L3b Eigenvalue Decomposition - Perturbation Theory

Ivan Slapničar

April 9, 2018

1 Eigenvalue Decomposition - Perturbation Theory

1.1 Prerequisites

The reader should be familiar with basic linear algebra concepts and facts about eigenvalue decomposition.

1.2 Competences

The reader should be able to understand and check the facts about perturbations of eigenvalues and eigenvectors.

1.3 Norms

In order to measure changes, we need to define norms. For more details and the proofs of the Facts below, see Section [ByDa14] and the references therein.

References

[ByDa14] R. Byers and B. N. Datta, Vector and Matrix Norms, Error Analysis, Efficiency, and Stability, in: L. Hogben, ed., 'Handbook of Linear Algebra', pp. 50.1-50.24, CRC Press, Boca Raton, 2014.

1.3.1 Definitions

Norm on a vector space X is a real-valued function $\| \ \| : X \to \mathbb{R}$ with the following properties:

1. $\|x\| \geq 0$ and $\|x\| = 0$ if and only if x is the zero vector *(Positive definiteness)*
2. $\|\lambda x\| = |\lambda| \|x\|$ *(Homogeneity)*
3. $\|x + y\| \leq \|x\| + \|y\|$ *(Triangle inequality)*

Commonly encountered vector norms for $x \in \mathbb{C}^n$ are:

- **Hölder norm** or p-**norm**: for $p \geq 1$, $\|x\|_p = \big(|x_1|^p + |x_2|^p + \cdots |x_n|^p\big)^{1/p}$,
- **Sum norm** or 1-**norm**: $\|x\|_1 = |x_1| + |x_2| + \cdots |x_n|$,
- **Euclidean norm** or 2-**norm**: $\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \cdots |x_n|^2}$,
- **Sup-norm** or ∞-**norm**: $\|x\|_\infty = \max\limits_{i=1,\ldots,n} |x_i|$.

Vector norm is **absolute** if $\||x|\| = \|x\|$.

Vector norm is **monotone** if $|x| \leq |y|$ implies $\|x\| \leq \|y\|$.

From every vector norm we can derive a corresponding **induced** matrix norm (also, **operator norm** or **natural norm**):

$$\|A\| = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|.$$

For matrix $A \in \mathbb{C}^{m\times n}$ we define:

- **Maximum absolute column sum norm**: $\|A\|_1 = \max\limits_{1\leq j\leq n} \sum_{i=1}^m |a_{ij}|$,
- __ Spectral norm__: $\|A\|_2 = \sqrt{\rho(A^*A)} = \sigma_{\max}(A)$ (largest singular value of A),
- **Maximum absolute row sum norm**: $\|A\|_\infty = \max\limits_{1\leq i\leq m} \sum_{j=1}^n |a_{ij}|$,
- **Euclidean norm** or **Frobenius norm**: $\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\operatorname{tr}(A^*A)}$.

Matrix norm is **consistent** if $\|A \cdot B\| \leq \|A\| \cdot \|B\|$, where A and B are compatible for matrix multiplication.

Matrix norm is **absolute** if $\||A|\| = \|A\|$.

1.3.2 Examples

```
In [1]: s=srand(421)
        x=rand(-4:4,5)
```

```
Out[1]: 5-element Array{Int64,1}:
          3
         -1
          2
          1
          1
```

```
In [2]: norm(x,1), norm(x), norm(x,Inf)
```

```
Out[2]: (8.0, 4.0, 3.0)
```

```
In [3]: A=rand(-4:4,7,5)
```

```
Out[3]: 7×5 Array{Int64,2}:
          2  -1   1  -2  -2
          2  -4   3   0   0
         -1  -4  -1  -2  -4
          2   0  -2  -2  -1
          4   4   3  -4  -3
         -3  -4   4  -3  -1
         -4  -4   4  -3   2
```

```
In [4]: norm(A,1), norm(A), norm(A,2), norm(A,Inf), vecnorm(A)
```

```
Out[4]: (21.0, 11.43978932236231, 11.43978932236231, 18.0, 16.492422502470642)
```

```
In [5]: norm(vec(A))
```

```
Out[5]: 16.492422502470642
```

1.3.3 Facts

1. $\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$ and $\|x\|_p$ are absolute and monotone vector norms.
2. A vector norm is absolute iff it is monotone.
3. **Convergence.** $x_k \to x_*$ iff for any vector norm $\|x_k - x_*\| \to 0$.
4. Any two vector norms are equivalent in the sense that, for all x and some $\alpha, \beta > 0$

$$\alpha\|x\|_\mu \leq \|x\|_\nu \leq \beta\|x\|_\mu.$$

   In particular:

- $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$,
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$,
- $\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty$.

