



UNIVERSITI TEKNOLOGI MALAYSIA

SME 3023 Applied Numerical Methods

Solution of Matrix Eigenvalue Problems

Abu Hasan Abdullah

Faculty of Mechanical Engineering

Sept 2012



1 Introduction

Outline

- 1 Introduction
- 2 Engineering Applications

- 1 Introduction
- 2 Engineering Applications
- 3 Definitions and Basic Facts
 - Eigenvalue & Eigenvector
 - Characteristic Equation
 - Standard & General Eigenvalue Problems

- 1 Introduction
- 2 Engineering Applications
- 3 Definitions and Basic Facts
 - Eigenvalue & Eigenvector
 - Characteristic Equation
 - Standard & General Eigenvalue Problems
- 4 Power Method

- 1 Introduction
- 2 Engineering Applications
- 3 Definitions and Basic Facts
 - Eigenvalue & Eigenvector
 - Characteristic Equation
 - Standard & General Eigenvalue Problems
- 4 Power Method
- 5 Faddeev-Leverrier Method



- 1 Introduction
- 2 Engineering Applications
- 3 Definitions and Basic Facts
 - Eigenvalue & Eigenvector
 - Characteristic Equation
 - Standard & General Eigenvalue Problems
- 4 Power Method
- 5 Faddeev-Leverrier Method
- 6 Matlab `eig` Function



- 1 Introduction
- 2 Engineering Applications
- 3 Definitions and Basic Facts
 - Eigenvalue & Eigenvector
 - Characteristic Equation
 - Standard & General Eigenvalue Problems
- 4 Power Method
- 5 Faddeev-Leverrier Method
- 6 Matlab `eig` Function
- 7 Bibliography



Introduction

- Eigenvalues play an important role in situations where the matrix is a transformation from one vector space onto itself.
- Systems of linear ordinary differential equations are the primary examples.
- The eigenvalues can correspond to critical values of stability parameters, or energy levels of atoms, or frequencies of vibration—see Figure 1.

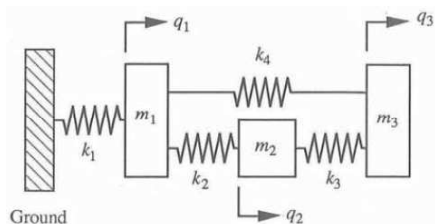


Figure 1 : Vibration of mass-spring system with three degrees of freedom.

The equations of motion for a system of masses and spring shown in Figure 1 are

$$\begin{aligned}m_1\ddot{q}_1 + (k_1 + k_2 + k_4)q_1 - k_2q_2 - k_4q_3 &= 0 \\m_2\ddot{q}_2 - k_2q_1 + (k_2 + k_3)q_2 - k_3q_3 &= 0 \\m_3\ddot{q}_3 - k_4q_1 - k_3q_2 + (k_3 + k_4)q_3 &= 0\end{aligned}\tag{1}$$



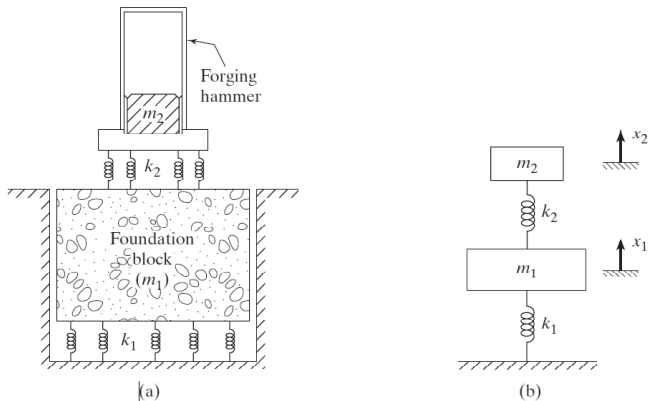


Figure 2 : Forging hammer.

Problem Statement:

A forging hammer of mass m_1 is mounted on a concrete foundation block of mass m_2 . The stiffness of the springs underneath the forging hammer and the foundation block are given by k_2 and k_1 , respectively (see Figure 2). The system undergoes simple harmonic motion at one of its natural frequencies ω , which are given by

$$\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \quad (a)$$

where ω^2 is the eigenvalue and $\vec{X}^T = \{x_1 \ x_2\}$ is the eigenvector or mode shape (displacement pattern) of the system. Determine the natural frequencies and mode shapes of the system for the following data:

Solution:

Work through the example.

