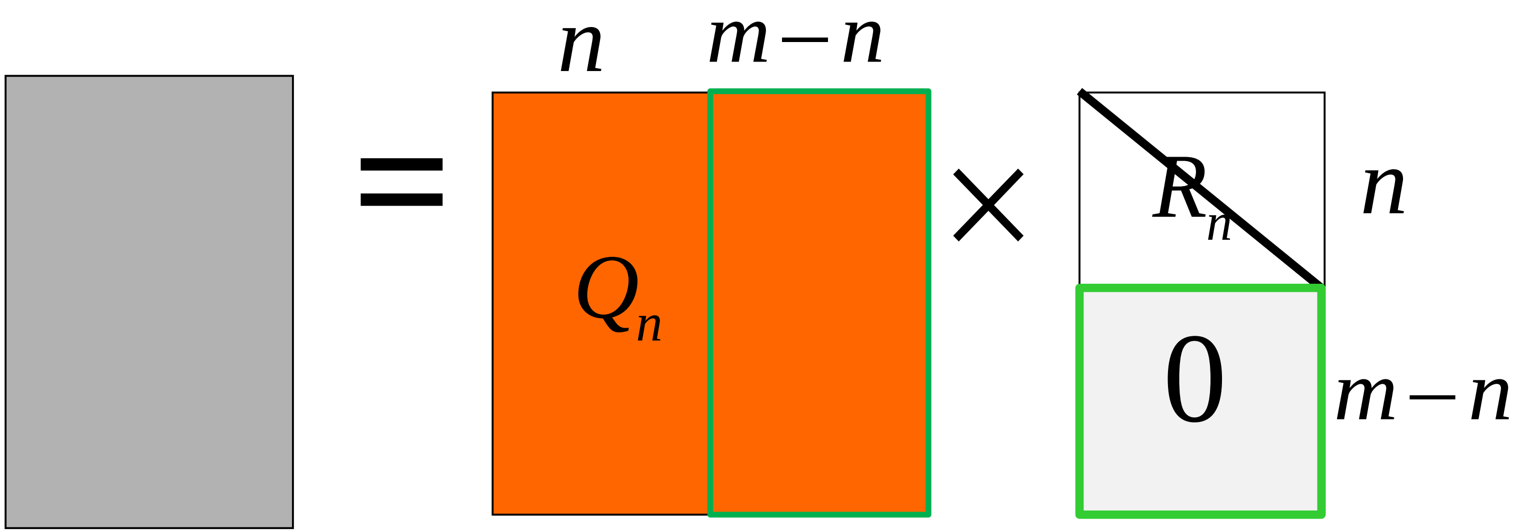




QR Factorization in «reduced» form (economy)