2. **Cauchy-Schwartz inequality.** $|x^*y| \leq \|x\|_2\|y\|_2$.
3. **Hölder inequality.** if $p, q \geq 1$ and $\frac{1}{p} + \frac{1}{q} = 1$, then $|x^*y| \leq \|x\|_p\|y\|_q$.
4. $\|A\|_1$, $\|A\|_2$ and $\|A\|_\infty$ are induced by the corresponding vector norms.
5. $\|A\|_F$ is not an induced norm.
6. $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$ and $\|A\|_F$ are consistent.
7. $\|A\|_1$, $\|A\|_\infty$ and $\|A\|_F$ are absolute. However, $\||A|\|_2 \neq \|A\|_2$.
8. Any two matrix norms are equivalent in the sense that, for all A and some $\alpha, \beta > 0$

$$\alpha\|A\|_\mu \leq \|A\|_\nu \leq \beta\|A\|_\mu.$$

   In particular:

- $\frac{1}{\sqrt{n}}\|A\|_\infty \leq \|A\|_2 \leq \sqrt{m}\|A\|_\infty$,
- $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2$,
- $\frac{1}{\sqrt{m}}\|A\|_1 \leq \|A\|_2 \leq \sqrt{n}\|A\|_1$.

6. $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$.
7. $\|AB\|_F \leq \|A\|_F \|B\|_2$ and $\|AB\|_F \leq \|A\|_2 \|B\|_F$.
8. If $A = xy^*$, then $\|A\|_2 = \|A\|_F = \|x\|_2 \|y\|_2$.
9. $\|A^*\|_2 = \|A\|_2$ and $\|A^*\|_F = \|A\|_F$.
10. For a unitary matrix U of compatible dimension,

$$\|AU\|_2 = \|A\|_2, \quad \|AU\|_F = \|A\|_F, \quad \|UA\|_2 = \|A\|_2, \quad \|UA\|_F = \|A\|_F.$$

11. For A square, $\rho(A) \leq \|A\|$.
12. For A square, $A^k \to 0$ iff $> \rho(A) < 1$.

In [6]:
```
# Absolute norms
norm(A,1), norm(abs.(A),1), norm(A,Inf), norm(abs.(A),Inf), vecnorm(A),
vecnorm(abs.(A)),  norm(A),norm(abs.(A))
```

Out[6]: (21.0, 21.0, 18.0, 18.0, 16.492422502470642, 16.492422502470642, 11.43978932236231, 15.5

In [7]:
```
# Equivalence of norms
m,n=size(A)
norm(A,Inf)\sqrt(n),norm(A), sqrt(m)*norm(A,Inf)
```

Out[7]: (0.12422599874998833, 11.43978932236231, 47.62352359916263)

In [8]:
```
norm(A), vecnorm(A), sqrt(n)*norm(A)
```

Out[8]: (11.43978932236231, 16.492422502470642, 25.580146573078384)

In [9]:
```
norm(A,1)\sqrt(m),norm(A), sqrt(n)*norm(A,1)
```

Out[9]: (0.12598815766974242, 11.43978932236231, 46.95742752749558)

In [10]:
```
# Fact 12
norm(A), sqrt(norm(A,1)*norm(A,Inf))
```

Out[10]: (11.43978932236231, 19.44222209522358)

In [11]:
```
# Fact 13
B=rand(n,rand(1:9))
vecnorm(A*B), vecnorm(A)*norm(B), norm(A)*vecnorm(B)
```

Out[11]: (15.01635865162152, 33.906961090532626, 26.12124331440825)

In [12]:
```
# Fact 14
x=rand(10)+im*rand(10)
y=rand(10)+im*rand(10)
A=x*y'
norm(A), vecnorm(A), norm(x)*norm(y)
```

```
Out[12]: (6.78328917727892, 6.783289177278921, 6.783289177278921)
```

```
In [13]: # Fact 15
         A=rand(-4:4,7,5)+im*rand(-4:4,7,5)
         norm(A), norm(A'), vecnorm(A), vecnorm(A')
```

```
Out[13]: (14.906901435908395, 14.906901435908395, 21.908902300206645, 21.908902300206645)
```

```
In [14]: # Unitary invariance - generate random unitary matrix U
         U,R=qr(rand(size(A))+im*rand(size(A)),thin=false);
```

```
In [15]: norm(A), norm(U*A), vecnorm(A), vecnorm(U*A)
```

```
Out[15]: (14.906901435908395, 14.9069014359084, 21.908902300206645, 21.908902300206645)
```

```
In [16]: # Spectral radius
         A=rand(7,7)+im*rand(7,7)
         maximum(abs,eigvals(A)), norm(A,Inf), norm(A,1), norm(A), vecnorm(A)
```

```
Out[16]: (5.2341579591715774, 6.827084930229757, 6.70185034881129, 5.514868120397622, 5.96805775
```

```
In [17]: # Fact 18
         B=A/(maximum(abs,eigvals(A))+2)
         @show maximum(abs,eigvals(B))
         norm(B^100)
```

```
maximum(abs, eigvals(B)) = 0.7235338222792927
```

```
Out[17]: 9.28450234348096e-15
```

1.4 Errors and condition numbers

We want to answer the question:

How much the value of a function changes with respect to the change of its argument?

