GRADUATE SEMINAR REPORT

ON

EIGENVALUE PROBLEMS

(submitted in partial fulfillment of M.Sc. degree in Mathematics)

BY

TEKLE GEMECHU

ADVISOR

PROF. DR. S.N. MURTHY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA UNIVERSITY

June , 2004

ADDIS ABABA

CONTENTS

Chapter-II: Generalized eigenvalue problems and numerical method of solutions

Preface

The study of **Eigenvalue Problems** has a fundamental place in the study of algebraic equations. They appear in science and are much applicable in physics.They arise from physical problems involving matrix equations such as mathematical problems leading to differential equations, wave motions ($u_{tt} = c^2u_{xx}$), heat equation ($u_t = k^2u_{xx}$) , Laplace equations ($u_{xx} + u_{yy} = 0$) in the two dimensional plane and transformations ($Tx = \lambda x$) and so on. The way such **Eigenvalue Problems** are tackled is explained under the introduction part (next section). The method starts from elementary method like solving the roots of the characteristic polynomial of a matrix A , then using iterative methods and finally grows to more advanced and sophisticated numerical methods involving step-wise similarity transformations of these times.

This seminar paper on the eigenvalue problems was compiled reading several books, Internet and other information.The topic presently under report has been suggested by my Advisor **Prof.Dr.S.Narasimha Murthy**.I am grateful to him for his suggestion and his continuous guidance during the preparation of the report.
It was my great desire and pleasure to get an M.Sc in mathematics. Wishes, hopes, needs (desires) and dreams will not enable one to seek it, they may enhance. Working day and night, morning and evening is also not enough to make it. It is through the nonslackenning (continuous) follow-ups and advises of lecturers, advisors, and librarians , computer Laboratory instructors, Secretaries and so on from the university members. So, I thank much, in particular my course advisor
Prof.Dr.S.N. Murthy, Dr.Demisu (Mathematics post graduate advisor),
Dr.Adnew (head Dept.) and R.Deumlich, Lab. instructors (Ato Abate, Biniam),
librarian (Ato Berhanu) and all others.

Tekle Gemechu

Chapter-I: Theoretical results on eigenvalue problems

1 .1 Introduction to eigenvalue problems

The purpose of this seminar paper is to study **algebraic (matrix) eigenvalue** problems and applying numerical methods for their solutions. Such problems always occur both in physical and social sciences.

We motivate some key issues discussed in this, with examples, theorems proofs and how some of them originate. The main concept of the chapter is the notion of **Eigenvalues and Eigenvectors**. The inputs are nonzero square (complex) matrices and the required are eigenvalues or eigenvectors.

The eigenvalue problems we consider are defined by the matrix equation $Ax = B\lambda x$ where A and B are nonzero square matrices of the same order n in $M_n(C)$ and x is nonzero vector in C^n and λ is a number in C, the case $B = I_n$ is our prime concern i.e., when $Ax = \lambda X$.

The seminar has two parts. The first part mainly discusses about the theoretical results of eigenvalues λ and eigenvectors x, numerical methods for solving for the roots of $P_A(t)$ (the characteristic polynomials), i.e.,the eigenvalues of A like **Newton's** method and **or any formula in (1.0)** for solving eigenvalues . Basic theorms such as **Cayley Hamilton, Gerschgorin's and Schur's** theorem for the bounds and locations of eigenvalues with out numerical methods. The spectral radius and norms as bounds for eigenvalues.The normal forms, Simpler forms of matrices, Singular value decomposition shall be discussed.

In the second part, we present advanced numerical methods like power method, inverse iteration, **Jacobi's, Given's ,Householder's, QR, LR** and finally, we compare these methods in summary and analyze the error encountered. Mathematica software application commands are used wherever needed and some of them are presented in 2:11 and finally we summarize our survey.

1.2 Basic symbols and formulae

C = the set of complex numbers

A, B, C, D will represent matrices

D = a diagonal matrix

D = dia (a, b, c, ...x)

$$||A|| = \max \left| \sum_{j=1}^{n} a_{i,j} \right| = \text{norm of matrix}$$

A ~B: A and B are Similar matrices

$M_n(C)$: The set of complex matrices of order n

$A^H = A^{-T}$: Conjugate transpose of A

E_λ = Eigenspace of λ (set of eigenvectors associated to λ)

λ_A = Spectrum of A.

LI = Linear independent vectors

EVP = Eigenvalue problem

CHT = Cayley Hamilton theorem

FTA = fundamental theorem of algebra.

$P_A(t)$ = Characteristic polynomial of matrix A.

$M_A(t)$ = Minimum polynomial of A.

$a(\lambda)$ = Algebraic multiplicity of λ as a root of $P_A(t)$

$g(\lambda)$ = Geometric multiplicity of λ.

ρ (A) = Rank of A

Cond (A) = condition no of matrix A

R (A) = spectral radius of A.

$\lambda(A)$ = Eigenvalue of A.

tr (A) = trace of A = $\sum_{i=1}^{n} \lambda_i$

DetA = $|A| = a_{11}A_{11} - a_{12}A_{12} + A_{13}A_{13} + ---- \pm a_{ln}A_{ln}$

$= \prod \lambda_i = (-1)^{1+j} \sum_{j=1}^{j=n} A_{1j}\, a_{1j}$

$A^{-1} = \dfrac{AdJ\ A}{|A|} = Inverse\ of\ A.$

$g(\lambda) = n - \rho(A - \lambda I) = DimE_{\lambda}$

$I = I_n$ = Identity matrix of order n

Numerical methods for approaximating roots (eigenvalues) for $P_A(t)$

Newton's method

$$x_{n+1} = x_n - \frac{p_A(x)}{p'_A(x)} \tag{1.0}$$

Regular falsi method

$$x_{n+1} = \frac{p_A(x_{n-1})\, x_n - x_{n-1}\, p_A(x_n)}{p_A(x_{n-1}) - p_A(x_n)}$$

Chebyshev's formula

$$x_{n+1} = x_n - \frac{p_A(x)}{P'_A(x)} - \frac{1}{2}\frac{p^2{}_A(x_n)p''(x_n)}{p'^3{}_A(x_n)}$$

R[X] = Set of all real valued functions.

Lim = the limit at infinity

Δ = forward difference operator

1.3.1 **Basic definitions**

Eigenvalue Problems: For A, B $\in M_n(c)/\{0\}$ and $\lambda \in C$, the general matrix equation $\mathbf{Ax = B}\lambda\mathbf{x}$ is called an eigenvalue problem (evp).

If $B = \mathbf{I_n}$, **then it is** $\mathbf{Ax} = \lambda\mathbf{x}$ or $(\mathbf{A}-\lambda\mathbf{x}) = \mathbf{0}$. **(1.1)**

The matrix $A - \lambda I$ is said to be characteristic matrix of A. Such matrix equations are needed when we need non- trivial solutions for systems of equations with several parameters.

1.3.2: **Geometrical description for eigenvalues and eigenvectors.**

Definition: 1. In equation, (1.1), λ is called an eigenvalue of A and $x \neq 0$ is an eigenvector associated to λ of A.

Example -1.1. Let A = $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$: $x = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$, then

$$\text{Ax} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix} = 2\begin{pmatrix} 1 \\ 3 \end{pmatrix} = 2x \tag{1.2}$$

$\lambda = 2$ is an eigenvalue and x = $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$ is an eigenvector asociated to $\lambda = 2$ of A.

This is a transformation $A : R^2 \to R^2$ given by y =$\mathbf{Ax = 2x}$ is a stretching by length in the same direction and (0,0) is the fixed point.

Graph

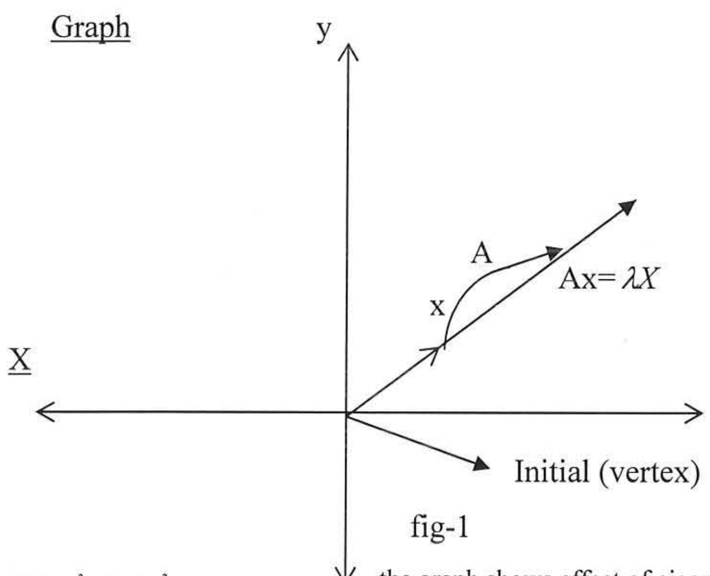


fig-1

$A : R^2 \to R^2$ the graph shows effect of eigenvalue (stretching)

Example – 1.2 Let D be the differential operator on $c^n[a, b] \subseteq R[x]$.

Let $x(t) = e^{3t}$, then $Dx(t) = 3x(t) = 3e^{3t}$.

3 is an eigenvalue, and $x(t) = e^{3t}$ is an eigenvector of D .

This is a transformation of points on $y(t) = e^{3t}$ all by 3 units (is stretching).

Mathematica command "Plot[{D[x[t],t],x[t]},{t,-100,1000}] " plots both functions.

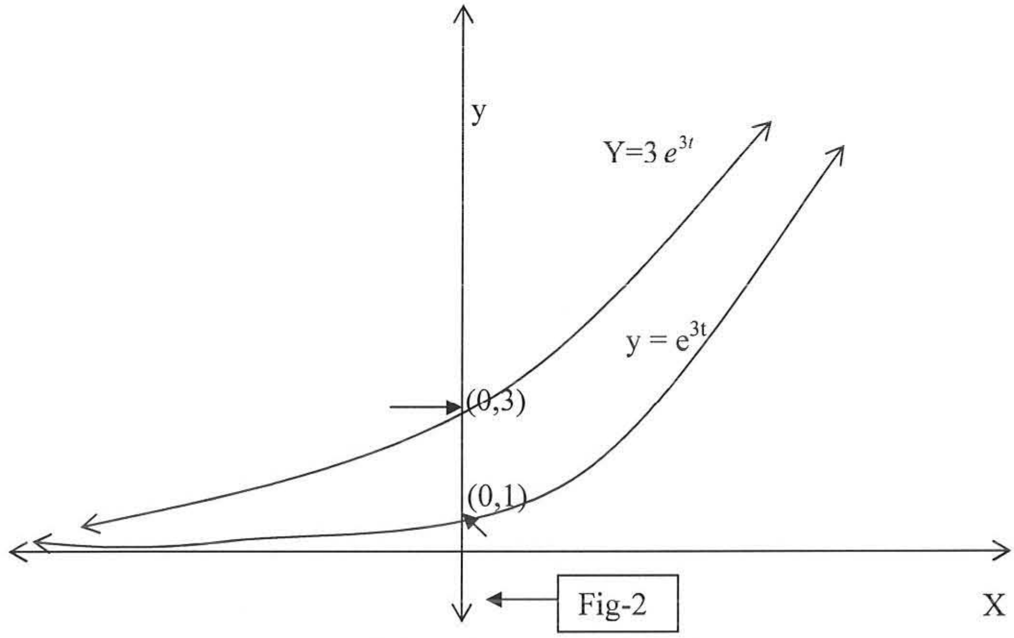


Fig-2

The graph shows effect of eigenvalue on f(x) i.e., stretching

Theorem-1.1: λ is an eigenvalue of A $\Rightarrow \lambda$ is also an eigenvalue of A^T and $(1/\lambda)$ is an eigenvalue of A^{-1}.

Proof: Let $Ax = \lambda x$

$$\Rightarrow x^H A^H = \lambda x^H = x^H \lambda \qquad (1.3)$$

$$\Rightarrow A^{-1}x = \lambda^{-1}x.$$ (hence the prove!)

Theorem-1.2: $\lambda = 1$ **is** the only eigenvalue for $A = \mathbf{I_n}$ **(identity matrix)**

Since $p_A(t) = (t-1)^n$, it is trivial.

1.3.3: Finding eigenvalues from characteristic polynomials.

a) By finding the characteristic polynomial and applying numerical methods such as

1) Newton's formula 2. Iteration

3) Regula falsi 4. Chebyshev's formula.

b) By similarity transformations and estimation by bounds.

c) advanced methods such as power method, Inverse iteration, Jacobi's,

d) By Householder,QR, and LR methods.

a . Finding eigenvalues using characteristic polynomials

Consider the eigenvalue problem (1.1) i.e. $(A-\lambda I)\,x = o$. The homogeneous equation, $(A- tI)\,x = 0$ has non-trivial solution $(x \neq o)$

$\Leftrightarrow$ Det (A-tI) = 0 $\Leftrightarrow$ A-tI is singular. Where

$$\text{Det}(A-tI) = t^n + a_{n-1}\,t^{n-1} + \text{----} + at + a_0 \qquad (1.4)$$

This is is called the characterstic polynomial of A denoted by $p_A(t)$.

Note that $p_A(t) = 0$ if and only if $t = \lambda$ is an eigenvalue of A.

1. If $\lambda_1, \lambda_2, ------\lambda_k$ *are roots of* $p_A(t)$, the corresponding eigenvector(s) x_k are solved by using λ_k *in* (1.1) *i.e* $(A-\lambda_k I)x_k = o$
2. The set of all eigenvalues of A denoted by $\lambda_A = \{\lambda_1, \lambda_2, -----\lambda_k\}$ is known as a spectrum of A .
3. The set $E_\lambda = \{x : Ax = \lambda x\}$ is called an eigenspace of λ of A or kernel of $(A-\lambda I)$ which generates C^2 since it is linear independent set of eigenvectors of A.
4. geometric multiplicity of λ is $g(\lambda)$ the dimension of the eigenspace i.e, $g(\lambda) = \text{Dim}\,E_\lambda = n - \rho(A-\lambda I) = Dim\ \ker\ (A-\lambda I)$
5. The multiplicity of λ as a root of $p_A(t)$ is its algebraic multiplicity $a(\lambda)$.

Note: If $A-\lambda I$ is singular ,then λ exists for A and hence Ker $(A-\lambda I) \neq \{\ \}$ or Dim $E_\lambda > 0$.

Example-1.3

let A = $\begin{pmatrix} 1 & 4 \\ 2 & 3 \end{pmatrix}$

Find eigenvalues and eigenvectors of A.

Solution: $P_A(t) = |A - tI| = t^2 - 4t + 5 = (t-5)(t+1), \lambda_1 = 5, \lambda_2 = -1.$

are eigenvalues. To find the eigenvectors

1. Take $\lambda=5$, then using (1.1), $(A-5I)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0 \Rightarrow x = t(1,1)^T : t \neq o$

$$E_5 = \{(1,1)^t\}, Dim\ E_5 = g(5) = 1 \tag{1.5}$$

2. Similarily,taking $\lambda = -1$, $(A-(-1I))\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = o \Rightarrow x = (4,-2)^t$

Dim $E_{-1} = g(-1) = 1$

In all cases, a (λ) = g (λ) = 1.

$\lambda_A = spectrum\ \ of\ \ A = \{-1, 5\}$ and

$E_5 \cup E_{-1} = \{ (1,1)^T, (4,-2)^T \}$ is linear independent and generates R^2.

The matrix A with full set (two) linear independent eigenvectors is called Simple matrix.

And when $a(\lambda) = g(\lambda) = 1$ A is said to be nonderogatory which is diagonable and normalizable, otherwise, derogatory.(This will be dealt later in normal forms).

Note that product of $(\lambda_i) = \det A$. Put $t = 0$ in $p_A(t)$ for the proof.

Newton's method or any numerical formula in (1.0))

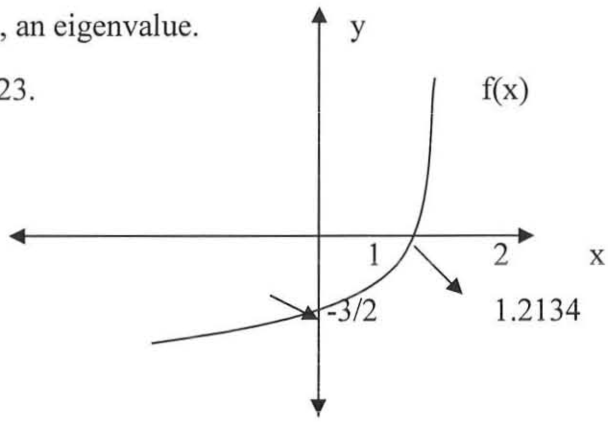
When the characteristic polynomial $p_A(t)$ can not be factored we can use Newton's method or any formula in (1.0) to approximate the eigenvalues of A as a roots of $P_A(t)$ and Gauss elimination for finding the eigenvectors can be used.

Example.1.4 If $f(t) = P_A(t) = (t^3 + t - 3)/2$ is polynomial for some matrix A, then the roots are approximated between 1 and 2 by Newton's iterative formula

as $\mathbf{x_{n+1} = x_n - (f(x_n) / f`(x_n))}$ with to = 1.2, (1.6)

$t_3 \approx 1.2134$ is a root i.e., an eigenvalue.

with f (1.2134) < 0.0023.



Mathematica

Pl ot [f [x], {x,-2,2}] gives the above graph of Newton's method. Solve [f [x]= =0,x] gives the exact root(evs)

Note: Newton's method is quadratic convergent and in this example convergence depends on initial guess x_0.

1.3.4 Existence and uniqueness of eigenvalues and eigenvectors

Theorem.1.3 A non-zero square matrix of order n has at least one eigenvalue (eigenvalues are not necessarily distinct) also the, eigenvectors for these eigenvalves may not be distinct.

Proof: by fundamental theorem of algebra (FTA) , $P_A(t)$ = det (A –tI) has atleast one root. Eigenvalues (roots) may not be distinct as a root of $p_A(t)$.

Example1.5 : 1 is the only eigenvalue for I_n since $p_A(t) = (1-t)^n$.

But there are many eigenvectors.

To prove that eigenvectors are not unique

Consider (A –tI) = 0 .Let x_1 and x_2 be solutions.

i.e., $Ax_1 = \lambda x_1 \quad and \quad Ax_2 = \lambda x_2$ (1.7)

$\Rightarrow \quad A\,x_1 + A\,x_2 = \lambda(x_1 + x_2)$

$\therefore\ x_1 + x_2$ is an eigenvector associated to λ. In general, $ax_1 + bx_2$ is an eigenvector as

$A(ax_1 + bx_2) = \lambda A x_1 + bAx_2 = a\lambda x_1 + b\lambda x_2 = \lambda(ax_1 + bx_2)$

There is no one to one correspondence between λ and x

Note: Distinct eigenvalues have different eigenvectors which are linear independent (LI).

Suppose λ_1 and λ_2 be the two eigenvalues with the same eigenvectors.

Let $\lambda_1 x = \lambda_2 x$, then

$(\lambda_1 - \lambda_2)x = 0$. so, $\lambda_1 = \lambda_2$ is a contradiction.

To show $\{x_1, x_2, \ldots, x_n\}$ is linear independent,

If n = 1,then it is trivally true.

Suppose $\sum c_i x_i = 0$.Multiplying this by A and λ_j, we get

$\sum c_i(\lambda_i - \lambda_j) = 0$.

Hence, $c_i = 0$ for all i.

Note: Eigenvalues of a matrix tell us some of its basic Properties like convergence and diagonability.

A matrix A said to be convergent when $\text{Lim}\|A^n\| = o$.

Basic theorems on characteristic and minimum polynomials

(The Cayley Hamilton's theorem)

Every square matrix of order n satisfies its characteristic polynomial $P_A(t)$, i.e, $P_A(A) = 0$.

Proof –1, Suppose $P_A(t) = \det(A - It) = 0$, then this implies

$$P_A(A) = \det(A - IA) = \det(0) = 0.$$

Proof of (case-2)

Let A be an n square matrix

Then $P_A(t) = |A - It| = t^n + a_{n-1}t^{n-1} + \text{------} + at + ta_o$

Let B (t) be the adjoint of A-tI

The elements of B (t) are cofactors of A-tI and are polynomials of degree $\leq n - 1$.

