9. EIGENVALUES: important in engineering applications such as structural engineering (cf. example of bridge). Associated with resonances of system, dynamical systems, stability ...

§9.1 Linear Algebra preliminaries

Def (eigenvalue, eigenvector) Let $A \in \mathbb{C}^{n\times n}$. A non-zero vector $x \in \mathbb{C}^n$ is an eigenvector of A and λ its corresponding eigenvalue if

$$Ax = \lambda x$$

Def (Linear independence) Let $\{v_1, v_2, \dots, v_n\}$ be a set of vectors. The set is *linearly independent* if:

$$\sum_{i=1}^{k} \alpha_k v_k = 0 \iff \alpha_i = 0,\ i = 1..k.$$

Theorem (Basis): Let $\{v_1, \dots, v_n\}$ be a set of n lin. indep vectors of $\mathbb{R}^n$. Then:

$$\forall x \in \mathbb{R}^n \ \exists \beta_i \text{ s.t. } \underline{x} = \sum_{i=1}^{n} \beta_i v_i$$

proof: $A = [v_1 | v_2 | \cdots | v_n] \in \mathbb{R}^{n\times n}$

$\{v_1, v_2, \dots, v_n\}$ lin indep $\iff \left(A\underline{\alpha} = \sum_{i=1}^{n} \alpha_i v_i = 0 \iff \underline{\alpha} = 0 \right)$

$\iff$ A is invertible (or nonsingular)

Then $\underline{x} = A\underline{\beta}$, where $\underline{\beta} = A^{-1}x$

Def (Dimension) The dimension of a vector subspace is the max. number of linearly independent vectors spanning the set.

Eigenvalue decomposition

For a square matrix $A \in \mathbb{C}^{n\times n}$ the eigenvalue decomposition is a factorization:

$$A = X \Lambda X^{-1} \qquad \text{(note: is not always possible!)}$$

$$\Leftrightarrow \quad AX = X\Lambda$$

$$\Leftrightarrow \quad A\,[x_1|x_2|\cdots|x_n] = [x_1|x_2|\cdots|x_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Def (characteristic polynomial) The characteristic poly of some $A \in \mathbb{C}^{m\times m}$ is

$$p_A(z) = \det(zI - A) = \text{m degree poly.}$$

Theorem: λ is an eigenvalue of $A \Leftrightarrow p_A(\lambda) = 0$

proof: λ eigenvalue $\Leftrightarrow \exists\, x \neq 0$ s.t. $\lambda x - Ax = 0$

$\Leftrightarrow \lambda I - A$ is singular

$\Leftrightarrow \det(\lambda I - A) = 0$.

This theorem means that even if $A \in \mathbb{R}^n$, spectrum maybe complex. (physically complex eigenvalues give oscillatory behaviours).

Def (Algebraic Multiplicity)

Fundamental theorem of algebra $\Rightarrow p_A(z) = (z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_m)$

The algebraic multiplicity is the multiplicity of λ as a root of p.(ie how many times $z-\lambda$ appears in $p_A(z)$.

Def (Geometric Multiplicity)

Let λ be an eigenvalue of A.

$E_\lambda = \{x \mid Ax = \lambda x\}$ = vector space

= eigenspace or invariant subspace of A. $[AE_\lambda \subset E_\lambda]$

geometric multiplicity of λ = $\dim E_\lambda$ = $\dim \text{null}(\lambda I - A)$.

Theorem: Let $A \in \mathbb{C}^{m \times m}$, A has m eigenvalues counted with algebraic multiplicity. (easy corollary of fundamental theorem of algebra)

Similarity transformations If $X \in \mathbb{C}^m$ is non-singular, then the map $A \to X^{-1}AX$ is a similarity transformation.

A and B are similar if there is a similarity transf s.t.

$A = X^{-1}BX$. ($\sim$ change of basis)

Theorem (similar matrices) if X is non-singular then A and $X^{-1}AX$ have the same $p_A(z)$, eigenvalues and algebraic / geom. multiplicities.

Proof:

$$p_{X^{-1}AX}(z) = \det(zI - X^{-1}AX) = \det(X^{-1}(zI - A)X)$$

$$= \det(zI - A) = p(z)$$

proves same eigenvalues and algebraic multiplicity

Also if E_λ is an eigenspace for A then $X^{-1}E_\lambda$ is an eigenspace for $X^{-1}AX$ as well.

Theorem. algebraic multiplicity $\geq$ geom. multiplicity.

Matrices for which geom. multiplicity $<$ algebraic mult are called *defective*.