Problem Statement:

The pin-ended column shown in Figure 3(a) is subjected to a compressive (axial) force, P . If the column is perturbed slightly as shown in Figure 3(b), as might occur from a slight vibration of the supports, it may not return to horizontal position even after removal of the disturbances; instead, the deflection might grow if the load is sufficiently large. Such load is called the **buckling load**.

Solution:

Work through the example.

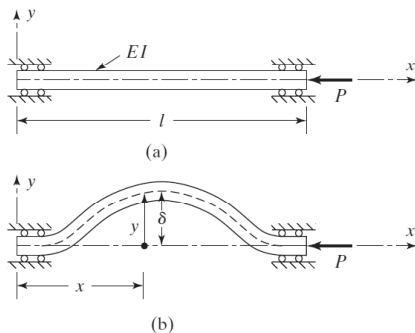


Figure 3 : Pin-ended column.

Definitions and Basic Facts

- A matrix is called **symmetric** if it is equal to its transpose,

$$\mathbf{A} = \mathbf{A}^T \quad \text{or} \quad a_{ij} = a_{ji} \quad (2)$$

- A matrix is called **Hermitian** or **self-adjoint** if it equals the complex conjugate of its transpose (its **Hermitian conjugate**, denoted by “†”)

$$\mathbf{A} = \mathbf{A}^\dagger \quad \text{or} \quad a_{ij} = a_{ji}^* \quad (3)$$

- A matrix is termed **orthogonal** if its transpose equals its inverse,

$$\mathbf{A}^T \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{A}^T = \mathbf{1} \quad (4)$$

and unitary if its Hermitian conjugate equals its inverse.

- A matrix is called **normal** if it **commutes** with its Hermitian conjugate,

$$\mathbf{A} \cdot \mathbf{A}^\dagger = \mathbf{A}^\dagger \cdot \mathbf{A} \quad (5)$$



Definitions and Basic Facts

Eigenvalue & Eigenvector

- An **eigenvalue** and **eigenvector** of a square matrix $[A]$ are a scalar λ , and a nonzero vector \vec{X} so that

$$[A]\vec{X} = \lambda\vec{X}$$

- Standard eigenvalue problem is defined by the homogeneous equations

$$(a_{11} - \lambda)x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 + a_{23}x_3 + \dots + a_{2n}x_n = 0$$

...

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + (a_{nn} - \lambda)x_n = 0 \tag{6}$$

and in matrix form

$$[A]\vec{X} = \lambda\vec{X} \tag{7}$$



Definitions and Basic Facts

Eigenvalue & Eigenvector

- In Eq. (7),

- 1 $[\mathbf{A}]$ is a **known** square, $(n \times n)$, matrix

$$[\mathbf{A}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (8)$$

- 2 \vec{X} is the **unknown** n -component vector,

$$\vec{X} = \left\{ \begin{array}{c} x_1 \\ x_2 \\ \dots \\ x_n \end{array} \right\} \quad (9)$$

- 3 λ is an **unknown** scalar, a.k.a. **eigenvalue**.

- Eq. (7) can be re-written as

$$[[\mathbf{A}] - \lambda[\mathbf{I}]]\vec{X} = \vec{0} \quad (10)$$

where $[\mathbf{I}]$ is the **identity matrix** of order n .

- Eq. (10) has **nontrivial** solution because it represents a system of n homogeneous equation in $n + 1$ unknowns. Thus the **determinant** of the coefficient matrix of \vec{X} in Eq. (10) must be zero:

$$|[\mathbf{A}] - \lambda[\mathbf{I}]| = 0 \quad (11)$$



Definitions and Basic Facts

Characteristic Equation

- If we expand Eq. (11) we get

$$\begin{vmatrix} (a_{11} - \lambda) & a_{12} & \dots & a_{1n} \\ a_{21} & (a_{22} - \lambda) & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & (a_{nn} - \lambda) \end{vmatrix} = 0 \quad (12)$$

which, upon further expansion, gives an n^{th} order polynomial in λ ,

$$P_n(\lambda) = a_n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_2 \lambda^2 + a_1 \lambda + a_0 = 0 \quad (13)$$

called the **characteristic equation**, where $a_0, a_1, a_2 \dots a_n$ are **coefficients of polynomial**. Assuming $\lambda_1, \lambda_2, \dots, \lambda_n$ are roots of the characteristic equation, then there will be n solutions to Eq. (7).