Thus, $B(t) = B_{n-1}t^{n-1} + \ldots\ldots + B_1 t + B_0$

B_i are n-square matrices. By properties of adjoints,

$(B(t))\,(A-tI)=|A-It|I$

(tI-A) ($B_{n-1}\,t^{n-1}$ +.......B_1t + B_0 (1.8)

$=(t^n+a_{n-1}t^{n-1}+----+a_1t+a_o)I$

$\Rightarrow B_{n-1}=I$, hence the following.

$B_{n-2}-AB_{n-I}=a_{n-1}I.$

$B_{n-3}-AB_{n-2}=a_{n-2}I$

$B_{n-k}\;-A\,B_{n-(K-1)}=a_{n-(k-1)}I$

$B_o-\quad AB_1\;=\;a_oI$

-A $B_o\;\;=a_oI$

Multiplying both sides by $A^n,A^{n-1},-----,A,I$

$A^n\;B_{n-1}\;=\;A^n$

$A^{n-1}B_{n-2}-A^nB_{n-1}=a_{n-1}A^{n-1}$

$A\,B_o-A\,B_1=aA$, .

-ABO $=a_oI$

adding both sides

$A_n\;+\;a_{n-1}\;A^{n-1}\;+\;-\;-\;-\;-\;+\;a_1A\;+\;a_oI\;=\;o$

Example;1.6 Let $A=\begin{pmatrix}1&2\\3&2\end{pmatrix}$,then

$P_A(A)=A^2-3A-4I=\begin{pmatrix}7&6\\9&10\end{pmatrix}-\begin{pmatrix}7&6\\9&10\end{pmatrix}=\begin{pmatrix}0&0\\0&0\end{pmatrix}$

<u>Theorem1.5:</u> The minimum polynomial $m_A(t)$ of A divides its characteristic polynomial $P_A(t)$. It is a polynomial of smallest degree such that $m_A(A)=0$.

Proof; if p(x) = m(x)g(x) + r(x),

then P(A) = 0.Hence r(x) = 0.

Application of Hamilton's theorem (CHT)

The cayley Hamilton's theorem is used to find inverse of A if it exists as $A^{-1} = -1/a_0 (A^{n-1} + a_{n-1}A^{n-2} + \dots\dots\dots + aI)$

Example $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ using CHT, $A^{-1} = -1/2 \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$

1.3.5;Theorems on estimating location and bounds of eigenvalues

Theorem 1.3.5.1 Let A be $(a_{ik})_{n,n}$ and λ be its eigenvalue .Then **Gerschgorin's** theorem (estimation by region) states that each eigenvalue λ, of A lies within one of the n Gerschgorin's disks (circles) given by

C:$| \lambda - a_{kk} | \leq |a_{k1}| + |a_{k2}| + - - - + |a_{kn}| = r > o$

Proof: consider $(A-\lambda I)\, x = 0$ is system of n linear equations.

Let $k \leq n$. And choose from these equations, the equation

$a_{k1}x_1 + a_{k2}x_2 + \dots + (a_{kk}x_k - \lambda x_k) + \dots. + a_{kn}x_n = 0$

$$(a_{kk}x_k - \lambda x_k) = a_{k1}x_1 + a_{k2}x_2 + \dots + \dots. + a_{kn}x_n \tag{1.9}$$

Let $|x_m| = \max \{ |x_1|, |x_2|, \dots, |x_n| \}$.

Hence, $|x_k/x_m| \leq 1$.

Applying triangular inequality to (1.9) we get $|\lambda - a_{kk}| \leq |a_{k1}| + \dots\dots. + |a_{kn}|$

Example1.3.5.1: let $A = \begin{pmatrix} 2 & 1 & 3 \\ 2 & 3 & 3 \\ 1 & 4 & 5 \end{pmatrix}$

The three Gerschporin's disks are ;

$$\begin{aligned} D_1 &: | \lambda - 2| \leq 1 + 3 = 4 \\ D_2 &: | \lambda - 3| \leq 2 + 3 = 5 \\ D_3 &: | \lambda - 5| \leq 1 + 4 = 5 \end{aligned}$$

These are the circles :

$$\begin{aligned} D_1 &: (x-2)^2 + y^2 \leq 4^2 = 16 \\ D_2 &: (x-3)^2 + y^2 \leq 5^2 = 25 \qquad \lambda = x + iy : i = \sqrt{-1} \\ D_3 &: (x-5)^2 + y^2 \leq 5^2 = 25 \end{aligned} \tag{1.10}$$

using mathematica" Eigenvalues [a] "gives 8.55479 ,. 7226 + 1.05563i, .7226 - 1.05563i as eigenvalues.

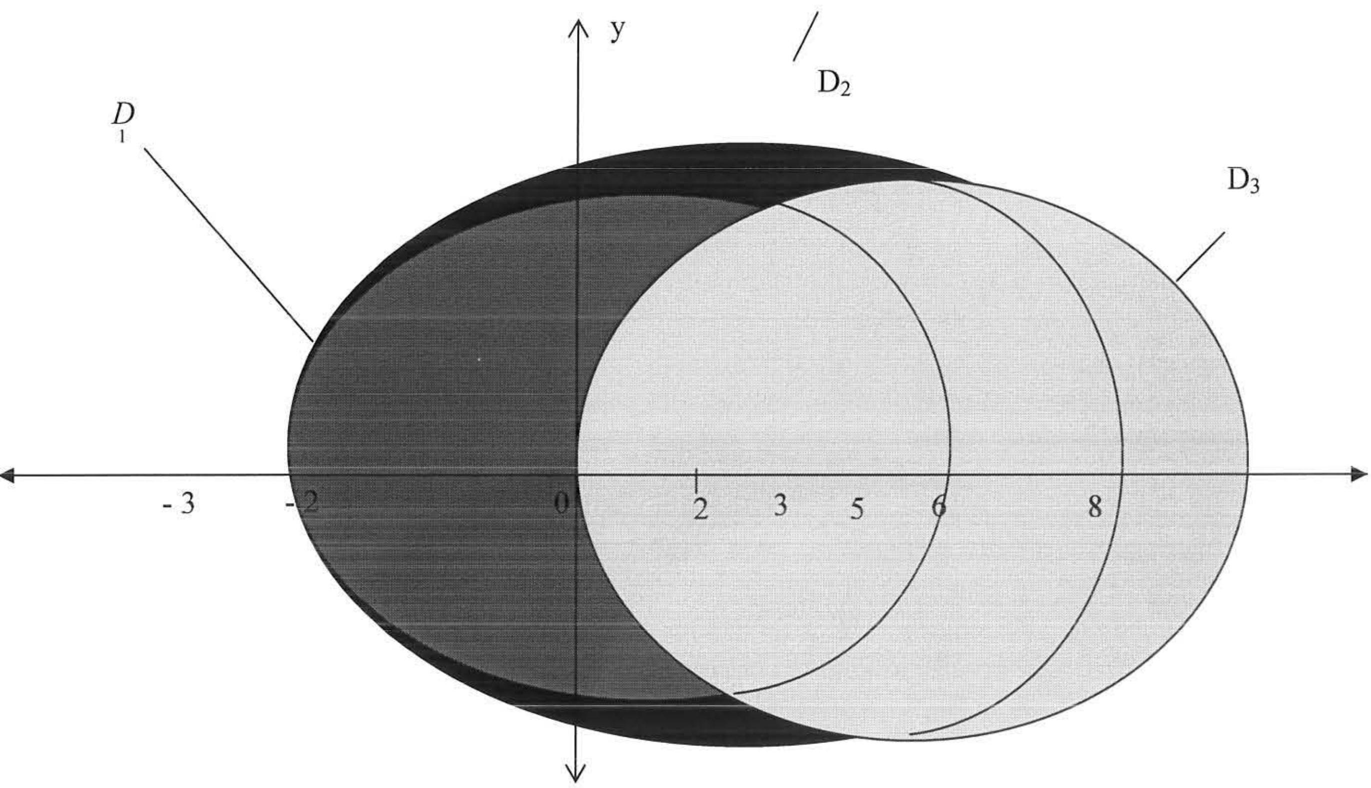


Fig-3

The figure gives the guess on bounds ,minimum and maximum estimation not the exact eigenvalues

Theorem: 1.3.5.2 Schur's theorem (estimation by bounds)

Let A be $(a_{ik})_{n,n}$ and λ be its eigenvalue.Then

$$\sum_{i=1}^{n} \left|\lambda_i\right|^2 \le \sum_{i=1}^{n} \sum_{j=1}^{n} \left|a_{ij}\right|^2 \quad and \quad \left|\lambda_k\right| \le \sqrt{\sum_{j=1}^{n} \sum_{J=1}^{n} \left|a_{ij}\right|^2}$$

For the above A,

$$\lambda_k \le \sqrt{(2^2+1^2+3^2)+(2^2+3^2+3^2)+(1^2+4^2+5^2)} = \sqrt{78}\text{ ,k} = 1,2,3.$$

proof: Let $U(U^T = U^{-1})$ be a unitary matrix and T be an upper triangular with $A = U^H TU$ i.e, A is similar to T.

Hence, spectrum (A) = spectrum (T) = $\{t_{11}, t_{22}, \ldots t_{nn}\}$.

$$\sum_{i=1}^{n} \lambda_i^2 = \sum_{i=1}^{n} t_{ii}^2 \leq \sum_{i=j=1}^{n} a_{ij}^2 = \text{Trace}\,(T^H T) = \text{Trace}\,(A^H A).$$

1.3.6:The bounds of eigenvalues by norms and spectral radius (R(A))

The three- norms of a matrix A as in (sec:1.2) are

$\|A\|_1$ = Max(column sum)

$\|A\|_\infty$ = Max(row sum) (1.11)

$\|A\|_2$ = The Euclidean norm.

The norm and the spectral radius **R(A)** tell us the upper bound for the eigenvalues.

$R(A) = \max\{\lambda_i\} \rightarrow$ tells convergence of a matrix

A matrix A converges if Lim $\|A^n\| = 0$.

Cond (A) = $\|A\|.\|A^{-1}\|$ $\rightarrow$ tells singularity (well-posedness)

Theorem1.3.6.1: $|\lambda_i| \leq \|A\|$ and $R(A) \leq \|A\|$ and

$R(A)\|A^{-1}\| \leq$ Cond(A)

Proof; using $\lambda x = Ax$,$|\lambda| \leq \|A\|$. The others follow by the definition.

Example1.3.6.1; $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ the values of the three norms

are; 3,3, $\sqrt{10}$; R (A) = 3 = max ({-1,3}) = max (spectrum of A)

$A^{-1} = -1/3 \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$, det(A) = -3 , $\det A^{-1} = \dfrac{-1}{3}$, Cond (A) =1.

1.3.7: Similarity method and diagonalization

Definition.3: A is said to be similar to B (A~B) if and only if there is a nonsingular matrix T such that $B = T^{-1}AT$.

A matrix A is called diagonable if there is a matrix T such that AT = TD.(A~D).

Example: 1.3.7.1 Normal matrix ($A^TA = AA^T$), Hermitian matrix ($A^T = A$) and all simple matrices are diagonable (we see this later).

Theorem1.3.7.1 similar matrices have the same eigenvalues.

Proof:-$P_B(t) = |T^{-1}AT\text{-}tI| = |T^{-1}(A\text{-}It)\,T)| = P_A(t)$.

Hence,since A and B have the same polynomials ,they will have the same roots.

Example 1.3.7.2 let $A = \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix} : T = \frac{1}{\sqrt{5}}\begin{pmatrix} -2 & 2 \\ 1 & 1 \end{pmatrix}\ B = \begin{pmatrix} 1 & 0 \\ 0 & 6 \end{pmatrix}$

then $T^{-1}AT = B \Rightarrow A \sim B$

The eigenvalues are 1 and 6,i.e.,the roots of $p_A(t) = (t\text{-}1)(t\text{-}6)$.

The General similarity transformation is given by $A_i = T_i^{-1}A^{i-1}T_i$.

We often use this in the forgoing sections.

The non-zero column of T are the eigenvectors of A.

To see this; let $Ax_1 = \lambda_1 x_1$

$Ax_2 = \lambda_2 x_2$

. .

. .

$Ax_n = \lambda_n x_n$.

Then, $A\,[x_1\ x_2\ x_3\ \ldots.\ x_n] = [x_1\ x_2 \ldots.x_n]D_\lambda$ (1.12)

We get $AT = TD_\lambda$

Therefore, $T^{-1}AT = D_\lambda$ or ($A \sim D_\lambda$).

T, the matrix of eigenvectors of A is called Modal matrix of A and D_λ is a diagonal Matrix whose elements are the eigenvalues.

We can say diagonalization is one method of finding eigenvalues.

Algorithms of diagonalization.

1. Determine the characteristic polynomial of the matrix A.
2. Find all the eigenvalues
3. Find all the eigenvectors.We may normalize them.
4. Write the matrix of order the same as of A with non-zero columns the linear independent eigenvectors in 3 .Say this matrix T.
5. Find T^{-1}
6. Obtin $T^{-1}AT = D_\lambda$ which is a diagonal matrix containing the eigenvalues of A and is also similar to A.We say A is diagonable .

Example:1.3.7.3 Consider the above example. $\lambda_1 = 1$, and $\lambda_2 = 6$.

The eigenvectors are $x_1 = (\frac{-2}{\sqrt{5}}, \frac{1}{\sqrt{5}})$ and $x_2 = (\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}})$ when normalized.

$T = [x_1, x_2]$ and $D_\lambda = B$ so that $T^{-1}AT = D_\lambda$.

The similarity theorem also helps to prove the following.

Theorem: 1.3.7.2 $g(\lambda) \leq a(\lambda)$.

Proof: let $p = g(\lambda)$ and $\{x_1, \ldots, x_p\}$ be linear independent (LI) eigenvectors of λ $(Ax_i = \lambda x_i)$.

We choose n-p additional LI vectors basis of c^n such that

$T = [x_1, \ldots, x_n]$ is nonsingular and $T^{-1}AT$ has the form

$$T^{-1}AT = \begin{bmatrix} \lambda I & B \\ 0 & C \end{bmatrix}.$$

Hence, $p_A(t) = \det(T^{-1}AT - tI) = (t-\lambda)^p \det(C - tI)$ which tells us that

λ has a multiplicity of atleast p.

Thus, $p = g(\lambda) \leq a(\lambda)$. Proved!.

Later, we see that if $a(\lambda) = 1$ then all elementary divisors of $p_A(t)$ are

Linear and if $g(\lambda) = 1$ we call A nonderogatory matrix as before.

Note:

1. For triangular and diagonal matrices eigenvalues are the diagonal elements.

   Let $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$, then $P_A(t) = (t-1)(t-2)(t-3) = 0$.

Gives t =1,2,3 to be the eigenvalues of A.

2. For Hermitian or symmetric matrices ($A^T = A$) eigenvalues are real and eigenvectors of distinct eigenvalues are orthogonal.

   Proof; Suppose $Ax = \lambda x$, then

   $A^H x = \lambda x$.But then we have by transposing

   $x^H A^H = \lambda^* x^H = x^H A$.

   So $\lambda = \lambda^*$. Hence λ is real. To show orthogonality,

let $Ax_1 = \lambda_1 x_1$ and $Ax_2 = \lambda_2 x_2$ as the set C is Hilbert space,

We get $(\lambda_1 - \lambda_2)\langle x_1, x_2 \rangle = 0$. i.e $\langle x_1, x_2 \rangle = 0$ as $\lambda_1 \neq \lambda_2$ (hence x_1 and x_2 are orthogonal)

3. Normal matrices ($A^H A = AA^H$).

   Theorem; 1.3.7.4 If A is normal with real eigenvalues then it is Hermitian.

   Proof; let $U^H AU = D$, then

   $(U^H AU)^H = D^H = D$ i.e., $A^H = A$.

   Where,$U = U^{-1}$ is a unitary matrix.

   Diagonal, Hermitian and unitary matrices are simple and are normal.

1.3.8: **Matrix normal forms**

A. The Jordan normal form: It is easy to see that not all matrices are diagonable.

For example, $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is not diagonable, for if it were, then

$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \sim \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}$ for some x, y in R and x, y must be 1.

Therefore, $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ which is 1= 0 is absurd.

The Jordan matrices are easy to deal with and solve the problems of diagonability.

Definition.4 A Jordan block $J_n(\lambda_i)$ is an (nxn) matrix with

1. each of whose diagonal elements are λ_i.
2. each of whose super diagonal entries are 1.
3. each of whose other entries are 0.

Example;1.3.8.1

$$J_4(2) = \begin{bmatrix} 2 & 1 & O & O \\ 0 & 2 & 1 & O \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \text{ and } J_1(2) = (2) : \quad \lambda = 2.$$

Definition:5 If A is (nxn) and B is an (mxm) matrix then the direct sum of A and B is $A \oplus B$ is the (n+m) x (n+m) matrix C;

where $c_{ij} = a_{ij}$ for $1 \leq i, j \leq n$ or $c_{ij} = b_{ij}$,if $j \leq m+n$ or $c_{ij} = 0$.

Example:1.3.8.2,
$$(2) \oplus \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 4 \end{bmatrix}$$

We say that J is a Jordan matrix if and only if J is a direct sum of its Jordan blocks.

Example;1.3.8.2

$$\begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \oplus (2) \oplus \begin{bmatrix} 4 & 1 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$= J_2(2) \oplus J_1(2) \oplus J_3(4) = \oplus J_{ni}(\lambda_i)$; i =1,2,3 is a Jordan matrix.

In general ,J is a Jordan matrix if and only if $J = \oplus J_{ni}(\lambda_i)$.

We also call this Jordan (normal) decomposition of J.

If J is (kxk) then $\sum_{i=1}^{i=m} n_i = k$. Where k is number of Jordan blocks. It is also true that the characteristic polynomial of J ,$P_J(t)$ is a multiple of $m_A(t)$ .Where, $m_A(t)$ is given by

$$(t-\lambda_1)^{n_1}(t-\lambda_2)^{n_2}\ldots(t-\lambda_n)^{n_m} = \det(J-tI) = \prod_{i=1}^{k} \det(J_i - It) \tag{1.13}$$

det (J_i-tI) of the Jordan block J_i are called elementary divisors of A (see theorem:1.5). If Det (J_i-tI) are all linear (a (λ_i) =1) and if also g (λ_i) =1,then A is diagonable and (nonderogatory, otherwise derogatory) and it is similar to a Jordan matrix (Jordan form exists). For the above J, a (2) = 2, a (4) = 3.

Hence, 2+1+3 = 6 = k, m =3 and $m_J(t) = (t-2)^2(t-2)(t-4)^3$.

Example;1.3.8. 3

$$J = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

is a Jordan matrix has eigenvalues 1 and -1

with multiplicities a (λ_1 = 1) = 2, a (λ_2= 1) = 3, a (λ_1 = 1)= 5 ,a(λ_2 = -1) = 4.

elementary divisors are;

$(1\text{-}t)^3$, $(1\text{-}t)^2$,(-1-t) ,(-1-t) ; $P_J(t) = (-1)^9\,(-1+t)^5(t+1)^4$.

The minimum polynomial $m_J(t)$ is the multiple LCM of elementary divisors and $m_J(t) = -(1-t)^3(t+1)^2$.

$E_1=\{e_1 , e_4\}$ and $E_{-1} = \{e_6, e_8, e_9\}$ are the eigenspaces.

Jordan's theorem: Every matrix is similar to some Jordan matrix.

There is a nonsingular matrix T with

$$J = T^{-1}AT = \begin{bmatrix} J_1 & 0 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & J_n \end{bmatrix} = \oplus J_{ni}^{(\lambda i)} \qquad (1.14)$$

The proof holds directly from similarity, diagolization algorithms and the above notes and examples. This theorem has good application as a simple proof for Hamilton's theorem.