$$A = QR$$

$$m > n$$

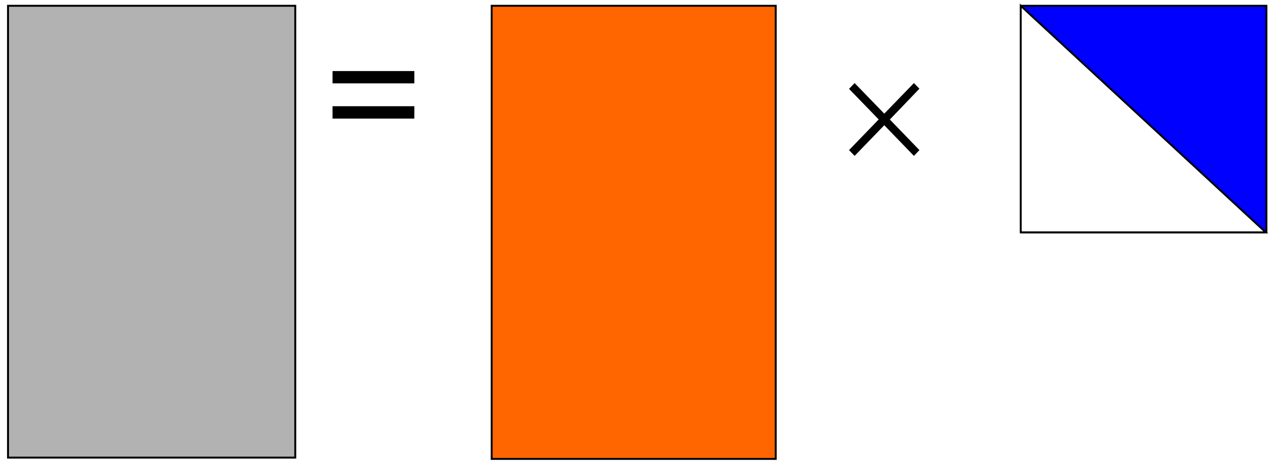


$$A \;\; m\times n \;,\; Q \;\; m\times m \;,\; R \;\; m\times n \;,\; Q_n \;\; m\times n \;,\; R_n \;\; n\times n$$

A è sempre MxN Q nel caso classico è MxM e R è MxN. Per ottenere Qn e Rn dobbiamo considerare che di R prendiamo come colonne sempre le n ma in questo caso prendiamo le prime n righe anche perchè nelle m-n troviamo solo 0, quindi di q ci serviranno solo le prime n colonne e di fatto possiamo non considerare le m-n successive. In ogni caso dalla figura si capisce. Ovviamente m>n.

Qui spiega che è possibile ottenere A usando la fattorizzazione in forma ridotta cioè invece di prendere interamente R e Q prendiamo solo una parte che chiamiamo Qn e Rn, nella slide precedente è possibile capire come si ottiene Rn e Qn.

QR Factorization in «reduced» form (economy)



Quindi A è ottenuto con il prodotto di questa roba qui.

$$A = Q_n R_n$$

$$A \quad m\times n \quad , \quad Q_n \quad m\times n \quad , \quad R_n \quad n\times n$$

```
[Qn,Rn]=qr(A,0)
```

In matlab con questo comando riusciamo a qualcosare Qn e Rn

Questa slide ripete il fatto che per ottenere R moltiplichiamo la trasposta di Q per A. otteniamo quindi R che per convenienza lo vediamo come un vettore dove al primo termine abbiamo Rn che è un blocco quadrato e sotto il blocco nullo, in particolare Rn è quadrato e triangolare superiore. Rn è di dimensione n mentre il blocco nullo è m-n

$$A = QR$$

$$Q^T A = R = \begin{pmatrix} R_n \\ 0 \end{pmatrix} \begin{array}{c} n \\ m-n \end{array}$$

Se il rango di A = r allora il rango di R = r, quindi se r è minore di n ovviamente vuol dire che sulla diagonale c'è qualche elemento nullo in Rn

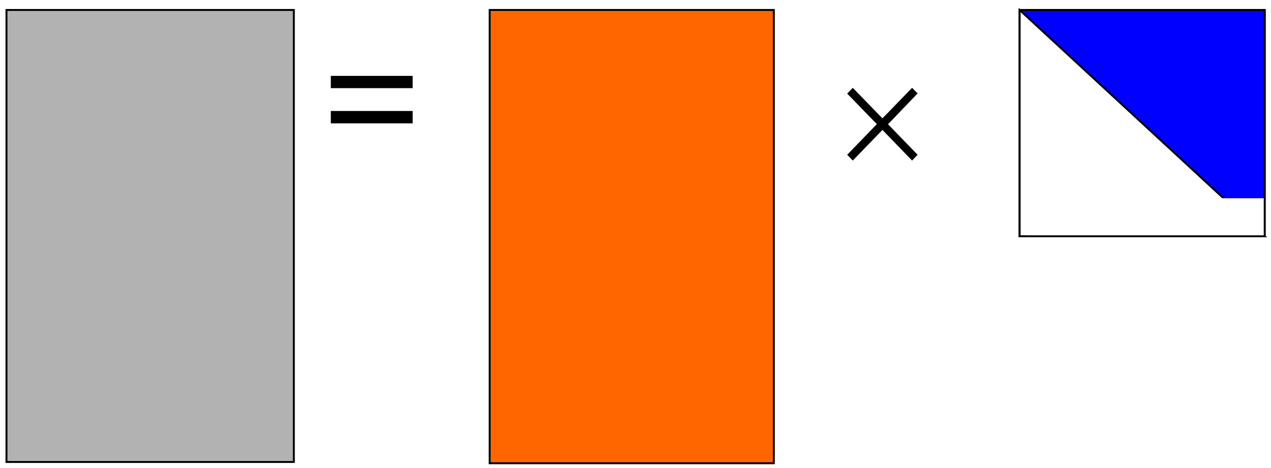
if $rank(A)=r$ then $rank\left(R_n\right) = r$