Example:

$$A = \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{bmatrix}$$

$$p_A(\lambda) = (\lambda-2)^3$$

$$p_B(\lambda) = (\lambda-2)^3$$

$\Rightarrow$ same eigenvalues and algebraic mult.

$(\lambda-2)$ (alg. mult of λ is 3)

Three lin indep eigenvectors of A are: $\Rightarrow$ geom mult is 3

$\underline{e}_1, \underline{e}_2, \underline{e}_3$.

For B the only eigenvector possible is $\underline{e}_1$. $\Rightarrow$ geom mult is 1.

Theorem A non defective $\Leftrightarrow$ $A = X \Lambda X^{-1}$ (diagonalizable)

Def A matrix Q is *unitary* if $Q^* Q = I$.

The columns of a unitary matrix are orthogonal.

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

theorem. Let $A \in \mathbb{C}^{n \times n}$.

$A = A^* \Rightarrow A = Q \Lambda Q^*$, Q unitary, Λ diagonal

(Hermitian) and eigenvalues are *real*.

Theorem: A is unitarily diagonalizable iff it is normal,

i.e. $A^* A = A A^*$.

Schur Factorization (behind famous QR algo)

$$A = QTQ^* \qquad T = (\nabla) \qquad Q = \text{unitary}$$

This is an eigenvalue revealing factorization because A and T are similar $\Rightarrow$ eigenvalues of A appear on diagonal of T.

Theorem: $\forall A \in \mathbb{C}^{n\times n}$ admits a Schur factorization

proof: By induction on size of A.

$n = 1$ trivial

Assume any matrix of size n has a Schur factorization.

Consider $A \in \mathbb{C}^{n+1 \times n+1}$, Let x be an eigenvector of A w/ eigenvalue λ.

$U = \left[\frac{x}{\|x\|}, \underbrace{\quad}_{\text{complete basis}}\right]$ = unitary.

Then: $U^* A U = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix}$. By induction hyp:

$C = VTV^*$, $T = (\nabla)$ V = unitary

Let $Q = U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$

Then $Q^* A Q = \begin{bmatrix} 1 & 0 \\ 0 & V^* \end{bmatrix} U^* A U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$

$$= \begin{bmatrix} 1 & 0 \\ 0 & V^* \end{bmatrix} \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$$

$$= \begin{bmatrix} \lambda & B \\ 0 & V^*C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$$

$$= \begin{bmatrix} \lambda & BV \\ 0 & T \end{bmatrix} = (\nabla). \qquad \text{QED}$$

This is an existence proof! it does not tell us how to compute Schur fact'n!

Note: all eigenvalue algorithms need to be iterative.

The reason is that finding eigenvalues $\Leftrightarrow$ finding roots of a poly.

($\Rightarrow$) characteristic poly

($\Leftarrow$) companion matrix of poly:

$$p(z) = z^m + a_{m-1} z^{m-1} + \cdots + a_1 z + a_0$$

$$A = \begin{bmatrix} 0 & & & -a_0 \\ 1 & \ddots & & -a_1 \\ & \ddots & 0 & \vdots \\ & & 1 & -a_{m-1} \end{bmatrix}$$

Can verify that: if z is a root of p:
$v = (1, z, z^2, \cdots, z^{m-1})$ is a left eigenvector of A with eigenvalue z:

$$vA = zv$$

And it is a well known fact that no general formula exists for computing the roots of a polynomial of degree ≥ 5, so we can only hope to approximate them through an iterative process.

§ 9.2 Power Method

Power method (or iteration) algorithm

$v^{(0)}$ = some vector with $\|v^{(0)}\|_\infty = 1 = v_{p_0}^{(0)}$

for $k = 1, 2, \cdots$

$w^{(k)} = A v^{(k-1)}$

$v^{(k)} = w^{(k)} / \|w^{(k)}\|_\infty$

$\lambda^{(k)} = w_p$, where $p =$ smallest index s.t. $|v_p^{(k-1)}| = \|v^{(k-1)}\|_\infty = 1$

The iteration stops when two successive iterates become close to within some tolerance.

The power method finds the largest eigenvalue (in magnitude) and its associated eigenvector.