1.4.1 Definitions

For function $f(x)$ and argument x, the **absolute error** with respect to the **perturbation** of the argument δx is

$$\|f(x+\delta x)-f(x)\| = \frac{\|f(x+\delta x)-f(x)\|}{\|\delta x\|}\|\delta x\| \equiv \kappa\|\delta x\|.$$

The **condition** or **condition number** κ tells how much does the perturbation of the argument increase. (Its form resembles derivative.)

Similarly, the **relative error** with respect to the relative perturbation of the argument is

$$\frac{\|f(x+\delta x)-f(x)\|}{\|f(x)\|}=\frac{\|f(x+\delta x)-f(x)\|\cdot\|x\|}{\|\delta x\|\cdot\|f(x)\|}\cdot\frac{\|\delta x\|}{\|x\|}\equiv\kappa_{rel}\frac{\|\delta x\|}{\|x\|}.$$

1.5 Peturbation bounds

1.5.1 Definitions

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

Pair $(\lambda,x)\in\mathbb{C}\times\mathbb{C}^{n\times n}$ is an **eigenpair** of A if $x\neq 0$ and $Ax=\lambda x$.

Triplet $(y,\lambda,x)\in\times\mathbb{C}^n\times\mathbb{C}\times\mathbb{C}^n$ is an **eigentriplet** of A if $x,y\neq 0$ and $Ax=\lambda x$ and $y^*A=\lambda y^*$.

Eigenvalue matrix is a diagonal matrix $\Lambda=\operatorname{diag}(\lambda_1,\lambda_2,\ldots,\lambda_n)$.

If all eigenvalues are real, they can be increasingly ordered. $\Lambda^{\uparrow}$ is the eigenvalue matrix of increasingly ordered eigenvalues.

τ is a **permutation** of $\{1,2,\ldots,n\}$.

$\tilde{A}=A+\Delta A$ is a **perturbed matrix**, where ΔA is **perturbation**. $(\tilde{\lambda},\tilde{x})$ are the eigenpairs of $\tilde{A}$.

Condition number of a nonsingular matrix X is $\kappa(X)=\|X\|\|X^{-1}\|$.

Let $X,Y\in\mathbb{C}^{n\times k}$ with $\operatorname{rank}(X)=\operatorname{rank}(Y)=k$. The **canonical angles** between their column spaces, θ_i, are defined by $\cos\theta_i=\sigma_i$, where σ_i are the singular values of the matrix

$$(Y^*Y)^{-1/2}Y^*X(X^*X)^{-1/2}.$$

The **canonical angle matrix** between X and Y is

$$\Theta(X,Y)=\operatorname{diag}(\theta_1,\theta_2,\ldots,\theta_k).$$

1.5.2 Facts

Bounds become more strict as matrices have more structure. Many bounds have versions in spectral norm and Frobenius norm. For more details and the proofs of the Facts below, see Section [Li14], and the references therein.

References

[Li14] R.-C. Li, Matrix Perturbation Theory, in: L. Hogben, ed., 'Handbook of Linear Algebra', pp. 21.1-21.20, CRC Press, Boca Raton, 2014.

1. There exists τ such that

$$\|\Lambda-\tilde{\Lambda}_\tau\|_2\leq 4(\|A\|_2+\|\tilde{A}\|_2)^{1-1/n}\|\Delta A\|_2^{1/n}.$$

2. **First-order perturbation bounds.** Let (y,λ,x) be an eigentriplet of a simple λ. ΔA changes λ to $\tilde{\lambda}=\lambda+\delta\lambda$, where

$$\delta\lambda=\frac{y^*(\Delta A)x}{y^*x}+O(\|\Delta A\|_2^2).$$

3. Let λ be a semisimple eigenvalue of A with multiplicitiy k, and let $X, Y \in \mathbb{C}^{n\times k}$ be the matrices of the corresponding right and left eigenvectors, that is, $AX = \lambda X$ and $Y^*A = \lambda Y^*$, such that $Y^*X = I_k$. ΔA changes the k copies of μ to $\tilde{\mu} = \mu + \delta\mu_i$, where $\delta\mu_i$ are the eigenvalues of $Y^*(\Delta A)X$ up to $O(\|\Delta A\|_2^2$.

