

PHYS-4007/5007: Computational Physics
Course Lecture Notes
Section VII

Dr. Donald G. Luttermoser
East Tennessee State University

Version 7.1

Abstract

These class notes are designed for use of the instructor and students of the course **PHYS-4007/5007: Computational Physics** taught by Dr. Donald Luttermoser at East Tennessee State University.

VII. Matrices and Solutions to Linear Equations

A. Introduction: Setting Up the Problem.

1. There may be times when you have a system of N linear equations with N unknowns:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N = b_1 \quad (\text{VII-1})$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N = b_2 \quad (\text{VII-2})$$

$$\vdots = \vdots$$

$$a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N = b_N \quad (\text{VII-3})$$

- a) In many cases, the a and b values are known, so your problem is to solve for all of the x values.
- b) To solve this problem, we must set the problem up as a **matrix equation**:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} \quad (\text{VII-4})$$

$$AX = B. \quad (\text{VII-5})$$

- c) The solution for the X vector is then found by inverting the A matrix:

$$A^{-1}AX = A^{-1}B \quad (\text{VII-6})$$

$$X = A^{-1}B. \quad (\text{VII-7})$$

2. Before discussing the techniques for carrying out such an inversion, we need to go over some basic linear algebra and discuss various types of matrices that one might encounter in physics.

3. Also, since matrices play a big role in quantum mechanics, we will use the formalism that is used in QM to describe vectors and matrices.

B. Linear Algebra.

1. In classical mechanics, **vectors** are typically defined in Cartesian coordinates as

$$\boldsymbol{\alpha} = \alpha_x \hat{\mathbf{x}} + \alpha_y \hat{\mathbf{y}} + \alpha_z \hat{\mathbf{z}}. \quad (\text{VII-8})$$

Note that one also can use the i, j, k notation for the unit vectors.

- a) Vectors are added via the component method such that

$$\boldsymbol{\alpha} \pm \boldsymbol{\beta} = (\alpha_x \pm \beta_x) \hat{\mathbf{x}} + (\alpha_y \pm \beta_y) \hat{\mathbf{y}} + (\alpha_z \pm \beta_z) \hat{\mathbf{z}}. \quad (\text{VII-9})$$

- b) However in quantum mechanics, often we will have more than 3 coordinates to worry about — indeed, sometimes there may be an infinite number of coordinates!
- c) As such, we will introduce a new notation (the so-called **bra-and-ket** notation) to describe vectors:

$$\begin{aligned} \boldsymbol{\alpha} &\equiv |\alpha\rangle && \text{(ket),} \\ \boldsymbol{\alpha}^* &\equiv \langle\alpha| && \text{(bra).} \end{aligned} \quad (\text{VII-10})$$

Note that the $*$ in the “bra” definition means take the complex conjugate (multiply all $i = \sqrt{-1}$ terms by -1) in vector α .

2. A **vector space** consists of a set of **vectors** ($|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \dots$), together with a set of (real or complex) **scalars** (a, b, c, \dots), which are subject to 2 operations:

- a) **Vector addition:** The *sum* of any 2 vectors is another vector:

$$|\alpha\rangle + |\beta\rangle = |\gamma\rangle. \quad (\text{VII-11})$$

i) Vector addition is **commutative**:

$$|\alpha\rangle + |\beta\rangle = |\beta\rangle + |\alpha\rangle. \quad (\text{VII-12})$$

ii) Vector addition is **associative**:

$$|\alpha\rangle + (|\beta\rangle + |\gamma\rangle) = (|\alpha\rangle + |\beta\rangle) + |\gamma\rangle. \quad (\text{VII-13})$$

iii) There exists a **zero** (or **null**) **vector**, $|0\rangle$, with the property that

$$|\alpha\rangle + |0\rangle = |\alpha\rangle, \quad (\text{VII-14})$$

for every vector $|\alpha\rangle$.

iv) For every vector $|\alpha\rangle$ there is an associated **inverse vector** ($|- \alpha\rangle$) such that

$$|\alpha\rangle + |- \alpha\rangle = |0\rangle. \quad (\text{VII-15})$$

b) **Scalar multiplication**: The *product* of any scalar with any vector is another vector:

$$a|\alpha\rangle = |\gamma\rangle. \quad (\text{VII-16})$$

i) Scalar multiplication is **distributive** with respect to vector addition:

$$a(|\alpha\rangle + |\beta\rangle) = a|\alpha\rangle + a|\beta\rangle, \quad (\text{VII-17})$$

and with respect to scalar addition:

$$(a + b)|\alpha\rangle = a|\alpha\rangle + b|\alpha\rangle. \quad (\text{VII-18})$$

ii) It is also **associative**:

$$a(b|\alpha\rangle) = (ab)|\alpha\rangle. \quad (\text{VII-19})$$

iii) Multiplications by the **null** and **unit vector** are

$$0|\alpha\rangle = |0\rangle; \quad 1|\alpha\rangle = |\alpha\rangle. \quad (\text{VII-20})$$

(Note that $|- \alpha\rangle = (-1)|\alpha\rangle$.)

c) A **linear combination** of the vectors $|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \dots$ is an expression of the form

$$a|\alpha\rangle + b|\beta\rangle + c|\gamma\rangle + \dots \quad (\text{VII-21})$$

i) A vector $|\lambda\rangle$ is said to be **linearly independent** of the set $|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \dots$ if it cannot be written as a linear combination of them (*e.g.*, unit vectors $\hat{\mathbf{x}}, \hat{\mathbf{y}},$ and $\hat{\mathbf{z}}$).

ii) A collection of vectors is said to **span** the space if *every* vector can be written as a linear combination of the members of this set.

iii) A set of *linearly independent* vectors that spans the space is called a **basis** $\implies \hat{\mathbf{x}}, \hat{\mathbf{y}},$ and $\hat{\mathbf{z}}$ (or i, j, k) define the Cartesian basis, which is a 3-D orthogonal basis.

iv) The number of vectors in any basis is called the **dimension** of the space. Here we will introduce the *finite* bases (analogous to unit vectors),

$$|e_1\rangle, |e_2\rangle, \dots, |e_n\rangle, \quad (\text{VII-22})$$

of any given vector:

$$|\alpha\rangle = a_1|e_1\rangle + a_2|e_2\rangle + \dots + a_n|e_n\rangle, \quad (\text{VII-23})$$

which is uniquely represented by the (ordered) n -tuple of its **components**:

$$|\alpha\rangle \leftrightarrow (a_1, a_2, \dots, a_n). \quad (\text{VII-24})$$

- v) It is often easier to work with components than with the abstract vectors themselves. Use whatever method to which you are most comfortable.

3. In 3 dimensions, we encounter 2 kinds of vector products: the *dot product* and the *cross product*. The latter does not generalize in any natural way to n -dimensional vector spaces, but the former *does* and is called the **inner product**.

- a) The inner product of 2 vectors ($|\alpha\rangle$ and $|\beta\rangle$) is a complex number (which we write as $\langle\alpha|\beta\rangle$), with the following properties:

$$\langle\beta|\alpha\rangle = \langle\alpha|\beta\rangle^* \quad (\text{VII-25})$$

$$\langle\alpha|\alpha\rangle \geq 0 \quad (\text{VII-26})$$

$$\langle\alpha|\alpha\rangle = 0 \Leftrightarrow |\alpha\rangle = |0\rangle \quad (\text{VII-27})$$

$$\langle\alpha|(b|\beta\rangle + c|\gamma\rangle) = b\langle\alpha|\beta\rangle + c\langle\alpha|\gamma\rangle \quad (\text{VII-28})$$

$$\langle\alpha|\beta\rangle = \sum_{n=1}^N \alpha_n^* \beta_n. \quad (\text{VII-29})$$

- b) A vector space with an inner product is called an **inner product space**.
- c) Because the inner product of any vector with itself is a non-negative number (Eq. VII-26), its square root is *real* — we call this the **norm** (think of this as the *length*) of the vector:

$$\|\alpha\| \equiv \sqrt{\langle\alpha|\alpha\rangle}. \quad (\text{VII-30})$$

- d) A *unit* vector, whose norm is 1, is said to be **normalized**.
- e) Two vectors whose inner product is zero are called **orthogonal** \Rightarrow a collection of mutually orthogonal normalized vectors,

$$\langle\alpha_i|\alpha_j\rangle = \delta_{ij}, \quad (\text{VII-31})$$

is called an **orthonormal set**, where δ_{ij} is the **Kronecker delta**.

