chapter 4 : computing eigenvalues

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 $2 - x_2 = 3, -1 + 2x_2 = 3x_2 \Rightarrow x_2 = -1$

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

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

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

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

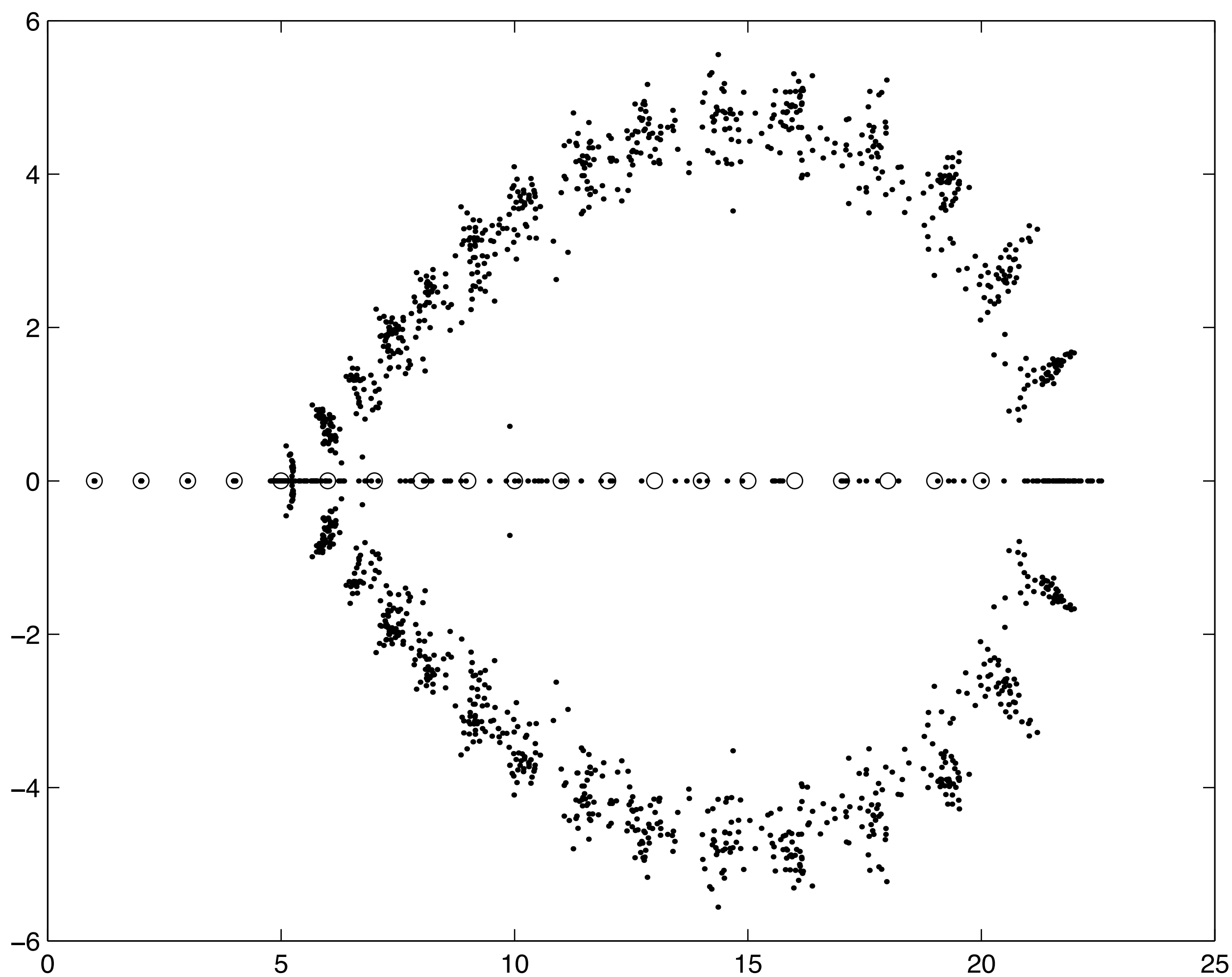
$A = \text{diag}(1, 2, \ldots, 20)$

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

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
```



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

def : Given any $x \neq 0$, define $R_A(x) = \dfrac{x^T A x}{x^T x}$: Rayleigh quotient.

note

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

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.

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

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

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

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

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

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

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

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

$$\nabla R_A(q_j) = \frac{2}{q_j^T q_j}((Aq_j)^T - R_A(q_j)q_j^T) = 2((\lambda_j q_j)^T - \lambda_j q_j^T) = 0$$

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

section 4.1 : power method

idea : v, Av, $A^2 v$, ...

algorithm

1. $v^{(0)}$: given , $||v^{(0)}||_2 = 1$
2. for $k = 1, 2, \ldots$
3. $\quad w = Av^{(k-1)}$ % if A is sparse, this can be done efficiently
4. $\quad v^{(k)} = w/||w||_2$ % this is done to avoid overflow/underflow
5. $\quad \lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ % $\lambda^{(k)} = \lambda_1 + O(||v^{(k)} - (\pm q_1)||^2)$: more soon

note : Suppose that $A = A_h$, where $(A_h v)_i = -D_+D_- v_i = \frac{1}{h^2}(-v_{i-1} + 2v_i - v_{i+1})$, assuming $h = \frac{1}{n+1}, v_0 = v_{n+1} = 0$. Then line 3, $w = Av$, can be coded as a loop.

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

This is more efficient than forming A_h and computing $w = A_h v$ by matrix-vector multiplication. This comment is also relevant for Computer Project 2, where $x_{n+1} = Bx_n + c$.

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

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

The $\pm$ depends on sign of λ_1.

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

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

$\Rightarrow v_{(k)} \sim \pm q_1$ as $k \to \infty$

If $q_1^T v^{(0)} = 0$, then the scheme converges to $\lambda_2, \pm q_2$. ok

note : The power method has some limitations.

1. it only gives the largest e-value λ_1