Quanto detto prima vuol dire che le ultime n-r righe di Rn sono vettori con tutti zero.

$$A = QR$$

if $rank(A) = rank(R_n) = r$

last n-r rows of R_n are zero vectors



Quindi in conseguenza a ciò possiamo riscrivere A = Qr Rr dove Qr è fomrata dalle prime r colonne di Q e Rr è fomrata dalle prime r righe di R.

$$A = QR$$

if $rank(A) = rank(R_n) = r$

↓

last n-r rows of R_n are zero vectors

Q_r formed by the first r columns of Q

R_r formed by the first r rows of R

are such that

$$A = Q_r R_r$$

Osservando questa fattorizzazione possiamo intepretare le colonne di A come una combinazione lineare delle colonne di Q.
E' possibile poi definire delle proprietà riferite al rango, in particolare se r è il rango di A allora possiamo dire che il sottoinseme delle prime r colonne di Q è una base ortonormale per il range di A, se r = n allora vuol dire che tutte le colonne lo sono.

$$\boxed{A = QR}$$

the columns of A are a **linear combination** of the columns of Q

if r is the **rank** of A

the subset of the first r columns of Q is an **orthonormal basis** for the **range** of A

Se r è minore di n allora diremo che sempre se r è il rank di A in questo caso l'isneme delle ultime m-r colonne di Q è una base ortonormaper per il complemente ortogonale del range di A, cioè lo spazio nullo della trapsosta di A.

$$A = QR$$

the columns of A are a **linear combination** of the columns of Q

if r is the **rank** of A

The subset of the last m-r columns of Q is an **orthonormal basis** for the **orthogonal complement of the range** of A **(i.e., the null space of A^T)**

Application 1

$$A = QR$$

$$A \in \Re^{n \times n}, x \in \Re^{n}, b \in \Re^{n}$$

Questo è un applicazione didattica. Se supponiamo che A è quadrata e possiamo usare la fattorizzazione per risolvere Ax = b

solving $Ax = b$

case: A is a square matrix

Poniamo QR al posto di A e scriviamo che Rx = y

$$QRx = b$$

posed $Rx = y$

Otteniamo quindi questa forma. e la risoluzione la problema è y = trasposta di Q * b

$$Qy = b$$

$$y = Q^T b$$

$$Rx = y$$

A questo punto noto y possiamo determinare Rx = y che è triangolare
e si risolve per sost all'indietro. Così determino x e ho risolto il problema di partenza.
Ovviamente non ci conviene usare questo metodo per una questione di complessità. E' migliore il metodo classico cioè Fattorizzazione LU.

Application 2

$$A = QR$$

$$A \in \Re^{m \times n},\ m > n$$

$$x \in \Re^{n}, b \in \Re^{m}$$

Supponiamo di dover risolvere il sistema sovradeterminato, in questo caso sappiamo che il problema nei termini dei minimi quadrati o Latest Square problem è un problema di minimizzare rispetto a x la norma 2 di quella roba. Le nostre ipotesi sono queste in alto a dx della slide e full rank cioè il rank di A = n. In questa sitauzione sappiamo già risolverlo moltiplicando entrambi i membri per la trasposta di A quindi x = (At*A)^-1 * At b <- sistema di equazioni normali.