- f) Components of vectors can be written as

$$a_i = \langle e_i | \alpha \rangle. \quad (\text{VII-32})$$

- g) For vectors that are co-linear and proportional to each other, the **Schwartz inequality** can be applied to these vectors:

$$|\langle \alpha | \beta \rangle|^2 \leq \langle \alpha | \alpha \rangle \langle \beta | \beta \rangle \quad (\text{VII-33})$$

and we can define the (complex) angle between $|\alpha\rangle$ and $|\beta\rangle$ by the formula

$$\cos \theta = \sqrt{\frac{\langle \alpha | \beta \rangle \langle \beta | \alpha \rangle}{\langle \alpha | \alpha \rangle \langle \beta | \beta \rangle}}. \quad (\text{VII-34})$$

4. A **linear transformation** (\hat{T} , the *hat* on an operator from this point forward will imply that the operator is a linear transformation — don't confuse it with the *hat* of a unit vector) takes each vector in a vector space and “transforms” it into some other vector ($|\alpha\rangle \rightarrow |\alpha'\rangle = \hat{T}|\alpha\rangle$), with the proviso that the operator is *linear*

$$\hat{T}(a|\alpha\rangle + b|\beta\rangle) = a(\hat{T}|\alpha\rangle) + b(\hat{T}|\beta\rangle). \quad (\text{VII-35})$$

- a) We can write the linear transformation of basis vectors as

$$\hat{T}|e_j\rangle = \sum_{i=1}^n T_{ij}|e_i\rangle, \quad (j = 1, 2, \dots, n). \quad (\text{VII-36})$$

This is also the definition of a **tensor**, as such, the operator \hat{T} is also a tensor.

- b) If $|\alpha\rangle$ is an arbitrary vector:

$$|\alpha\rangle = a_1|e_1\rangle + \dots + a_n|e_n\rangle = \sum_{j=1}^n a_j|e_j\rangle, \quad (\text{VII-37})$$

then

$$\hat{T}|\alpha\rangle = \sum_{j=1}^n a_j(\hat{T}|e_j\rangle) = \sum_{j=1}^n \sum_{i=1}^n a_j T_{ij} |e_i\rangle = \sum_{i=1}^n \left(\sum_{j=1}^n T_{ij} a_j \right) |e_i\rangle. \quad (\text{VII-38})$$

\hat{T} takes a vector with components a_1, a_2, \dots, a_n into a vector with components

$$a'_i = \sum_{j=1}^n T_{ij} a_j. \quad (\text{VII-39})$$

- c) If the basis is orthonormal, it follows from Eq. (VII-36) that

$$T_{ij} = \langle e_i | \hat{T} | e_j \rangle, \quad (\text{VII-40})$$

or in matrix notation

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & & \vdots \\ T_{n1} & T_{n2} & \cdots & T_{nn} \end{pmatrix}. \quad (\text{VII-41})$$

- d) The sum of 2 linear transformations is

$$(\hat{S} + \hat{T})|\alpha\rangle = \hat{S}|\alpha\rangle + \hat{T}|\alpha\rangle, \quad (\text{VII-42})$$

or, again, in matrix notation,

$$\mathbf{U} = \mathbf{S} + \mathbf{T} \Leftrightarrow U_{ij} = S_{ij} + T_{ij}. \quad (\text{VII-43})$$

- e) The *product* of 2 linear transformations ($\hat{S}\hat{T}$) is the net effect of performing them in succession — first \hat{T} , the \hat{S} . In matrix notation:

$$\mathbf{U} = \mathbf{S}\mathbf{T} \Leftrightarrow U_{ik} = \sum_{j=1}^n S_{ij} T_{jk}; \quad (\text{VII-44})$$

this is the standard rule for matrix multiplication — to find the ik^{th} element of the product, you look at the i^{th} row of \mathbf{S} and the k^{th} column of \mathbf{T} , multiply corresponding entries, and add.

- f) The **transpose** of a matrix ($\tilde{\mathbf{T}}$) is the same set of elements in \mathbf{T} , but with the rows and columns interchanged:

$$\tilde{\mathbf{T}} = \begin{pmatrix} T_{11} & T_{21} & \cdots & T_{n1} \\ T_{12} & T_{22} & \cdots & T_{n2} \\ \vdots & \vdots & & \vdots \\ T_{1n} & T_{2n} & \cdots & T_{nn} \end{pmatrix}. \quad (\text{VII-45})$$

Note that the transpose of a column matrix is a row matrix!

- g) A square matrix is **symmetric** if it is equal to its transpose (reflection in the main diagonal — upper left to lower right — leaves it unchanged); it is **antisymmetric** if this operation reverses the sign:

$$\text{SYMMETRIC: } \tilde{\mathbf{T}} = \mathbf{T}; \quad \text{ANTISYMMETRIC: } \tilde{\mathbf{T}} = -\mathbf{T}. \quad (\text{VII-46})$$

- h) The (complex) **conjugate** (\mathbf{T}^*) is obtained by taking the complex conjugate of every element:

$$\mathbf{T}^* = \begin{pmatrix} T_{11}^* & T_{12}^* & \cdots & T_{1n}^* \\ T_{21}^* & T_{22}^* & \cdots & T_{2n}^* \\ \vdots & \vdots & & \vdots \\ T_{n1}^* & T_{n2}^* & \cdots & T_{nn}^* \end{pmatrix}; \quad \mathbf{a}^* = \begin{pmatrix} a_1^* \\ a_2^* \\ \vdots \\ a_n^* \end{pmatrix}. \quad (\text{VII-47})$$

- i) A matrix is **real** if all its elements are real and **imaginary** if they are all imaginary:

$$\text{REAL: } \mathbf{T}^* = \mathbf{T}; \quad \text{IMAGINARY: } \mathbf{T}^* = -\mathbf{T}. \quad (\text{VII-48})$$

- j) A square matrix is **Hermitian** (or **self-adjoint** as defined by $\mathbf{T}^\dagger \equiv \tilde{\mathbf{T}}^*$) if it is equal to its Hermitian conjugate; if Hermitian conjugation introduces a minus sign, the matrix is **skew Hermitian** (or **anti-Hermitian**):

$$\text{HERMITIAN: } \mathbf{T}^\dagger = \mathbf{T}; \quad \text{SKEW HERMITIAN: } \mathbf{T}^\dagger = -\mathbf{T}. \quad (\text{VII-49})$$

- k) With this notation, the inner product of 2 vectors (with respect to an orthonormal basis), can be written in matrix form:

$$\langle \alpha | \beta \rangle = \mathbf{a}^\dagger \mathbf{b}. \quad (\text{VII-50})$$

- l) Matrix multiplication is not, in general, commutative ($\mathbf{ST} \neq \mathbf{TS}$) — the difference between 2 orderings is called the **commutator**:

$$[\mathbf{S}, \mathbf{T}] \equiv \mathbf{ST} - \mathbf{TS}. \quad (\text{VII-51})$$