Definitions and Basic Facts

Characteristic Equation

- Corresponding to each distinct eigenvalue λ_i , a nontrivial solution of linear equations in Eq. (7) can be determined through

$$[\mathbf{A}]\vec{X}^{(i)} = \lambda_i\vec{X}^{(i)} \quad (14)$$

where

$$\vec{X}^{(i)} = \begin{Bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \dots \\ x_n^{(i)} \end{Bmatrix} \quad (15)$$

is called the **eigenvector** corresponding to eigenvalue λ_i .



Definitions and Basic Facts

Standard & General Eigenvalue Problems

Standard Eigenvalue Problem

- Eigenvalue problem as expressed by Eq. (7), repeated below,

$$(a_{11} - \lambda)x_1 + a_{12}x_2 + \dots + a_{1n}x_n = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 + \dots + a_{2n}x_n = 0$$

...

$$a_{n1}x_1 + a_{n2}x_2 + \dots + (a_{nn} - \lambda)x_n = 0$$

and in matrix form

$$[\mathbf{A}]\vec{X} = \lambda\vec{X}$$

is known as the **standard eigenvalue problem**.

General Eigenvalue Problem

- Many physical problems, however, are expressed as the **general eigenvalue problem** given by

$$[\mathbf{A}]\vec{X} = \lambda[\mathbf{B}]\vec{X} \quad (16)$$

- General eigenvalue problem of Eq. (16) can be reduced to the standard eigenvalue problem of Eq. (7). We shall deal with this conversion later!



Power Method

- Since each eigenvector is associated with an eigenvalue, we often refer to an eigenvector X and eigenvalue λ that correspond to one another as an **eigenpair**.
- Many of the “real world” applications are primarily interested in the **dominant eigenpair**.
- The **dominant eigenvector** of a matrix is an eigenvector corresponding to the eigenvalue of largest magnitude (for real numbers, largest absolute value) of that matrix. The method that is used to find this eigenvector is called the **power method**.



Power Method

Sequencing the Iteration

Like the Jacobi and Gauss-Seidel methods, the **power method** for approximating eigenvalues is iterative.

- ① First we assume that the matrix \mathbf{A} has a dominant eigenvalue with corresponding dominant eigenvectors.
- ② Then we choose an initial approximation of one of the dominant eigenvectors of \mathbf{A} . This initial approximation must be a nonzero vector in R^n .
- ③ Finally we form the sequence given by

$$\begin{aligned}x_1 &= \mathbf{A}x_0 \\x_2 &= \mathbf{A}x_1 = \mathbf{A}(\mathbf{A}x_0) = \mathbf{A}^2x_0 \\x_3 &= \mathbf{A}x_2 = \mathbf{A}(\mathbf{A}^2x_0) = \mathbf{A}^3x_0 \\&\vdots \\x_k &= \mathbf{A}x_{k-1} = \mathbf{A}(\mathbf{A}^{k-1}x_0) = \mathbf{A}^kx_0\end{aligned}\tag{17}$$



Power Method

Sequencing the Iteration—Example 1

We demonstrate those sequences on the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix}$$

and, for Step 1, arbitrarily choose

$$x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$x_1 = \mathbf{A}x_0 = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 9 \\ 5 \end{bmatrix}$$

$$\Rightarrow x'_1 = \begin{bmatrix} 1.8 \\ 1 \end{bmatrix}$$

$$x_2 = \mathbf{A}x_1 = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1.8 \\ 1 \end{bmatrix} = \begin{bmatrix} 11.4 \\ 5.8 \end{bmatrix}$$

$$\Rightarrow x'_2 = \begin{bmatrix} 1.965517241 \\ 1 \end{bmatrix}$$

$$x_3 = \mathbf{A}x_2 = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1.965517241 \\ 1 \end{bmatrix} = \begin{bmatrix} 11.89655172 \\ 5.96551724 \end{bmatrix}$$

$$\Rightarrow x'_3 = \begin{bmatrix} 1.994219653 \\ 1 \end{bmatrix}$$

$$x_4 = \mathbf{A}x_3 = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1.994219653 \\ 1 \end{bmatrix} = \begin{bmatrix} 11.98265896 \\ 5.99421965 \end{bmatrix}$$

$$\Rightarrow x'_4 = \begin{bmatrix} 1.99903568 \\ 1 \end{bmatrix}$$

It looks like an eigenvector is

$$x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$



Power Method

Sequencing the Iteration-Example 1 (continued)

