chapter 4 : computing eigenvalues

4.1 introduction

problem : Given A, find λ and $x \neq 0$ such that $Ax = \lambda x$.

λ : e-value (e.g. frequency, growth rate, energy level)

x : e-vector (e.g. normal mode, principal component, bound state)

thm : Assume A is real and symmetric. Then the e-values λ_i are real and the e-vectors q_i form an orthonormal basis, i.e. $q_i^T q_j = 0$ for $i \neq j$, $||q_i||_2 = 1$, and any x can be written as a linear combination of the q_i. (pf : omit)

ex : $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$

$$f_A(\lambda) = \det \begin{pmatrix} 2-\lambda & -1 \\ -1 & 2-\lambda \end{pmatrix} = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = (\lambda-3)(\lambda-1)$$

$\lambda_1 = 3$: $Ax = 3x \Rightarrow \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 3\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

choose $x_1 = 1$, then $2x_1 - x_2 = 3x_1 \Rightarrow x_2 = -1, -x_1 + 2x_2 = 3x_2$ ok

$q_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow ||q_1||_2 = 1$

$\lambda_2 = 1$: $Ax = x \Rightarrow \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

choose $x_1 = 1$, then $2x_1 - x_2 = x_1 \Rightarrow x_2 = 1, -x_1 + 2x_2 = x_2$ ok

$q_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow ||q_2||_2 = 1$, $q_1^T q_2 = 0$ ok

obvious method for computing e-values

step 1. form $f_A(\lambda) = \det(A - \lambda I)$

step 2. solve $f_A(\lambda) = 0$ by the methods of chapter 2

ex : $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\tilde{A} = \begin{pmatrix} 1+\epsilon & 0 \\ 0 & 1-\epsilon \end{pmatrix}$: perturbed matrix

$$f_A(\lambda) = (1-\lambda)^2 = \lambda^2 - 2\lambda + 1 = 0 \Rightarrow \lambda = 1$$

$$f_{\tilde{A}}(\lambda) = (1+\epsilon-\lambda)(1-\epsilon-\lambda) = \lambda^2 - 2\lambda + 1 - \epsilon^2 = 0 \Rightarrow \lambda = 1 \pm \epsilon$$

1. A change in the elements of A of size ϵ leads to a change in the e-values of size ϵ.

2. A change in the coefficients of $f_A(\lambda)$ of size ϵ^2 leads to a change in the roots of size ϵ.

3. Hence the roots of $f_A(\lambda)$ depend sensitively on the coefficients, and this implies that the obvious method for computing e-values is unstable.

<u>ex</u> (Wilkinson)