It can also be shown that one can write the following commutator relation:

$$[\hat{A}\hat{B}, \hat{C}] = \hat{A}[\hat{B}, \hat{C}] + [\hat{A}, \hat{C}]\hat{B}. \quad (\text{VII-52})$$

- m) The transpose of a product is the product of the transpose *in reverse order*:

$$(\tilde{\mathbf{ST}}) = \tilde{\mathbf{T}}\tilde{\mathbf{S}}, \quad (\text{VII-53})$$

and the same goes for Hermitian conjugates:

$$(\mathbf{ST})^\dagger = \mathbf{T}^\dagger \mathbf{S}^\dagger. \quad (\text{VII-54})$$

- n) The **unit matrix** is defined by

$$\mathbf{1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \quad (\text{VII-55})$$

In other words,

$$\mathbf{1}_{ij} = \delta_{ij}. \quad (\text{VII-56})$$

- o) The **inverse** of a matrix (written \mathbf{T}^{-1}) is defined by

$$\mathbf{T}^{-1}\mathbf{T} = \mathbf{T}\mathbf{T}^{-1} = \mathbf{1}. \quad (\text{VII-57})$$

- i) A matrix has an inverse if and only if its **determinant** is nonzero; in fact

$$\mathbf{T}^{-1} = \frac{1}{\det \mathbf{T}} \tilde{\mathbf{C}}, \quad (\text{VII-58})$$

where \mathbf{C} is the matrix of **cofactors**.

- ii) The cofactor of element T_{ij} is $(-1)^{i+j}$ times the determinant of the submatrix obtained from \mathbf{T} by erasing the i^{th} row by the j^{th} column.

- iii) As an example for taking the inverse of a matrix, let's assume that \mathbf{T} is a 3x3 matrix of form

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix}. \quad (\text{VII-59})$$

Its determinant is then

$$\begin{aligned} \det \mathbf{T} &= |\mathbf{T}| = \begin{vmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{vmatrix} \\ &= T_{11} \begin{vmatrix} T_{22} & T_{23} \\ T_{32} & T_{33} \end{vmatrix} - T_{12} \begin{vmatrix} T_{21} & T_{23} \\ T_{31} & T_{33} \end{vmatrix} \\ &\quad + T_{13} \begin{vmatrix} T_{21} & T_{22} \\ T_{31} & T_{32} \end{vmatrix} \\ &= T_{11} (T_{22}T_{33} - T_{23}T_{32}) - T_{12} (T_{21}T_{33} - T_{23}T_{31}) \\ &\quad + T_{13} (T_{21}T_{32} - T_{22}T_{31}) . \end{aligned} \quad (\text{VII-60})$$

- iv) For this 3x3 matrix, the matrix of cofactors is given by

$$\mathbf{C} = \begin{pmatrix} \begin{vmatrix} T_{22} & T_{23} \\ T_{32} & T_{33} \end{vmatrix} & -\begin{vmatrix} T_{21} & T_{23} \\ T_{31} & T_{33} \end{vmatrix} & \begin{vmatrix} T_{21} & T_{22} \\ T_{31} & T_{32} \end{vmatrix} \\ -\begin{vmatrix} T_{12} & T_{13} \\ T_{32} & T_{33} \end{vmatrix} & \begin{vmatrix} T_{11} & T_{13} \\ T_{31} & T_{33} \end{vmatrix} & -\begin{vmatrix} T_{11} & T_{12} \\ T_{31} & T_{32} \end{vmatrix} \\ \begin{vmatrix} T_{12} & T_{13} \\ T_{22} & T_{23} \end{vmatrix} & -\begin{vmatrix} T_{11} & T_{13} \\ T_{21} & T_{23} \end{vmatrix} & \begin{vmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{vmatrix} \end{pmatrix}. \quad (\text{VII-61})$$

- v) The transpose of this cofactor matrix is then (see Eq. VII-45)

$$\tilde{\mathbf{C}} = \begin{pmatrix} \begin{vmatrix} T_{22} & T_{32} \\ T_{23} & T_{33} \end{vmatrix} & -\begin{vmatrix} T_{12} & T_{32} \\ T_{13} & T_{33} \end{vmatrix} & \begin{vmatrix} T_{12} & T_{22} \\ T_{13} & T_{23} \end{vmatrix} \\ -\begin{vmatrix} T_{21} & T_{31} \\ T_{23} & T_{33} \end{vmatrix} & \begin{vmatrix} T_{11} & T_{31} \\ T_{13} & T_{33} \end{vmatrix} & -\begin{vmatrix} T_{11} & T_{21} \\ T_{13} & T_{23} \end{vmatrix} \\ \begin{vmatrix} T_{21} & T_{31} \\ T_{22} & T_{32} \end{vmatrix} & -\begin{vmatrix} T_{11} & T_{31} \\ T_{12} & T_{32} \end{vmatrix} & \begin{vmatrix} T_{11} & T_{21} \\ T_{12} & T_{22} \end{vmatrix} \end{pmatrix}. \quad (\text{VII-62})$$

- vi) A matrix without an inverse is said to be **singular**.

- vii) The inverse of a product (assuming it exists) is the product of the inverses *in reverse order*:

$$(\mathbf{ST})^{-1} = \mathbf{T}^{-1}\mathbf{S}^{-1}. \quad (\text{VII-63})$$

- p) A matrix is **unitary** if its inverse is equal to its Hermitian conjugate:

$$\text{UNITARY: } \mathbf{U}^\dagger = \mathbf{U}^{-1}. \quad (\text{VII-64})$$

- q) The **trace** of a matrix is the sum of the diagonal elements:

$$\text{Tr}(\mathbf{T}) \equiv \sum_{i=1}^m T_{ii}, \quad (\text{VII-65})$$

and has the property

$$\text{Tr}(\mathbf{T}_1\mathbf{T}_2) = \text{Tr}(\mathbf{T}_2\mathbf{T}_1). \quad (\text{VII-66})$$

5. A vector under a linear transformation that obeys the following equation:

$$\hat{T}|\alpha\rangle = \lambda|\alpha\rangle, \quad (\text{VII-67})$$

is called an **eigenvector** of the transformation, and the (complex) number λ is called the **eigenvalue**.

- a) Notice that any (nonzero) multiple of an eigenvector is still an eigenvector with the same eigenvalue.
- b) In matrix form, the eigenvector equation takes the form:

$$\mathbf{T}\mathbf{a} = \lambda\mathbf{a} \quad (\text{VII-68})$$

(for nonzero \mathbf{a}), or

$$(\mathbf{T} - \lambda\mathbf{1})\mathbf{a} = \mathbf{0}. \quad (\text{VII-69})$$

(here $\mathbf{0}$ is the **zero matrix**, whose elements are all zero.)

- c) If the matrix $(\mathbf{T} - \lambda\mathbf{1})$ had an *inverse*, we could multiply both sides of Eq. (VII-69) by $(\mathbf{T} - \lambda\mathbf{1})^{-1}$, and conclude that $\mathbf{a} = \mathbf{0}$. But by assumption, \mathbf{a} is *not* zero, so the matrix $(\mathbf{T} - \lambda\mathbf{1})$ must in fact be singular, which means that its determinant vanishes:

$$\det(\mathbf{T} - \lambda\mathbf{1}) = \begin{vmatrix} (T_{11} - \lambda) & T_{12} & \cdots & T_{1n} \\ T_{21} & (T_{22} - \lambda) & \cdots & T_{2n} \\ \vdots & \vdots & & \vdots \\ T_{n1} & T_{n2} & \cdots & (T_{nn} - \lambda) \end{vmatrix} = 0. \quad (\text{VII-70})$$

- d) Expansion of the determinant yields an algebraic equation for λ :

$$C_n\lambda^n + C_{n-1}\lambda^{n-1} + \cdots + C_1\lambda + C_0 = 0, \quad (\text{VII-71})$$

where the coefficients C_i depend on the elements of \mathbf{T} . This is called the **characteristic equation** for the matrix — its solutions determine the eigenvalues. Note that it is an n^{th} -order equation, so it has n (complex) roots.