Any Jordan matrix J similar to A is also called a Jordan form of A.If J has m Jordan blocks ($J = \oplus J_{ni}(\lambda_i)$;I =1,…,m),then there will be a total of (m!) Jordan forms by permutting each blocks $J_{ni}(\lambda_i)$. And if A is diagonable, then so is its Jordan form. (we say it is also normalizable)

Here is numerical example;

If $A = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$, then the following is complete list of

the Jordan forms. There are 3 Jordan blocks. Therefore, there must be $3! = 6$ Jordan forms. These are;

1. $J_1 = J_2(2) \oplus J_1(3) \oplus J_2(4)$.
2. $J_2 = J_1(3) \oplus J_2(4) \oplus J_2(2)$.
3. $J_3 = J_2(4) \oplus J_2(2) \oplus J_1(3)$.
4. $J_2(2) \oplus J_2(4) \oplus J_1(3)$
5. $J_1(3) \oplus J_2(2) \oplus J_2(4)$
6. $J_2(4)\ J_1 \oplus J_1(3) \oplus J_2(2)$.

and the eigenvalues are $\lambda_1, \lambda_2, \lambda_3 = 2, 3, 4$ respectively.

If we partition the matrix T in accordance with the Jordan normal form J

As $T = [T_1^{\,1},\dots,T^k a(\lambda_k)]$, then from

$T^{-1}AT = J$, $AT^i a(\lambda_k) = T^i a(\lambda_k) J_i$.

Denoting the columns of $T^i a(\lambda_k)$.

By t_m, $T^i a(\lambda_k) = [t_1,\dots,t_{a(\lambda j)}]$. So, we get from

$$(A-\lambda_i I)\,([t_1,\dots,t_{a(\lambda j)}].) = [t_1,\dots,t_j]\begin{bmatrix} 0 & 1 & & & 0 \\ & . & 1 & & \\ & & . & 1 & \\ & & & . & 1 \\ & & & & 0 \end{bmatrix} \tag{1.15}$$

$(A-\lambda_i I)\, t_m = t_{m-1}$, $(A-\lambda_i I)\, t_1 = 0$. t_1, the first column of $T^i_{a(\lambda j)}$ is an eigenvectorof λ_i. The remaining $t_m (m = 2,3,\dots,j)$ are called principal vectors corresponding to λ_i and we note that with each block J_j there is associated eigenvector and principal vectors .

All together, for (nxn) matrix A, we obtain a basis for C^n (the columns of T) which consists entirely of eigenvector and principal vectors. In this case we could find all eigenvectors and eigenvalues from all the Jordan blocks one by one for each block.

B. Frobenieus normal form

As Jordan blocks are the building blocks of the Jordan matrices, the Frobenius normal form (also called the rational normal forms) are built from Frobenius matrice (F is also called normal form) of the form

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_0 \\ 1 & . & . & 0 & -a_1 \\ & . & & . & . \\ & & . & 0 & . \\ 0 & & & 1 & -a_{n-1} \end{bmatrix}$$

The last column consists of the first n-1 coefficients of $m_A(t)$ of degree n. Such matrices are studied in Krylov sequence of vectors. These vectors satisfy

1. $t_i = At_{i-1}$.
2. $\{t_i\}$ is LI. Where t_0 is the initial vector in c^n and A is (nxn) matrix. If one forms $T = [t_0, t_1, \ldots, t_{m-1}]$ and the matrix F, then the $\rho(T) = m =$ number

of LI. vectors t_i.

And $AT = A[t_0, t_1, \ldots, t_{m-1}] = [t_1, \ldots, t_m] = [t_0, t_1, \ldots, t_{m-1}]F = TF$ (1.16)

Hence, AT = TF implies $T^{-1}AT = F$. i.e, F is similar to A.So, every eigenvalue of F is of A. If $x = Tz$ for $z \neq 0$, $Ax = ATz = TFz = \lambda x$.

Theorem:1.3.8.1 F is nonderogatory and the minimal polynomial of F is $P_F(t) = a_0 + a_1 t + \ldots + a_n t^n$.

proof: $(-1)^n \det(F-tI) = a_0 + a_1 t + \ldots + a_n t^n = P_F(t)$ ' proved!.

Theorem:1.3.8.2 For every (nxn) matrix A there is a nonsingular (nxn) matrix T with

$$T^{-1}AT = F = \begin{bmatrix} F_1 & & 0 \\ & F_2 & \\ 0 & & F_n \end{bmatrix}$$

is called Frobenius normal form of A. Each matrix F_i is Frobenius matrix.

For the proof we consider as in the Jordan normal form and set

$S_i^{-1} F_i S_i = J_i$(is a similarity transformation).

since the Jordan normal form is unique up to permutation of Jordan blocks

$$j' = \begin{bmatrix} J_1 & & & 0 \\ & J_2 & & \\ 0 & & J_3 & \\ & & & \ddots \\ & & & & J_n \end{bmatrix} \tag{1.17}$$

$$= \begin{bmatrix} S_1 & & & \\ & S_2 & & \\ & & S_3 & \\ & & & S_n \end{bmatrix}^{-1} \begin{bmatrix} F_1 & & \\ & F_2 & \\ & & F_n \end{bmatrix} \begin{bmatrix} S_1 & & & \\ & S_2 & & \\ & & S_3 & \\ & & & S_n \end{bmatrix}$$

From Jordan's theorem, there is U such that $U^{-1}AU = J$.

The matrix $T = US^{-1}$ with $S = \begin{bmatrix} s_1 & & & & \\ & s_2 & & & \\ & & s_3 & & \\ & & & \ddots & \\ & & & & s_n \end{bmatrix}$

transforms A to the desired form.

<u>Example:1.3.8.4</u> For the matrix in example 1.3 .8.3 of Jordan,

the characteristic polynomials of F_1 ,F_2 and F_3 are

$p_{F_1}(t) = (1-t)^3(-1-t)^2 = -(t^5 - t^4 - 2t^3 + 2t^2 + t - 1)$

$p_{F_2}(t) = (1-t)^2(-1-t) = -(t^3 - t^2 - t + 1)$; $p_{F_3}(t) = -(t+1)$.

Using coefficients of the F_i, we find that

$$F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, F_2 = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, F_3 = [-1]$$

Thus F = $\begin{bmatrix} F_1 & & \\ & F_2 & \\ & & F_3 \end{bmatrix}$ is the Frobenius normal form of A.

and eigenvalues are -1 and 1.This is one way of solving eigenvalue problems simplifying Newton's method.

C. The Schur's normal form and singular value decompositions.

Theorem (Schur's): For every (nxn) matrix there is a unitary (nxn) matrix U such that $U^H AU = \begin{bmatrix} \lambda_1 & * & \cdot & \cdot & * \\ & \lambda_2 & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & * \\ 0 & & & & \lambda_n \end{bmatrix}$

λ_i are eigenvalues of A not necessarily distinct .

The proof is by induction.

For i = n =1, it is trival.

Suppoes it holds for (n-1) x (n-1) matrix.

Let λ_1 be eigenvalue of A and $x_1 \neq 0$ be its eigenvector with $||X||^2 = x^T x = 1$, $Ax_1 = \lambda_1 x_1$.

Then we can find additional vectors $x_2, x_3, \ldots, x_n$ forming orthonormal basis of c^n and the (nxn) matrix $X = [x_1, x_2, x_3, \ldots, x_n]$ with columns x_i is thus unitary ($x^T x = 1$).

Since $X^H AXe_1 = X^H Ax_1 = \lambda_1 X^H x_1 = \lambda_1 e_1$, the matrix $X^H AX$ has the form

$X^H AX = \begin{bmatrix} \lambda_1 & a \\ 0 & A_1 \end{bmatrix}$ where A_1 is a matrix of (n-1)x(n-1)

and a^H is in c^{n-1} by induction hypothesis there is a unitary n-1 x n-1 matrix U_1 such that $U_1{}^H AU_1 = \begin{bmatrix} \lambda_1 & * & \cdot & \cdot & * \\ & \lambda_2 & \cdot & \cdot & \vdots \\ & & & \cdot & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & * \\ 0 & & & & \lambda_n \end{bmatrix}$

The matrix U = $X.\begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}$ then is a unitary nxn satisfying

$$U^HAU = \begin{bmatrix} \lambda_1 & * & \cdot & \cdot & * \\ & \lambda_2 & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & * \\ 0 & & & & \lambda_n \end{bmatrix}$$

D. Singular values and singular value decomposion

A matrix B is said to be positive definite (semi positive) definite if and only if eigenvalues are all positive or non-negative. Given an (nxn) matrix A, the (nxn) matrix A^HA is positive semidefinite, since $x^H(A^HA)x = \|Ax\|^2_2 \geq 0$ for any x in c^n. Its eigenvalues can be written as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$.

Hence, let $\lambda_k = \sigma_k^2$ with $\sigma_k \geq 0$ and are called singular values of A.

Replacing A by A^HA above we get $\sigma_1 = \text{Max}\ (\|Ax\|_2.\|x\|_2\ ) = \|A\|_2$ and

$\sigma_n = \min\ (\|Ax\|_2.\|x\|_2)$ for x in $c^n /\{0\}$.In particular, if A is nonsingular, one has

$1/\sigma_n = \max\|x\|_2\ .\|Ax\|_2 = \text{Max}\|A^{-1}y\|_2.\|y\|_2 = \|A^{-1}\|_2$

i.e, Cond (A) = $\|A\|_2.\|A^{-1}\|_2 = \sigma_1/\sigma_n$. The smallest singular value σ_n of A gives the distance of A to the nearest singular matrix.

Singular value decomposion theorem (SVD)

Let A be an (mxn) matrix. Then

a) there exist an (mxm) unitary matrix U and an (nxn) unitary matrix V such that $U^HAU = D'$ is a diagonal matrix of the form $D' = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$, where

$D = \text{dia}\ (\sigma_1, \ldots, \sigma_r)$.

b) the nonvanishing singular values of A^H are the numbers $\sigma_1, \ldots, \sigma_r$,

r is ρ (A). $A = UDV^H$ is called singular value decomposion of A.

Proof: We start by induction on m and n. For m = 0 or n = 0 nothing to prove. Assume it holds for (m-1xn-1) matrices and A is (mxm).

Let σ_1 be the largest eigenvalue of A. If $\sigma_1 = 0$,then A = O.

Suppose $\sigma_1 > 0$.

Let x $\neq$ 0 be an eigenvector of A^HA for $(\sigma_1)^2$ with $\|x\|_2 = 1$.

$A^HAx_1 = \sigma_1{}^2x_1$.

We can find n-1 additional vectors $x_1,\ldots,x_n$ in C^n such that the(nxn) matrix $X = [\, x_1,\ldots,x_n]$ becomes unitary ($X^HX = I_n$).

By virtue of $\|Ax\|_2{}^2 = x_1{}^HAx_1 = \sigma_1{}^2x_1{}^Hx_1 = \sigma_1{}^2 > 0$.

We can find additional vectors $y_2\ldots y_n$ as $y_1 = 1/\sigma_1(Ax_1)$ with $\|y_1\|$ is well defined.

So the (mxm) matrix $Y = [\, y_{1,\ldots,}\; y_n]$ is Unitary.($Y^HY = I_m$).

Now $Y^HAXe_1 = Y^HAx_1 = \sigma_1 Y^Hy_1 = \sigma_1 e$ (1.18)

and $(Y^HAX)^He_1 = Y^HA^HXe_1 = X^HAy_1 = X^HA^HAx_1/\sigma_1$.

So Y^HAX has the form $Y^HAX = \begin{bmatrix} \sigma_1 & 0 \\ 0 & A' \end{bmatrix}$ where A' is (m-1-by-m-1) matrix.

By the induction hypothesis, there is a unitary (m-1x m-1) matrix U' and a unitary n-1x n-1 matrix V' Such that

$$U'^HA'V' = D'' = \begin{bmatrix} D' & 0 \\ o & 0 \end{bmatrix}$$

$D' = \operatorname{diag}(\sigma_2,\ldots,\sigma_r)$ and $\sigma_2 \geq ,\ldots, \geq \sigma_r$.

With D'' an (m-1x n-1) diagonal matrix indicated.

$$U = Y.\begin{bmatrix} 1 & 0 \\ 0 & U' \end{bmatrix}$$

$$V = X.\begin{bmatrix} 1 & 0 \\ 0 & V' \end{bmatrix}$$ and we have

$$U^HAV = \begin{bmatrix} 1 & 0 \\ 0 & U^H \end{bmatrix} Y^HAX \begin{bmatrix} 1 & 0 \\ 0 & V' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U'^H \end{bmatrix}$$

$$\begin{bmatrix}\sigma_1 & 0\\ 0 & A'\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & V'\end{bmatrix}=\begin{bmatrix}\sigma_1 & 0\\ 0 & D'\end{bmatrix}=\begin{bmatrix}D & 0\\ o & 0\end{bmatrix}=D'$$

and r = rankA= rankD' = rank U^HAV.

$D'^HD' = \text{dia}\,(\sigma_1^2 \ldots \sigma_r^2, 0,0,\ldots,0) = V^H(A^HA)V$, $\sigma_1^2,\ldots,\sigma_r^2$

are nonvanishing eigenvalues of A^HA since $\sigma_1^2=\lambda_{\max}(A^HA)$

we also have $\sigma_1 \geq \sigma_2$.

The unitary matrices U and V have the following properties.

The columns of U are m orthonormal eigenvectors of A^HA and that of V are of AA^H .i.e.,one way of solving evp (1.1).

E. Reduction to simpler forms

The most common method of finding eigenvalues and eigenvectors of a dense matrix A is by finite number of similarity transformations

$$A = A_0 \rightarrow A_1 \rightarrow \rightarrow \ldots \rightarrow A_m \tag{1.19}$$

$A_i = T_i^{-1}A_{i-1}T_i$, $i = 1,2 \ldots m$.

One first transforms the matrix A into a matrix B of simpler form, by

$B = A_m = T^{-1}AT$, $T = T_1T_2\ldots T_m$ and then solve

$By = \lambda y$.

For $x = Ty = T_1T_2\ldots T_my$, since $B = A_m = T^{-1}AT$, we then have

$Ax = \lambda x$.

And at this we can apply numerical formula in (1.0) or methods in (unit-II).

The matrix B is similar to A and is chosen as follows

1. finding eigenvalues of B needs only few operations
2. the evp for B is less worse conditioned for B than for A.
3. Small changes (perturbations) in B do not impair the eigenvalues of B nor of A.That means

$B + \Delta B = T^{-1}(A + \Delta A)\,T$

And $\Delta A = T \dot{\Delta} B\ T^{-1}$

$||B|| = \text{Cond}(T)||A||$, for large Cond (T) >1, evp for B will be not well conditioned as for A. The most important reduction methods are Householder,Jacobi and Given's rotations which will be seen later. For these we use Householder matrix(unitary matrix(U), orthogonal matrix $T^{-1} = T^{H}$), Elimination matrix (E), Hessenberg matrices (H) and Permutation (P) matrices with the forms.

$$P = P^{-1} = \begin{bmatrix} 1 & & & \\ & . & & 0 \\ & & 1 & \\ 0 & & & . \\ & & & & 1 \end{bmatrix} \quad E = E^{-1} = \begin{bmatrix} 1 & & & & \\ & . & & & \\ & & 1 & & \\ & -l_{j+1,j} & & . & \\ & -l_{nj} & & & 1 \end{bmatrix} \qquad (1.20)$$

$$H = \begin{bmatrix} * & . & . & . & * \\ * & . & & & . \\ 0 & . & . & & . \\ . & . & . & . & . \\ 0 & 0 & & * & * \end{bmatrix} \quad U = \begin{bmatrix} 1-2w_1^2 & -2w_1w_2 & & -2w_1w_n \\ -2w_2w_1 & 1-w_2^2 & & -2w_2w_1 \\ & & & \\ -2w_nw_1 & -2w_nw_2 & & 1-2w_n^2 \end{bmatrix}$$

$U = U^{-1} = U^{T} = I\text{-}ww^{T}: U^2 = I$ (U is orthogonal and unitary)

And the final simple form will be Hermitian (Tridiagonal, Diagonal) or Jacobi matrix from which eigenvalues are easily obtained.

CHAPTER-TWO

Generalized eigenvalue problems and numerical methods of solution

2.1.**Power method (simple vector iteration):** Recall the generalized eigenvalue problem (evp) in (1.1).i.e., ($Ax = \lambda Bx$, and consider when $B = I$ as before) .Finding eigenvalues by characteristic polynomials is un reliable. Because the coefficients and the roots have to be determined in round arithmetic errors. Also, solving higher order determinant matrices and polynomials of higher degree is not easy. So we need advanced numerical methods.
One such is the **power method** by Von Mises.The simplest eigenvalue problem (evp) is to compute just the dominating eigenvalue along with its eigenvector. The power method is the simplest iterative method for this task.

1. It determines the eigenvalues one by one starting by the dominant one (the spectral radius of A).
2. It is important when we need to find only the one dominant eigenvalue or only one of its corresponding eigenvectors for practical cases like Markov chain's process related with transition matrices in statistics.And also where low frequencies are needed in physics.
3. We consider the case A is symmetric (eigenvalues are real), semi-simple matrix (non-difective) i.e, eigenvectors are linear independent(LI). But,the method holds in most cases or one uses an alternate method like QR.
4. The elementary divisors are assumed to be linear although it holds generally.(refer to Jordan decomposition theorem (sec-1.3.8 or Schur's theorem).
5. This method needs no factorization of A into QR, LR, LU.(see Sec-2.2,2.7)

Procedure: (a). Let $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots \lambda_n$ be the eigenvalues and $u_1, u_2, u_3, \ldots u_n$ be the corresponding eigenvectorsof A.

wlog, let also $|\lambda_1| > |\lambda_2| \geq |\lambda_3|, \ldots \geq \ldots \geq |\lambda_n|$.

i.e , λ_1 = dominant , λ_2 = subdominant.

(b). choose x (initial vector) $\neq 0$ and write it as $x = \sum c_i u_i$ (i = 1to n) as alinear combination of the eigenvectors or the u_i's of A.

(c) . premultiply x in (b) by $A^1, A^2, A^3, \ldots A^n$ (by powers of A) to get

$x_1 = Ax = \sum c_i \lambda^1_i u_i = A^1 x$

$x_2 = Ax_1 = \sum c_i \lambda^2_i u_i = A^2 x$ (i = 1 to n)

. . ..

. . .

$$x_m = Ax_{m-1} = A^m \sum c_i u_i = \sum c_i u_i (\lambda_i)^m = A^m x \tag{2.1}$$

c_i's are called normalization constants.

$$x_m = Ax_{m-1} = A^m x = \sum c_i \lambda_i^m u_i = \lambda_1^m (c_1 u_1 + \sum c_i (\lambda_i / \lambda_1)^m u_i : i = 2,3,\ldots n \tag{2.2}$$

If m is large enough, by iteration x_m tends to

$c_1 \lambda_1^m u_1 = A^m x = A(A^{m-1}x) = Ax_{m-1} = \lambda x_{m-1} = x_m$. i.e, dividing both sides of (2.2) by λ_1^m we get the result.

($c_1 \lambda_1^m u_1 = A^m x$ as a limit and $x_m = \lambda x_{m-1}$ as a limit)

Hence,the iteration will converge to u_1 if x has a component in direction of u_1. $\langle x, u_1 \rangle \neq 0$ or $c_1 \neq 0$.Therefore, λ_1 is obtained from the two successive iterations by dividing the corresponding elements of x_m and x_{m-1} ($x_m = \lambda x_{m-1}$) . As for u_1,it is directly proportional to x_m. ($x_m = \lambda_1^m c_1 u_1$ as a limit).

(d). if x and u_1 or x_1 are perpendicular,then the method will not converge in exact arithmetic and if they are parallel,then increasing sequence of vectors parallel to x_1 will be generated. The conditions occur when x is choosen blindly.

Von Mise's Theorem: If the matrix A has n linear independent eigenvectors (A is simple) and if the dominant eigenvalue is λ_1,then if x_0 has component in u_1 or x_1 direction then the $\lim A^m x_0 / \lambda_1^m = c_1 u_1$.

Proof:The need that x_0 has a component in the direction of u_1 or x_1 is assurance that , $c_1 \neq 0$. If y is any vector $\langle y, u_1 \rangle \neq 0$,then from $Ay = \lambda_1 y$ and (2.1) we get $\lambda_1 = \lim y^T x_{m+1}/ y^T x_m$ with $y^T x_{m+1} = y^T A\, x_m = \lambda_1 x_m$ called schwarz constants. We also can apply root test taking $a_n = (\lambda_{i/\lambda 1})^m$ and get $\lim(a_n)^{1/n}$ < 1 only if $\lambda_{i/\lambda 1} < 1$ for all i in N for convergence of the method.

The Power method in short (simple vector iteration in short).