4. Perturbations of an inverse matrix are as follows: if $\|A\|_p < 1$, then $I - A$ is nonsingular and

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

with

$$\|I - A\|_p \leq \frac{1}{1 - \|A\|_p}, \qquad \|(I - A)^{-1} - I\|_p \leq \frac{\|A\|_p}{1 - \|A\|_p}.$$

5. **Geršgorin Circle Theorem.** If $X^{-1}AX = D + F$, where $D = \operatorname{diag}(d_1, \ldots, d_n)$ and F has zero diagonal entries, then

$$\sigma(A) \subseteq \bigcup_{i=1}^{n} D_i,$$

where

$$D_i = \left\{ z \in \mathbb{C} : |z - d_i| \leq \sum_{j=1}^{n} |f_{ij}| \right\}.$$

Moreover, by continuity, if a connected component of D consists of k circles, it contains k eigenvalues.

6. **Bauer-Fike Theorem.** If A is diagonalizable and $A = X\Lambda X^{-1}$ is its eigenvalue decomposition, then

$$\max_i \min_j |\tilde{\lambda}_i - \lambda_j| \leq \|X^{-1}(\Delta A)X\|_p \leq \kappa_p(X)\|\Delta A\|_p.$$

7. If A and $\tilde{A}$ are diagonalizable, there exists τ such that

$$\|\Lambda - \tilde{\Lambda}_\tau\|_F \leq \sqrt{\kappa_2(X)\kappa_2(\tilde{X})}\|\Delta A\|_F.$$

If Λ and $\tilde{\Lambda}$ are real, then

$$\|\Lambda^\uparrow - \tilde{\Lambda}^\uparrow\|_{2,F} \leq \sqrt{\kappa_2(X)\kappa_2(\tilde{X})}\|\Delta A\|_{2,F}.$$

8. If A is normal, there exists τ such that $\|\Lambda - \tilde{\Lambda}_\tau\|_F \leq \sqrt{n}\|\Delta A\|_F$.

9. **Hoffman-Wielandt Theorem.** If A and $\tilde{A}$ are normal, there exists τ such that $\|\Lambda - \tilde{\Lambda}_\tau\|_F \leq \|\Delta A\|_F$.

10. If A and $\tilde{A}$ are Hermitian, for any unitarily invariant norm $\|\Lambda^\uparrow - \tilde{\Lambda}^\uparrow\| \leq \|\Delta A\|$. In particular,

$$\max_i |\lambda_i^\uparrow - \tilde{\lambda}_i^\uparrow| \leq \|\Delta A\|_2,$$
$$\sqrt{\sum_i (\lambda_i^\uparrow - \tilde{\lambda}_i^\uparrow)^2} \leq \|\Delta A\|_F.$$

11. **Residual bounds.** Let A be Hermitian. For some $\tilde{\lambda} \in \mathbb{R}$ and $\tilde{x} \in \mathbb{C}^n$ with $\|\tilde{x}\|_2 = 1$, define **residual** $r = A\tilde{x} - \tilde{\lambda}\tilde{x}$. Then $|\tilde{\lambda} - \lambda| \leq \|r\|_2$ for some $\lambda \in \sigma(A)$.

12. Let, in addition, $\tilde{\lambda} = \tilde{x}^* A \tilde{x}$, let λ be closest to $\tilde{\lambda}$ and x be its unit eigenvector, and let

$$\eta = \operatorname{gap}(\tilde{\lambda}) = \min_{\lambda \neq \mu \in \sigma(A)} |\tilde{\lambda} - \mu|.$$

If $\eta > 0$, then

$$|\tilde{\lambda} - \lambda| \leq \frac{\|r\|_2^2}{\eta}, \quad \sin\theta(x, \tilde{x}) \leq \frac{\|r\|_2}{\eta}.$$

13. Let A be Hermitian, $X \in \mathbb{C}^{n \times k}$ have full column rank, and $M \in \mathcal{H}_k$ having eigenvalues $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$. Set $R = AX - XM$. Then there exist $\lambda_{i_1} \leq \lambda_{i_2} \leq \cdots \leq \lambda_{i_k} \in \sigma(A)$ such that

$$\max_{1 \leq j \leq k} |\mu_j - \lambda_{i_j}| \leq \frac{\|R\|_2}{\sigma_{\min}(X)},$$
$$\sqrt{\sum_{j=1}^{k} (\mu_j - \lambda_{i_j})^2} \leq \frac{\|R\|_F}{\sigma_{\min}(X)}.$$

(The indices i_j need not be the same in the above formulae.)