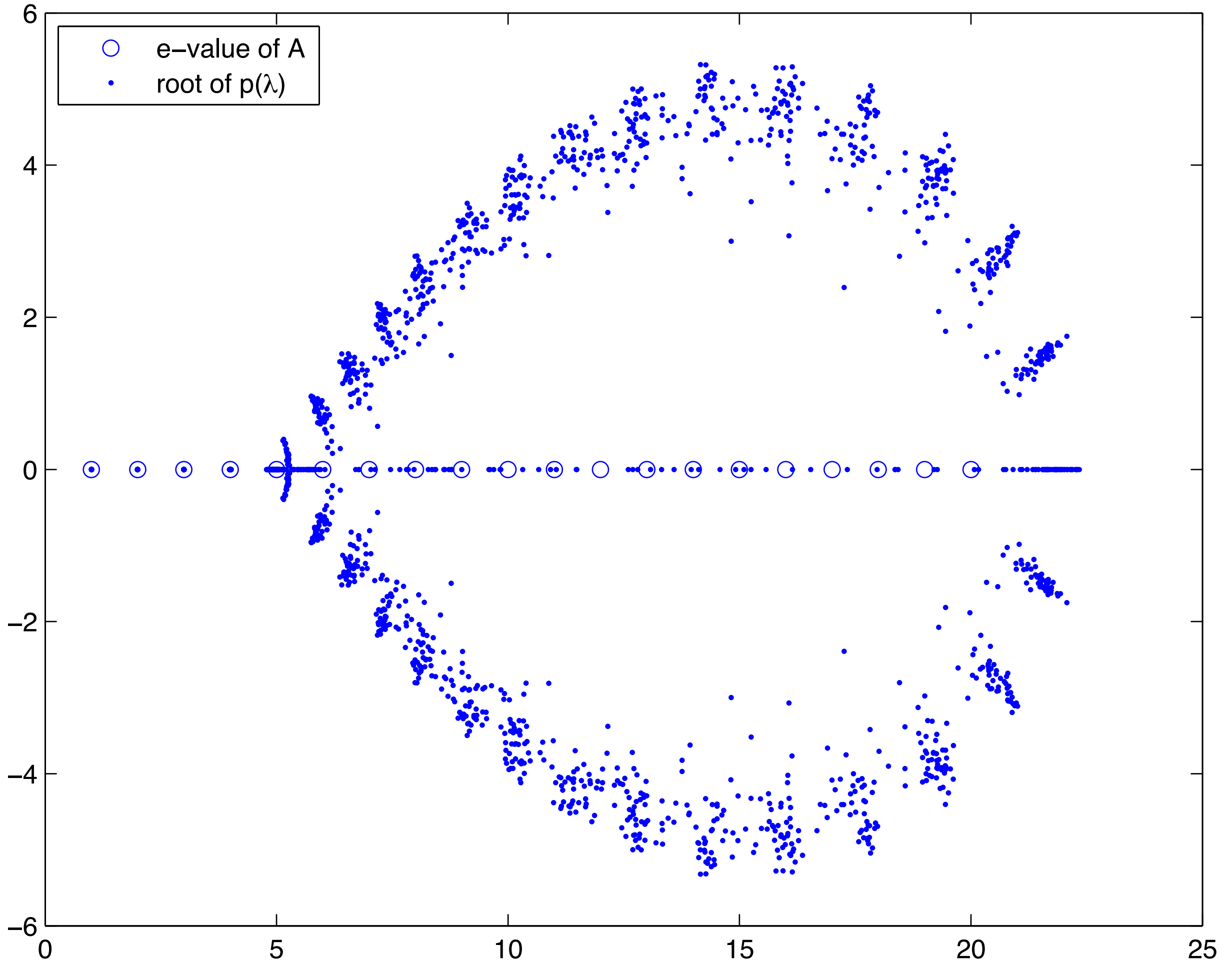
$A = \mathrm{diag}(1, 2, \ldots, 20)$

$f_A(\lambda) = (1-\lambda)(2-\lambda)\cdots(20-\lambda) = \sum_{k=0}^{20} a_k \lambda^k$

set $\tilde{a}_k = a_k(1 + 10^{-10}\epsilon_k)$, $\epsilon_k \in (0,1)$: random , $p(\lambda) = \sum_{k=0}^{20} \tilde{a}_k \lambda^k$, roots $= ?$

<u>Matlab</u>

```
plot(zeros(1,20),'o'); hold on;
for i=1:100
  r = roots(poly(1:20).*(ones(1,21)+1e-10*randn(1,21)));
  plot(r,'.'); axis([0,25,-6,6]);
end
```



1. This example shows that the roots of the characteristic polynomial are very sensitive to perturbations in the coefficients, and hence solving $f_A(\lambda) = 0$ numerically is not a practical method for computing e-values.

2. question : What method does Matlab use to compute the roots of $p(\lambda)$?

<u>def</u> : Given any $x \neq 0$, define $R_A(x) = \dfrac{x^TAx}{x^Tx}$: <u>Rayleigh quotient</u>.