Here is a sketch of the convergence proof.
For simplicity we assume A is diagonalizable. We also assume that the largest eigenvalue (in magnitude) of A is simple.
Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of A ordered s.t.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

Let $q_1, q_2, q_3, \ldots, q_n$ the associated eigenvectors (lin indep)

Then given $v^{(0)}$ (The starting vector) $\exists \beta_i$ s.t.

$$v^{(0)} = \sum_{i=1}^{n} \beta_i q_i.$$

$$Av^{(0)} = \sum_{i=1}^{n} \beta_i \lambda_i q_i$$

$$A^k v^{(0)} = \sum_{i=1}^{n} \beta_i \lambda_i^k q_i = \lambda_1^k \beta_1 q_1 + \lambda_1^k \sum_{i=2}^{n} \beta_i \left(\frac{\lambda_i}{\lambda_1}\right)^k q_i$$

It's not hard to see that when we apply the power method, we have:

$$v^{(k)} = \frac{A^k v^{(0)}}{\|A^k v^{(0)}\|} = \frac{\lambda_1^k \left(\beta_1 q_1 + \sum_{i=1}^{2} \beta_i \left(\frac{\lambda_i}{\lambda_1}\right)^k q_i\right.}{|\lambda_1|^k \left\|\beta_1 q_1 + \sum_{i=2}^{n} \beta_i \left(\frac{\lambda_i}{\lambda_1}\right)^k q_i\right\|_\infty}$$

Now since $\lim_{k \to \infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0, \; i = 2, \ldots, n,$

we have $v^{(k)} \to$ a multiple of q_1.
and the rate of convergence is linear with ratio $\left|\frac{\lambda_2}{\lambda_1}\right|$.

Drawback of power method: (for both vector & eigenvalue)

- nobody guarantees initial vol has component in direction of largest eigenvector
- computes only largest eigenvalue & eigenvector
  - (we will see how to fix this)
- convergence rate can be very slow if $\lambda_1 \sim \lambda_2$.

Advantages

- Algorithm works even when A is nondiagonalizable.
- simplicity.
- Uses only matrix vector prod.

It's possible to get faster linear convergence of the eigenvalues in the case where A is real & symmetric:

Power method (symmetric case)

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$

for $k = 1, 2, \dots$

$$w^{(k)} = A v^{(k-1)}$$
$$v^{(k)} = w^{(k)} / \|w^{(k)}\|$$
$$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$$

faster than in general case.

In this case we can expect: $|\lambda^{(k)} - \lambda_1| = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)$

and $|v^{(k)} - q_1| = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{k}\right)$

↑ depends on sign of multiple of q_1 we converge to.

The reason behind the update in $\lambda^{(k)}$ is the so called Rayleigh quotient:

$$r(x) = \frac{x^T A x}{x^T x}.$$

For real symmetric matrices it can be shown that:

$$\lambda_{min} \leq r(x) \leq \lambda_{max}$$

So if $v^{(k)}$ is a good approx to q_1, then:

$$v^{(k)T} A v^{(k)} = \frac{v^{(k)T} A v^{(k)}}{v^{(k)T} v^{(k)}} \approx \frac{q_1^T A q_1}{q_1^T q_1} = \lambda_1 \frac{q_1^T q_1}{q_1^T q_1}$$

1.

The power method can be modified to zoom in and find the closest eigenvalue to some μ. The idea is that:

if v is an eigenvector of A with eigenvalue λ

then v ——— $(A-\mu I)^{-1}$ ——— $(\lambda-\mu)^{-1}$

So if we apply power method to matrix $(A-\mu I)^{-1}$ the method will converge to the largest eigenvalue $|\lambda_j - \mu|^{-1}$, a.k.a the closest to μ. This method is:

Inverse iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\|=1$

for $k=1,2,\ldots$

| solve $(A-\mu I)w^{(k)} = v^{(k-1)}$ for $w^{(k)}$ (= apply $(A-\mu I)^{-1}$)
| $v^{(k)} = w^{(k)}/\|w^{(k)}\|$
| $\lambda^{(k)} = v^{(k)T} A v^{(k)}$

(here we wrote the version for symmetric matrices, but the same modification can be carried out in the original method)

Convergence is still linear, but once we control μ, we can control the convergence rate.

Suppose λ_{J_1} is the closest eigenvalue of A to μ

λ_{J_2} —— next to closest ——

$$|\lambda_{J_1} - \mu| < |\lambda_{J_2} - \mu| \leq |\lambda_j - \mu| \quad \text{for } j \neq J_1$$

then:

$$|\lambda^{(k)} - \lambda_{J_1}| = O\left(\left|\frac{\mu - \lambda_{J_1}}{\mu - \lambda_{J_2}}\right|^{2k}\right)$$

and

$$\| v^{(k)} - (\pm q_1) \| = O\left(\left| \frac{\mu - \lambda_{J_1}}{\mu - \lambda_{J_2}} \right|^k \right)$$

So why not use variable shifts μ that get closer to the eigenvalue of interest?