14. If, additionally, $X^* X = I$ and $M = X^* A X$, and if all but k of A's eigenvalues differ from every one of M's eigenvalues by at least $\eta > 0$, then

$$\sqrt{\sum_{j=1}^{k} (\mu_j - \lambda_{i_j})^2} \leq \frac{\|R\|_F^2}{\eta \sqrt{1 - \|R\|_F^2 / \eta^2}}.$$

15. Let $A = \begin{bmatrix} M & E^* \\ E & H \end{bmatrix}$ and $\tilde{A} = \begin{bmatrix} M & 0 \\ 0 & H \end{bmatrix}$ be Hermitian, and set $\eta = \min |\mu - \nu|$ over all $\mu \in \sigma(M)$ and $\nu \in \sigma(H)$. Then

$$\max |\lambda_j^{\uparrow} - \tilde{\lambda}_j^{\uparrow}| \leq \frac{2\|E\|_2^2}{\eta + \sqrt{\eta^2 + 4\|E\|_2^2}}.$$

16. Let

$$\begin{bmatrix} X_1^* \\ X_2^* \end{bmatrix} A \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix}, \quad \begin{bmatrix} X_1 & X_2 \end{bmatrix} \text{ unitary}, \quad X_1 \in \mathbb{C}^{n \times k}.$$

Let $Q \in \mathbb{C}^{n \times k}$ have orthonormal columns and for a Hermitian $k \times k$ matrix M set $R = AQ - QM$. Let $\eta = \min |\mu - \nu|$ over all $\mu \in \sigma(M)$ and $\nu \in \sigma(A_2)$. If $\eta > 0$, then

$$\| \sin \Theta(X_1, Q) \|_F \leq \frac{\|R\|_F}{\eta}.$$

1.5.3 Example - Nondiagonalizable matrix

In [18]:
```
A=[-3 7 -1; 6 8 -2; 72 -28 19]
```

Out[18]:
```
3×3 Array{Int64,2}:
 -3    7  -1
  6    8  -2
 72  -28  19
```

In [19]:
```
# (Right) eigenvectors
λ,X=eig(A)
λ
```

Out[19]:
```
3-element Array{Float64,1}:
 -6.0
 15.0
 15.0
```

In [20]:
```
X
```

Out[20]:
```
3×3 Array{Float64,2}:
  0.235702   0.218218  -0.218218
 -0.235702   0.436436  -0.436436
 -0.942809  -0.872872   0.872872
```

In [21]:
```
cond(X)
```

Out[21]:
```
9.091581997455394e7
```

In [22]:
```
# Left eigenvectors
λ1,Y=eig(A')
```

Out[22]:
```
(Complex{Float64}[-6.0+0.0im, 15.0+2.00883e-7im, 15.0-2.00883e-7im], Complex{Float64}[0
```

In [23]:
```
# Try k=2,3
k=3
Y[:,k]'*A-λ[k]*Y[:,k]'
```

Out[23]:
```
1×3 RowVector{Complex{Float64},Array{Complex{Float64},1}}:
 2.34268e-7+1.94885e-7im  9.60124e-15-3.9217e-15im  5.8567e-8+4.87212e-8im
```

In [24]:
```
ΔA=rand(3,3)/20
B=A+ΔA
```

Out[24]:
```
3×3 Array{Float64,2}:
 -2.98154    7.01968  -0.969565
  6.00433    8.00049  -1.99528
 72.0195   -27.9878   19.0199
```

```
In [25]: norm(ΔA)

Out[25]: 0.05083596017679965

In [26]: μ,Z=eig(B)

Out[26]: ([-6.07659, 15.5997, 14.5157], [0.237236 0.19424 -0.238936; -0.2348 0.389826 -0.478897;

In [27]: # Fact 2
         δλ=μ[1]-λ[1]

Out[27]: -0.07659245719926133

In [28]: k=1
         Y[:,k]'*ΔA*X[:,k]/(Y[:,k]⋅X[:,k])

Out[28]: -0.07696005334382856 + 0.0im
```