<u>note</u>

1. For $x = q_i$, $R_A(q_i) = \dfrac{q_i^TAq_i}{q_i^Tq_i} = \dfrac{q_i^T\lambda_i q_i}{q_i^Tq_i} = \lambda_i$.

2. For $x \approx q_i$, $R_A(x)$ is an approximation to λ_i and we can derive an error estimate by Taylor expansion. First recall some notation.

$$f(x_1, x_2) = f(a_1, a_2) + \frac{\partial f}{\partial x_1}(a_1, a_2)(x_1 - a_1) + \frac{\partial f}{\partial x_2}(a_1, a_2)(x_2 - a_2) + \cdots$$

$$f(x) = f(a) + \nabla f(a) \cdot (x - a) + O(||x - a||^2) \quad , \quad \nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)$$

$$R_A(x) \ = \ R_A(q_i) \ + \ \nabla R_A(q_i) \cdot (x - q_i) \ + \ O(||x - q_i||^2)$$

$$\nabla R_A(x) \ = \ \nabla\left(\frac{x^TAx}{x^Tx}\right) = \frac{x^Tx \cdot \nabla(x^TAx) \ - \ x^TAx \cdot \nabla(x^Tx)}{(x^Tx)^2}$$

$$\nabla(x^Tx) \ = \ \nabla(x_1^2 + x_2^2) \ = \ (2x_1, 2x_2) \ = \ 2x^T$$

$$x^TAx \ = \ (x_1, x_2)\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \ = \ a_{11}x_1^2 \ + \ 2a_{12}x_1x_2 \ + \ a_{22}x_2^2$$

$$\nabla(x^TAx) \ = (2a_{11}x_1 \ + \ 2a_{12}x_2, 2a_{12}x_1 \ + \ 2a_{22}x_2) \ = \ 2(Ax)^T$$

$$\nabla R_A(x) \ = \ \frac{x^Tx \cdot 2(Ax)^T - \ x^TAx \cdot 2x^T}{(x^Tx)^2} \ = \ \frac{2}{x^Tx}((Ax)^T - R_A(x)x^T)$$

$$\nabla R_A(q_i) \ = \ \frac{2}{q_i^Tq_i}((Aq_i)^T - R_A(q_i)q_i^T) \ = \ 2((\lambda_i q_i)^T - \lambda_i q_i^T) \ = \ 0$$

$$\Rightarrow \ R_A(x) \ = \ \lambda_i \ + \ O(||x - q_i||^2) \ \ : \ \ \text{quadratic appoximation}$$

<u>4.2 power method</u> 16 Thu 3/16

idea : $v\,,\, Av\,,\, A^2v\,,\, \ldots$

<u>algorithm</u>

1. $v^{(0)}$: given , $||v^{(0)}||_2 = 1\,,\ \lambda^{(0)} = (v^{(0)})^TAv^{(0)}$
2. for $k = 1, 2, \ldots$
3. $\quad w \ = \ Av^{(k-1)}$ % this can be done efficiently if A is sparse
4. $\quad v^{(k)} \ = \ w/||w||_2$ % this is done to avoid overflow/underflow
5. $\quad \lambda^{(k)} \ = \ (v^{(k)})^TAv^{(k)}$ % $\lambda^{(k)} \to \lambda_1$: largest e-value of A in absolute value

$\underline{\text{ex}}$: $A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix} \Rightarrow \begin{cases} \lambda_1 = 5.214320 \\ \lambda_2 = 2.460811 \\ \lambda_3 = 1.324869 \end{cases}$

$v^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \frac{1}{\sqrt{3}} \Rightarrow \lambda^{(0)} = (v^{(0)})^T A v^{(0)} = 5$

power method

k	$\lambda^{(k)}$	$\lvert\lambda^{(k)} - \lambda_1\rvert$	$\dfrac{\lvert\lambda^{(k)} - \lambda_1\rvert}{\lvert\lambda^{(k-1)} - \lambda_1\rvert}$
0	5.000000	0.214320	-
1	5.181818	0.032502	0.151650
2	5.208193	0.006127	0.188513
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
∞	λ_1	0	$(\lambda_2/\lambda_1)^2$

thm : Assume that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$ and $q_1^T v^{(0)} \neq 0$.