- i) Some of these root may be duplicates, so all we can say for certain is that an $n \times n$ matrix has *at least one* and *at most* n distinct eigenvalues.
- ii) In the cases where duplicates exist, such states are said to be **degenerate**.
- iii) To construct the corresponding eigenvectors, it is generally easiest simply to plug each λ back into Eq. (VII-68) and solve (by hand) for the components of \mathbf{a} (see Examples VII-2 and VII-3).

6. In many physical problems involving matrices in both classical mechanics and quantum mechanics it is desirable to carry out a (real) orthogonal similarity transformation or a unitary transformation to reduce the matrix to its diagonal form (*i.e.*, all non-diagonal elements equal to zero).

- a) If eigenvectors span the space, we are free to use them as a basis

$$\begin{aligned}
 \hat{T}|f_1\rangle &= \lambda_1|f_1\rangle \\
 \hat{T}|f_2\rangle &= \lambda_2|f_2\rangle \\
 &\dots \\
 \hat{T}|f_n\rangle &= \lambda_n|f_n\rangle
 \end{aligned}$$

- b) The matrix representing \hat{T} takes on a very simple form in this basis, with the eigenvalues strung out along the main

diagonal and all other elements zero:

$$\mathbf{T} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}. \quad (\text{VII-72})$$

c) The (normalized) eigenvectors are equally simple:

$$\mathbf{a}^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{a}^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \dots, \mathbf{a}^{(n)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}. \quad (\text{VII-73})$$

- d) A matrix that can be brought to **diagonal form** (Eq. VII-72) by change of basis is said to be **diagonalizable**.
- e) In a geometrical sense, diagonalizing a matrix is equivalent to rotating the bases of a matrix about some point in the space until all of the off-diagonal elements go to zero. If \mathbf{D} is the diagonalized matrix of matrix \mathbf{M} , the operation that diagonalizes \mathbf{M} is

$$\mathbf{D} = \mathbf{S}\mathbf{M}\mathbf{S}^{-1}, \quad (\text{VII-74})$$

where matrix \mathbf{S} is called a similarity transformation. Note that the inverse of the **similarity matrix** can be constructed by using the eigenvectors (in the old basis) as the columns of \mathbf{S}^{-1} :

$$(\mathbf{S}^{-1})_{ij} = (\mathbf{a}^{(j)})_i. \quad (\text{VII-75})$$

- f) There is great advantage in bringing a matrix to diagonal form — it is much easier to work with. Unfortunately,

not every matrix can be diagonalized — **the eigenvectors have to span the space for a matrix to be diagonalizable.**

7. The Hermitian conjugate of a linear transformation (called a **Hermitian transformation**) is that transformation \hat{T}^\dagger which, when applied to the *first* member of an inner product, gives the same result as if \hat{T} itself had been applied to the *second* vector:

$$\langle \hat{T}^\dagger \alpha | \beta \rangle = \langle \alpha | \hat{T} \beta \rangle \quad (\text{VII-76})$$

(for all vectors $|\alpha\rangle$ and $|\beta\rangle$).

- a) Note that the notation used in Eq. (VII-76) is commonly used but incorrect: $\hat{T}|\beta\rangle$ actually means $\hat{T}|\beta\rangle$ and $\langle \hat{T}^\dagger \alpha | \beta \rangle$ means the inner product of the vector $\hat{T}^\dagger |\alpha\rangle$.
- b) Note that we can also write

$$\langle \alpha | \hat{T} \beta \rangle = \mathbf{a}^\dagger \mathbf{T} \mathbf{b} = (\mathbf{T}^\dagger \mathbf{a})^\dagger \mathbf{b} = \langle \hat{T}^\dagger \alpha | \beta \rangle. \quad (\text{VII-77})$$

- c) In quantum mechanics, a fundamental role is played by Hermitian transformations ($\hat{T}^\dagger = \hat{T}$). The eigenvectors and eigenvalues of a Hermitian transformation have 3 crucial properties:
- i) **The eigenvalues of a Hermitian transformation are real.**
 - ii) **The eigenvectors of a Hermitian transformation belonging to distinct eigenvalues are orthogonal.**
 - iii) **The eigenvectors of a Hermitian transformation span the space.**

Example VII-1. Given the following two matrices:

$$\mathbf{A} = \begin{pmatrix} -1 & 1 & i \\ 2 & 0 & 3 \\ 2i & -2i & 2 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 2 & 0 & -i \\ 0 & 1 & 0 \\ i & 3 & 2 \end{pmatrix},$$

compute (a) $\mathbf{A} + \mathbf{B}$, (b) \mathbf{AB} , (c) $[\mathbf{A}, \mathbf{B}]$, (d) $\tilde{\mathbf{A}}$, (e) \mathbf{A}^* , (f) \mathbf{A}^\dagger , (g) $\text{Tr}(\mathbf{B})$, (h) $\det(\mathbf{B})$, and (i) \mathbf{B}^{-1} . Check that $\mathbf{BB}^{-1} = \mathbf{1}$. Does \mathbf{A} have an inverse?

Solution (a): Sum the respective elements of the matrix:

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} -1 & 1 & i \\ 2 & 0 & 3 \\ 2i & -2i & 2 \end{pmatrix} + \begin{pmatrix} 2 & 0 & -i \\ 0 & 1 & 0 \\ i & 3 & 2 \end{pmatrix} = \boxed{\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 3 \\ 3i & (3 - 2i) & 4 \end{pmatrix}}.$$

Solution (b): Multiply rows of \mathbf{A} by columns of \mathbf{B} :

$$\begin{aligned} \mathbf{AB} &= \begin{pmatrix} (-2 + 0 - 1) & (0 + 1 + 3i) & (i + 0 + 2i) \\ (4 + 0 + 3i) & (0 + 0 + 9) & (-2i + 0 + 6) \\ (4i + 0 + 2i) & (0 - 2i + 6) & (2 + 0 + 4) \end{pmatrix} \\ &= \boxed{\begin{pmatrix} -3 & (1 + 3i) & 3i \\ (4 + 3i) & 9 & (6 - 2i) \\ 6i & (6 - 2i) & 6 \end{pmatrix}}. \end{aligned}$$

Solution (c): $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$, we already have \mathbf{AB} ,

$$\begin{aligned} \mathbf{BA} &= \begin{pmatrix} (-2 + 0 + 2) & (2 + 0 - 2) & (2i + 0 - 2i) \\ (0 + 2 + 0) & (0 + 0 + 0) & (0 + 3 + 0) \\ (-i + 6 + 4i) & (i + 0 - 4i) & (-1 + 9 + 4) \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 3 \\ (6 + 3i) & -3i & 12 \end{pmatrix}; \\ [\mathbf{A}, \mathbf{B}] &= \begin{pmatrix} -3 & (1 + 3i) & 3i \\ (4 + 3i) & 9 & (6 - 2i) \\ 6i & (6 - 2i) & 6 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 3 \\ (6 + 3i) & -3i & 12 \end{pmatrix} \end{aligned}$$

$$= \begin{bmatrix} \begin{pmatrix} -3 & (1+3i) & 3i \\ (2+3i) & 9 & (3-2i) \\ (-6+3i) & (6+i) & -6 \end{pmatrix} \end{bmatrix}.$$

Solution (d): Transpose of \mathbf{A} — flip \mathbf{A} about the diagonal:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \begin{pmatrix} -1 & 2 & 2i \\ 1 & 0 & -2i \\ i & 3 & 2 \end{pmatrix} \end{bmatrix}.$$