1.5.4 Example - Jordan form

```
In [29]: n=10
         c=0.5
         J=Bidiagonal(c*ones(n),ones(n-1),true)

Out[29]: 10×10 Bidiagonal{Float64}:
          0.5  1.0   ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅
           ⋅   0.5  1.0   ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅
           ⋅    ⋅   0.5  1.0   ⋅    ⋅    ⋅    ⋅    ⋅    ⋅
           ⋅    ⋅    ⋅   0.5  1.0   ⋅    ⋅    ⋅    ⋅    ⋅
           ⋅    ⋅    ⋅    ⋅   0.5  1.0   ⋅    ⋅    ⋅    ⋅
           ⋅    ⋅    ⋅    ⋅    ⋅   0.5  1.0   ⋅    ⋅    ⋅
           ⋅    ⋅    ⋅    ⋅    ⋅    ⋅   0.5  1.0   ⋅    ⋅
           ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅   0.5  1.0   ⋅
           ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅   0.5  1.0
           ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅    ⋅   0.5

In [30]: # Accurately defined eigenvalues
         λ=eigvals(J)

Out[30]: 10-element Array{Float64,1}:
          0.5
          0.5
          0.5
          0.5
          0.5
          0.5
```

```
 0.5
 0.5
 0.5
 0.5
```

```
In [31]: # Only one eigenvector
         X=eigvecs(J)
```

```
Out[31]: 10×10 Array{Float64,2}:
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
In [32]: x=eigvecs(J)[:,1]
         y=eigvecs(J')[:,1]
```

```
Out[32]: 10-element Array{Float64,1}:
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 1.0
```

```
In [33]: y'*full(J)-0.5*y'
```

```
Out[33]: 1×10 RowVector{Float64,Array{Float64,1}}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
In [34]: # Just one perturbed element in the lower left corner
         ΔJ=sqrt(eps())*[zeros(n-1);1]*eye(1,n)
```

```
Out[34]: 10×10 Array{Float64,2}:
 0.0          0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0          0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.49012e-8  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
In [35]: B=J+ΔJ
         μ=eigvals(B)

Out[35]: 10-element Array{Complex{Float64},1}:
          0.335062+0.0im
          0.366562+0.0969484im
          0.366562-0.0969484im
          0.449031+0.156866im
          0.449031-0.156866im
          0.550969+0.156866im
          0.550969-0.156866im
          0.633438+0.0969484im
          0.633438-0.0969484im
          0.664938+0.0im

In [36]: # Fact 2
         maximum(abs,λ-μ)

Out[36]: 0.16493848891190788

In [37]: y'*ΔJ*x/(y⋅x)

Out[37]: Inf
```

However, since B is diagonalizable, we can apply Bauer-Fike theorem to it:

```
In [38]: μ,Y=eig(B)

Out[38]: (Complex{Float64}[0.335062+0.0im, 0.366562+0.0969484im, 0.366562-0.0969484im, 0.449031+

In [39]: cond(Y)

Out[39]: 1.1068834616372582e7

In [40]: norm(inv(Y)*ΔJ*Y), cond(Y)*norm(ΔJ)

Out[40]: (0.16493848884660858, 0.16493848884660872)
```

1.5.5 Example - Normal matrix

In [41]:
```
using SpecialMatrices
```

In [42]:
```
n=6
C=Circulant(rand(-5:5,n))
```

Out[42]:
```
6×6 SpecialMatrices.Circulant{Int64}:
  0   1  -2   4   3  -1
 -1   0   1  -2   4   3
  3  -1   0   1  -2   4
  4   3  -1   0   1  -2
 -2   4   3  -1   0   1
  1  -2   4   3  -1   0
```

In [43]:
```
λ=eigvals(full(C))
```

Out[43]:
```
6-element Array{Complex{Float64},1}:
  3.5+6.06218im
  3.5-6.06218im
  5.0+0.0im
 -4.5+2.59808im
 -4.5-2.59808im
 -3.0+0.0im
```

In [44]:
```
ΔC=rand(n,n)*0.0001
```

Out[44]:
```
6×6 Array{Float64,2}:
 5.65113e-6  8.18593e-6  2.91374e-5  5.23826e-5  2.19303e-6  4.12383e-5
 7.38086e-5  3.304e-5    1.7022e-5   8.15197e-5  6.55649e-5  8.27823e-6
 2.52795e-5  8.71817e-5  1.60833e-5  4.22353e-5  5.85002e-5  8.79271e-5
 8.74645e-5  3.37922e-5  9.03781e-5  1.06812e-5  5.55664e-5  4.83179e-5
 4.21087e-5  5.83319e-5  1.72333e-5  3.66189e-5  2.85959e-5  4.56073e-5
 2.85931e-5  6.22676e-5  8.46095e-5  7.5246e-5   8.68421e-5  3.14651e-5
```

In [45]:
```
@show norm(ΔC)
μ=eigvals(C+ΔC)
```

```
norm(ΔC) = 0.00028746042358312345
```

Out[45]:
```
6-element Array{Complex{Float64},1}:
      3.5+6.06212im
      3.5-6.06212im
  5.00028+0.0im
 -4.50004+2.59808im
 -4.50004-2.59808im
 -3.00007+0.0im
```

1.5.6 Example - Hermitian matrix

In [46]:
```
m=10
n=6
A=rand(m,n)
# Some scaling
D=diagm((rand(n)-0.5)*exp(20))
A=A*D
```

Out[46]:
```
10×6 Array{Float64,2}:
 2.05475e8  4.09074e7  -9.13798e7  -5.90804e7      6.08877e6      -1.62603e7
 9.21091e7  3.41178e7  -9.99822e7  -1.26239e8      1.131e7        -2.09524e7
 1.84566e8  1.12501e7  -1.54004e8  -1.23988e8      5.19292e7      -1.51885e7
 4.61826e7  6.83336e6  -9.70188e7  -9.81788e7      1.40293e7      -1.63346e7
 1.40357e8  4.58961e6  -7.90101e7  -4.57018e7      3.64459e6  -69010.1
 8.72923e7  3.38662e7  -1.62994e8  -1.41557e8  11011.1            -7.06146e6
 1.83502e8  2.84315e7  -1.56965e8  -1.29304e8      8.22864e6      -2.39611e7
 2.54658e7  3.93192e7  -1.22663e8  -7.99995e7      5.5123e7       -2.89255e7
 2.06163e8  4.39879e7  -1.15831e7  -2.02923e8      2.71868e7      -3.03529e7
 2.09821e8  4.79292e7  -7.49529e7  -1.40758e8      3.11803e7      -3.29736e7
```