This gives the following algo:

Rayleigh quotient iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$

$\lambda^{(0)} = v^{(0)T} A v^{(0)}$ = Rayleigh quot.

for $k = 1, 2, \ldots$

- solve $(A - \lambda^{(k-1)} I) w^{(k)} = v^{(k-1)}$ (apply $(A - \lambda^{(k-1)} I)^{-1}$)
- $v^{(k)} = w^{(k)} / \| w^{(k)} \|$
- $\lambda^{(k)} = v^{(k)T} A v^{(k)}$ (update guess with Rayleigh quot)

Convergence of RQI is one of the rare cases where one gets cubic convergence!

If the start vector $v^{(0)}$ is sufficiently close to the eigenvector q_J:

$$\| v^{(k+1)} - (\pm q_J) \| = O\left(\| v^{(k)} - (\pm) q_J \|^3 \right)$$

$$| \lambda^{(k+1)} - \lambda_J | = O\left(| \lambda^{(k)} - \lambda_J |^3 \right)$$

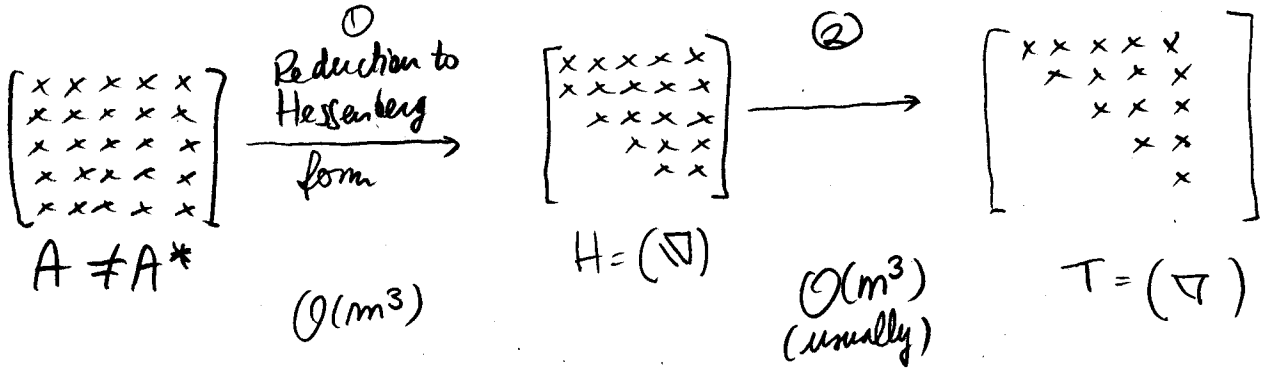
§9.3 Reduction to Hessenberg or tridiagonal form

Motivation: We would like to somehow compute Schur form of A by successive multiplications by orthogonal matrices: ($AQ = QT$)

$$T_j = Q_j^* Q_{j-1}^* \cdots Q_2^* Q_1^* A Q_1 Q_2 \cdots Q_j$$

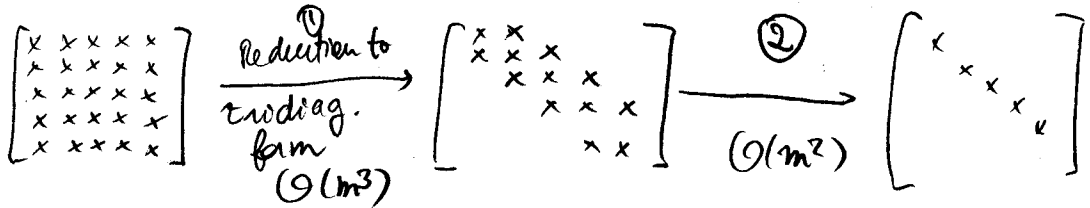
so that $T_j \to T = (\triangledown)$ as $j \to \infty$.

Usually this is split into 2 phases:



② could run forever, however convergence is usually obtained in m steps and each step in ② takes $O(m^2)$ flops.