Solution (e): Complex conjugate of \mathbf{A} — multiply each i term by -1 in \mathbf{A} :

$$\mathbf{A}^* = \begin{bmatrix} \begin{pmatrix} -1 & 1 & -i \\ 2 & 0 & 3 \\ -2i & 2i & 2 \end{pmatrix} \end{bmatrix}.$$

Solution (f): Hermitian of \mathbf{A} :

$$\mathbf{A}^\dagger \equiv \tilde{\mathbf{A}}^* = \begin{bmatrix} \begin{pmatrix} -1 & 2 & -2i \\ 1 & 0 & 2i \\ -i & 3 & 2 \end{pmatrix} \end{bmatrix}.$$

Solution (g): Trace of \mathbf{B} :

$$\text{Tr}(\mathbf{B}) = \sum_{i=1}^3 B_{ii} = 2 + 1 + 2 = \boxed{5}.$$

Solution (h): Determinant of \mathbf{B} :

$$\det(\mathbf{B}) = 2(2 - 0) - 0(0 - 0) - i(0 - i) = 4 - 0 - 1 = \boxed{3}.$$

Solution (i): Inverse of \mathbf{B} :

$$\mathbf{B}^{-1} = \frac{1}{\det(\mathbf{B})} \tilde{\mathbf{C}},$$

where

$$\mathbf{C} = \begin{pmatrix} \begin{vmatrix} 1 & 0 \\ 3 & 2 \end{vmatrix} & -\begin{vmatrix} 0 & 0 \\ i & 2 \end{vmatrix} & \begin{vmatrix} 0 & 1 \\ i & 3 \end{vmatrix} \\ -\begin{vmatrix} 0 & -i \\ 3 & 2 \end{vmatrix} & \begin{vmatrix} 2 & -i \\ i & 2 \end{vmatrix} & -\begin{vmatrix} 2 & 0 \\ i & 3 \end{vmatrix} \\ \begin{vmatrix} 0 & -i \\ 1 & 0 \end{vmatrix} & -\begin{vmatrix} 2 & -i \\ 0 & 0 \end{vmatrix} & \begin{vmatrix} 2 & 0 \\ 0 & 1 \end{vmatrix} \end{pmatrix} = \begin{pmatrix} 2 & 0 & -i \\ -3i & 3 & -6 \\ i & 0 & 2 \end{pmatrix},$$

then

$$\boxed{\mathbf{B}^{-1} = \frac{1}{3} \begin{pmatrix} 2 & -3i & i \\ 0 & 3 & 0 \\ -i & -6 & 2 \end{pmatrix}}.$$

$$\begin{aligned} \mathbf{B}\mathbf{B}^{-1} &= \frac{1}{3} \begin{pmatrix} (4+0-1) & (-6i+0+6i) & (2i+0-2i) \\ (0+0+0) & (0+3+0) & (0+0+0) \\ (2i+0-2i) & (3+9-12) & (-1+0+4) \end{pmatrix} \\ &= \frac{1}{3} \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad \checkmark \end{aligned}$$

If $\det(\mathbf{A}) \neq 0$, then \mathbf{A} has an inverse:

$$\det(\mathbf{A}) = -1(0+6i) - 1(4-6i) + i(-4i-0) = -6i-4+6i+4 = 0.$$

As such, $\boxed{\mathbf{A} \text{ does \underline{not} have an inverse.}}$

Example VII-2. Find the eigenvalues and normalized eigenvectors of the following matrix:

$$\mathbf{M} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Can this matrix be diagonalized?

Solution:

$$\begin{aligned} 0 = \det(\mathbf{M} - \lambda \mathbf{1}) &= \begin{vmatrix} (1-\lambda) & 1 \\ 0 & (1-\lambda) \end{vmatrix} \\ &= (1-\lambda)^2 \end{aligned}$$

$$\boxed{\lambda = 1} \quad (\text{only one eigenvalue}).$$

From Eq. (VII-68) we get

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = 1 \cdot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} .$$

We get two equations from this eigenvector equation:

$$\begin{aligned} a_1 + a_2 &= a_1 \\ a_2 &= a_2 . \end{aligned}$$

The second equation tells us nothing, but the first equation shows us that $a_2 = 0$. We still need to figure out the value for a_1 . We can do this by normalizing our eigenvector $\mathbf{a} = |\alpha\rangle$:

$$\begin{aligned} 1 &= \langle \alpha | \alpha \rangle = \sum_{i=1}^2 |a_i|^2 \\ &= |a_1|^2 + |a_2|^2 = |a_1|^2 \end{aligned}$$

or $a_1 = 1$. Hence our normalized eigenvector,

$$|\alpha\rangle = \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} ,$$

is

$$\boxed{\mathbf{a} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} .}$$

Since these eigenvectors do not span the space (as described on page VII-4, §A.2.c.ii.), this matrix cannot be diagonalized.

Example VII-3. Find the eigenvalues and eigenvectors of the following matrix:

$$\mathbf{M} = \begin{pmatrix} 2 & 0 & -2 \\ -2i & i & 2i \\ 1 & 0 & -1 \end{pmatrix} .$$

Solution:

The characteristic equation is

$$\begin{aligned}
 |\mathbf{M} - \lambda \mathbf{1}| &= \begin{vmatrix} (2 - \lambda) & 0 & -2 \\ -2i & (i - \lambda) & 2i \\ 1 & 0 & (-1 - \lambda) \end{vmatrix} \\
 &= (2 - \lambda) \begin{vmatrix} (i - \lambda) & 2i \\ 0 & (-1 - \lambda) \end{vmatrix} - 0 - 2 \begin{vmatrix} -2i & (i - \lambda) \\ 1 & 0 \end{vmatrix} \\
 &= (2 - \lambda)[(i - \lambda)(-1 - \lambda) - 0] - 2[0 - (i - \lambda)] \\
 &= (2 - \lambda)(-i - i\lambda + \lambda + \lambda^2) + 2i - 2\lambda \\
 &= -2i - 2i\lambda + 2\lambda + 2\lambda^2 + i\lambda + i\lambda^2 - \lambda^2 - \lambda^3 + 2i - 2\lambda \\
 &= -\lambda^3 + (1 + i)\lambda^2 - i\lambda = 0 .
 \end{aligned}$$

To find the roots to this characteristic equation, factor out a λ and use the quadratic formula solution equation:

$$\begin{aligned}
 0 &= -\lambda^3 + (1 + i)\lambda - i\lambda \\
 &= [-\lambda^2 + (1 + i)\lambda - i]\lambda \\
 \lambda_1 &= 0 \\
 \lambda_{2,3} &= \frac{-(1 + i) \pm \sqrt{(1 + i)^2 - 4i}}{-2} \\
 &= \frac{-(1 + i) \pm \sqrt{(1 + 2i - 1) - 4i}}{-2} \\
 &= \frac{-(1 + i) \pm \sqrt{-2i}}{-2} .
 \end{aligned}$$

However note that $(1 - i)^2 = -2i$. As such, the equation above becomes

$$\begin{aligned}
 \lambda_{2,3} &= \frac{-(1 + i) \pm \sqrt{(1 - i)^2}}{-2} \\
 &= \frac{-(1 + i) \pm (1 - i)}{-2} \\
 \lambda_2 &= \frac{-(1 + i) - (1 - i)}{-2} = \frac{-2}{-2} = 1 \\
 \lambda_3 &= \frac{-(1 + i) + (1 - i)}{-2} = \frac{-2i}{-2} = i ,
 \end{aligned}$$

so the roots of λ (*i.e.*, the eigenvalues) are 0, 1, and i . Now, let's call the components of the first eigenvector $|\alpha\rangle$ (a_1, a_2, a_3) which corresponds to eigenvalue $\lambda_1 = 0$. The eigenvector equation becomes

$$\begin{pmatrix} 2 & 0 & -2 \\ -2i & i & 2i \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = 0 \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

which yield 3 equations:

$$\begin{aligned} 2a_1 - 2a_3 &= 0 \\ -2ia_1 + ia_2 + 2ia_3 &= 0 \\ a_1 - a_3 &= 0. \end{aligned}$$