solving $Ax=b$

case: overdetermined system

LS full rank problem

$$\min_x \|Ax-b\|_2$$

$$rank(A)=n$$

Un altra tenica di risoluzione passa proprio per la fattorizzazione QR, riscriviamo il problema ricordandoci della proprietà cioè la norma non cambia se moltiplico il residuo (Ax-b) per una matrice ortoganle, in questo caso scegliamo Qt. Quello che otteniamo come ultimo termine è semplicemente quello che otteniamo facendo il prodotto.

$$\min_x \|Ax-b\|_2 = \min_x \|Q^T(Ax-b)\|_2 = \min_x \boxed{\|Q^T Ax - Q^T b\|_2}$$

Il termine in verde è un vettore ottenuto come sottrazione di due vettori. Ora guardiamo con attenzione prima il primo e poi il secondo. Qt * A lo possiamo scrivere come R quindi otteniamo Rx, la figura mostra com'è fatto R cioè solito vettore con due blocchi.

$$Q^T Ax = Rx = \begin{pmatrix} R_n \\ 0 \end{pmatrix} x = \begin{pmatrix} R_n x \\ 0 \end{pmatrix}$$

$$Q^T b = \begin{pmatrix} c \\ d \end{pmatrix} \begin{array}{c} n \\ m-n \end{array}$$

Il secondo termine invece è un vettore e lo spezzo due pezzi, le prime n componenti le chiamo c e le m-n le chiamo d.

Application 2

$$A = QR$$

$A \in \Re^{m \times n},\ m > n$

$x \in \Re^{n}, b \in \Re^{m}$

qui dice che stando a quanto appena detto fondamentalmente
sappiamo che la diff tra quei due vettori visti prima da il vettore in cerchiato di nero. Ne voglio calcolare
la norma 2 (rosso), quindi la norma due di quel vettore è na somma della norma due del blocco superiore
e la norma due del blocco inferiore.

solving $Ax=b$

case: overdetermined system

LS full rank problem

$$\min_x \|Ax - b\|_2$$

$rank(A)=n$

$$Q^T A x = \begin{pmatrix} R_n x \\ 0 \end{pmatrix} \qquad Q^T b = \begin{pmatrix} c \\ d \end{pmatrix} \begin{array}{c} n \\ m-n \end{array}$$

$$Q^T A x - Q^T b = \begin{pmatrix} R_n x - c \\ -d \end{pmatrix}$$

$$\|Q^T A x - Q^T b\|_2 = \left\| \begin{pmatrix} R_n x - c \\ -d \end{pmatrix} \right\|_2$$

$$\|Q^T A x - Q^T b\|_2^2 = \|R_n x - c\|_2^2 + \|-d\|_2^2$$

Application 2

$$A = QR$$

$$A \in \Re^{m \times n},\ m > n$$

$$x \in \Re^{n},\ b \in \Re^{m}$$

Determinare il minimo è = a determinare il min del quadrato della stessa quantità. Questo lo dice un teorema di analisi.
Poiché la norma due è la radice quadrata di una somma di quadrati di compoenenti elevando al quadrato eliminiamo la radice
e rimane solo la somma di quadrati. Ed è quello che faccio qui riscrivendo il probl con i quadrati

case: overdetermined system

solving $Ax=b$

LS full rank problem

$$\min_x \|Ax-b\|_2$$

$rank(A)=n$

$$\|Q^T Ax - Q^T b\|_2^2 = \|R_n x - c\|_2^2 + \|-d\|_2^2$$

Qui abbiamo riscritto il problema tenendo conto che questa differenza da il vettore cerchiato di rosso della slide precedente e che la norma di quel vettore è la somma dellle norme al quadrato delle singole componenti. Ora noi dobbiamo minimizzare questo ma in realtà minimizziamo solo il primo termine.

$$\min_x \|Ax-b\|_2^2 = \min_x \|Q^T Ax - Q^T b\|_2^2 = \min_x \|R_n x - c\|_2^2 + \|d\|_2^2$$

perché d non è un termine con la x.
Lo spezzamento della norma come la somma delle norme è fattibile granzei al fatto che abbiamo elevato al quadrato.
Per minimizzare il primo elemento: quello è il residuo di un sistema quadrato in particolare questo sistema (sottolineato)

$$R_n x_{LS} = c$$

$$\rho_{LS}^2 = \|d\|_2^2$$

Rn è la matr quadrata triang superiore non singolare e quindi risolvo il probl rispetto alla solzuione xls (ls perché usiamo i minimi quadrati.)
Ro^2 è il quadrato della lunghezza ro del vettore che è = alla norma quadrata di d.