1. The approximation begins with an initial guess vector x_0.
2. x_k's are normalized dividing by their largest component or their norm
3. The Rayligh sequence $x_k = A^k x_0$ is generated if x_0 and x_1 or u_1 are parallel .
4. The Rayligh quotient $q_k = x_k^T A x_k / \|x_k\|^2$ gives the approximate

dominant value (λ_1). This ratio is obtained from

$$\lambda_k x_k = A x_k$$

$$x_k^T \lambda_k x_k = \lambda_k \|x_k\|^2 = x_k^T A x_k \text{ or } \lambda_k = x_k^T A x_k / \|x_k\|^2 \tag{2.3}$$

putting $k = 1, \lambda_1 = x_1 A x_1/\|x_1\|^2$ .To obtain x_1,use

$\lim A^m x_0 / \lambda_1^m = \lim x_m / \lambda_1^m = c_1 x_1, c_1 \neq 0$ (Von Mise's theorem)

And $c_1 x_1$ is an eigenvector implies that x_1 is also an eigenvector.

Note that if $q = \langle x, Ax\rangle / \langle x, x\rangle = x^T A x / x^T x$, then

$\lambda_{max} = \max[q] = R(A) =$ the dominant eigenvalue.

Let A be Hermitian.Then A = UAU* and U is a unitary matrix.

Let Y = U*X.

i.e, $x^*Ax = x^*UDU^*x = Y^*DY = \sum_{j=1}^{j=k} \lambda_j |y_j|^2$

Hence, $\lambda_k y^*y < x^*Ax < \lambda_1 y^*y$: $y^*y = x^*x$

U is unitary. Inner product is preserved.

Therefore, $\lambda_k < x^*Ax/x^*x = q < \lambda_1 =$ dominant.

From this, $1 < (\lambda_k)^{-1} q) < \lambda_1/\lambda_k.$

$\lambda_k/\lambda_1 < 1$ tells convergence.

Example:2.1 Apply power method on a symmetric matrix $A = \begin{bmatrix} 5 & -2 & -4 \\ -2 & 2 & 2 \\ -4 & 2 & 5 \end{bmatrix}$

Choose $x^T = (1,0,0)$.

Then, $x_1 = Ax = (5,-2,-4)^T$

$$X_2 = Ax_1 = (45,-22,-44)^T \quad , \quad X_3 = Ax_2 = (445,-222,-444)^T$$

For large m, the iteration gives the convergence

$\lambda_1 \approx 445/45 \approx -222/-22 \approx -444/-44 \approx 10$

Hence, u_1 (the dominant eigenvector) is proportional to x_m and is found (using Von Mise's theorem or Rayligh quotient) to be $(2,-1,-2)^T$.

To obtain λ_2 we assume u′ is reciporical vector of u_1 i.e, $\langle u_1, u' \rangle = 0$ and $\langle u_i, u' \rangle$ is not zero for i is not equal to 1 or $\langle u_i, u'_i \rangle = 0$ but $\langle u_i, u_j \rangle$ is not zero for i and j not equal.They are linear independent (LI). And we apply power method to $A_1 = A - \lambda_1 u_1 u'^*$ and we take $u' = u_1 / \| u_1 \|$ for symmetric matrices.

And use $A_1 = A - \lambda_1 u_1 u_1^* / \| u_1 \|$

Choose $x^T = (0,1,0)$ and we will see that no dominant eigenvalue as $\lambda_2 = \lambda_3 = 1$

A_1 has the same eigenvalues and eigenvectors except we found λ_1 and u_1.

Now λ_2 is the dominant.No easy way to find u′ except the case of Hermitian or symmetric in which the power method is continuously applied.This is called deflation method.(See Sec-2.3 : QR or Jacobi's method) .And if A is not symmetric we consider Schur's normal form (decomposition) to get λ_1 and A_1.

mathematica "Eigensystems[A]" gives set of eigenvalues {1, 1, 10 } and set of eigenvectors { (1,0,1), (1,2,0), (2,-1,2) }.

Mathematica Subroutine (Power Method).

To compute the dominant value λ_1 and its associated eigenvector V_1 for the (n×n) matrix A. It is assumed that the n eigenvalues have the dominance property $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$.

```
PowerMeth[A0_, V0_, ε_, m_] :=
 Module[{A = N[A0], c1, count, err, λ, λ0, X, X0 = N[V0], Y},
  maxsize[W_] := Module[{w = Sort[W], msize},
    If[ Abs[w[[-1]]] ≥ Abs[w[[1]]],
     msize = w[[-1]],
     msize = w[[1]]];
    Return[msize]; ];
  norm[V_] := √(V.V);
  λ0 = 0;
  count = 0;
  While[ count ≤ m,
   count = count + 1;
   Y = A.X0;
   λ = maxsize[Y];
   X = (1/λ) Y;
   Print[NumberForm[λ, 6], " , ", PaddedForm[X, {6, 6}]];
   err = Max[{ Abs[λ - λ0], norm[X - X0]}];
   If[ err < ε, Return[{λ, X}] ];
   X0 = X;
   λ0 = λ; ];
  Return[{λ, X}]; ];
```

Example 1. Use the power method to find the dominant eigenvalue and eigenvector for the matrix $A = \begin{pmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{pmatrix}$.

Solution .For illustration purposes we will set the maximum number of iterations to be 50 and $\epsilon = 0.000001$.

$$A = \begin{pmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{pmatrix};$$

```
X0 = {1, 1, 1};
Print["A = ", MatrixForm[A], ",   X0 = ", MatrixForm[X0]];
{λ, X} = PowerMeth[A, X0, 0.000001, 50];
Print[""];
Print["A = ", MatrixForm[A]];
Print["The `dominant` eigenpair is"];
Print["λ = ", λ, ",   X = ", MatrixForm[X]];
```

$$A = \begin{pmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{pmatrix}, \quad X_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

```
12. , { 0.500000, 0.666667, 1.000000}
5.33333 , { 0.437500, 0.625000, 1.000000}
4.5 , { 0.416667, 0.611111, 1.000000}
4.22222 , { 0.407895, 0.605263, 1.000000}
4.10526 , { 0.403846, 0.602564, 1.000000}
4.05128 , { 0.401899, 0.601266, 1.000000}
4.02532 , { 0.400943, 0.600629, 1.000000}
4.01258 , { 0.400470, 0.600313, 1.000000}
4.00627 , { 0.400235, 0.600156, 1.000000}
4.00313 , { 0.400117, 0.600078, 1.000000}
4.00156 , { 0.400059, 0.600039, 1.000000}
4.00078 , { 0.400029, 0.600020, 1.000000}
4.00039 , { 0.400015, 0.600010, 1.000000}
4.0002 , { 0.400007, 0.600005, 1.000000}
4.0001 , { 0.400004, 0.600002, 1.000000}
4.00005 , { 0.400002, 0.600001, 1.000000}
4.00002 , { 0.400001, 0.600001, 1.000000}
4.00001 , { 0.400000, 0.600000, 1.000000}
4.00001 , { 0.400000, 0.600000, 1.000000}
```

<

```
4. , { 0.400000, 0.600000, 1.000000}
4. , { 0.400000, 0.600000, 1.000000}
4. , { 0.400000, 0.600000, 1.000000}
```

$$A = \begin{pmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{pmatrix}$$

The `dominant` eigenpair is

$$\lambda = 4., \quad X = \begin{pmatrix} 0.4 \\ 0.6 \\ 1. \end{pmatrix}$$

That is ϵ close to the dominant eigenvalue $\lambda = 4$ as a limit when n tends to infinity and corresponding eigenvector $X = \begin{pmatrix} \frac{2}{5} \\ \frac{3}{5} \\ 1 \end{pmatrix}$

The following problems arise in this method

1. How can we choose the starting vector x_0?.
2. Does the method always converge to the desired result?
3. Can we speed (accelarate) the method?

Convergence of power method

Depends on the factor (ratio) $(\lambda_i/\lambda_1)^m$ goes to 0.If this is not small slow convergence.The number of iterations to get a desired degree of convergence depends upon both rate of convergence and how large c_1 is which in turn depends on x_0.Hence, c_1 and in particular λ_2/λ_1 affect convergence. It is not fast convergent and also more limitted precision than inverse power method (sec-2.2) which is invoked to refine the numerical values and full precision.

Choice of the initial vector $x = x_0$

We can blindly choose $\mathbf{x_0}$ with c_1 not zero.But we choose usually 1 in the position of maximum component of x_m and 0 else where.Even the largest component of x_m can vary.It is the largest component of the eigenvector. λ_{max} is the ratio of successive largest components of x_m and x_{m-1} same as

$\lim x^T x_m / x^T x_{m-1}$ and $x^T x_m = x^T A x_{m-1}$ is called Schwartz constant.The x_m's are normalized (to make the largest component 1) before the next iteration.If x_0 is an eigenvector,then we need another choice for starting the iteration.

The power method fails due to the following.

1. The initial guess x_0 has no component in the direction of $u_1(c_{1=0})$.
2. If eigenvalues have the same maginitude (no dominant).Like matrix I_n. (we have to consider the case the eigenvalues are distinct).
3. complex eigenvalue(s) due to computational problems as eigenvalues may appear in conjugate pairs but the same in magnitude hence no dominant.

More examples on power method.

2.2a) Find the dominant eigenvalue of A.

$$A = \begin{bmatrix} 1 & 1 & .5 \\ 1 & 1 & .25 \\ .5 & .25 & 2 \end{bmatrix} \quad \text{and } x_0 = (1,1,1)^T$$

In the table below each x_m was calculated using (2.1) and then normalized to make its largest component 1 before the next iteration. λ is the normalizing factor and represents the ratio of the largest component of x_m to x_{m-1}. For example, λ_1 is ratio of third component of unnormalized x_1 to x_0. Where $x_m = A\,x_{m-1}$.These calculations were made using computer.The true $\lambda = 2.5365258$ and $x_1 = (.74822116,.64966116,1)$ is obtained at 28[th] iteration.

m	x_m(normalized)	λ	M	x_m(normalized)	λ
1.	(.9091,.8182,1)	2.75	16.	(.7483,.6497,1)	2.536584
5.	(.7651,.6674,1)	2.5587918	18.	(.7482,.6497,1)	2.5365456
10.	(.7494,.6508,1)	2.5380029	19.	(.7482,.6497,1)	2.5365374
15.	(.7483,.6497,1)	2.5366256	20.	(.7482,.6497,1)	2.5365323
			28	(.7482,.6497,.1)	2.5365258

Table 2-1

The table shows eigenvalues and eigenvectors computed at some iterations using (2.1)

2.2b). Apply power method to the above (2.2a) for $x_0 = (-.64966116,.7482216,0)^t$

which is orthogonal to x_1.Since A is symmetric x_0 has no component in the direction of x_1.Initially the iteration converges to a value nearly equal to $\lambda_2 = 1.4801215$ since $|\lambda_2/\lambda_3|$ is large.

M	X_m(normalized)	λ	M	X_m(normalized)	λ
1.	(-.7154,-.7154,1)	-.1377753	20	(-.6356,-.8044,1)	1.4807007
5.	(-.6369,-.8058,1)	1.4801216	30	(-.4008,-.5576,1)	1.5931941
10.	(-.6369,-.8058,1)	1.4801240	40	(.7180,.6179,1)	2.4976711
15.	(-.6368,-.8057,1)	1.4801606	50	(.7481,.6495,1)	2.5363412

Table2-2

The table shows eigenvalues and eigenvectors computed at some iterations using (2.1)

Using mathematica, "EigenSystems[A]", eigenvalues are; 2.53653 ,1.48012, -0.0166473 and eigenvectors are; (-.531483,-.461473,-.710329), (-.444281,-.562109,-.710329), (.721207,-.686349,-.093728).

Accelaration of rate of convergence.

(a) **The δ^2 process** :In this application we assume that λ_1,λ_2 are real eigenvalues and $-\lambda_1, -\lambda_2$ are not eigenvalues.

Let e_i be the i^{th} column of the identity matrix I ,then the schwarz constant

$$e_i^T x_m = \sum_{i=1}^{n} c_i \lambda_i^m .$$

$$e_i^T x_{m+1} / e_i^T x_m = (c_1\lambda_1^{m+1} + \sum_{i=2}^{n} c_i \lambda_i^{m+1}) / (c_1\lambda_1^m + \sum_{i=2}^{n} c_i \lambda_i^m)$$

Dividing numerator and denominator by $c_1\lambda_1^m$ and expanding the denominator in power series, we get

$e_i^T x_{m+1} / e_i^T x_m = \lambda_1 + \beta_i (\lambda_2/\lambda_1)^m$ +[terms in $(\lambda_3/\lambda_1)^m \ldots (\lambda_n/\lambda_1)^m$ and higher power].

If we are nearer to convergence,the terms in brackets are small and we have

$\lambda_1 \approx R_m - \beta_i r^m$.

$R_m = e_i^T x_{m+1} / e_i^T x_m$ and $r = \lambda_2/\lambda_1$

We get a better approximation for $\lambda_1 \approx R_{m+2} - (\Delta R_{m+1})^2 / \Delta^2 R_m$

If we put m=10 and i = 3 in example (2.2a), we get $\lambda_1 \approx 2.5365266$

Which is better than achieved after 20 iterations.

b.Rayleigh quotient (for symmetric matrices)

If A is a symmetric matrix

then the eigenvectors are orthogonal and if we need them to be

orthonormal , $x^T_m A x_m = x^T_m x^T_{m+1} = \sum_{i=1}^{i=n} c_i^2 \lambda_i^{2m+1}$ (2.4).

since $||x_m|| = 1$.

$$x^T_m x_m = \sum_{i=1}^{i=n} c_i^2 \lambda_i^{2m}$$

(2.5)

and we can write as $x^T_m A x_m / x^T_m x_m = x^T_m x^T_{m+1} / x^T_m x_m$

$= \lambda_1 + O(\lambda_i / \lambda_1)^{2m}$

by comparison for arbitrary vector y

$y^T x_{m+1} / y^T x_m = \lambda_1 + O(\lambda_i / \lambda_1)^m$ (2.6)

since higher order terms are smaller it gives better approximation for λ_1 than power method. Consider m = 11 in example (2.2a),then

$x^T_{11} = (.74888011, .65035358, 1)$

And $x^T_{12} = (.74860561, .65006512, 1)$. For these vectors

$\lambda_1 \approx 2.5365256$ (obtained at 12th iteration) which is better

approximation than δ^2-process.It converges quadratically .

Note: the error in power method is $e = \max|\lambda - \lambda_0| = |x - x_0|$

2.2. Inverse iteration (Inverse power (Weilandt's) method).

1. Refinement of power method and solves problems of power method.
2. It may need factorization of A into LU (see Sec-2.7).
3. used to find the smallest eigenvalue starting with initial guess to it called shift.
4. it is a power method applied on A^{-1} but faster than it.

5. The properties of A are as before except A has to be invertible.

If A is a nonsingular matrix with eigenvalues $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_n$, then A^{-1} has eigenvalues $\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1}, \ldots, \lambda_n^{-1}$.

Hence, if A-vI is nonsingular then the eigenvalues of $(A-vI)^{-1}$ are $(\lambda_1-v)^{-1}, (\lambda_2-v)^{-1}, \ldots, (\lambda_n-v)^{-1}$. (see theorem:1.1)

If we now apply power method to $(A-vI)^{-1}$ and starting with x_0 as before, (as in power method),

$$x_m = (A-vI)^{-m} x_0 = \sum_{i=1}^{i=n} c_i\,(\lambda_i - v)^{-m} x_i \tag{2.7}$$

If now v which is called the shift is close to one of the distinct eigenvalues λ_j, which may be a multiple eigenvalue but which we assume of multiplicity 1, and if λ_j is separated from all other eigenvalues so that $|\lambda_j - v| < |\lambda_i - v|$, then the term $c_j(\lambda_j-v)^{-m}x_j$ will dominate the sum in (2.7) for large m.The vector x_m then becomes the best approximation for the eigenvector of λ_j.This shows that v is very close to λ_j.

The inverse power method is applied in two different contexts.

I. The first is in **an iterative method** to find eigenvalues and eigenvectors using generalized Rayleigh quotient iteration (2.6) or (2.7).

We compute the Rayleigh quotient as follows

$\mu_1 = x^T_o A x_0 / x_0^T x_0$ followed by

$x_1 = k_1(A-\mu_1 I)^{-1} X_0$ where k_1 is a normalizing factor μ_1 choosen so that $\|x_1\| = 1$ and μ_1 plays the role of v.

Ingeneral ,we compute $\mu_{i+1} = x^T_i A x_i / \|x^T_i\|$

and $x_{i+1} = k_{i+1}(A-\mu_1 I)^{-1} x_i$.

Under appropriate hypotheses it can be shown that the sequence $\{\lambda_i , x_i\}$ converges to an eigenvalue –eigenvector pair $\{\lambda, v\}$ which depends on the initial approximation x_0.In practice,the inverse $(A-\mu_1 I)^{-1}$ is not computed explicitly.Instead,the set of linear equations

$(A-\mu_{i+1} I)z_{i+1} = x_i$ is solved, then we use

$x_{i+1} = k_{i+1} z_{i+1}$ (2.8)

Even though the coefficient matrix (A- μ_i I) approaches singularity as μ_i approaches an eigenvalue, this does not detract from the accuracy of the calculation provided that partial pivoting is used for (2.8).

II. The second context is used to find an eigenvector corresponding to a calculated eigenvalue.We use LU decomposition as follows. We assume that we have computed an accurate approximation ξ to eigenvalue λ and wish to compute the corresponding eigenvector x .

Since $\xi \neq \lambda$, (A- ξ I) is non-singular there is a permutation matrix P obtained by interchanging rows or columns of I_n such that p(A- ξ I) = LU and $P^2 = I$ and $P^{-1} = P$. So we have

$$P(A-\xi I) = LU \qquad (2.9).$$

We now wish to solve (A- ξ I)$z_1 = P^{-1}LUZ_1 = x_0$.

One way which has been very successful in practice is to choose x_0 so that $L^{-1}px_0 = u = (1,1,1,\ldots.,1)^T$ and we start by solving $Uz_1 = u$.

After we normalize z_1 we apply one more iteration involving forward and backward substitution.And obtain z_2 a very accurate approximate to the eigenvector x.This method (inverse power) will let to converge to the shift v not to λ_{max} as before (power method). It has a linear convergence.

Example-2-3

Apply inverse power method to find an eigenvector x of

$$A = \begin{bmatrix} -120 & 90.86 & 0 & 0 \\ 90.86 & -157.2 & -67.59 & 0 \\ 0 & -67.59 & -124 & 46.26 \\ 0 & 0 & 46.26 & -78.84 \end{bmatrix}$$ with a shift ξ = -253.601.

A is symmetric and in a tridiagonal form.

The LU factorization of A- ξ I is

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ .66804 & -.56387 & .14800 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 136.01 & 90.86 & 0 & 0 \\ 0 & -67.59 & 132.01 & 46.26 \\ 0 & 0 & 46.26 & 177.17 \\ 0 & 0 & 0 & -.13653 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Set $u = (1,1,1,1)^T$. Solve

$Uz_1 = u$, gives $z_1 = [-.33254,.4979,.28067,-.73244]^T$

Solve then $LUz_2 = pz_1$ to obtain $z_2 = [-.2953.3,.44205,.24915,-.65059]^T$.

ξ is an approximation to the true $\lambda = -255.9996$.The eigenvector to this is $x = [-.6681,1,.56363,-.14718]^T$.A similar normalization of z_2 gives $x' = [-.66809,1,.56362,-.14718]^T$ which agrees with x to within rounding error.

Mathematica "EigenSystems[A]", gives eigenvalues ; -255.9996 ,-144.025, -64.32 ,-16.0128 and eigenvectors ; (-.49996,.7483,.4219,-.1101), (-.5000,.1322,-.6979,.4953), (.5000,.5724,-.5233,-.3853). (-.4999,-.3080,-.2472,-.7706)

2.3.The Jacobi method (Eigenvalues of symmetric matrices)

Jacobi's method is an easily understood algorithm for finding all eigenpairs for a symmetric matrix. It is a reliable method that produces uniformly accurate answers for the results. For matrices of order up to 10×10, the algorithm is competitive with more sophisticated ones. If speed is not a major consideration, it is quite acceptable for matrices up to order 20×20. A solution is guaranteed for all real symmetric matrices when Jacobi's method is used. This limitation is not severe since many practical problems of applied mathematics and engineering involve symmetric matrices. From a theoretical viewpoint, the method embodies techniques that are found in more sophisticated algorithms. For instructive purposes, it is worthwhile to investigate the details of Jacobi's method.