The first equation gives $a_3 = a_1$, the second gives $a_2 = 0$, and the third is redundant with the first equation. We can find the values for a_1 and a_3 by normalizing:

$$\begin{aligned} 1 &= \langle \alpha | \alpha \rangle = \sum_{i=1}^3 |a_i|^2 \\ &= |a_1|^2 + |a_2|^2 + |a_3|^2 = |a_1|^2 + |a_1|^2 \\ &= 2|a_1|^2, \end{aligned}$$

or $a_1 = a_3 = (1/\sqrt{2}) = \sqrt{2}/2$. Hence our eigenvector for λ_1 is

$$|\alpha\rangle = \mathbf{a} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \text{ for } \lambda_1 = 0.$$

For the second eigenvector, let's call it $|\beta\rangle = \mathbf{b}$, we have

$$\begin{pmatrix} 2 & 0 & -2 \\ -2i & i & 2i \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = 1 \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

which yield the equations:

$$\begin{aligned} 2b_1 - 2b_3 &= b_1 \\ -2ib_1 + ib_2 + 2ib_3 &= b_2 \\ b_1 - b_3 &= b_3, \end{aligned}$$

with the solutions $b_3 = (1/2)b_1$ and $b_2 = [(1 - i)/2]b_1$. Normalizing gives

$$\begin{aligned}
 1 &= \langle \beta | \beta \rangle = \sum_{i=1}^3 |b_i|^2 \\
 &= |b_1|^2 + |b_2|^2 + |b_3|^2 \\
 &= |b_1|^2 + \left(\frac{1+i}{2}\right) \left(\frac{1-i}{2}\right) |b_1|^2 + \frac{1}{4} |b_1|^2 \\
 &= |b_1|^2 + \left(\frac{1+i-i+1}{4}\right) |b_1|^2 + \frac{1}{4} |b_1|^2 \\
 &= \frac{4}{4} |b_1|^2 + \frac{2}{4} |b_1|^2 + \frac{1}{4} |b_1|^2 \\
 &= \frac{7}{4} |b_1|^2 ,
 \end{aligned}$$

or $b_1 = (2/\sqrt{7})$. So $b_2 = [(1 - i)/\sqrt{7}]$ and $b_3 = (1/\sqrt{7})$ giving our final eigenvector for λ_2 as

$$|\beta\rangle = \mathbf{b} = \frac{\sqrt{7}}{7} \begin{pmatrix} 2 \\ (1-i) \\ 1 \end{pmatrix}, \text{ for } \lambda_2 = 1 .$$

Finally, the third eigenvector (call it $|\gamma\rangle = \mathbf{c}$) is

$$\begin{pmatrix} 2 & 0 & -2 \\ -2i & i & 2i \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = i \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} ic_1 \\ ic_2 \\ ic_3 \end{pmatrix} ,$$

which gives the equations:

$$\begin{aligned}
 2c_1 - 2c_3 &= ic_1 \\
 -2ic_1 + ic_2 + 2ic_3 &= ic_2 \\
 c_1 - c_3 &= ic_3 ,
 \end{aligned}$$

with the solutions $c_3 = c_1 = 0$, with c_2 undetermined. Once again, we can normalize our eigenvector to determine this undetermined c_2 coefficient:

$$\begin{aligned}
 1 &= \langle \gamma | \gamma \rangle = \sum_{i=1}^3 |c_i|^2 \\
 &= |c_1|^2 + |c_2|^2 + |c_3|^2 = |c_2|^2 ,
 \end{aligned}$$

or $c_2 = 1$, which gives our third eigenvector:

$$|\gamma\rangle = \mathbf{c} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \text{ for } \lambda_3 = i .$$

C. Computational Linear Algebra.

1. Here, we will be solving the set of equations as described in Eqs. (VII-1,2,3).
 - a) However for the present, let N = number of unknowns and M = number of equations.
 - b) If $N = M$, there's a good chance we can obtain a unique solution.
 - i) However, if one or more of the M equations is a linear combination of the others \implies **row degeneracy** occurs \implies we will not be able to obtain a unique solution.
 - ii) Or, if all equations contain variables in exactly the same linear combination \implies **column degeneracy** occurs \implies no unique solution can be found.
 - iii) For square ($N = M$) matrices, row degeneracy implies column degeneracy and vice versa.
 - iv) If a set of equations are degenerate, the matrix is said to be **singular**.

- c) Numerically, other things can go wrong:
 - i) If some equations are close to being a linear combination of the other equations in the set, roundoff errors may render them linear dependent at some stage of the calculations.
 - ii) Accumulated roundoff errors in the solution can swamp the true solution \implies this can occur if N is too large \implies **direct substitution of the solution back into the original equations can verify this.**
- d) Linear sets with $2 < N < \sim 50$ can be routinely solved in single precision without resorting to sophisticated methods \implies in double precision, $N \rightarrow 200$ without worrying about roundoff error.
- e) As we have discussed, solution to a set of linear equations involve inverting matrices. To write the most efficient matrix inverter, one needs to know how numbers are stored. Assuming we have matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{pmatrix} .$$

- i) **Column storage** (which IDL calls “row-major storage”):

$$a_{11}, a_{21}, \dots, a_{M1}, a_{12}, a_{22}, \dots, a_{M2}, \dots, a_{1N}, a_{2N}, \dots, a_{MN} .$$

\implies Fortran and IDL use this method.

- ii) **Row storage** (which IDL calls “column-major storage”):

$$a_{11}, a_{12}, \dots, a_{1N}, a_{21}, a_{22}, \dots, a_{2N}, \dots, a_{M1}, a_{M2}, \dots, a_{MN} .$$

\implies C and C++ use this method.

- iii) The techniques we will be discussing here are designed with column storage in mind.

2. The basic process of solving *linear* systems of equations is to eliminate variables until you have a single equation with a single unknown.
3. For *nonlinear* problems, an iterative scheme is developed that solves a linearized version of the equations.
4. Equations that do not depend upon time are called **autonomous** systems, that is

$$\mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}) . \quad (\text{VII-78})$$

- a) If initial conditions are of the form $x_i(t) = x_i(0)$ for all i ($1 \leq i \leq N$) and t , the solution points in the N -dimensional space of the variables are called **steady state**.
- b) If we start at steady state, we stay there forever.
- c) Locating steady states for linear equations (or ODE's) is important since they are used in stability analysis problems.
- d) It is easy to see that $\mathbf{x}^* = [x_1^*, \dots, x_N^*]$ is a steady state if and only if

$$\mathbf{f}(\mathbf{x}^*) = 0 , \quad (\text{VII-79})$$

or

$$f_i(x_1^*, \dots, x_N^*) = 0, \quad \text{for all } i, \quad (\text{VII-80})$$

since this implies that $d\mathbf{x}^*/dt = 0$.

e) Hence, locating steady states reduces to the problem of solving N equations in the N unknowns x_i^* .

f) This problem is also called “finding roots of $f(x)$.”