2. $v^{(k)}$, $\lambda^{(k)}$ converge linearly and the convergence factor $\left|\frac{\lambda_2}{\lambda_1}\right|$ may not be small

recall : linear convergence means $||v^{(k)} - (\pm q_1)|| \leq C||v^{(k-1)} - (\pm q_1)||$

section 4.2 : inverse iteration

idea : apply power method to A^{-1} , $(A - \mu I)^{-1}$, μ : shift

1. $Aq_i = \lambda_i q_i \Rightarrow A^{-1} q_i = \lambda_i^{-1} q_i$

The largest e-value of A^{-1} is λ_n^{-1}, so the vectors $v^{(k)}$ converge to $\pm q_n$.

2. $(A - \mu I)q_i = (\lambda_i - \mu)q_i \Rightarrow (A - \mu I)^{-1} q_i = (\lambda_i - \mu)^{-1} q_i$

The largest e-value of $(A - \mu I)^{-1}$ is $|\lambda_J - \mu|^{-1}$, where λ_J is the e-value of A closest to μ, so the vectors $v^{(k)}$ converge to $\pm q_J$.

3. $w = A^{-1}v \Rightarrow Aw = v$, $w = (A - \mu I)^{-1} v \Rightarrow (A - \mu I)w = v$

algorithm

1. $v^{(0)}$: given , $||v^{(0)}||_2 = 1$
2. for $k = 1, 2, \ldots$
3. solve $(A - \mu I)w = v^{(k-1)}$ % e.g. LU factorization, etc.
4. $v^{(k)} = w/||w||_2$
5. $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ % why not $(A - \mu I)^{-1}$?

thm : Assume that λ_J is the e-value of A closest to μ and λ_K is the next closest, i.e. $|\lambda_J - \mu| < |\lambda_K - \mu| < |\lambda_i - \mu|$ for $i \neq J, K$, and $q_J^T v^{(0)} \neq 0$.

Then $||v^{(k)} - (\pm q_J)|| = O\left(\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|^k\right)$, $|\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|^{2k}\right)$.

pf : as before , $\lambda_1 \to \frac{1}{\lambda_J - \mu}$, $\lambda_2 \to \frac{1}{\lambda_K - \mu}$ $\Rightarrow$ $\left|\frac{\lambda_2}{\lambda_1}\right| \to \left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|$ ok

note : Using a suitable shift μ, any e-value of A can be obtained and the convergence factor $\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|$ can be made arbitrarily small.

Rayleigh quotient iteration

idea : update μ

algorithm

1. $v^{(0)}$: given , $||v^{(0)}||_2 = 1$, $\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$
2. for $k = 1, 2, \ldots$
3. solve $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$
4. $v^{(k)} = w/||w||_2$
5. $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

thm : If $v^{(0)}$ is sufficiently close to an e-vector q_J, then

$$\left.\begin{aligned} ||v^{(k+1)} - (\pm q_J)|| &= O(||v^{(k)} - (\pm q_J)||^3) \\ |\lambda^{(k+1)} - \lambda_J| &= O(|\lambda^{(k)} - \lambda_J|^3) \end{aligned}\right\} \ : \ \text{cubic convergence}$$

pf : omit

ex : $A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}$, $\lambda_1 = 5.214319743377$, $v^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \frac{1}{\sqrt{3}}$

k	power method	shifted inverse iteration , $\mu = 5$	Rayleigh quotient iteration
0	5.0	5.0	5.0
1	5.181818	5.213114	5.213114
2	5.208192	5.214312617	5.214319743184
	2	6	10