We shall use orthogonal (unitary) matrices for similarity transformation of the matrix A with an infinite sequences of transformations of the general form (1.19)

$A^0 = A$ and $A^i \rightarrow A^{i+1} = (T^i)^H A^i T^i$ converging not to a tridiagonal but to a diagonal matrix D,

$$D = \begin{bmatrix} \lambda_1 & & \\ & \cdot & \\ & & \lambda_n \end{bmatrix}$$

Theorem-2-1:If A is symmetric matrix ,then there is an orthogonal matrix Q such that $Q^TAQ = D$ is a diagonal matrix with eigenvalues the diagonal elements.

Our technique is to find a sequence $\{s_k\}$ of orthogonal matrices with the property that

$\lim S_1.S_2 \ldots S_K = Q$ as k tends to infinity.

We shall use notation $T_K = S_K.S_{K-1} \ldots S_1 A S_1.S_2 \ldots S_K.T_0 = D_K$. (2.10)

We denote the elements of T_k by $t^{(k)}_{ij}$ and of S_k by $s^{(k)}_{ij}$

Where S_k has the form $\begin{bmatrix} 1 & . & 0 & 0 & . & 0 \\ 0 & . & \cos\theta & \sin\theta & . & 0 \\ . & . & -\sin\theta & \cos\theta & . & . \\ 0 & . & . & . & . & 0 \\ . & . & . & . & . & . \\ 0 & . & 0 & 0 & 0 & 1 \end{bmatrix}$ with θ an angle of rotation

is a rotation matrix is an orthogonal matrix.

Define $v_k = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} t_{ij}^{(k)2}$; k = 0,1,…and $i \neq j$ (2.11)

$$w_k = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} (t_{ij}^{k})^2 : \quad k = 0,1,2,\ldots.n \tag{2.12}$$

Here $||T_k||^2 = w_k$ and v_k is sum of square of off-diagonal elements of T_k.

Our object is to choose the sequence $\{S_k\}$ so that

$W_{k+1} = w_k$ and $v_{k+1} < v_k$ for all k (2.13)

And $\lim v_k = 0$. In which case $\lim T_k = D$.

Let t^{k-1}_{pq} be a non-zero off diagonal element of T_{k-1}.

We wish to choose S_k so that $t^k_{pq} = 0$.We sall show that (2.13) is satisfied.

Let $s^k_{pp} = s^k_{qq} = \cos\theta_K$,$s^k_{ii} = 1$,$i \neq p,q$. (2.14)

$S^k_{pq} = -s^k_{qp} = \sin\theta_K$,$s^k_{ij} = 0$.

The orthogonal matrix defined by (2.14) is called a plane rotation matrix since the linear transformation defined by S_k consists of a rotation of the axes of the P^{th} and q^{th} coordinates via an angle θ_K .From (2.10)

$T_k = S^T{}_k T_{K-1} S_{k.}$.

So we have using (2.14) the rotation formula;

$t^k_{pj} = t^{k-1}_{pj}\cos\theta_K - t_{qi}{}^{k-1}\sin\theta_K$, $j \neq p, q$. (2.15)

$t_{qj}{}^k = t_{pj}{}^{k-1}\sin\theta_K + t_{qj}{}^{k-1}\cos\theta_K$.

$t_{ip}{}^k = t_{ip}{}^{k-1}\cos\theta_K - t_{ip}{}^{k-1}\sin\theta_K$. $i \neq p,q$ (2.16)

$t_{tq}{}^k = t_{tp}{}^{k-1}\sin\theta_K + t_{tq}{}^{k-1}\cos\theta_K$

$t_{pp}{}^k = t_{pp}{}^{k-1}\cos\theta_K{}^2 + t_{qq}{}^{k-1}\sin\theta_K{}^2 - 2t_{pq}{}^{k-1}\sin\theta_K\cos\theta_K$ (2.17)

$t_{qq}{}^k = t_{qq}{}^{k-1}\cos\theta_K{}^2 + t_{pp}{}^{k-1}\sin\theta_K{}^2 - 2t_{pq}{}^{k-1}\sin\theta_K\cos\theta_K$ (2.18)

$t_{pq}{}^k = t_{qp}{}^k = 1/2(t_{pp}{}^{k-1} - t_{qq}{}^{k-1})\sin 2\theta_K + t_{pq}{}^{k-1}\cos 2\theta_K$.

$t_{ij}{}^k = t_{ij}{}^{k-1}$,i is not equal to p, j is not equal to q (2.19)

Now we shall choose θ_K to make $t_{pq}{}^k$ (off diagonal elements) vanish. By (2.19), we get $\alpha = \cot 2\theta_K = (t_{qq}{}^{k-1} - t_{pp}{}^{k-1})/2t_{pq}{}^{k-1}$

(2.20)

i.e, θ_K always exists .In practice we do not calculate θ_K ,but $\sin\theta_K$ and $\cos\theta_K$ required by (2.15), (2.16), (2.17).

The calculation is as follows.

Let $T = \tan\theta_K$ .Using the relation(identity)

$\tan^2\theta_K + 2\tan\theta_K \cot 2\theta_K - 1 = 0$,choose T

a smaller root (corresponding to smaller angle of rotation $\theta_K \leq \Pi/4$)

of $t^2 + 2At - 1 = 0$ (21)

Put $A = \cot 2\theta_K$,and $t = \tan\theta_K$.

$T = 1/(|A| + \sqrt{(1+A^2)})\text{sign}(A)$. (22)

then $C = \cos\theta_K = 1/\sqrt{(1+T^2)}$,and

$S = \sin\theta_K = TC$. (23)

Equations (2.15 to (2.17) can be formulated to give the following equations which give more accurate results in computations by expressing all elements of T_k as perturbations of elements of T_{k-1}.

$t_{pp}{}^{k} = t_{pp}{}^{k-1} - h$, $t_{qq}{}^{k} = t_{qq}{}^{k-1} + h$, $t_{pq}{}^{k} = 0$ (24)

$t_{jp}{}^{k} = t_{pj}{}^{k} = t_{pj}{}^{k-1} - S(t_{qj}{}^{k-1} + T' t_{pj}{}^{k-1})$.

$t_{jq}{}^{k} = t_{qj}{}^{k} = t_{qj}{}^{k-1} + S(t_{pj}{}^{k-1} - T' t_{qj}{}^{k-1})$. $T' = (\tan\theta_K)/2 = s/(1+C)$.

and $h = T t_{pq}{}^{k-1}$. (25)

Using (2.15,2.16,2.17),we can easily calculate that,independent of θ_k,

$(t_{pj}{}^{k})^2 + (t_{qj}{}^{k})^2 = (t_{pj}{}^{k-1})^2 + (t_{qj}{}^{k-1})^2$

and $(t_{ip}{}^{k})^2 + (t_{iq}{}^{k})^2 = (t_{ip}{}^{k-1})^2 + (t_{iq}{}^{k-1})^2$

$(t_{pp}{}^{k})^2 + (t_{qq}{}^{k})^2 + (t_{pq}{}^{k})^2 + (t_{qp}{}^{k})^2$

$= (t_{pp}{}^{k-1})^2 + (t_{qq}{}^{k-1})^2 + (t_{pq}{}^{k-1})^2 + t_{qp}{}^{k-1})^2$ (26)

But with θ_K chosen as in (2.20), $t_{pq}{}^{k} = t_{qp}{}^{k} = 0$ (off-diagonal elements vanish). Therefore, 26 shows that $w_{k-1} = w_k$ and $v_k < v_{k-1}$.

Infact 26 implies since $t_{pq}{}^{k-1} = t_{qp}{}^{k-1}$ as all T_k are symmetric,the off-diagonal sum of squares is reduced by $2(t_{pq}{}^{k-1})^2$ and the square sum of diagonal elements is increased by $2(t_{pq}{}^{k-1})^2$.

The Jacobi iteration in short:

1. choose a nonzero off diagonal element
2. calculate $\sin\theta_k$ and $\cos\theta_k$ from(23)
3. calculate those elements in T_k which differ from those in T_{k-1} using (24).
4. note that an off-diagonal element made zero at one stage will

generally become non-zero at some later stage. Our main objective is

to make them zero.

The process can be continued as long as desired till off-diagonal elements are 0.

5. convergence of the method depends on choosing off-diagonal elements (large in magintude).($\sqrt{v_k}$) will be reduced.
6. The limit of sum of the off-diagonal elements converges to 0 since it decreases monotonically, hence convergent.The method converges quadratically when the transformation is excuted in suitable order and the sum converges to 0.
7. the eigenvectors of A are readily calculated using this iteration.
8. the columns of the orthogonal matrix Q used to reduce A to a diagonal matrix D obtained after convergence) are the eigenvectors

of A.To calculate these eigenvectors ,we must calculate the product of S_k matrices in (2.10).This can be done as

$T_k = R_k^T A R_k$ (from 1.19,1.20)

$R_k = S_1.S_2\ldots S_k$.

Thus,$R_{K+1} = R_k S_{k+1}$.And from (2.14),we can calculate the elements of $R_{k+1} = [r_{ij}^{k+1}]$ in terms of those of $R_k = [r_{ij}^{k}]$:

$r_{ip}^{k+1} = r_{ip}^{k}\cos\theta_k - r_{iq}^{k}\sin\theta_k$

$r_{iq}^{k+1} = r_{ip}^{k}\sin\theta_k + r_{iq}^{k}\cos\theta_k$

$r_{ij}^{k+1} = r_{ij}^{k}$ starting with $R_1 = S_1$. (27)

9. the other problem in the use of Jacobi method is how to choose the off-diagonal element to be annihilated at each stage.We choose that

with greatest maginitude as it results in the greatest reduction of the off-diagonal norm $\sqrt{v_k}$ at all stages.

10. it is more convenient for symmetric matrices to annihilate all the subdiagonal elements in serial order,starting with the element at the (2, 1),

proceeding down the first column, continuing with the (3,2) element, etc.This is called the serial jacobi method and can be shown to converge provided the rotation angle θ_K satisfies $|\theta_K| < 45^{\circ}$.

11. it is used to give **all eigenpairs** for symmetric matrices.

Example:2.4 Apply Jacobi's method on the matrix of example-2.2a

The largest off-diagonal element is 1 i.e ,at (1,2)

Using (23) and (24), $\sin\theta_1 = \dfrac{-1}{\sqrt{2}}$; $\cos\theta_1 = \dfrac{1}{\sqrt{2}}$.

Using (25),we compute $T_1 = S_1^T A S_1 = \begin{bmatrix} 2 & 0 & 3/4\sqrt{2} \\ 0 & 0 & -1/4\sqrt{2} \\ 3/4\sqrt{2} & -1/4\sqrt{2} & 2 \end{bmatrix}$

By continuing the process we obtain

$$\lim T_k = D = \begin{bmatrix} 2.5365258 & O & 0 \\ 0 & -.0166473 & 0 \\ 0 & 0 & 1.4801215 \end{bmatrix}$$

We read the eigenvales simply from the diagonal of D.

And using (27) and (2.10)

$$\lim S_1.S_2\ldots S_K = Q. = \begin{bmatrix} .53148338 & -.72120712 & -.44428106 \\ .46147338 & .68634982 & -.56210938 \\ .71032933 & .09372796 & .69760117 \end{bmatrix}$$

mathematica "Eigensystems[a]" gives eigenvalues 2.53653 ,1.48012, -0.0166473 and eigenvectors (-.531483,-.461473,-.710329), (-.444281,-.562109,-.710329), (.721207,-.686349,-.093728).

Columns of Q are eigenvectors as A= QAQ^{-1}.

12. the accuracy of Jacobi method depends on how accurately the roots of $\sin\theta_K$ and $\cos\theta_K$ in (23) and (24) are calculated and how round

off errors accumulate.Stability will occur if these values are correct with less round off error.

13. the Jacobi method has been widely applied on digital computers to find eigenvalues and eigenvectors of symmetric matrices of higher order n with simple compact program.It yeilds all eigenvalues and eigenvectors.Columns of Q are eigenvectors as $A = QAQ^{-1}$.

14. The main disadvantage is that it is an infinite iterative method(sequence) with a lengthy computation (not tridiagonalization but diagonalization).

2.4.Given's method (Reduction of Hermitian to tridiagonal form).

The chain of transformations is as in (1.19)

$$A = A_0 \rightarrow A_1 \rightarrow \rightarrow \ \ldots \rightarrow A_m.$$

$A_i = T_i^{-1} A_{i-1} T_i$, $i = 1,2\ldots m$ with the form S_k or

$$T_i = \begin{bmatrix} 1 & & & & 0 \\ & \cos\varphi & & e^{-i\psi}\sin\varphi & \\ & & 1 & & \\ & e^{i\psi}\sin\varphi & & \cos\varphi & \\ 0 & & & & 1 \end{bmatrix}$$

a unitary or an orthogonal to be used as a tansformation matrix in the processes.

This method involves two methods both of which are finite process.The basis of this method is to use an orthogonal matrix not to diagonalize A, but to tridiagonalize A, i.e ,to reduce A to a form in which the only non-zero elements are on the main diagonals directly above or below it, as shown by fig-2-4 below.We will then consider how to find the eigenvalues of such matrix.

$$\begin{bmatrix} a_{11} & a_{12} & o & o & o \\ a_{21} & a_{22} & a_{23} & o & o \\ o & a_{32} & & & o \\ 0 & 0 & & & a_{n-1,n} \\ o & 0 & o & a_{n,n-1} & a_{nn} \end{bmatrix}$$

fig:2-4 a tridiagonal matrix

The orthogonal (unitary) matrices S_k used in the Jacobi method had the property that only the p^{th} and q^{th} rows and columns of T_{k-1} were changed in calculating

$T_k = S^T{}_k T_{k-1} S_k$.

We can annihilate off-diagonal elements while keeping previously annihilated elements zero.First note S_k as defined in (2.10) with θ_k unspecified as yet) ,we can annihilate, instead of $t_{pq}{}^{k-1}$, $t_{rq}{}^{k-1}$ with $r \neq p$ or q (and by symmtry $t_{qr}{}^{k-1}$ is annihilated also). This follows by (2.15)

$t^k{}_{rq} = t^{k-1}{}_{rp}\sin\theta_k + t^{k-1}{}_{rq}\cos\theta_k$ (is a rotation formula)

which will be satisfied if $\sin\theta_k = -\alpha\ t^{k-1}{}_{rq}$ and $\cos\theta_k = \alpha\ t^{k-1}{}_{rp}$

with $\alpha = 1/[\ t^{k-1}{}_{rp}{}^2 + t^{k-1}{}_{rq}{}^2]^{1/2}$ as $(\sin\theta_k)^2 + (\cos\theta_k)^2 = 1$ (28)

Let us denote the matrix whose elements are given by (2.10) with $\sin\theta_k$ and $\cos\theta_k$ as in (28) by the triplet (p,q,r) these three distinct integers denoting the row (column) of significance in the transformation. Inparticular we consider the sequence of

transformations $(p,q,r) = (2,i,1) \quad i = 3,4,\ldots,n.$ (29)

applied in succession to the original matrix A.Each triplet defines a transformation which we apply by premultiplying the current matrix by the transpose of the matrix in (2.10) and postmultiplying by the matrix itself.Denoting this matrix by S_{pqr}, we have after the firt step

$S^T{}_{123} A S_{231}$ (30)

Thus,the transformation (2,3,1) annihilates the element in the first row (column) with p and q in (2.10) being 2 and 3,respectively.In general the (2,i,1) transformation annihilates the element in the first column (row) and i^{th} column (row) and each annihilated element remains zero since the (2,i,1) transformation changes only the second row (column)

and the i^{th} column (row) but does not affect the previously annihilated elements.Therefore, after the sequence of transformation in (29) all elements in the first row (column) except the first two(a_{11} and a_{12}) are zero.

Next we consider the sequence (p,q,r) = (3,i,2) i = 4,…,n. (31)
which annihilates the elements of the second row (column) from a_{24} to a_{2n} .By the same case as before and using (31),this leaves all annihilated elements in the first and second rows (columns) zero.Our general algorithm is that follows;

(p,q,r) = (j,i ,j-1) j = 2,…n i = j+1,…,n. (32)

and in so doing we get a new symmetric matrix B with the form

$$B = \begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 \\ c_1 & b_2 & c_2 & 0 & 0 \\ 0 & c_2 & & & 0 \\ 0 & 0 & & & c_{n-1} \\ 0 & 0 & 0 & c_{n-1} & b_n \end{bmatrix} \text{ a tridiagonal form} \qquad (33)$$

From (33) the number of transformations required is (n-2).(n-1)/2, and the total number of operations required is of the order of $4/3(n^2)$ plus, (n-2).(n-1)/2 square roots required by (28).Here again(second process),we shall show that B has a form effective to find its eigenvalues.We may then apply QR method or consider the matrix

$$\lambda I - B = \begin{bmatrix} -b_1+\lambda & -c_1 & 0 & 0 & 0 \\ -c_1 & -b_2+\lambda & -c_2 & 0 & 0 \\ 0 & -c_2 & & & 0 \\ 0 & 0 & & & -c_{n-1} \\ 0 & 0 & 0 & -c_{n-1} & -b_n+\lambda \end{bmatrix} \qquad (34)$$

If we denote the principal minor of order i of (34) by $f_{n-i}(\lambda)$,we can easily show that by using Det(λI -B) we get

$f_{n-(i+1)}(\lambda) = (\lambda - b_{i+1}) f_{n-i}(\lambda) - c^2_i \, f_{n-(i-1)}(\lambda)$ i = 1,…,n.

$f_n(\lambda) = 1$ and $f_{n-1}(\lambda) = -b_1 + 1$.

The characterstic equation is $f_0(\lambda) = 0$. (35)

we can solve (35) by Newton's method or any formula in (1.0)

(since A is symmetric eigenvalues are real)

$f_0, f_1, f_2, \ldots$ is called sturm sequence.

And if $f_k(x) = 0$,then $f_{k-1}(x).f_{K+i}(x) < 0$.

This helps us for guessing existence of roots to apply Newton's method .

Example 2.5: Apply Givens' method to the matrix A of Example 2.2a

The only off-tridiagonal element is a_{13} , so that there is only one transformation to perform.

From (28) with p = 2, q = 3, r = 1

$$\alpha = \frac{2}{\sqrt{5}}\ ,\ \sin\theta\ = -\frac{1}{\sqrt{5}}\ ,\cos\theta = \frac{2}{\sqrt{5}}$$

Using these to get the matrix S of (33), we then calculate

$$B\ =\ S^T AS\ =\ \begin{bmatrix} 1 & \frac{\sqrt{5}}{2} & 0 \\ \frac{\sqrt{5}}{2} & 1.40 & .55 \\ 0 & .55 & 1.60 \end{bmatrix}$$ is a tridiagonal

Then from (34) we get from minors of λI -B

$f_3(\lambda) = 1$, $f_2(\lambda)\ = \lambda$-1 ; $f_1(\lambda) = \lambda^2 - 2.4\lambda + .15$

$f_0(\lambda) = \lambda^3 - 4\lambda^2 + 3.6875\lambda + .0625$

with $f_0(\lambda)$ the characteristic equation of A.

$f_1(\lambda)$, $f_3(\lambda)$, $f_3(\lambda)$ is sturm sequence.1 is a root of $f_2(\lambda)$ as $f_2(1) = 0$.

$F_3(\lambda) = 1 > 0$, $f_1(1) < 0$.

$\Rightarrow \quad F_3(1).f_1(1) < 0$

Then we can apply newton's method $\lambda_{n+1} = \lambda_n - f(\lambda_n) / f'(\lambda_n)$

to $f_0(\lambda) = 0$ to obtain the roots(eigenvalues of A) .These are

2 .53653 ,1.48012, -0.0166473

2-5 Householder's method (Reduction to tridiagonal matrix)

This method, a variation of Givens' method, enables us to reduce A to tridiagonal form with about half as much computation as Givens' method requires. The technique is to reduce a whole row and column (except for the tridiagonal elements) to zero at a time.