5. We shall now discuss the various numerical techniques used in solving sets of linear equations.

D. Gaussian Elimination

1. The problem of solving $f_i(\{x_j\}) = 0$ is divided into two important classes:

- a) Techniques used for linear equations such as **Gaussian elimination** and **matrix inversion**.
- b) Techniques used for nonlinear equations such as **Newton’s Method**.
- c) With Gaussian elimination, we set up the N linear equations with N unknowns in the form of Eqs. (VII-1,2,3):

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1N}x_N & - & b_1 & = & 0 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2N}x_N & - & b_2 & = & 0 \\ \vdots & & \vdots & & & & \vdots & & \vdots & & \vdots \\ a_{N1}x_1 & + & a_{N2}x_2 & + & \cdots & + & a_{NN}x_N & - & b_N & = & 0 \end{array} \quad (\text{VII-81})$$

or in matrix form

$$\mathbf{A} \mathbf{x} - \mathbf{b} = 0, \quad (\text{VII-82})$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}; \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix}.$$

(VII-83)

- d) One can then go through a process of eliminating variables by adding or subtracting one equation to or from the other equations.

Example VII-4. Take the equations

$$\begin{aligned} 2x_1 + x_2 &= 4 \\ 4x_1 - x_2 &= 2. \end{aligned}$$

We want to eliminate x_1 from the second equation — we multiply the first equation by 2 and subtract the first equation from the second equation giving

$$-3x_2 = -6, \quad \text{or} \quad x_2 = 2.$$

This step is known as *forward elimination*. For larger sets of equations, the forward elimination procedure eliminates x_1 from the second equation, then eliminates x_1 and x_2 from the third equation, and so on. The last equation will only contain the variable x_N , which can then be solved. We then carry out a *back-substitution*, x_N is then plugged back into the $N - 1$ equation to solve for x_{N-1} . In the example above, substitute $x_2 = 2$ into the first equation giving

$$2x_1 + 2 = 4 \quad \text{or} \quad x_1 = 1.$$

- e) This method of solving systems of linear equations is called **Gaussian elimination**. A portion of a Fortran 77 code

that might perform such a Gaussian elimination would be written as

* Forward elimination

```
DO K = 1, N-1      % Go to column (k) operate
  DO I = K+1, N    % on the rows (i) below column k.
    COEFF = A(I,K) / A(K,K)
    DO J = K+1, N
      A(I,J) = A(I,J) - COEFF * A(K,J)
    ENDDO
    A(I,K) = COEFF
    B(I) = B(I) - COEFF * B(K)
  ENDDO
ENDDO
```

Then the back-substitution is performed via

* Back-substitution

```
X(N) = B(N) / A(N,N)  % Start from bottom and work
DO I = N-1, 1, -1     % work upward. (Note: This loop
  SUM = B(I)          % goes from n-1 to 1 in steps of -1.)
  DO J = I+1, N       % Skip lower triangular part.
    SUM = SUM - A(I,J)*X(J)
  ENDDO
  X(I) = SUM/A(I,I)
ENDDO
```

- f) Gaussian elimination is a simple procedure, yet it has its pitfalls. Consider the set of equations

$$\begin{aligned}\varepsilon x_1 + x_2 + x_3 &= 5 \\ x_1 + x_2 &= 3 \\ x_1 + x_3 &= 4\end{aligned}$$

In the limit $\varepsilon \rightarrow 0$, the solution is $x_1 = 1, x_2 = 2, x_3 = 3$. For these equations, the forward elimination step would start by multiplying the first equation by $(1/\varepsilon)$ and sub-

tracting it from the second and third equations, giving

$$\begin{array}{rcccccccl} \varepsilon x_1 & + & x_2 & & + & x_3 & & = & 5 \\ & + & (1 - 1/\varepsilon)x_2 & & - & (1/\varepsilon)x_3 & & = & 3 - 5/\varepsilon \\ & & -(1/\varepsilon)x_2 & & + & (1 - 1/\varepsilon)x_3 & & = & 4 - 5/\varepsilon \end{array}$$

i) Of course, if $\varepsilon = 0$ we have big problems, since the $(1/\varepsilon)$ factors blow up.

ii) Even if $\varepsilon \neq 0$, but is small, we are going to have serious roundoff problems. In this case, $1/\varepsilon \gg 1$, so the equations above become

$$\begin{array}{rcccccccl} \varepsilon x_1 & + & x_2 & & + & x_3 & & = & 5 \\ & & -(1/\varepsilon)x_2 & & - & (1/\varepsilon)x_3 & & = & -5/\varepsilon \\ & & -(1/\varepsilon)x_2 & & - & (1/\varepsilon)x_3 & & = & -5/\varepsilon \end{array}$$

At this point it is clear that we may not proceed since the second and third equations are now identical \implies 3 unknowns with only 2 equations.

g) Fortunately, there is a simple fix; we can just interchange the order of the equations before doing the forward elimination:

$$\begin{array}{rcccccccl} x_1 & + & x_2 & & & & & = & 3 \\ \varepsilon x_1 & + & x_2 & + & x_3 & & & = & 5 \\ x_1 & & & & + & x_3 & & = & 4 \end{array}$$

i) The next step of forward elimination gives

$$\begin{array}{rcccccccl} x_1 & + & x_2 & & & & & = & 3 \\ & & (1 - \varepsilon)x_2 & + & x_3 & & & = & 5 - 3\varepsilon \\ & & -x_2 & & + & x_3 & & = & 4 - 3 \end{array}$$

ii) Roundoff eliminates the ε terms giving

$$\begin{array}{rcccccccl} x_1 & + & x_2 & & & & & = & 3 \\ & & x_2 & + & x_3 & & & = & 5 \\ & & -x_2 & + & x_3 & & & = & 1 \end{array}$$

- iii) The second step of forward elimination removes x_2 from the third equation using the second equation,

$$\begin{array}{rcl} x_1 + x_2 & = & 3 \\ x_2 + x_3 & = & 5 \\ 2x_3 & = & 6 \end{array}$$

- iv) You can easily substitute back giving $x_1 = 1, x_2 = 2, x_3 = 3$.

- h) Algorithms that rearrange the equations when they spot small diagonal elements are said to *pivot*. The price of pivoting is just a little extra bookkeeping in the program, but it is essential to use pivoting in all but the smallest matrices.
- i) Even with pivoting, one cannot guarantee being safe from roundoff problems when dealing with very large matrices. The program below performs Gaussian elimination with pivoting.

```

      subroutine ge(aa,bb,n,np,x)
* Perform Gaussian elimination to solve aa*x = bb
* Matrix aa is physically np by np but only n by n is used (n <= np)
      parameter( nmax = 100 )
      real aa(np,np),bb(np),x(np)
      real a(nmax,nmax), b(nmax)
      integer index(nmax)
      real scale(nmax)
*
      if( np .gt. nmax ) then
        print *, 'ERROR - Matrix is too large for ge routine'
        stop
      end if
*
      do i=1,n
        b(i) = bb(i)          ! Copy vector
        do j=1,n
          a(i,j) = aa(i,j)    ! Copy matrix
        end do
      end do
*

```



```

* !!!!! Forward elimination !!!!!
*
  do i=1,n
    index(i) = i
    scalemax = 0.
    do j=1,N
      scalemax = amax1(scalemax,abs(a(i,j)))
    end do
    scale(i) = scalemax
  end do
*
  do k=1,N-1
    ratiomax = 0.
    do i=k,n
      ratio = abs(a(index(i),k))/scale(index(i))
      if( ratio .gt. ratiomax ) then
        j=i
        ratiomax = ratio
      end if
    end do
    indexk = index(j)
    index(j) = index(k)
    index(k) = indexk
    do i=k+1,n
      coeff = a(index(i),k)/a(indexk,k)
      do j=k+1,n
        a(index(i),j) = a(index(i),j) - coeff*a(indexk,j)
      end do
      a(index(i),k) = coeff
      b(index(i)) = b(index(i)) - a(index(i),k)*b(indexk)
    end do
  end do
*
* !!!!! Back substitution !!!!!
*
  x(n) = b(index(n))/a(index(n),n)
  do i=n-1,1,-1
    sum = b(index(i))
    do j=i+1,n
      sum = sum - a(index(i),j)*x(j)
    end do
    x(i) = sum/a(index(i),i)
  end do
*
  return
end