Qui mostra una sorta di strada alternativa usando la fattorizzazione QR ridotta.

Application 2

$$A = Q_n R_n$$

reduced QR factorization

$$A \in \Re^{m\times n},\ m > n$$

$$x \in \Re^n,\ b \in \Re^m$$

LS full rank problem

alternative way

$$\min_x \|Ax-b\|_2$$

$$rank(A)=n$$

Quindi la soluzione xls (x è la solu del problema dei minimi quadrati) è quel vettore tale che Axls è la proiezione ortogonale di b sul range di A. La formula di sotto esprime questo (la prima formula)

the solution x_{LS} is such that Ax_{LS} is the orthogonal projection of b onto the $range\,(A)$

$$Ax_{LS} = \boxed{Q_n Q_n^T} b$$

orthogonal projector

Ora moltiplico ambo i membri per Qt, quelli cerchiati corrispondono alla matrice identità. Quindi ottengo la formula finale.
dove Qnt * b = c

$$Q^T\ {}^{*}Q_n R_n x_{LS} = Q^T Q_n Q_n^T b$$

$$\boxed{R_n x_{LS} = Q_n^T b}$$

* Qui esprime A in termine di Qn Rn

Qui ricorda quali sono i passi dell'algoritmo dei minimi quadrati per calcolare la fattorizzazione QR. (a titolo di esempio prende la forma ridotta.)

Application 2

$$A = Q_n R_n$$

reduced QR factorization

$$A \in \Re^{m\times n},\ m > n$$

$$x \in \Re^{n}, b \in \Re^{m}$$

LS full rank problem

$$\min_x \|Ax - b\|_2$$

1) Calcolo la fattorizzazione QR ridotta di A
2) Calcola il vettore C
3) Risolve il sistema triangolare superiore.

Least Squares Algorithm via QR

1. Compute the reduced QR factorization of A
2. Compute the vector $c = Q_n^T b$
3. Solve the upper triangular system

$$\boxed{R_n x_{LS} = c}$$

N.B la fattorizzazione QR da molte info sullo spazio delle colonne, ma poco sulle righe. Infatti la fattorizzazione più usata è un altra che da molte info anche sulle righe oltre che sulle colonne.

Quanto sto per dire riguarda come posso determinare Xls a partire dalla formula del problema da risolvere al punto 3.

Application 2

$$A = Q_n R_n$$

$$A \in \Re^{m \times n}, \; m > n$$

Poiché abbiamo detto che il rango di Rn è = al rango di A che è = n e Rn è quadrata di rango n allora è invertibile.
Posso quindi moltiplicare ambo i membri per Rn^-1 e ottengo il primo termine.
L'inversa * la trasposta da la pseudo inversa.

reduced QR factorization

$$x \in \Re^n, b \in \Re^m$$

LS full rank problem

$$\min_x \|Ax - b\|_2$$

$$R_n x_{LS} = Q_n^T b$$

pseudoinverse via QR

$$x_{LS} = \boxed{R_n^{-1} Q_n^T} b = A^+ b$$

$$A^+ = R_n^{-1} Q_n^T$$

fondamentalmente calcolando Xls ho la soluione del sistema triangolare superiore.

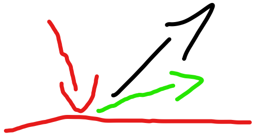
Ora vediamo un algoritmo per la fattorizzazione QR.

Il rifletotre di Householder è un algoritmo che calcola la fattorizzazione QR, l'idea si basa su delle matrici ortogonali che si chiamano riflettori di Householder. L'idea è che costriamo una matrice ortogonale, a partire da un vettore, ruotando questo vettore in mod tale che lo porta su unasse, cioè azzera tutte le componenti tranne una. Si costruiscono nel modo indicato nella prima forma.