Let v be a vector such that $v^T v = 1$ (36)

Then it is easy to see that the matrix $P = I - 2vv^T$ is orthogonal and symmetric. In particular we choose v_k to be a vector whose first $k-1$ components are zero, so that

$$V_k^T = (0,0,0,\dots, v_k^{(k)}, v_k^{(k+1)}, \dots, v_n^{(k)}) \qquad (37)$$

Then with $P_k = I - 2v_k v_k^T$

We define $A_k = P_k^T A_{k-1} P_k \quad k = 2, \dots, n-1$, and $A_1 = A$

Now suppose that the symmetric matrix $A_{k-1} = [g_{ij}]$ has zeros in its first $k-2$ rows and columns except for the tridiagonal elements:

$$A_{k-1} = \begin{array}{l} \\ \\ \\ \\ \\ \\ \text{row } k\text{-}1 \\ \\ \\ \\ \end{array} \begin{bmatrix} g_{11} & g_{12} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ & & \ddots & & & & & \vdots \\ g_{12} & g_{22} & g_{23} & \ddots & & & & \vdots \\ & \ddots & \ddots & \ddots & & & & \vdots \\ 0 & g_{23} & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & g_{k-2,k-2} & g_{k-2,k-1} & 0 & \cdots & 0 \\ \vdots & & \ddots & g_{k-2,k-1} & g_{k-1,k-1} & \cdots & \cdots & g_{k-1,n} \\ \vdots & & & \ddots 0 & \vdots & \ddots & & \vdots \\ \vdots & & & \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & g_{k-1,n} & \cdots & \cdots & g_{nn} \end{bmatrix} \qquad (38)$$

The matrix P_k has the form

$$P_k = I - 2v_k\, v_k^T = \begin{bmatrix} 1 & 1 & \cdots & \cdots & \cdots & 0 \\ \ddots & \ddots & & & & \vdots \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & 1-2\left(v^{(k)}{}_k\right)^2 & \cdots\cdots & -2v^{(n)}{}_k v^{(k)}{}_k \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 \cdots & \cdots & 0 & -2v^{(n)}{}_k v^{(k)}{}_k & \cdots\cdots & 1-2\left(v^{(n)}{}_k\right)^2 \end{bmatrix} \qquad (39)$$

Using (36) to (39) we can verify that A_k has zeros in the positions shown as zero for A_{k-1}. Our object is to choose the $n-k+1$ numbers $v_k^{(k)}, \dots, v_k^{(n)}$ to satisfy (36) and so that the $n-k$ off-tridiagonal elements in row (column) $k-1$ of A_k are zero. We define $S = \sum_{j=k}^{n} g_{k-1,j}^2$ (40)

and then let $\left(v_k^{(k)}\right)^2 = \frac{1}{2}\left(1 \pm \frac{g_{k-1,k}}{\sqrt{S}}\right)$ and $v_k^{(j)} = \pm\frac{g_{k-1.j}}{2v_k^{(k)}\sqrt{S}}$, $j=k+1,\dots,n$

where the plus or minus sign will be chosen below. The motivation for (40) can be found in the algebra leading to the proof that the desired n – k elements in the (k – 1) st row (column) of A_k are zero and that (41) is satisfied.Proceeding as above at each step, we arrive at a tridiagonal matrix A_{n-1}.

The accuracy of this method depends naturally on the accuracy of the matrices P_k, and these in turn depend upon the accuracy of the components of (39). The key to making this accuracy as great as possible is to make $v_k^{(k)}$ as accurate as possible, which we do by choosing the sign in (40) to be that of g_{k-1}, thus avoiding a possible loss of significant figures resulting from the subtraction of almost equal quantities. We then use the same sign in (40). The total number of operations is of the order of $\frac{2}{3}n^3$, compared with $\frac{4}{3}n^3$ for Given's method,it is twice. At each stage it would appear that two square roots are required one for $\sqrt{S}$ and one for $\left[\left(v_k^{(k)}\right)^2\right]^{1/2}$. However, by arranging the calculations properly, the latter or these two need not be calculated. Therefore, Householders method requires (n – 2) square roots compared with (n– 2)(n-1)/2 for Givens' method. For large matrices, then, Householder's method is a more efficient way than Givens' method to reduce a symmetric matrix to tridiagonla form. The discussion in Sec. 2-4 on finding the eigenvalues and eigenvectors of tridiagonal matrices also applies here. Calculation of the eigenvectors of A from the eigenvectors of a tridiagonal matrix found using Householder's method is considered in QR method sec.2-7.

Example 2-6; Apply Householder's method to the matrix of Example 2.2a

As in example:2-4, there is only one step to perform.

We have $S = 1^2 + \left(\frac{1}{2}\right)^2 = \frac{5}{4}$, $\sqrt{S} \approx 1.11803$

Since $a_{12} = 1$, we choose the + sign in (41) and get

$$v_2^{(2)} \approx \left[\frac{1}{2}\left(1+\frac{1}{1.11803}\right)\right]^{1/2} \approx .97325\ ,$$

$$v_2^{(3)} \approx \frac{1}{4\times(1.11803)\times .97325} \approx .22975$$

(In fact the calculation of the square root required to get $v_2^{(2)}$ can be avoided, as mentioned above . The best way to proceed with the computation is to note that

$A_{k-1}P_k = A_{k-1} - 2w_k v_k^T$ with $w_k = A_{k-1}v_k$,

and then to use the result

$A_k = P_k^T A_{k-1} P_k = A_{k-1} - 2v_k q_k^T - 2q_k v^T_k$

with $q_k = w_k - (v_k^T w_k)v_k$. In this example then we compute

$W^T{}_2 = v_2^T A = [1.08813\ \ 1.03069\ \ .70281]$

Then $v_2^T w_2 = 1.16459$ and $q_2^T = [1.08813\ -.10275\ \ .43525]$

$$\text{Finally } A_2 = \begin{bmatrix} 1 & -1.11803 & 0 \\ -1.11803 & 1.40000 & -.55000 \\ 0 & -.55000 & 1.60000 \end{bmatrix} = B$$

Then we can find Det(B-tI) and apply Newton's method as in Given's method or QR method to get the eigenvalues 2 .53653 ,1.48012, -0.0166473 Which except for round off and some sign changes is the same matrix as B in example2:4. Because only a single orthogonal transformation is needed in this case, we expect Given's and Householder's methods to lead to essentially the same tridiagonal matrix. But for higher-order matrices, this will not be the case. Note the desirability of computing double precision scalar products in this method in order to minimize round off.

2.6.1 Methods for nonsymmetric matrices (Hyman's method) (Reduction to lower Hessenberg form)

Our approach to nonsymmetric matrices will be similar to that of Given's and Householder's methods in the sense that we shall perform a series of transformations on the matrix A similarity more often than orthogonal in order to reduce A to a matrix B with the same eignvalues as A but whose eigenvalues are more easily calculable. In the method to be considered the matrix B will be tridiagonal, as in the case of Givens' and Householder's methods. In Sec.2-6-2 we shall consider techniques for reducing A to a matrix B which has the form shown in Fig. 2.2. Such a matrix is said to be in super triangular or, more commonly, in lower Hessenberg form (where as the transpose of such a matrix is said to be in upper Hessenberg form).

$$\begin{bmatrix} b_{11} & b_{12} & & & & \\ b_{21} & b_{22} & b_{23} & & O & \\ \vdots & & \ddots & \ddots & & \\ \vdots & & & \ddots & \ddots & \\ \vdots & & & & \ddots & b_{n-1,n} \\ b_{n1} & \cdots & \cdots & \cdots & \cdots & b_{nn} \end{bmatrix}$$

Figure:2.2 a matrix in lower Hessenberg form.

A vital aspect of the calculation of the eigenvalues of nonsymmetric matrices is the stability of the calculation with respect to the growth of round off errors. The details of the round off-error analysis of the methods we are about to consider are beyond the scope of this seminar paper. We shall content ourselves with making some general comments on this matter in what follows; for more details the reader is referred to the references mentioned in the Bibliographic Notes.

2-6-2 Supertriangularization (Reduction to Hessenberg form)

If Givens' or Householder's method is applied to nonsummetric matrix A, the result is a Hessenberg matrix, as shown in Fig-2.2 . However it is possible to reduce a general matrix A to Hessenberg form with considerably less computation than is required by the methods mentioned above. This can be achieved by using Gaussian elimination instead of orthogolnal transformations. Moreover, by using partial pivoting, this process can be made extremely stable with respect to round off error.

As in Given's and Householder's methods, we proceed row by row to make the off-tridiagonal elements (in the upper triangle) zero. Suppose we have done this for rows 1, . . . , k-1. For convenience let the elements in the matrix at this stage still be denoted by a_{ij}. Then for row k we

1. Select the largest element a_{kl} in magnitude among a_{k}, a_{k+1}, ..., a_{kn} and interchange columns k + 1 and l.

2. Calculate $m_{kj} = -\dfrac{a_{kj}}{a_{k,k+1}} \quad j = k+2,...,n$ (41)

   where, because of step 1, $|m_{kj}| \leq 1$.

3. Add m_{kj} times column k + 1 to column j, $j = k + 2, ...,n$

Step 1 and each part of step 3 are equivalent to post multiplying the matrix by elementary column matrices. Therefore, to complete the similarity transformation for row k, it is only necessary to premultiply by the inverses of these (nonsingular) elementary matrices. It is quite easy to see that the zero elements in rows 1, ..., k - 1 remain zero. Therefore, performing this algorithm for k = 1, ..., n-2 results in a matrix B = $[b_{ij}]$ in Hessenberg form. The stability of this method with respect to round off results from the fact that the m_{kj} in (41) are all no greater than 1 in magnitude.

The number of multiplications and divisions required is $\frac{2}{3}n^3 + 0(n^2)$

While Householder's method for nonsymmetric matrices requires $\frac{4}{3}n^3 + 0(n^2)$ operations plus (n - 2) square roots.

In the next section we shall consider a powerful method for calculating all the eigenvalues of a matrix in Hessenberg form. In the remainder of this section, however, we shall consider how we can calculate an eignvalue and eigenvector of such a matrix. We assume throughout the remainder of this section that no $b_{i,i+1}$ is 0, for if so we can then consider a reduced matrix .

The system of equations $(B - \lambda I)\, x = 0$ can be written

$$
\begin{aligned}
(b_{11}-\lambda)x_1 + b_{12}x_2 &= 0 \\
b_{21}x_1 + (b_{22}-\lambda)x_2 + b_{23}x_3 &= 0 \qquad (42) \\
b_{n1}x_1 + b_{n2}x_2 + \dots + (b_{nn}-\lambda)x_n &= 0
\end{aligned}
$$

Now let us assume a value of λ and set $x_1 = 1$ in the first equation. The first n-1 equations can then be solved recursively for $x_2,\ldots,x_n$. But the last equation will be satisfied only if λ is an eigenvalue . Denote the value of the left-hand side of the last equation by F(λ). We shall now show that F(λ) is multiple of the characteristic equation.

The matrix B - λI has the form

$$B - \lambda I = \begin{bmatrix} b_{11} - \lambda & b_{12} & & & \\ b_{21} & b_{22} - \lambda & b_{23} & & O \\ \vdots & & \ddots & \ddots & \\ \vdots & & & \ddots & b_{n-1,n} \\ \vdots & & & & \ddots \\ b_{n1} & \cdots & \cdots & \cdots & b_{nn} - \lambda \end{bmatrix} \tag{43}$$

For i = 2, ..., n we multiply the i^{th} column by x_i as found above and add this to the first column, thereby obtaining the matrix

$$\begin{bmatrix} 0 & b_{12} & & & \\ \vdots & b_{22} - \lambda & b_{23} & & O \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & \vdots & & \ddots & b_{n-1,n} \\ & \vdots & & & \ddots \\ F(\lambda) & b_{n2} & \cdots & \cdots & b_{nn} - \lambda \end{bmatrix} \tag{45}$$

We have then

$$|B - \lambda I| = F(\lambda)\prod_{i=1}^{n-1} b_{i,i+1} = cF(\lambda) \tag{46}$$

where c is nonzero since we have assumed no $b_{i,i+1} = 0$. Since our object is to find a zero of F(λ), we can use one of the techniques in (1.0) of unit-I to find this zero. Since the eigenvalues of B may be complex, an appropriate choice is Newton's method in (1.0), which can find a complex root starting with real approximations and which has good convergence properties. Alternatively, we could compute one or two derivatives of F(λ) and use the Newton-based method of Sec.1.To compute F(λ), we differentiate (42) with respect to λ, taking into account that the x_i, i = 1,..., n, are function of λ.

We get the system $(b_{11} - \lambda)x'_1 - \lambda x_1 + b_{12}x'_2 = 0$

$$b_{21}x'_1 + (b_{22} - \lambda)x'_2 - \lambda x_2 + b_{23}x'_3 = 0$$

$$b_{n1}x'_1 + b_{n2}x'_2 + \ldots + (b_{nn} - \lambda)x'_n - \lambda x_n = 0 \qquad (47)$$

Setting $x_1 = 1$, $x'_1 = 0$, we solve (47) successively for $x'_2,\ldots,x'_n$ using the previously computed values $x_2,\ldots,x_n$. The value of the left-hand side of the last equation is then $F'(\lambda)$. We proceed similarly for $F''(\lambda)$.

It might seem that the accuracy of this procedure would be severely curtailed if any of the $b_{i,i+1}$ are very small in magnitude because of the necessity of dividing by $b_{i,i+1}$ in the recursion used to solve the first n - 1 equations of (42). But, in fact, it can be shown that there is little correlation between the accuracy of the method and the magnitude of the $b_{i,i+1}$, $i = 1, \ldots, n - 1$.

Having computed an eigenvalue of B by the procedure above, we compute the corresponding eigenvector by inverse iteration (Sec.2-2), which is quite efficient for Hessenberg matrices.

Once we have computed one eignvalue λ_1, we apply an implicit deflation and find a zero of

$F_1(\lambda) \equiv F(\lambda)/(\lambda - \lambda_1)$. More generally, once we have computed $\lambda_1, \ldots, \lambda_p$,

we seek a zero of $F_p(\lambda) \equiv \dfrac{F(\lambda)}{\prod_{i==1}^{P}(\lambda - \lambda_i)}$ (48)

evaluating $F(\lambda)$ by (42). The Newton-based method can be computed using the formulas in (1.0) as $\lambda_{n+1} = \lambda_n - F(\lambda_n)/F'(\lambda_n)$

Example 2:7 Apply the Gaussian elimination and deflation methods of this section to the matrix

$$A = \begin{bmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{bmatrix}$$

Interchanging the second and third columns of the matrix and then eliminating the element in the (1, 3) position, we obtain the matrix

$$\begin{bmatrix} 2 & 3 & 0 \\ 1 & 1 & \frac{5}{3} \\ 1 & -1 & \frac{7}{3} \end{bmatrix} \qquad (49)$$

Then premulitplying by the inverses of the elementary matrices used to derive (49), we obtain

$$B = \begin{bmatrix} 2 & 3 & 0 \\ \frac{1}{3} & -\frac{5}{3} & \frac{11}{9} \\ 1 & 1 & \frac{5}{3} \end{bmatrix}$$

To calculate an eigenvalue of B let us take as initial approximations $\lambda = 0$, $\lambda = -\frac{1}{2}$, and $\lambda = \frac{1}{2}$. Then using equations (42) we calculate

$cF(\lambda) = Det(B- \lambda I) = Det(A-\lambda I) = -\lambda^3 + 2\lambda^2+5\lambda-6$. $c = 3/11$.

$F(0) = -\frac{18}{11}$, $F\left(-\frac{1}{2}\right) = -\frac{189}{88}$, and $\lambda = .912537$. Convergence to $\lambda_1 = 1$ is very rapid (five iterations).

Now suppose we have found $\lambda_1 = 1$. Then, to calculate the next eigenvalue of B, we use the same initial approximations, and using (47), we calculate

$F_1(0) = -\frac{18}{11}$, $F\left(-\frac{1}{2}\right) = -\frac{63}{44}$, and $F_1\left(\frac{1}{2}\right) = \frac{75}{44}$. Using the same method again, λ_2 converges to –2 in one iteration. Similarly λ_3 = converge s to 3 in one iteration.

Eigensystem[A] gives eigenvalues -2,1,3 and eigenvectors (-11,-1,-14), (1,1,1) and (-1,1,1).

2-7 The LR and QR algorithms

The basis of these methods is the successive factorization of a sequence of matrices $\{A_k\}$, all of which have the same form as the original matrix $A_1 = A$; for example, if A is tridiagonal, so is every A_k.The key to these methods is the observation that if A_1 is factored into the product F_1G_1, where F_1 is nonsingular, then if we multiply F_1 and G_1 in reverse order, the matrix $A_1 = G_1F_1$ has the same eigenvalues as A_1. This is true because

$$A_2 = G_1F_1 = F_1^{-1}A_1F_1 \qquad (50)$$

So that A_1 and A_2 are similar. Now A_2 itself can likewise be decomposed into $A_2 = F_2G_2$, and in this way we define a sequence of matrices

$$A_k = F_kG_k = G_{k-1}F_{k-1} \qquad k = 2, 3, \ldots n$$

$$A_1 = A = F_1G_1 \qquad (51)$$

The following properties of the matrices A_k ,F_k, and G_k are of interest to us:

All the matrices A_k are similar and therefore have the same eigenvalues.

This follows from (50).

Let $E_k = F_1 \ldots F_k$, so that E_k is nonsingular. Since, as in (50),

$A_{k+1} = F_k^{-1}A_kF_k$,

it follows inductively that

$$A_{k+1} = E_k^{-1}A_1E_k \qquad (52)$$

Let $H_k = G_k \ldots G_1$. Since $F_jG_j = G_{j-1}F_{j-1}$,

$$E_kH_k = F_1 \ldots F_{k-1}F_kG_kG_{k-1} \ldots G_1 = F_1 \ldots F_{k-1}G_{k-1}F_{k-1}G_{k-1} \ldots G_1 = E_{k-1}A_kH_{k-1}.$$

Hence using (52) with k replaced by k - 1, $E_kH_k = A_1E_{k-1}H_{k-1}$.

By repetition of this process we arrive at $E_kH_k = A_1^k$ (53)

The LR and QR algorithms come from the following two factorizations of A.

LR : If we assume that there exists a unique triangular decomposition for each A_k,

$A_k = L_kU_k$ with L_k unit lower triangular and hence nonsingular, then

$F_k = L_k$, and $G_k = U_k$ defines the LR transformation. Such a decomposition may not necessarily exist even if A_k is nonsingular. In this case, all we know is that for some permutation matrix P, $PA_k = L_kU_k$.Hence, the above assumption is nontrivial.

QR: If we assume that it is possible to decompose an arbitrary real matrix A into a product QR, where Q is orthogonal and R is upper triangular with nonnegative diagonal elements, and that when A is nonsingular, this decomposition is unique, then $F_k = Q_k$, $G_k = R_k$ defines the QR transformation.

In contrast to the LR case, we now show that QR decomposition always exists by construction.

To obtain this decomposition, we apply to A a sequence $\{P_k\}$ of Householder transformations, as in Sec. 2.5, making sure at each stage that the diagonal element becomes nonnegative.

This is always possible since such transformations are determined up to a sign.

Thus, we find that

$$P_{n-1}P_{n-2} \dots P_1A = R \tag{54}$$

Since each P_j is orthogonal, we have that A = QR, where

$$Q = [P_{n-1} \ \dots \ P_1]^T$$

Since each P_j is uniquely determined by the nonnegativity condition if the diagonal element $a^{(j)}_{jj}$ is not 0 when P_j is applied, it follows that when A is nonsingular, the decomposition is unique.

Since the product of triangular matrices is triangular and that of orthogonal matrices is orthogonal, we have that E_k is the lower triangular or orthogonal factor of A_1^k.. Hence the convergence of either the LR or QR process is determined by the behavior of the sequence $\{E_k\}$, since $A_{k+1} = E^{-1}{}_kA_1E_k$.

<u>**Theorem2:2**</u> If $\{E_k\}$ converges to a nonsingular matrix E_∞ as $k\to\infty$, and if each G_k is an upper triangular matrix, then $\lim_{k\to\infty} A_k$ exists and is an upper triangular matrix.

proof: Since $\{E_k\}$ converges, the following limits also exist

$$\lim_{k \to \infty} F_k = \lim_{k \to \infty} E^{-1}{}_{k-1} E_k = I \tag{55}$$

$$G_\infty = \lim_{k \to \infty} G_k = \lim_{k \to \infty} A_{k+1} F^{-1}{}_k = \lim_{k \to \infty} E^{-1}{}_k A_1 E_{k-1}$$

$$= E^{-1}{}_\infty A_1 E_\infty$$

exists and is upper triangular, which proves the theorem.