In [47]:
```
A=cor(A)
```

Out[47]:
```
6×6 Array{Float64,2}:
  1.0         0.264937   0.301115   -0.296486  -0.0579805  -0.187906
  0.264937    1.0        0.267789   -0.435122   0.121425   -0.696386
  0.301115    0.267789   1.0        -0.221459  -0.0135589  -0.318058
 -0.296486   -0.435122  -0.221459    1.0       -0.145986    0.524904
 -0.0579805   0.121425  -0.0135589  -0.145986   1.0        -0.531814
 -0.187906   -0.696386  -0.318058    0.524904  -0.531814    1.0
```

In [48]:
```
ΔA=cor(rand(m,n)*D)*1e-5
```

Out[48]:
```
6×6 Array{Float64,2}:
  1.0e-5       -4.13624e-6  -4.2146e-7   -3.04769e-7  -4.09441e-6  -1.00836e-6
 -4.13624e-6    1.0e-5      -3.09945e-6  -3.87776e-6   5.30034e-6  -3.9518e-7
 -4.2146e-7    -3.09945e-6   1.0e-5       7.09858e-6  -7.31398e-6   1.11485e-6
 -3.04769e-7   -3.87776e-6   7.09858e-6   1.0e-5      -5.31101e-6  -1.48879e-6
 -4.09441e-6    5.30034e-6  -7.31398e-6  -5.31101e-6   1.0e-5      -2.58469e-6
 -1.00836e-6   -3.9518e-7    1.11485e-6  -1.48879e-6  -2.58469e-6   1.0e-5
```

In [49]:
```
B=A+ΔA
```

Out[49]:
```
6×6 Array{Float64,2}:
  1.00001     0.264933   0.301115   -0.296486  -0.0579846  -0.187907
  0.264933    1.00001    0.267786   -0.435126   0.12143    -0.696386
  0.301115    0.267786   1.00001    -0.221451  -0.0135663  -0.318057
```

```
 -0.296486   -0.435126  -0.221451    1.00001   -0.145992    0.524903
 -0.0579846   0.12143   -0.0135663  -0.145992   1.00001    -0.531816
 -0.187907   -0.696386  -0.318057    0.524903  -0.531816    1.00001
```

```
In [50]: λ,U=eig(A)
         μ=eigvals(B)
         [λ  μ]
```

```
Out[50]: 6×2 Array{Float64,2}:
          0.156967  0.156978
          0.564736  0.564741
          0.697134  0.697143
          0.76263   0.762641
          1.25555   1.25556
          2.56298   2.56299
```

```
In [51]: norm(ΔA)
```

```
Out[51]: 2.710736332267096e-5
```

```
In [52]: # Residual bounds - how close is μ, y to λ[2],X[:,2]
         k=3
         μ=round(λ[k],2)
         y=round.(U[:,k],2)
         y=y/norm(y)
```

```
Out[52]: 6-element Array{Float64,1}:
          -0.7213
           0.430776
           0.240433
           0.0500902
          -0.470848
          -0.110199
```

```
In [53]: μ
```

```
Out[53]: 0.7
```

```
In [54]: # Fact 9
         r=A*y-μ*y
```

```
Out[54]: 6-element Array{Float64,1}:
           0.00329292
           0.000291516
           0.000633525
          -0.000910423
           0.00090635
           0.0027153
```

In [55]:
```
minimum(abs,μ-λ), norm(r)
```

Out[55]:
```
(0.0028657129459148667, 0.004511417460003287)
```

In [56]:
```
# Fact 10 - μ is Rayleigh quotient
μ=y⋅(A*y)
r=A*y-μ*y
```

Out[56]:
```
6-element Array{Float64,1}:
  0.00122361
  0.00152735
  0.0013233
 -0.00076672
 -0.00044445
  0.00239916
```

In [57]:
```
η=min(abs(μ-λ[k-1]),abs(μ-λ[k+1]))
```

Out[57]:
```
0.06549932581058704
```

In [58]:
```
μ-λ[k], norm(r)^2/η
```

Out[58]:
```
(-3.1513140098526904e-6, 0.0001850783226625135)
```

In [59]:
```
# Eigenvector bound
# cos(θ)
cosθ=dot(y,U[:,k])
# sin(θ)
sinθ=sqrt(1-cosθ^2)
sinθ,norm(r)/η
```