Algorithm for the QR factorization

Si chiamano riflettori perché agiscono come se fosse uno specchio, cioè creano una riflessione che ha direzione u (per intenderci u è la linea verde), questo specchio riflette la direzione del vettore.

Householder reflector



$$H = I - \frac{2}{\|u\|_2^2} u u^T$$

H is a **symmetric orthogonal** matrix

$$Hx = x - \frac{2}{\|u\|_2^2} u u^T x$$

$$Hx = x - \tau u$$

Quindi dato x che è un vettore il prodotto di H*x mi porta a questa fornula. Questo vettore H*x è il vettore riflesso rispetto ad u. Il secondo termine è un multiplo di u e lui chiama tau. (quella lettera greca.)

Questo è un po la situazione: abbiamo la direzione u dato lo specchio, arriviamo con x ci calcoliamo h facciamo h*x e otteniamo il vettore rosso.
Questo vale in generale, ma come lo usa Housholder? Lavora al contrario, se tu arrivi con un x puoi costruire uno specchio tale che riflette x e lo manda su un asse?

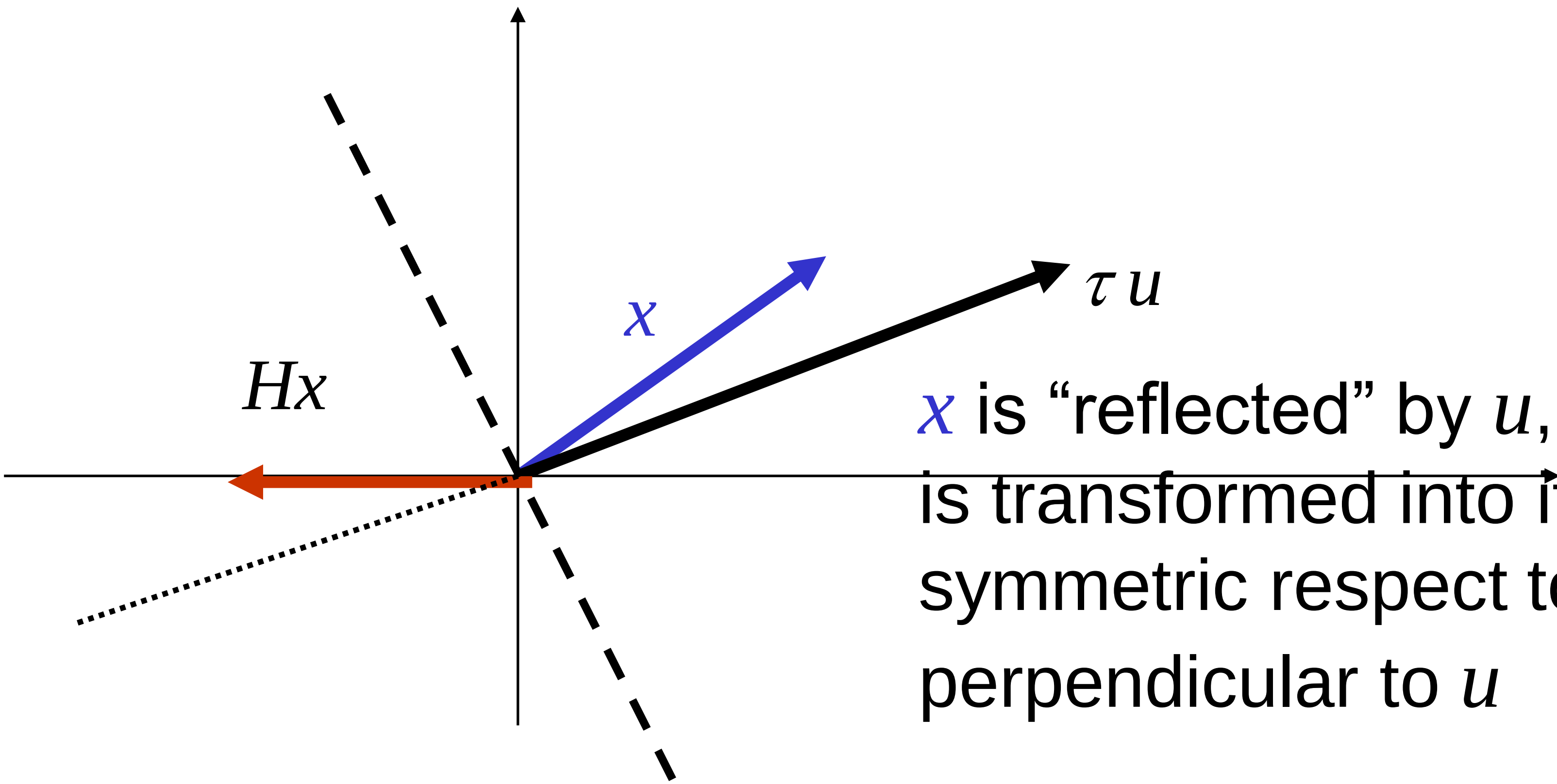
Householder reflector

$$Hx = x - \tau u$$

Per fare questo ti costruisci u che è lo specchio opportuno, da u ti costruisci h fai h*x e risolvi.

$$Hx = x - \frac{2}{\|u\|_2^2} u u^T x$$

the vector x is projected onto u, multiplied by 2 and subtracted from x



x is “reflected” by u, i.e. is transformed into its symmetric respect to the perpendicular to u

Il riflettore è quindi utile per azzerare alcune componenti di un vettore x tramite un opprotuna riflessione (quindi ci dice che possiamo usare i riflettori appunto per il nostro scopo cioè azzerare la componente che vogliamo).

Householder reflector

Per farlo abbiamo x e decidiamo la k-esima componente che non vogliamo azzerare. Allora prendiamo x e sommiamo il vettore ek cioè il k-esimo vettore della base canonica, cioè tutti 0 tranne nella k-esima, lo moltiplichiamo per la norma 2 di x e otteniamo quindi u.