An investigation of the convergence of the E_k in general is beyond the scope of this seminar topic. However, we shall prove a quite general theorem for the QR case, since this is the case of practical importance, and quote some results for cases not covered by the theorem.

First we make the following definition.If the sequence $\{A_k\}$ produced by either algorithm tends to upper triangular form even though elements above the main diagonal may not converge, we say that the algorithm converges essentially. If the sequence $\{A_k\}$ converges, we say that the algorithm converges.

Theorem 2-3: Let the real $(n \times n)$ matrix $A_1 = H\Lambda H^{-1}$, where $\Lambda = \text{diag}\{\lambda_1, ..., \lambda_n\}$. If $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_N| > 0$, and if $H^{-1} = Y$ has an LU factorization L_yU_y, then the QR algorithm converges essentially.

PROOF : $A^k{}_1 = H\Lambda^k H^{-1} = H\Lambda^k L_y U_y = H(\Lambda^k L_y \Lambda^{-k})(\Lambda^k U_y)$ (56)

Now $\Lambda^k L_y \Lambda^{-k} = I + B_k$, where

$$(B_k)_{ij} = \begin{cases} l_{ij}\left(\dfrac{\lambda_i}{\lambda_j}\right)^k & i > j; l_{ij} \in L_y \\ 0 & i \le j \end{cases}$$

we thus have that

$$A^k{}_1 = H(I + B_k)(\Lambda^k U_y) \tag{57}$$

where, since $|\lambda_i/\lambda_j| < 1$, $i > j$, $B_k \to 0$ as $k \to \infty$.

Now, by our previous construction, H can be decomposed into Q_xR_x, where R_x has positive diagonal elements.

Therefore, $A^k{}_1 = Q_xR_x(I + B_k)(\Lambda^k U_y) = Q_x(I + R_xB_kR_x^{-1})(R_x \Lambda^k U_y)$ (58)

Since $B_k \to 0$, $I + R_xB_kR_x^{-1}$ will eventually become nonsingular and hence will have a unique factorization $\overline{Q}_k\overline{R}_k$, where $\overline{Q}_k \to I, \overline{R}_k \to I$ as $k \to \infty$.

Thus $A^k{}_1 = \left(Q_x\widetilde{Q}_k\right)\left(\widetilde{R}_k R_x \Lambda^k U_y\right)$ (59)

This need not be the QR factorization of $A^k{}_1$ since the diagonal of the second factor need not be positive because of Λ^k and U_y. We therefore define diagonal orthogonal matrices D_1 and D_2 such that $D_1\Lambda$ and D_2U_y have positive diagonal elements.

Therefore, $Q_x\widetilde{Q}_k D^{-1}{}_2 D^{-k}{}_1$ is the orthogonal factor of $A^k{}_1$.

Hence, with $Q_x\widetilde{Q}_k D^{-1}{}_2 D^{-k}{}_1$ playing the role of E_k,

$$A_{k+1} = D^k{}_1 D_2 \overline{Q}^T{}_K Q^T{}_x A_1 Q_x \overline{Q}_k D^{-1}{}_2 D^{-k}{}_1 \rightarrow D^k{}_1 D_2 Q^T{}_x A_1 Q_x D^{-1}{}_2 D^{-k}{}_1$$

$$= D^k{}_1 (D_2 R_x \Lambda R^{-1}{}_x D^{-1}{}_2) D^{-k}{}_1 \qquad (60)$$

as $k \rightarrow \infty$ since $\overline{Q}_k \rightarrow I$ and $A_1 = Q_x R_x \Lambda R^{-1}{}_x Q^T{}_x$.

If $D^k{}_1$ converges so does A_k. But $D_1 = \mathrm{diag}\,(\pm 1, ..., \pm 1)$, and so $D^k{}_1$ will not converge if any diagonal element is negative.

However, this has no effect on the diagonal elements of $R_x\Lambda R^{-1}{}_x$ (nor on the magnitudes of the elements in the upper triangle). Hence, we have essential convergence. Further more, since the diagonal of $R_x\Lambda R^{-1}{}_x = \Lambda$, we see that the eignevalues appear on the diagonal in decreasing order of magnitude.Further results on convergence, which we state without proof, are the following.

If, in addition to the hypotheses of Theorem 2:3, X has an LU factorization, then the LR algorithm converges. If Y does not have an LU factorization, then since Y is nonsingular, there exists a permutation matrix P such that PY does have such a factorization. It is then not difficult to show that the QR algorithm still converges essentially. However, in contrast to the case of Theorem 2;3, the eigenvalues do not appear on the diagonal in decreasing order of magnitude. In fact, the sequence $\{A_k\}$ always converges to block triangular form, each diagonal block having roots of equal magnitude. In general, this does not cause problems. Only when there are distinct eigenvalues of equal modulus, do we have convergence to a form which is troublesome. Thus, even with real multiple eigenvalues, the convergence is to triangular form, while for multiple complex-conjugate pairs of eigenvalues, the limiting form of A_k will yield these roots in a string of 2 x 2 sub matrices along the diagonal. Hence, except for rare cases, the liming form of A_k is some modification of the form

$$\begin{bmatrix} \lambda_1 & x & \dots & x & x & x & \dots & x & x \\ 0 & \lambda_2 & \dots & x & x & x & \dots & x & x \\ \dots\dots & & \dots\dots & & \dots\dots & & \dots\dots & \dots & \\ 0 & 0 & \dots & \lambda_m & x & x & \dots & x & x \\ 0 & 0 & \dots & 0 & & & \dots & x & x \\ 0 & 0 & \dots & 0 & B_1 & & \dots & x & x \\ \dots\dots & & \dots\dots & & \dots\dots & & \dots\dots & \dots & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & & \\ & & & & & & & B_1 & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & & \end{bmatrix} \qquad (61)$$

Where m + 2L = n,(m,n,L $\in$ N) each B_j is a 2 x 2 real submatrix with complex conjugate eigenvalues which are eigenvalues of A_1, and the real eigenvalues λ_i and the B_j may appear in any order along the diagonal. Cases in which this limiting form is not achieved almost never arise in practice since, as we shall see below, the QR transformations are modified by shifts, which greatly reduce the possibility of distinct eigenvalues of equal modulus.Returning to Theorem 2;3, we see that

$a^{(k)}{}_{ii} \rightarrow \lambda_i \ as \ k \rightarrow \infty.$

In fact, it can be shown that

$$a^{(k)}{}_{ii} = \lambda_i + 0\left(r^k{}_i\right) \qquad a^{(k)}{}_{i+1,i} = 0\left(r^k{}_i\right) \qquad (62)$$

Where $r_i = \max\left(\left|\frac{\lambda_i}{\lambda_{i-1}}\right|, \left|\frac{\lambda_{i+1}}{\lambda_i}\right|\right) \qquad \lambda_0 = \infty \qquad \lambda_{n+1} = 0$

Which is a linear convergence. In particular, $r_n = |\lambda_n / \lambda_{n-1}|$, and our strategy is to try to modify the algorithm to make r_n very small so that $a_{nn} \rightarrow \lambda_n$ and $a_{n,n-1} \rightarrow 0$ very rapidly. Looking ahead a little, if the A_k were upper Hessenberg or tridiagonal, then once $a_{n,n-1}$ converged to 0 to with good accuracy, we would have computed

$\lambda_n = a_{nn}$.

Then we could deflate the matrix and work only with the matrix of order n - 1 consisting of the first n - 1 rows and columns of A_k. Thus successive deflations would result in less and less work to compute succeeding QR transformations.Recalling our discussion of the power method (Sec. 2;1 and 2.2), we know that the eigenvalues of A_k -pI are $\lambda_i - p_i$:i= 1, ..., n, so that if we can choose an appropriate value of p, the ratio $|(\lambda_n\text{-}p)/(\lambda_{n-1} - p)|$ will converge to 0 very rapidly. A good estimate of λ_n would serve this purpose very well. In order to allow us to

choose the best estimate at each iteration, we must work with $A_k = P_k I$, involving the variable shift p_k. This gives us the modified algorithm

$$\begin{aligned} &\text{Factor } A_k - p_k I \text{ into } F_k G_k \\ &\qquad\qquad\qquad K = 1, 2, 3, \ldots \\ &\text{Form } A_{k+1} = G_k F_k + p_k I \end{aligned} \tag{63}$$

Then, as before,

$$A_{k+1} = F^{-1}{}_k \ldots F^{-1}{}_1 A_1 F_1 \ldots F_k = E^{-1}{}_k A_1 E_k$$

and
$$E_k H_k = \prod_{j=1}^{k} \left(A_1 - P_j I\right)$$

If we define $\phi_k(\lambda) = \prod_{j=1}^{k} (\lambda - p_j)$, the proof of Theorem 2;3 will carry over to the modified algorithm when $\lambda^k{}_i$ is replaced by $\phi_k(\lambda_i)$. Thus, provided the P_j are chosen so that

$|\phi_k(\lambda_i)| \neq |\phi_k(\lambda_s)|, i \neq s,$ for K > s and so that $\phi_k(\lambda_i) \neq 0$, then the modified algorithm will converge. We defer until later our discussion of the choice of P_j.

2.7-2 The simple QR algorithm

As indicated above, we shall restrict our attention to the QR algorithm. We do this because the LR algorithm is numerically unstable. This is clearly true for the general situation since the LU decomposition of a matrix without pivoting may lead to disaster. While it is possible to modify this algorithm to include pivoting, the theoretical basis of the convergence proof is lost, and, in fact, simple examples can be constructed for which convergence does not take place. Even when the LU decomposition without pivoting can be justified, as in the case of positive definite symmetric matrices, in practice there is a loss of accuracy. Hence, even though the QR algorithm is more expensive than the LR algorithm, its superior numerical properties more than make up for the extra labor.If we count the number of operations involved in a single FG transformation applied to a full matrix, we find that it is of the order of n^3. Since many iterations may be necessary before convergence, this could be quite an expensive task. Fortunately, the workload decreases substantially if we work with Hessenberg or tridiagonal matrices, which we can do in as much as the QR transformation leaves these forms invariant. The work involved in a single transformation is of the order of n^2 operations for Hessenberg matrices and of n operations for triadiagonal matrices. We shall henceforth restrict our discussion to such matrices, where we further assume that all elements on the sub diagonal are nonzero, since otherwise we could decompose our problem into smaller ones.We now distinguish between two cases. In this section, we shall deal with the case where we know that all the eigenvalues are real. In Sec. 2: 6-2 we dealt with the case where some of the eigenvalues may be complex. In the first case there is a very efficient scheme for implementing the QR transformation.Since A_k is now assumed to be Hessenberg or tridiagonal form, we need not use Householder transformations to transform A_k to upper triangular form but can use plane rotations.Recalling the notation of Sec. 2;4, we apply the sequence of transformations (i, i + 1,i), i = 1, ..., n-1 to $A_k - p_kI$, where premultiplication by the corresponsiding matrix $S^T_{i\,mi+1.i}$ annihilates the element in the ith column and $(i + 1)^{th}$ row. We thus have

$$S^T_{n-1,n,n-1} \dots S^T_{2,3,2} S^T_{1,2,1} (A_k - p_k I) = R_k$$ so that

$$Q_k = S_{1,2,1} \dots S_{n-1,n,n-1}, \text{ and } A_{k+1} = R_k S_{1,2,1} \dots S_{n-1,n,n-1} + p_k I \qquad (64)$$

Each such transformation takes about $4n^2$ multiplications and n - 1 square roots.

The shift p_k is determined from the eigenvalues μ_k and v_k of T_k, the bottom 2 x 2 submatrix of A_k. If both are real, we take p_k to be μ_k or v_k according as $|\mu_k - a^{(k)}{}_{nn}|$ *or* $|v_k - a^{(k)}{}_{nn}|$ is smaller. Otherwise we set $p_k = \mathrm{Re}\ \mu_k$.

If our matrix A_1 is symmetric and tridiagonal, then since the QR transformation preserves symmetry, all subsequent matrices A_k will be symmetric and hence tridiagonal . In this case, the QR algorithm with shifts is very efficient. Thus the combined algorithm of first reducing a symmetric matrix to tridiagonal form by Householder transformations and then applying the QR algorithm is probably the most effective way to evaluate all the eigenvalues of a symmetric matrix.

QR: in short, $A_i = Q_iR_i$ and $A_{i+1} = Q_i^T A_i Q_i$

<u>Example</u> 2: 8 Apply the simple QR algorithm to find all the (real) eigenvalues of the symmetric tridiagonal matrix A, where

$$A = A_1 = \begin{bmatrix} 120.0 & -90.86 & 0 & 0 \\ -90.86 & 157.2 & 67.59 & 0 \\ 0 & 67.59 & 124.0 & -46.26 \\ 0 & 0 & -46.26 & 78.84 \end{bmatrix} \qquad (65)$$

The computations below were carried out to about 14 significant figures.
We give the results rounded to five figures. We give the transformation from A_1 to A_2 in full detail. After that, we shall only give the shifts, p_k, and the matrices A_k.

$$P_1 = 49.943; \quad S^T{}_{121} = \begin{bmatrix} .61061 & -.79193 & 0 & 0 \\ .79193 & .61061 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^T{}_{121}(A_1 - P_1 I) = \begin{bmatrix} 114.73 & -140.42 & -53.527 & 0 \\ 0 & -6.4629 & 41.271 & 0 \\ 0 & 67.590 & 74.057 & -16.260 \\ 0 & 0 & 16.260 & 28.390 \end{bmatrix}$$

$$S^T{}_{232} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -.155\ .185 & .99546 & 0 \\ 0 & -.99546 & -.095185 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^T{}_{232}S^T{}_{121}(A_1 - p_1 I) = \begin{bmatrix} 114.73 & -140.42 & -53.527 & 0 \\ 0 & 67.898 & 69.792 & -46.050 \\ 0 & 0 & -48.133 & 4.4032 \\ 0 & 0 & -46.260 & 28.897 \end{bmatrix}$$

$$S^T{}_{343} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -.72099 & -.69294 \\ 0 & 0 & .69294 & -.72099 \end{bmatrix}$$

R_1=

$$S^T{}_{343}S^T{}_{232}S^T{}_{121}(A_1 - p_1 I) = \begin{bmatrix} 114.73 & -140.42 & -53.527 & 0 \\ 0 & 67.898 & 69.792 & -46.050 \\ 0 & 0 & 66.759 & -23.198 \\ 0 & 0 & 0 & -17.783 \end{bmatrix}$$

$$R_1 S_{121} = \begin{bmatrix} 181.26 & 5.1182 & -53.527 & 0 \\ -53.771 & 41.459 & 69.792 & -46.050 \\ 0 & 0 & 66.759 & -23.198 \\ 0 & 0 & 0 & -17.783 \end{bmatrix}$$

$$R_1 S_{121} S_{232} = \begin{bmatrix} 181.26 & -53.771 & 0 & 0 \\ -53.771 & 65.529 & -47.914 & -46.050 \\ 0 & 66.456 & -6.3544 & -23.198 \\ 0 & 0 & 0 & -17.783 \end{bmatrix}$$

$$A_2 - P_1 I = R_1 S_{121} S_{232} S_{343} = \begin{bmatrix} 181.26 & -53.771 & 0 & 0 \\ -53.771 & 65.529 & 66.456 & 0 \\ 0 & 66.456 & 20.657 & 12.323 \\ 0 & 0 & 12.323 & 12.822 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 231.20 & -53.771 & 0 & 0 \\ -53.771 & 115.47 & 66.456 & 0 \\ 0 & 66.456 & 70.600 & 12.323 \\ 0 & 0 & 12.323 & 62.765 \end{bmatrix}$$

$$P_2 = 53.752 \quad A_3 = \begin{bmatrix} 251.32 & -23.029 & 0 & 0 \\ -23.029 & 136.29 & 38.237 & 0 \\ 0 & 38.237 & 28.568 & -2.8312 \\ 0 & 0 & -2.8312 & 63.861 \end{bmatrix}$$

$$P_3 = 64.087 \quad A_4 = \begin{bmatrix} 255.18 & -9.6151 & 0 & 0 \\ -9.6151 & 140.163 & 24.060 & 0 \\ 0 & 24.060 & 20.690 & -.0047934 \\ 0 & 0 & -.0047934 & 64.003 \end{bmatrix}$$

$$P_4 = 64.003 \quad A_5 = \begin{bmatrix} 255.86 & -3.9842 & 0 & 0 \\ -3.9842 & 142.45 & 14.730 & 0 \\ 0 & 14.730 & 17.730 & <10^{-10} \\ 0 & 0 & <10^{-10} & 64.003 \end{bmatrix}$$

We now deflate and continue on $\overline{A}_5$, the 3 x 3 submatrix of A_5, then

$$P_5 = 16.014 \quad \overline{A}_6 = \begin{bmatrix} 255.96 & -2.1128 & 0 \\ -2.1128 & 144.07 & -.00010338 \\ 0 & -.00010338 & 16.013 \end{bmatrix}$$

$$P_6 = 16.013 \qquad \overline{A}_7 = \begin{bmatrix} 255.99 & -1.1273 & 0 \\ -1.1273 & 144.04 & <10^{-10} \\ 0 & <10^{-10} & 16.013 \end{bmatrix}$$

We now deflate again and continue on $\overline{A}$, the 2 x 2 sub matrix of $\overline{A}$,.

$$P_7 = 144.03 \,, \quad A_8 = \begin{bmatrix} 256.00 & <10^{-10} \\ <10^{-10} & 144.03 \end{bmatrix}$$

We are now through, and the eigenvalues, read off the diagonal of the full matrix A_8, are as given in "EigenSystems[A]" as 256.00 ,144.03,16.013, 64.003

Note that they are not in decreasing order of magnitude.

We see that each premultiplication by $S^T_{i,i+1}$ affects only rows i and i + 1 and, similarly, each post multiplication by $S_{j,j+1.j}$ affects only columns j and j + 1. From this we see that if A is tridiagonal, then R is a band matrix of bandwidth [0,2]. This implies that the storage requirements in this case are 0(n) rather than $0(n^2)$ and similarly for the number of operations per iteration. We also notice that each A_k is tridiagonal and symmetric, as expected.

We also notice that at the same time that the element in the lower right-hand corner of the matrix is converging to an eigenvalue, the other diagonal elements are also converging although at a slower rate. Similarly, the other off-diagonal elements are tending to 0 although not as fast as $a_{n,\, n-1}$. Thus, subsequent iterations on the later deflated matrices converge in less iterations than are required for the earlier ones.

We point out one final item connected with the organization of the computation. We have applied formulas (64) in a straightforward fashion here. This requires saving the matrices $S_{i,i+1,i,}$ i = 1, ... n-1, or more accurately, the two values, sin θ_i and cos θ_i which determine $S_{i,i+1,i}$. However, we can reorganize the computation to avoid saving these values by noticing that, for example, after computing $S^T_{232}S^T_{121}$

$(A_1 = p_1 I)$, the first two columns remain unchanged and no information contained in $S^T_{232}S^T_{121}$

$(A_1 = p_1 I)S_{121}$ before premultiplying by S^T_{343} and therefore we need not save S_{121} any more. In general, we post multiply by $S_{i-1,\, i-1}$ after the premultiplication $S^T_{i,i+1,\, i,}$ i = 2, ..., n-1 and finally, post multiply by $S_{n-1,\, n,n-1}$. This saves about 2n storage locations.

Further savings in computation and storage can be obtained in the symmetric tridiagonal case if we take into account the fact that each A_k is of the same form.

2:8 Analysis of errors in calculated eigenvalues and eigenvectors

In most of the methods considered above, we do not compute the eigenvalues directly from the matrix itself but from a similar matrix B to which the transformation is made. Actualy, we do not intend to compare the accuracy of the methods as in iterations but we do compare their fastness. In the process of similarity there will occur round off errors due to perturbation.However, by backward error analysis, it can be shown that for all the methods discussed in this chapter B is similar to A + δA, where the elements of δA are small. In the course of the computation of B from A, round off errors enter, so that B is not strictly similar to A. Thus, in the reduction of the given matrix A to Hessenberg or tridiagonal form by Householder transformations using t-digit floating-point binary arithmetic, it can be shown that

$$\|\delta A\|_E \le \gamma n^2 2^{-t} \|A\|_E \tag{66}$$

where γ is a constant of order 1.If inner products are accumulated to double precision, the bound (66) becomes

$$\|\delta A\|_E \le yn2^{-t} \|A\|_E \tag{67}$$

The important question to consider is the effect of perturbations of the elements of A on its eigenvalues.