If $A = A^*$ (A Hermitian)



② each step takes m flops (if no eigenvectors are needed)
$\sim m$ steps total = $O(m^2$ flops)

In Burden & Faires: ① = §9.3 Householder's Method
② = §9.4 The QR algorithm

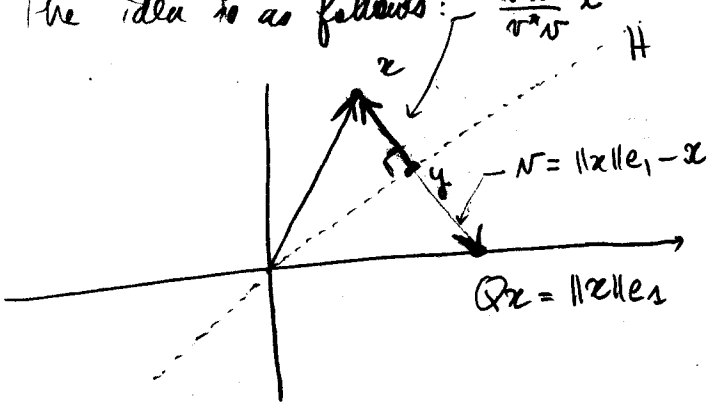
Phase ① can be made using orthogonal transformations called Householder reflectors (which are extremely useful in linear algebra ...)

Householder reflectors are orthogonal transformations that zero out part of a vector.

Given $x = \begin{bmatrix} x \\ x \\ x \\ \vdots \\ x \end{bmatrix}$ we can build a Q with $Q^*Q = I$ s.t.

$$Qx = \begin{bmatrix} \|x\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1.$$

The idea is as follows:



The $\perp$ proj of x onto v is:

$$\frac{vv^*}{v^*v}x$$

Thus:

$$y = x - \frac{vv^*}{v^*v}x$$

and:

$$\boxed{Qx = x - 2\frac{vv^*}{v^*v}x}$$

Another geometric interpretation:

y is projection $\perp$ of x onto H

Qx is reflection of x accross H.

It's amazing that a simple factor of 2 converts a projector (not full rank) into an orthogonal matrix!

$$\boxed{Q = I - 2\frac{vv^*}{v^*v}}$$

Q is orthogonal:

$$Q^*Q = \left(I - 2\frac{vv^*}{v^*v}\right)\left(I - 2\frac{vv^*}{v^*v}\right)$$

$$= I - 4\frac{vv^*}{v^*v} + 4\left(\frac{vv^*}{v^*v}\right)\left(\frac{vv^*}{v^*v}\right) = I.$$

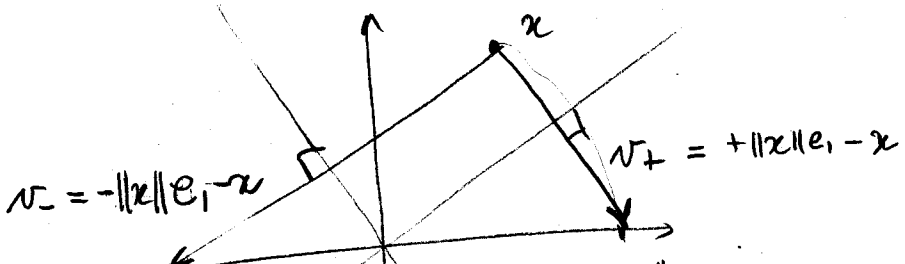
More generally we can introduce 0 anywhere in the vector.
A useful pattern is: $\overset{m}{\hookrightarrow}$

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & Q \end{bmatrix}, \text{ where } Q \text{ is a Householder refl.}$$

Q_k will leave untouched the first m entries:

$$x = \begin{bmatrix} x \\ x \\ x \\ x \\ x \\ x \end{bmatrix} \begin{matrix} \} m \\ \} m-m \end{matrix} \longrightarrow Q_k x = \begin{bmatrix} x \\ x \\ x \\ x \\ 0 \\ \vdots \\ 0 \end{bmatrix} \} \text{ unchanged}$$

There is a slight detail w.r.t. HH reflectors: we could have chosen Q s.t. $Qx = -\|x\| e_1$, so which HH refl is better:
$Q_+ x = +\|x\| e_1$ or the former?