The corresponding eigenvalue is

$$\frac{x^T Ax}{x^T x} = \frac{\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}}{\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \frac{[30]}{[5]} \implies \lambda = 6$$

Power Method

Algorithm & Matlab Implementation

The power method iteration in effect takes successively higher powers of matrix times initial starting vector

Algorithm: Power Method

x_0 = arbitrary nonzero vector

for $k = 1, 2, \dots$

$$y_k = \mathbf{A}x_{k-1}$$

$$x_k = y_k / \|y_k\|_\infty$$

end

Implementing the algorithm in Matlab:

Matlab Session

```
>> A = hilb(5)      % matrix A
>> x0 = ones(5,1)  % nonzero vector
>> yk = A*x0
>> xk = yk/norm(yk)
```

- Compare the new value of x with the original.
- Repeat the last two lines (**hint**: use the scroll up button).
- Compare the newest value of x with the previous one and the original. Notice that there is less change between the second two.
- Repeat the last two commands over and over until the values stop changing.



- Let \mathbf{A} be an $n \times n$ matrix. The determination of eigenvalues and eigenvectors requires the solution of

$$\mathbf{A}X = \lambda X$$

where λ is the eigenvalue corresponding to the eigenvector X . The values λ must satisfy the equation

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

Hence λ is a root of an n th degree polynomial $P(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$, which we write in the form

$$P(\lambda) = \lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} + \dots + c_{n-2}\lambda^2 + c_{n-1}\lambda + c_n \quad (18)$$

The Faddeev-Leverrier algorithm is an efficient method for finding the coefficients c_k of the polynomial $P(\lambda)$. As an additional benefit, the inverse matrix \mathbf{A}^{-1} is obtained at no extra computational expense.



Faddeev-Leverrier Method

Trace of Matrix \mathbf{A}

- The trace of the matrix \mathbf{A} , written $\text{Tr}[\mathbf{A}]$, is

$$\text{Tr}[\mathbf{A}] = a_{1,1} + a_{2,2} + \dots + a_{n,n} \quad (19)$$

- The algorithm generates a sequence of matrices $\mathbf{B}_{k=1}^n$ and uses their traces to compute the coefficients of $P(\lambda)$,

$$\begin{array}{lll} \mathbf{B}_1 = \mathbf{A} & \text{and} & p_1 = \text{Tr}[\mathbf{B}_1] \\ \mathbf{B}_2 = \mathbf{A}(\mathbf{B}_1 - p_1\mathbf{I}) & \text{and} & p_2 = \frac{1}{2}\text{Tr}[\mathbf{B}_2] \\ \vdots & \dots & \vdots \\ \mathbf{B}_k = \mathbf{A}(\mathbf{B}_{k-1} - p_{k-1}\mathbf{I}) & \text{and} & p_k = \frac{1}{k}\text{Tr}[\mathbf{B}_k] \\ \vdots & \dots & \vdots \\ \mathbf{B}_n = \mathbf{A}(\mathbf{B}_{n-1} - p_{n-1}\mathbf{I}) & \text{and} & p_n = \frac{1}{n}\text{Tr}[\mathbf{B}_n] \end{array} \quad (20)$$



Faddeev-Leverrier Method

Characteristic Polynomial

- Then the characteristic polynomial, Eq. 18, can be re-written as

$$P(\lambda) = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \dots + p_{n-2}\lambda^2 + p_{n-1}\lambda + p_n \quad (21)$$

- and the inverse matrix is given by

$$\mathbf{A}^{-1} = \frac{1}{p_n}(\mathbf{B}_{n-1} - p_{n-1}\mathbf{I}) \quad (22)$$



Faddeev-Leverrier Method

Example 1

Problem Statement:

Use Faddeev's method to find the characteristic polynomial and inverse of the matrix

$$\begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & -1 & 2 \end{bmatrix}$$

Solution:

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

$$\mathbf{B}_1 = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & -1 & 2 \end{bmatrix}$$

$$p_1 = \text{Tr}[\mathbf{B}_1] = 6$$



Solution (continued):

$$\mathbf{B}_2 = \mathbf{A}(\mathbf{B}_1 - p_1\mathbf{I})$$

$$\begin{aligned}\mathbf{B}_2 &= \begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & -1 & 2 \end{bmatrix} \left(\begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & -1 & 2 \end{bmatrix} - 6 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} -6 & 1 & -3 \\ 3 & -8 & -3 \\ -1 & 1 & -8 \end{bmatrix}\end{aligned}$$

$$p_2 = \frac{1}{2}\text{Tr}[\mathbf{B}_2] = \frac{1}{2}(-22) = -11$$

Solution (continued):

$$\begin{aligned} \mathbf{B}_3 &= \mathbf{A}(\mathbf{B}_2 - p_2\mathbf{I}) \\ \mathbf{B}_3 &= \begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & -1 & 2 \end{bmatrix} \left(\begin{bmatrix} -6 & 1 & -3 \\ 3 & -8 & -3 \\ -1 & 1 & -8 \end{bmatrix} - (-11) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix} \end{aligned}$$

$$p_3 = \frac{1}{3}\text{Tr}[\mathbf{B}_3] = \frac{1}{3}(18) = 6$$

The characteristic polynomial is thus

$$P(\lambda) = \lambda^3 - \sum_{i=1}^3 p_i \lambda^{n-i} = -6 + 11\lambda - 6\lambda^2 + \lambda^3$$



Faddeev-Leverrier Method

Example 1

Solution (continued):

The inverse matrix is

$$\begin{aligned} \mathbf{A}^{-1} &= \frac{1}{p_3} (\mathbf{B}_{3-1} - p_{3-1} \mathbf{I}) \\ &= \frac{1}{6} \left(\begin{bmatrix} -6 & 1 & -3 \\ 3 & -8 & -3 \\ -1 & 1 & -8 \end{bmatrix} - (-11) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \\ &= \frac{1}{6} \left(\begin{bmatrix} -6 & 1 & -3 \\ 3 & -8 & -3 \\ -1 & 1 & -8 \end{bmatrix} - \begin{bmatrix} -11 & 0 & 0 \\ 0 & -11 & 0 \\ 0 & 0 & -11 \end{bmatrix} \right) \\ &= \frac{1}{6} \begin{bmatrix} 5 & 1 & -3 \\ 3 & 3 & -3 \\ -1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{6} & \frac{1}{6} & \frac{1}{2} \end{bmatrix} \end{aligned}$$



Matlab eig Function

Example 1

Problem Statement:

$$\begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

Solution:

Matlab Session

```
>> A = [-149 -50 -154; 537 180 546; -27 -9 -25]
>> [X,lambda] = eig(A)
>> lambda1 = lambda(1,:)
>> lambda2 = lambda(2,:)
>> lambda3 = lambda(3,:)
>> X1 = X(:,1)
>> X2 = X(:,2)
>> X3 = X(:,3)
>> LHS1 = A*X1
>> RHS1 = lambda1*X1
>> % Check if LHS=RHS
```



Bibliography

- ① STEVEN C. CHAPRA, RAYMOND P. CANALE (2006): *Applied Numerical Methods for Engineers, 5ed*, ISBN 007-124429-8, McGraw-Hill
- ② SINGIRESU S. RAO (2002): *Applied Numerical Methods for Engineers and Scientists*, ISBN 0-13-089480-X, Prentice Hall
- ③ DAVID KINCAID AND WARD CHENEY (1991): *Numerical Analysis: Mathematics of Scientific Computing*, ISBN 0-534-13014-3, Brooks/Cole Publishing Co.
- ④ STEVEN C. CHAPRA (2005): *Applied Numerical Methods with MATLAB for Engineers and Scientists*, ISBN 007-124484-0, McGraw-Hill
- ⑤ WILLIAM J. PALM III (2005): *Introduction to Matlab 7 for Engineers*, ISBN 007-123262-1, McGraw-Hill
- ⑥ JOHN D. ANDERSON, JR. (1995): *Computational Fluid Dynamics–The Basics with Applications*, ISBN 007-113210-4, McGraw-Hill