Let us first consider the symmetric case, where both A and B, and hence also δA, are symmetric. If the eigenvalues of these matrices are α_i, β_i, and δ_i, respectively, arranged in non-increasing order, it can be shown that

$$\alpha_i + \delta_n \le \beta_i \le \alpha_i + \delta_n \tag{68}$$

If the elements of δA are all less than δ in magnitude, we have further that

$$-n\delta \le \delta_n \le \delta_i \le n\delta$$

So that

$$-n\delta \le \beta_i - \alpha_i \le n\delta \tag{69}$$

These results hold even when there are multiple eigenvlaues, and so it follows that the eigenvlalue problem for a symmetric matrix is always well conditioned.

In the nonsymmetric case, we may have ill conditioning. Let α_i be a simple eignevalue of A and x_i and y_i the corresponding right and left eigenvectors normalized so that $\|x_i\|_2 = \|y_i\|_2 = 1$. Then δA tends to the null matrix, A + δA has an eignevalue $\alpha_i + \delta\alpha_i$ such that

$$\delta\alpha_i < \frac{y_i^T \delta A x_i}{y^T{}_i x_i} \tag{70}$$

Thus, for δA small enough, $|\delta x_i| \le \frac{\|\delta A\|}{|y^T{}_i x_i|}$

where $(y_i)^T x_i$ is the cosine of the angle θ, between y_i and x_i. Matrices exist with simple eignenvalues for which the cos θ_i are arbitrarily small, and any such eigenvalue is very sensitive to perturbations in the elements.

Still, if $\|\delta A\|_2 \le \delta$, we have that $|\delta\alpha_i| \le \delta/|\cos\theta_i|$ and the right-hand side is linear in δ.

If α_i is not a simple root, the situation may be worse. For example, the matrix

$$A = \begin{bmatrix} c & 0 \\ 1 & c \end{bmatrix} \tag{71}$$

has the double eigenvalue $\alpha_i = \alpha_2 = c$, while the perturbed matrix

$$\begin{bmatrix} c & \delta \\ 1 & c \end{bmatrix} = A + \delta I \tag{72}$$

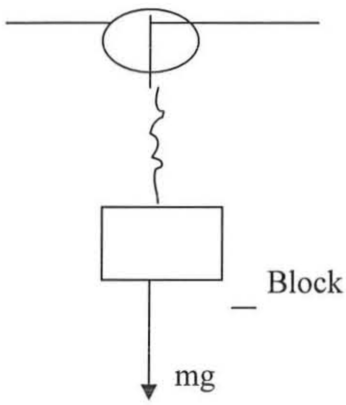
has the double eignevalues $c + \delta^{1/2}$.

We see that for a particular matrix A, some eigenvalues may be sensitive while other are not.

For simple eigenvalues, it depends on the angle θ_i between the corresponding right and left eigenvectors. In case cos θ_i is not small, α_i is well determined and a good algorithm should give accurate results.

2:9 Application of eigenvalue proplems

1. consider the vibrating vertical spring



This motion is defined by the boundary value second order differential equation,

$$y''+\lambda y = o : y(\pi) = y(o) = o \qquad (73)$$

Y = 0 means no motion .

This is a linear second order differential equation with constant coefficients. Hence, let y (t) = e^{pt}

$\therefore$ y'' + λ y(t) = $(P^2 + \lambda)e^{pt} = 0 \Rightarrow p^2 + \lambda = o$

$p = \pm\sqrt{\lambda}\, i$

$\Rightarrow$ y(t) = $c_1 \; e^{-\sqrt{\lambda}\,it} + c_2 \; e^{\sqrt{\lambda}\,it}$

$= A \cos\sqrt{\lambda}\, t + B \sin\sqrt{\lambda}\, t$

And applying boundary conditions, we get y(t) = Sin(wt).

$\sqrt{\lambda} = w$ is the eigenvalue since Dy = wy and it is found to be n $\in$N.

w is the natural frequency (eigenvalue) of the spring that represents the general behavior of the motion and has minimum value 1.The solution here is un stable since the frequency or the eigenvalues vary with n in N.

2. Application to Markov Chains' process

In the study of Markov chains the elements of the transition matrix are the probabilities of moving from any state to any other state. A Markov process can be described by a square matrix whose entries are all positive and the column sums are all equal to 1. For example, a 3×3 transition matrix looks like

$$A = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

where $p_{1,1} + p_{2,1} + p_{3,1} = 1$, $p_{1,2} + p_{2,2} + p_{3,2} = 1$ and $p_{1,3} + p_{2,3} + p_{3,3} = 1$. The initial state vector is

$$P_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ y_0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z_0 \end{pmatrix}.$$

The computation

$$\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} p_{1,1} x_0 \\ p_{2,1} x_0 \\ p_{3,1} x_0 \end{pmatrix} = x_0 \begin{pmatrix} p_{1,1} \\ p_{2,1} \\ p_{3,1} \end{pmatrix}$$

shows how the x_0 is redistributed in the next state. Similarly we see that

$$\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ y_0 \\ 0 \end{pmatrix} = \begin{pmatrix} p_{1,2} y_0 \\ p_{2,2} y_0 \\ p_{3,2} y_0 \end{pmatrix} = y_0 \begin{pmatrix} p_{1,2} \\ p_{2,2} \\ p_{3,2} \end{pmatrix}$$

shows how the x_0 and y_0 are redistributed in the next state.

and

$$\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} = \begin{pmatrix} p_{1,3} z_0 \\ p_{2,3} z_0 \\ p_{3,3} z_0 \end{pmatrix} = z_0 \begin{pmatrix} p_{1,3} \\ p_{2,3} \\ p_{3,3} \end{pmatrix}$$

shows how the x_0, y_0 and z_0 are redistributed in the next state.

Therefore, the distribution for the next state is

$$P_1 = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = A P_0$$

A recursive sequence is generated using the general rule

$P_{k+1} = A P_k$ for $k = 0, 1, 2, \ldots$.

We desire to know the limiting distribution $P = \lim_{k\to\infty} P_k$. Since we will also have $\lim_{k\to\infty} P_{k+1} = P$ we obtain the relation

$$P = \lim_{k\to\infty} P_{k+1} = A\left(\lim_{k\to\infty} P_k\right) = A P$$

From which it follows that

$$P = A P$$

Therefore the limiting distribution P is the eigenvector corresponding to the dominant eigenvalue $\lambda_1 = 1$. The following subroutine reminds us of the iteration used in the power method.

Mathematica Subroutine (Markov Process).

```
Markov[P0_, n_] := Module[{},
  T = z = y = x = Table[0, {i, 0, n}];
  P = Transpose[{x, y, z}];
  P[[1]] = P0;
  Print["P"0, " = ", MatrixForm[P[[1]]]];
  For[i = 1, i ≤ n, i++,
   T[[i+1]] = i;
   P[[i+1]] = A.P[[i]];
   Print["P"i, " = ", MatrixForm[P[[i+1]]]]; ];
  Return[P[[i]]]; ]
```

Example . Let $P_0 = (x^{(0)}, y^{(0)}, z^{(0)})^T$ record the number of people in a certain city who use brands X, Y, and Z, respectively.
Each month people decide to keep using the same brand or switch brands.
The probability that a user of brand X will switch to brand Y or Z is 0.3 and 0.3, respectively.

The probability that a user of brand Y will switch to brand X or Z is 0.3 and 0.2, respectively.
The probability that a user of brand Z will switch to brand X or Y is 0.1 and 0.3, respectively.
The transition matrix for this process is $P_{k+1} = A P_k$ or

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ z^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.2 & 0.6 \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ z^{(k)} \end{pmatrix}$$

Assume that the initial distribution $P_0 = (2000, 6000, 4000)^T$.

1. Find the first few terms in the sequence $\{P_k\}$.
2. Verify that $\lambda_1 = 1$ is the dominant eigenvalue of A.
3. Verify that a corresponding eigenvector is $V_1 = (3000, 4500, 4500)^T$.
4. Conclude that the limiting distribution is $\lim_{k\to\infty} P_k = V_1$.

Solution

1. Enter the matrix A and vector P_0 and use the subroutine Markov to find the first few terms in the sequence $\{P_k\}$.

```
A = ( 0.4  0.3  0.1
      0.3  0.5  0.3
      0.3  0.2  0.6 );
P0 = {2000, 6000, 4000};
Print["A = ", MatrixForm[A]];
Z = Markov[P0, 11];
```

$$A = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.2 & 0.6 \end{pmatrix}$$

$$P_0 = \begin{pmatrix} 2000 \\ 6000 \\ 4000 \end{pmatrix}$$
$$P_1 = \begin{pmatrix} 3000. \\ 4800. \\ 4200. \end{pmatrix}$$
$$P_2 = \begin{pmatrix} 3060. \\ 4560. \\ 4380. \end{pmatrix}$$
$$P_3 = \begin{pmatrix} 3030. \\ 4512. \\ 4458. \end{pmatrix}$$

$$P_4 = \begin{pmatrix} 3011.4 \\ 4502.4 \\ 4486.2 \end{pmatrix}$$

$$P_5 = \begin{pmatrix} 3003.9 \\ 4500.48 \\ 4495.62 \end{pmatrix}$$

$$P_6 = \begin{pmatrix} 3001.27 \\ 4500.1 \\ 4498.64 \end{pmatrix}$$

$$P_7 = \begin{pmatrix} 3000.4 \\ 4500.02 \\ 4499.58 \end{pmatrix}$$

$$P_8 = \begin{pmatrix} 3000.12 \\ 4500. \\ 4499.87 \end{pmatrix}$$

$$P_9 = \begin{pmatrix} 3000.04 \\ 4500. \\ 4499.96 \end{pmatrix}$$

$$P_{10} = \begin{pmatrix} 3000.01 \\ 4500. \\ 4499.99 \end{pmatrix}$$

$$P_{11} = \begin{pmatrix} 3000. \\ 4500. \\ 4500. \end{pmatrix}$$

2. Verify that $\lambda_1 = 1$ is an eigenvalue of **A**.

```
Print["A = ", MatrixForm[A]];
Print["The eigenvalues are  ", Eigenvalues[A]];
```

$$A = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.2 & 0.6 \end{pmatrix}$$

```
The eigenvalues are  {1., 0.3, 0.2}
```

3. Verify that $\lambda_1 = 1$ is an eigenvalue of **A** and a corresponding eigenvector is $V_1 = (3000, 4500, 4500)^T$.

```
λ = 1;
X = {3000, 4500, 4500};
Print["A = ", MatrixForm[A]];
Print["The eigenvalue is  λ = ", λ];
Print["An eigenvector is  X = ", MatrixForm[X]];
Print["A X = ", MatrixForm[A], MatrixForm[X], " = ", MatrixForm[A.X]];
Print["λ X = ", λ MatrixForm[X], " = ", MatrixForm[λ X]];
```

$$A = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.2 & 0.6 \end{pmatrix}$$

The eigenvalue is $\lambda = 1$

An eigenvector is $X = \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix}$

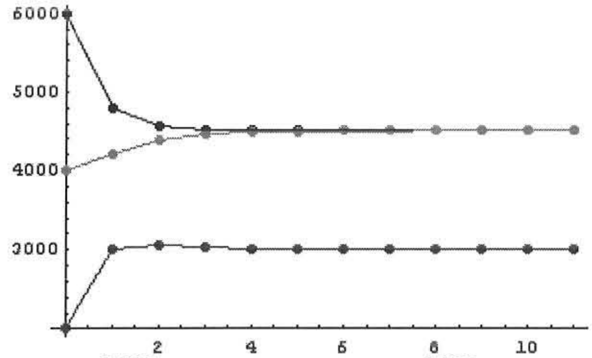
$$A\,X = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.2 & 0.6 \end{pmatrix} \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix} = \begin{pmatrix} 3000. \\ 4500. \\ 4500. \end{pmatrix}$$

$$\lambda\,X = \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix} = \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix}$$

4. The iteration in part (a) appears to be converging to $V_1 = (3000, 4500, 4500)^T$.

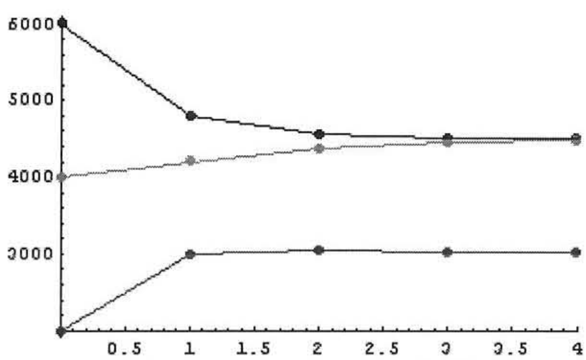
.We can graph the situation !

```
Needs["Graphics`Colors`"];
pts1 = Transpose[{T, Transpose[P][[1]]}];
pts2 = Transpose[{T, Transpose[P][[2]]}];
pts3 = Transpose[{T, Transpose[P][[3]]}];
gr1 = ListPlot[pts1, PlotStyle → {Red, PointSize[0.02]}, DisplayFunction → Identity];
gr2 = ListPlot[pts2, PlotStyle → {Blue, PointSize[0.02]}, DisplayFunction → Identity];
gr3 = ListPlot[pts3, PlotStyle → {Green, PointSize[0.02]}, DisplayFunction → Identity];
gr4 = ListPlot[pts1, PlotJoined → True, PlotStyle → Red, DisplayFunction → Identity];
gr5 = ListPlot[pts2, PlotJoined → True, PlotStyle → Blue, DisplayFunction → Identity];
gr6 = ListPlot[pts3, PlotJoined → True, PlotStyle → Green, DisplayFunction → Identity];
Show[gr1, gr2, gr3, gr4, gr5, gr6, PlotRange → All, DisplayFunction → $DisplayFunction];
Print["P0 = ", MatrixForm[P0], ",  lim(k→∞) Pk = P = ", MatrixForm[X]];
```



$$P_0 = \begin{pmatrix} 2000 \\ 6000 \\ 4000 \end{pmatrix}, \quad \lim_{k\to\infty} P_k = P = \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix}$$

```
Needs["Graphics`Colors`"];
pts1 = Transpose[{T, Transpose[P][[1]]}];
pts2 = Transpose[{T, Transpose[P][[2]]}];
pts3 = Transpose[{T, Transpose[P][[3]]}];
gr1 = ListPlot[pts1, PlotStyle → {Red, PointSize[0.02]}, DisplayFunction → Identity];
gr2 = ListPlot[pts2, PlotStyle → {Blue, PointSize[0.02]}, DisplayFunction → Identity];
gr3 = ListPlot[pts3, PlotStyle → {Green, PointSize[0.02]}, DisplayFunction → Identity];
gr4 = ListPlot[pts1, PlotJoined → True, PlotStyle → Red, DisplayFunction → Identity];
gr5 = ListPlot[pts2, PlotJoined → True, PlotStyle → Blue, DisplayFunction → Identity];
gr6 = ListPlot[pts3, PlotJoined → True, PlotStyle → Green, DisplayFunction → Identity];
Show[gr1, gr2, gr3, gr4, gr5, gr6, PlotRange → {{0, 4}, All}, DisplayFunction → $DisplayFunction];
Print["P0 = ", MatrixForm[P0], ",  lim_{k→∞} Pk = P = ", MatrixForm[X]];
```



$$P_0 = \begin{pmatrix} 2000 \\ 6000 \\ 4000 \end{pmatrix}, \quad \lim_{k\to\infty} P_k = P = \begin{pmatrix} 3000 \\ 4500 \\ 4500 \end{pmatrix}$$

2:10 Summary;

The methods of finding eigenvalues by characteristic polynomial (Newton's method),similarity method and estimate by bounds are not efficient for many reasons. Further a large degree polynomial is difficult to solve for roots and also errors occur due to rounding . It is not simple also to calculate the matrix T of similarity transformation. The method of bounds does not give exact value(s).

The refinement in these are the power,Wielandt's,Jacobi's,Given's, QR and LR methods. Almost all of reduction steps (tridiagonal,diagonal,normal or Hessenberg form) involve similarity.Reductions minimize number of operations.The Householder reduction is the most powerful method to Jacobi and Given's and is followed by QR for finding all eigenvalues including complex ones, which is cubic convergent.The Jacobi method is quadratic convergent. Diagonal matrices are the simplest forms. Given's method involves twice number of operations as Householder.Jacobi method is an infinite process but converges to a diagonal matrix.Hence,the Householder's method is recommended for large order n to tridiagonalize giving orthogonal set of eigenvectors by QR.

We have determined how classical methods are used to find eigenvalues of tridiagonal or Hessenberg matrix by seeking zeros of polynomials. The prototype is iterative and Wielandt's is a refinement but QR is the best method of all, also to LR for calculating eigenvalues.The power method gives only the one dominant value.It is slow linear convergent to one dominant (λ_1) and is used conditionally where Wielandt's is preferred to it. After Householder, we can apply QR or use bisection or Newton's method (1.0) for finding the roots of characteristic polynomial of A, but the method of Hyman is mostly recommended. The LR method involves elimination and LU decomposition and hence can't be applied to arbitrary matrix.The stability of each method mainly depends on the type of matrix (like Hermitian, tridiagonal).

The Evps are applicable in practical problems.For instance, in wave motion,heat conduction,Population statistics (markov chains' process) and so on.

To conclude, since every method has both positive and disadvantage sides, we would like to say and try to get most powerful method and fast computer aided programs for the future.

2:11 Mathematica program

"Matrix Solver" gives eigenvalues and eigenvectors".

Sol = EigenSystem[N[A]];

Print["{xi} = ",$Sol_{[1]}$];

Print[" "];

Prin["{xi} = ",TableForm[Transpose[$Sol_{[2]}$];

Eigenvectors[a] gives eigenvectors

Eigenvalues[a] gives eigenvalues .

CharacteristicPolynomial[a,x] gives characteristic polynomial of A

Nsolve[CharacteristicPolynomial[a,x] = = 0,x] finds roots

"Newton's method for some approximate roots"

g[x] = D[CharacteristicPolynomial[a,x],x] gives derivative

P[x] = x-CharacteristicPolynomial[a,x]/g[x]

X = a

x_1 = p[x]

c = x_1

c = p[c]

d = N[p[c]]

v = N[p[d]]

NewtonRaphson[a,b]

Print["v = ,"v] ; gives approximate root at fourth step.

EigenSystem[a]//N lists eigenvalues and eigenvectors

" Matrix operation"

JordanDecomposition[A]

SchurDecomposition[A]

LUDecomposition[A]

LUBackSubstitution[LUDecomposition[A],b]

QRDecomposition[A]

HessenbergDecomposition[A]

Minors[A] finds minors of matrix A

Minors[a,k] finds minor of order k

RowReduce[A]

NullSpace[A]

Dot[A,B] = A.B: computes the product

Inverse[A] gives inverse

Transpose[A]

Det[A] gives determinant

Tr[A] gives the trace of A

MatrixPower[A,n] gives power of A

SingularValues[A] gives singular values of A

References

1. Chard L.Burden, Numerical Analysis ,second edition ,
   U.S.A. ,1981
2. Sastry,S.S, Itroductory methods to numerical analysis, third edition,
   Prentice Hall of India ,2002
3. B.Baskaran, Engineering Mathematics, first edition , Avadi , 2000
4. R.Balakrrishnan Modern Algebra, Annamali Unversity, 1998
5. Seymour Lipschutz, Linear Algebra (schwam's), Second edition
   Newyork, 1974
6. Edvard D.Nering , Linear Algebra (matrix theory) , NewYork , 1970
   A S Deif , Advanced Matrix theory for scientists and Engineering
   Great Britain,1982
7. C.G.Broyden, Basic Matrix Theory , New York , 1975
8. Jack L.Goldberg, Matrix Theory with Application,
   U.S.A., 1991

10. N.J.Pullman, Matrix Theory with Application, U.S.A.,1976