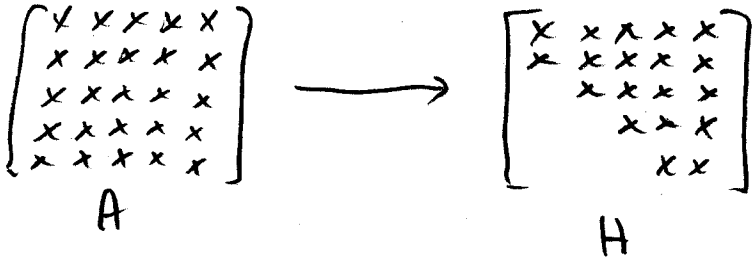
To avoid numerical cancellation in first component one usually takes $v = -\text{sgn}(x_1)\|x\| e_1 - x$

But since $Q = I - 2\frac{vv^*}{v^*v}$ we can eliminate the $-$ sign:

$$\boxed{v = \text{sgn}(x_1)\|x\| e_1 - x}.$$

Note
- each application of HH refl Q (or Q^*) is $O(m^2)$.
- we don't need to store Q, only v. Therefore successive applications of different HH refl $Q_1 Q_2 \dots Q_j$ can be stored with j vectors only.

Back to original problem:



how do we introduce zeros?

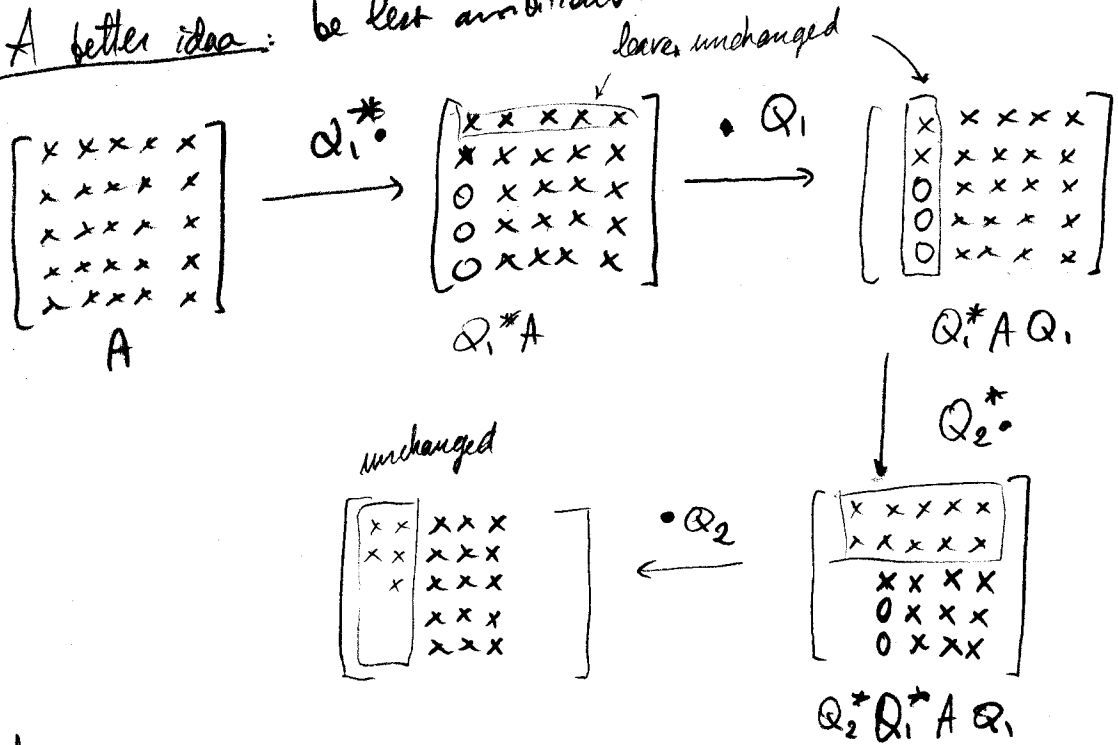
A first idea:

$$Q_1^* A = \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \xrightarrow{\times Q_1} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} = Q_1^* A Q_1$$

each col is lin comb of cols of $Q_1^* A$ so zeros disappear.

does not work!

A better idea: be less ambitious:



etc...

at the end we get (after $n-2$ steps)

$$H = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{bmatrix} = Q_{n-2}^* Q_{n-1}^* \cdots Q_1^* A Q_1 Q_2 \cdots Q_{n-2}$$

Householder reduction to Hessenberg form

for $k = 1 .. n-2$

$x = A(k+1:m, k)$

$v_k = \text{sign}(x(1)) \|x\|_2 e_1 + x$

$v_k = v_k / \|v_k\|_2$

$A(k+1:m, k:m) = A(k+1:m, k:m) - 2 v_k (v_k^* A(k+1:m, k:m))$

$A(1:m, k+1:m) = A(1:n, k+1:n) - 2 (A(1:n, k+1:n) v_k) v_k^*$

When $A = A^*$ Hermitian, we can do similar procedure -

$Q_1^* A Q_1$ will also be symmetric so any zeros we introduced in columns of A will also appear as zeros in corresp. rows.