Out[59]:
```
(0.006425330622266883, 0.053156865324295983)
```

In [60]:
```
# Residual bounds - Fact 13
U=eigvecs(A)
Q=round.(U[:,1:3],2)
# Orthogonalize
X,R=qr(Q)
M=X'*A*X
R=A*X-X*M
μ=eigvals(M)
R
```

Out[60]:
```
6×3 Array{Float64,2}:
  0.000644802  -0.00110021   -0.00136248
  0.00618014    0.000421687  -0.000294417
```

```
 0.000779744  -0.000216846  -0.00101336
-0.0055574    -0.000117687   0.000397763
 0.00643536    0.00230102    0.0014369
-0.00838315   -0.00150849   -0.000402537
```

In [61]:
```
λ
```

Out[61]:
```
6-element Array{Float64,1}:
 0.156967
 0.564736
 0.697134
 0.76263
 1.25555
 2.56298
```

In [62]:
```
μ
```

Out[62]:
```
3-element Array{Float64,1}:
 0.157059
 0.697144
 0.564749
```

In [63]:
```
# The entries of μ are not ordered - which algorithm was called?
issymmetric(M)
```

Out[63]:
```
false
```

In [64]:
```
M=Hermitian(M)
R=A*X-X*M
μ=eigvals(M)
```

Out[64]:
```
3-element Array{Float64,1}:
 0.157059
 0.564749
 0.697144
```

In [65]:
```
η=λ[4]-λ[3]
```

Out[65]:
```
0.06549617449657719
```

In [66]:
```
norm(λ[1:3]-μ), vecnorm(R)^2/η
```

Out[66]:
```
(9.31361491205613e-5, 0.0029951059103223037)
```

In [67]:
```
# Fact 13
M=A[1:3,1:3]
H=A[4:6,4:6]
E=A[4:6,1:3]
# Block-diagonal matrix
B=cat([1,2],M,H)
```

Out[67]:
```
6×6 Array{Float64,2}:
 1.0        0.264937  0.301115   0.0        0.0        0.0
 0.264937  1.0        0.267789   0.0        0.0        0.0
 0.301115  0.267789  1.0         0.0        0.0        0.0
 0.0        0.0        0.0        1.0       -0.145986   0.524904
 0.0        0.0        0.0       -0.145986   1.0       -0.531814
 0.0        0.0        0.0        0.524904  -0.531814   1.0
```

In [68]:
```
η=minimum(abs,eigvals(M)-eigvals(H))
μ=eigvals(B)
[λ μ]
```

Out[68]:
```
6×2 Array{Float64,2}:
 0.156967  0.322199
 0.564736  0.698823
 0.697134  0.744951
 0.76263   0.854027
 1.25555   1.55623
 2.56298   1.82377
```

In [69]:
```
2*norm(E)^2/(η+sqrt(η^2+4*norm(E)^2))
```

Out[69]:
```
0.9133785164898836
```

In [70]:
```
# Eigenspace bounds - Fact 14
B=A+ΔA
μ,V=eig(B)
```

Out[70]:
```
([0.156978, 0.564741, 0.697143, 0.762641, 1.25556, 2.56299], [-0.0482105 -0.306212 ...
```

In [71]:
```
# sin(Θ(U[:,1:3],V[:,1:3]))
X=U[:,1:3]
Q=V[:,1:3]
cosθ=svdvals(sqrtm(Q'*Q)*Q'*X*sqrtm(X'*X))
sinθ=sqrt.(1-cosθ.^2)
```

Out[71]:
```
3-element Array{Float64,1}:
 7.59521e-7
 4.28887e-6
 7.03148e-5
```

```
In [72]: # Bound
         M=Q'*A*Q

Out[72]: 3×3 Array{Float64,2}:
           0.156967    -2.16968e-6   2.56127e-6
          -2.16968e-6   0.564736    -1.12995e-6
           2.56127e-6  -1.12995e-6   0.697134

In [73]: R=A*Q-Q*M

Out[73]: 6×3 Array{Float64,2}:
           1.7645e-6    3.2951e-7   -1.0646e-6
          -1.7178e-6    1.20688e-7  -1.61113e-6
           5.73265e-6   2.1145e-6    2.88202e-6
           2.51765e-6   3.3477e-7    2.48137e-6
          -1.45408e-6   9.07386e-7   1.91377e-6
           1.49139e-6  -7.85053e-7  -9.79858e-9

In [74]: eigvals(M), λ

Out[74]: ([0.156967, 0.697134, 0.564736], [0.156967, 0.564736, 0.697134, 0.76263, 1.25555, 2.562

In [75]: η=abs(eigvals(M)[3]-λ[4])
         vecnorm(sinθ), vecnorm(R)/η

Out[75]: (7.044952733878622e-5, 4.451575856422721e-5)
```