is useful for "zeroing" some components of a vector x by means of a suitable reflection

Avuto u calcoliamo H facciamo h*x che ha tutte le ocmponenti nulle tranne la k-esima.

$$u = x + \|x\|_2 \, e_k$$

the reflector "zeroes" all the components of x **except** the k-th

Hx has all components equal to zero, **except** the k-th

Questo è un esempio dove non vogliamo azzerare la prima. Calcoliamo quindi prima u.

Exercise: given a vector x build the reflector that zeroes all the components **except the first**

$$x = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

first step: compute the vector u

$$u = x + \|x\|_2 \, e_1$$

$$u = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} + 3 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$u = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$$

Ora calcoliamo il riflettore H e facciamo h*x, quello che otteniamo è appunto il vettore hx

second step: compute the reflector H

$$x = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} \qquad u = \begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$$

$$Hx = x - \frac{2}{\|u\|_2^2} uu^T x$$

$$H = I - \frac{2}{\|u\|_2^2} uu^T$$

Hx has all components equal to zero, **except** the first one

Se interessano i calcoli i link è il video e andare al minuto 18 circa
https://web.microsoftstream.com/video/58808ba1-ea67-4701-9f41-ef9817904784

$$Hx = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} - 2\frac{15}{30}\begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$$

$$Hx = \begin{pmatrix} -3 \\ 0 \\ 0 \end{pmatrix}$$

Ora abbiamo il riflettore, come lo usiamo? In pratica una seuquenza di riflessioni di Householder viene applicata alle colonne di A per produrre (per colonne) la matrice R. La j-esima riflessione azzera la prte sotto la diagonale della j-esima colonna e produce la j-esima colonna di R. R è un amtrice triangolare superiore.