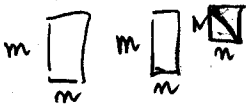
Applying HH ref l on both sides has exactly same cost, so reduction to tridiagonal form is relatively cheap.

The QR factorization [We assume square real matrices in following]

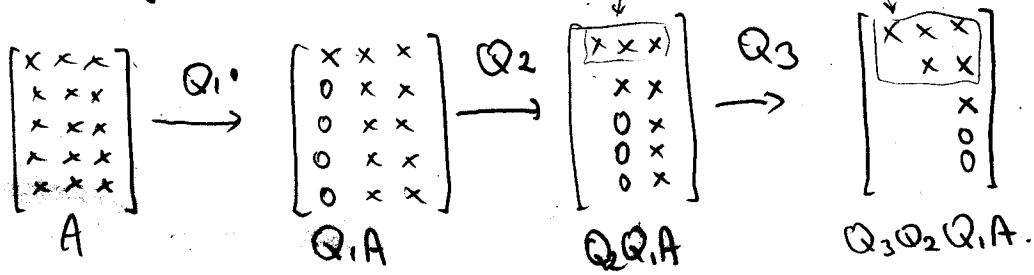
One other important matrix factorization that can be easily obtained with Householder reflectors is the QR fact:

Let $A \in \mathbb{R}^{m \times n}$, then $\exists$ Q unitary and R $\left(\nabla\right)$ s.t. $(Q^* Q = I)$

$$A = Q R$$



Idea: apply Householder reflectors on the left of A to obtain upper triangular matrix:



we will not see details. Keep in mind that in Matlab:

$[Q, R] = qr(A, 0)$ gives such a factorization.

Note: if A is full column rank (i.e. columns are lin indep) then $\text{range}(Q) = \text{range}(A)$.

Unnormalized Simultaneous iteration

Idea: apply power method to many vectors at once.

If $A^k v^{(0)} \to q_1$ (power method) — largest eigenvalue in magnitude

we can expect: $\text{span}\{A^k v_1^{(0)}, A^k v_2^{(0)}, \ldots, A^k v_n^{(0)}\}$

$\to \text{span}\{q_1, q_2, \ldots, q_n\}$

(Simultaneous iteration (= block power method))

largest n eigenvalues in magnitude

In matrix notation:

$$V^{(0)} = \begin{bmatrix} | & | & & | \\ v_1^{(0)} & v_2^{(0)} & \cdots & v_n^{(0)} \\ | & | & & | \end{bmatrix}$$

$$V^{(k)} = A^k V^{(0)} = \begin{bmatrix} | & | & & | \\ v_1^{(k)} & v_2^{(k)} & \cdots & v_n^{(k)} \\ | & | & & | \end{bmatrix}$$

Since all we want is a basis for Range $(V^{(k)})$ (or column space) we use QR fact:

$$\tilde{Q}^{(k)} \tilde{R}^{(k)} = V^{(k)}$$

If we assume:

- $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > |\lambda_{n+1}| \geq \cdots \geq |\lambda_m|$
- all leading principal submatrices of $\tilde{Q}^T V^{(0)}$ are non singular

where $\tilde{Q} = [q_1 \; q_2 \cdots q_n]$ (n columns) $= m$ = matrix with eigenvectors corresp. to $\lambda_1 \cdots \lambda_n$ as columns.

this condition is similar to condition in power method that initial vector has a component in direction of "largest" eigenvector.

Then it can be shown that:

$$\| q_j^{(k)} - \pm q_j \| = O(C^k)$$

where $C = \max_{1 \leq k \leq n} \frac{|\lambda_{k+1}|}{|\lambda_k|}$

(convergence ratio is determined by the eigenvalues that are the closest.)

Simultaneous iteration:

Obviously if we keep multiplying by A, we can have components that quickly become large in magnitude.
~> as in power method we need to renormalize every time we apply A:

pick $\tilde{Q}^{(0)} \in \mathbb{R}^{m \times n}$ with orthonormal cols (e.g. from qr facto of a random matrix)

for $k = 1, 2, \ldots$
$\quad Z = A \tilde{Q}^{(k-1)}$
$\quad \tilde{Q}^{(k)} \tilde{R}^{(k)} = Z$

It is not hard to show that $\tilde{Q}^{(k)}$ is precisely same matrix for which $\tilde{Q}^{(k)} R = A^k \tilde{Q}^{(0)}$ (provided we use same initial guess)

thus convergence study is the same.