Then $\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$, $|\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$.

pf : $v^{(0)} = \alpha_1 q_1 + \alpha_2 q_2 + \cdots + \alpha_n q_n$, where $\alpha_i = q_i^T v^{(0)}$

$$\begin{aligned} v^{(k)} = \beta_k A^k v^{(0)} &= \beta_k \left(\alpha_1 A^k q_1 + \alpha_2 A^k q_2 + \cdots + \alpha_n A^k q_n\right) \\ &= \beta_k \left(\alpha_1 \lambda_1^k q_1 + \alpha_2 \lambda_2^k q_2 + \cdots + \alpha_n \lambda_n^k q_n\right) \\ &= \beta_k \lambda_1^k \left(\alpha_1 q_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k q_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k q_n\right) \end{aligned}$$

$\Rightarrow$ $v^{(k)} \to \pm q_1$ as $k \to \infty$, $\pm$ depends on sign(λ_1) ok

note

1. The error is reduced by a constant factor in each step.

2. If $\alpha_1 = q_1^T v^{(0)} = 0$, then $v^{(k)} \to \pm q_2$, $\lambda^{(k)} \to \lambda_2$.

3. If $A = \frac{1}{h^2}\text{trid}(-1, 2, -1)$, the matrix form of $-D_+D_-$, then $w = Av$, can be coded as a loop.

for $i = 1 : n$; $w_i = (-v_{i-1} + 2v_i - v_{i+1})/h^2$; end % assuming $v_0 = v_{n+1} = 0$

This is more efficient than forming A and computing $w = Av$ by direct matrix-vector multiplication.

<u>question</u> : How can the other eigenvalues be obtained?

<u>4.3 inverse power method</u>

idea : apply power method to A^{-1}

1. $Aq_i = \lambda_i q_i \;\Rightarrow\; A^{-1}q_i = \lambda_i^{-1} q_i$: same e-vectors , reciprocal e-values

The largest e-value of A^{-1} is λ_n^{-1}.

2. $w = A^{-1}v \;\Leftrightarrow\; Aw = v$

<u>algorithm</u>

1. $v^{(0)}$: given , $||v^{(0)}||_2 = 1$
2. for $k = 1, 2, \ldots$
3. solve $Aw = v^{(k-1)}$ % e.g. LU factorization, etc.
4. $v^{(k)} = w/||w||_2$
5. $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

<u>ex</u> : $A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix} \Rightarrow \begin{cases} \lambda_1 = 5.214320 \\ \lambda_2 = 2.460811 \\ \lambda_3 = 1.324869 \end{cases}$, $v^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \frac{1}{\sqrt{3}}$

17 Thu 3/18

<u>inverse power method</u>

k	$\lambda^{(k)}$	$\lvert\lambda^{(k)} - \lambda_3\rvert$	$\dfrac{\lvert\lambda^{(k)} - \lambda_3\rvert}{\lvert\lambda^{(k-1)} - \lambda_3\rvert}$
0	5.000000	3.675131	-
1	3.816327	2.491457	0.677923
2	1.864903	0.540034	0.216754
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
∞	λ_3	0	hw

<u>summary</u>

1. power method : $\lambda^{(k)} \to \lambda_1$, convergence factor $= (\lambda_2/\lambda_1)^2$

   inverse power method : $\lambda^{(k)} \to \lambda_n$, convergence factor = hw

2. How to compute the other λ_i? Several methods are discussed in Math 571.

a) shifted inverse power method : $(A - \mu I)^{-1}$

b) QR method

   $A^{(0)} = Q^{(0)} R^{(0)}$, Q : orthogonal $(Q^T Q = I)$, R : upper triangular

   $A^{(1)} = R^{(0)} Q^{(0)} = Q^{(1)} R^{(1)}$: same e-values as A

   $\ldots$

   $A^{(k)} \to$ diagonal

3. recall : What method does Matlab use to compute the roots of a polynomial?

   answer : `type roots` , ...