Algorithm for the QR factorization

$$\boxed{H_n \cdots H_2 H_1 A = R}$$

a sequence of **Householder reflections** is applied to the columns of A to get (columnwise) the matrix R

the j-th reflection zeroes the part below the diagonal of the j-th column and gives the j-th column of R

Qui capiamo meglio come sono le H, H1 azzera la prima colonna tranne il primo elemento, H2 azzera la seconda colonna tranne i primi due lementi etc.

Algorithm for the QR factorization

$$H_n \cdots H_2 H_1 A = R$$

$$\underset{A}{\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix}} \xrightarrow{H_1} \underset{H_1 A}{\begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix}} \xrightarrow{H_2} \underset{H_2 H_1 A}{\begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & 0 & \times \\ & 0 & \times \\ & 0 & \times \end{pmatrix}} \xrightarrow{H_3} \underset{H_3 H_2 H_1 A}{\begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \\ & & 0 \\ & & 0 \end{pmatrix}}$$

Alla fine otteniamo R triangolare superiore

“zero” the first column of A, except the first element

Per azzerare la prima colonna, chiamiamola a allora calcoliamo u, h e possiamo fare H1 * A

$$A \quad \begin{pmatrix} \boxed{\begin{matrix} \times \\ \times \\ \times \\ \times \\ \times \end{matrix}} & \begin{matrix} \times \\ \times \\ \times \\ \times \\ \times \end{matrix} & \begin{matrix} \times \\ \times \\ \times \\ \times \\ \times \end{matrix} \end{pmatrix} \xrightarrow{H_1} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix} \quad H_1 A$$

$$a$$

$$u = a + \|a\|_2 \, e_1$$

$$H_1 = I - \frac{2}{\|u\|_2^2} u u^T$$

“zero” the second column of A, **except** the first and second elements

Come visto prima ma ora lo dobbiamo fare per la seconda colonna tranne i primi due lementi.

$$\begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix}$$

b

Usiamo sempre e1 perché consideriamo ad ogni passo un vettore ceh ha sempre una lunghezza inferiore rispetto al vettore del passo precedente cioè si prende il vettore a partire dalla riga successiva a quella non azzerata al passo precedente.

$$u = b + \|b\|_2 \, e_1$$

$$\mathrm{H}_2 = I - \frac{2}{\|u\|_2^2} u u^T$$

Qui ci costruiamo una matrice H2 che dato TUTTO A lascia inalterato tutto tranne b.

Questa prima riga ha 1 numero e il vettoe nullo di n-1 componenti
La seocnda riga ha il vettore nullo di n-1 componenti (è un vett colonna). Poi c'è h2

$$H_2 = \begin{pmatrix} 1 \; 1 & 0 \; n \\ 0 & \mathrm{H}_2 \end{pmatrix}$$

n

Si procede poi in questo modo con tutti gli n H

n-1 x n-1

Ricordiamo che il prodotto di matrici ortogonali è una matrice ortogonale.
Quanto vito è quello che fa la fattorizzazione QR.

Algorithm for the QR factorization

$$H_n \cdots H_2 H_1 A = R$$

$$Q = \left(H_n \cdots H_2 H_1\right)^T$$

the **product** of **orthogonal** matrices is an **orthogonal** matrix

Questa è la complessità di tempo e l'indicie di condizionamento è = all'indicie di condizionamento di A

Algorithm for the QR factorization

time complexity

$$T(m,n) = n^2(m - n/3)$$

accuracy (relative error) proportional to

$$\kappa_2(A)$$

$$\kappa_2(A) = \|A\|_2 \|A^+\|_2$$

Questo è il confrnot tra la il problema LS via fattorizzazione QR e risoluzione del sisrtema triangola e sotto abbiamo LS via costruzione e risoluzione del sistema delle equazioni normali. Qui si vede che via fattorizzazione costa di più ma ha un indice di condizonamento minore e quindi la si preferisce.

LS through the QR factorization and solving the triangular system

$$T(m,n) = n^2(m - n/3) + n^2/2 + nm$$

conditioning: $\kappa_2(A)$

LS through the system of normal equations

$$T(m,n) = (n^2/2)(m + n/3)$$

conditioning: $\kappa_2(A)^2$