QR algorithm (pure version, seldom implemented as is)

$A^{(0)} = A$
for $k = 1, 2, \ldots$
$\quad Q^{(k)} R^{(k)} = A^{(k-1)}$
$\quad A^{(k)} = R^{(k)} Q^{(k)}$

factors are recombined in reverse order.

Equivalence of QR algorithm and Simultaneous iteration

Of course this equivalence could only happen if we consider Simultaneous iteration on *all* eigenvectors ($m = n$)

To make equivalence obvious we rewrite algorithms:

Simultaneous iteration

$\tilde{Q}^{(0)} = I$; $A^{(0)} = A$

for $k = 1, 2, \ldots$

$Z = A\tilde{Q}^{(k-1)}$

$\tilde{Q}^{(k)}\tilde{R}^{(k)} = Z$

$A^{(k)} = \tilde{Q}^{(k)T} A \tilde{Q}^{(k)}$

QR algorithm

$A^{(0)} = A$; $\tilde{Q}^{(0)} = I$

for $k = 1, 2, \ldots$

$Q^{(k)}R^{(k)} = A^{(k-1)}$

$A^{(k)} = R^{(k)}Q^{(k)}$

$\tilde{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$

We will show by induction that both algorithms generate the same matrices $\tilde{Q}^{(k)}$, $A^{(k)}$ and $\tilde{R}^{(k)} \equiv R^{(k)}R^{(k-1)}\cdots R^{(1)}$, with:

i) $A^k = \tilde{Q}^{(k)}\tilde{R}^{(k)}$

ii) $A^{(k)} = \left(\tilde{Q}^{(k)}\right)^T A \tilde{Q}^{(k)}$

proof: $k=0$ trivial since for both methods $\tilde{Q}^{(0)} = I$, $A^{(0)} = A$, $R^{(0)} = I$

$$I = A^0 = \tilde{Q}^{(0)}\tilde{R}^{(0)} = I\,I.$$

- Case $k \geq 1$ for Simult. iter.:

$$A^k = AA^{k-1} \underset{\text{induction i)}}{=} A\tilde{Q}^{(k-1)}\tilde{R}^{(k-1)} \underset{\text{iteration } 2,\ldots}{=} \tilde{Q}^{(k)}\underbrace{R^{(k)}\tilde{R}^{(k-1)}}_{=\tilde{R}^{(k)}}.$$

- Case $k \geq 1$ for QR algo:

$$A^k = AA^{k-1} \underset{\text{induction i)}}{=} A\tilde{Q}^{(k-1)}\tilde{R}^{(k-1)} \underset{\text{induction ii)}}{=} \tilde{Q}^{(k-1)}A^{(k-1)}\tilde{R}^{(k-1)}$$

$$\underset{\text{iteration}}{=} \underbrace{\tilde{Q}^{(k-1)}Q^{(k)}}_{=\tilde{Q}^{(k)}}\underbrace{R^{(k)}\tilde{R}^{(k-1)}}_{=\tilde{R}^{(k)}}$$

Finally to show ii) for QR algo:

$$A^{(k)} = R^{(k)} Q^{(k)} = Q^{(k)T} A^{(k-1)} Q^{(k)}$$

↑ QR algo ↑ QR algo

$$= (Q^{(k)})^T (\tilde{Q}^{(k-1)})^T A \tilde{Q}^{(k-1)} Q^{(k)}$$

$$= \tilde{Q}^{(k)T} A \tilde{Q}^{(k)} \quad \text{qed.}$$

↑ induction

Here is how QR algorithm can be expected to converge:

- QR constructs orthonormal basis of $A^k \to$ finds eigenvectors
- diagonal elements of $A^{(k)}$ are Rayleigh quotients:

$$A^{(k)} = (Q^{(k)})^T A Q^{(k)}$$

So they should $\to$ eigenvalues

- strictly triangular part are "generalized" Rayleigh quotients. Since eigenvectors are $\perp$, these should $\to 0$.

We are not done with QR. A more practical version would be as follows: Shifted QR

$A^{(0)} = Q^{(0)} A (Q^{(0)})^T = (\equiv)$ (Reduction to tridiag form)

for $k = 1, 2, \ldots$

| pick shift $\mu^{(k)}$ (usually $\mu^{(k)} = A^{(k-1)}_{mm}$)
| $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$
| $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$
| "deflation" to lock in eigenvalues that have converged