```

2. Working with Matrices.

- a) It is easy to obtain **determinants** of a matrix using Gaussian elimination. After completing forward elimination,

one simply computes the product of the coefficients of the diagonal elements. Take the equations in **Example VII-4**, the matrix is

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 4 & -1 \end{bmatrix}.$$

With forward elimination, these equations become

$$\begin{aligned} 2x_1 + x_2 &= 4 \\ -3x_2 &= -6 \end{aligned}$$

The products of the coefficients of the diagonal elements of this matrix is $(2)(-3) = -6$, which is the determinant of \mathbf{A} above. However, it should be noted that this method is slightly more complicated when pivoting is used. If the number of points is odd, the determinant is the *negative* of the product of the coefficients of the diagonal elements.

b) Matrix Inverse and Gaussian Elimination.

i) Recall the linear equation in Eq. (VII-82),

$$\mathbf{A} \mathbf{x} - \mathbf{b} = 0, \quad (\text{VII-84})$$

where we solved for the vector \mathbf{x} by Gaussian elimination. Note, however, that we could also have solved it with a little matrix algebra:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}, \quad (\text{VII-85})$$

where \mathbf{A}^{-1} is the matrix inverse of \mathbf{A} .

ii) It shouldn't surprise you that the inverse of a matrix is computed by repeated applications of Gaussian elimination (or a variant called LU decomposition).

- iii) As we have already discussed, the **inverse of a matrix** is defined by

$$\mathbf{A} \mathbf{A}^{-1} = \mathbf{I}, \quad (\text{VII-86})$$

where \mathbf{I} is the *identity matrix*:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (\text{VII-87})$$

- iv) Defining the column vectors

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}; \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}; \quad \cdots; \quad \mathbf{e}_N = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (\text{VII-88})$$

we may write the identity matrix as a row vector of column vectors,

$$\mathbf{I} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_N]. \quad (\text{VII-89})$$

- v) If we solve the linear set of equations,

$$\mathbf{A} \mathbf{x}_1 = \mathbf{e}_1, \quad (\text{VII-90})$$

the solution vector \mathbf{x}_1 is the first column of the inverse matrix \mathbf{A}^{-1} .

- vi) If we proceed this way with the other \mathbf{e} 's, we will compute all of the columns of \mathbf{A}^{-1} . In other words, our matrix inverse equation (Eq. VII-86) is solved by writing it as

$$\mathbf{A} [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_N] = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_N]. \quad (\text{VII-91})$$

vii) After computing the \mathbf{x} 's, we build \mathbf{A}^{-1} as

$$\mathbf{A}^{-1} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_N] . \quad (\text{VII-92})$$

viii) It is usually not necessary to write your own routines to do matrix inverse since virtually all programming languages has routines that will do this for you. For instance, **Matlab** has the built in function `inv(A)`, **IDL** has `INVERT(A)`, **Fortran** uses the *Numerical Recipes* `LUDCMP` subroutine which can be freely downloaded (**LINPACK** also has inverting matrices routines), and **C** has similar routines to **Fortran**.

ix) A handy formula to remember involves the inverse of a 2 x 2 matrix:

$$\mathbf{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} . \quad (\text{VII-93})$$

For larger matrices the formulas quickly become very messy.

c) Singular and Ill-Conditioned Matrices.

i) A matrix that has no inverse is said to be **singular**, *e.g.*,

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

And remember, a singular matrix has a determinant of zero.

ii) Sometime a matrix is not singular but is so close to being singular that roundoff errors may *push it over the edge*. A trivial example would be

$$\begin{bmatrix} 1 + \varepsilon & 1 \\ 2 & 2 \end{bmatrix} ,$$

where $\varepsilon \ll 1$.

iii) The *condition* of a matrix indicates how close it is from being singular; a matrix is said to be **ill-conditioned** if it is almost singular.

iv) Formally, the condition criterion is defined as the normalized *distance* between a matrix and the nearest singular matrix. All of the programming languages mentioned above also have the ability of returning this normalized distance with either the inverse function or a separate function.

E. LU Decomposition.

1. Suppose we are able to write a matrix as the product of 2 matrices,

$$\mathbf{L} \cdot \mathbf{U} = \mathbf{A}, \quad (\text{VII-94})$$

where \mathbf{L} is *lower triangular* (has elements only on the diagonal and below) and \mathbf{U} is *upper triangular* (has elements only on the diagonal and above).

2. In the case of a 3 x 3 matrix \mathbf{A} , we would have

$$\begin{bmatrix} \alpha_{11} & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ 0 & \beta_{22} & \beta_{23} \\ 0 & 0 & \beta_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \quad (\text{VII-95})$$

- a) We can use a decomposition such as Eq. (VII-94) to solve the linear set

$$\mathbf{A} \cdot \mathbf{x} = (\mathbf{L} \cdot \mathbf{U}) \cdot \mathbf{x} = \mathbf{L} \cdot (\mathbf{U} \cdot \mathbf{x}) = \mathbf{b} \quad (\text{VII-96})$$

by first solving for the vector \mathbf{y} such that

$$\mathbf{L} \cdot \mathbf{y} = \mathbf{b} \quad (\text{VII-97})$$

and then solving

$$\mathbf{U} \cdot \mathbf{x} = \mathbf{y} . \quad (\text{VII-98})$$

- b)** The advantage to this method is that the solution of a triangular set of equations is quite trivial. Thus Eq. (VII-97) can be solved by *forward-substitution* as follows,

$$\begin{aligned} y_1 &= \frac{b_1}{\alpha_{11}} \\ y_i &= \frac{1}{\alpha_{ii}} \left[b_i - \sum_{j=1}^{i-1} \alpha_{ij} y_j \right] \quad i = 2, 3, \dots, N. \end{aligned} \quad (\text{VII-99})$$

- c)** Then Eq. (VII-98) can then be solved by *back-substituting* exactly as in

$$\begin{aligned} x_N &= \frac{y_N}{\beta_{NN}} \\ x_i &= \frac{1}{\beta_{ii}} \left[y_i - \sum_{j=i+1}^N \beta_{ij} x_j \right] \quad i = N-1, N-2, \dots, 1. \end{aligned} \quad (\text{VII-100})$$

- d)** Equations (VII-99) and (VII-100) total (for each right-hand side **b**) N^2 executions of an inner loop containing one multiply and one add. If we have N right-hand sides which are the unit column vectors (which is the case when we are inverting a matrix), then taking into account the leading zeros reduces the total execution count of Eq. (VII-99) from $\frac{1}{2}N^3$ to $\frac{1}{6}N^3$, while Eq. (VII-100) is unchanged.
- e)** Notice that, once we have the LU decomposition of \mathbf{A} , we can solve with as many right-hand sides as we then care to, one at a time. This is a distinct advantage over the Gaussian elimination scheme described earlier.

3. Performing the LU Decomposition.

- a) How do we solve for **L** and **U** given **A**? First, we write out the i, j^{th} component of Eqs. (VII-94) and (VII-95). That component is always the sum beginning with

$$\alpha_{i1}\beta_{1j} + \cdots = a_{ij}.$$

- b) The number of terms in the sum depends, however, on whether i or j is the smaller number. We have, in fact, the 3 cases:

$$i < j : \alpha_{i1}\beta_{1j} + \alpha_{i2}\beta_{2j} + \cdots + \alpha_{ii}\beta_{ij} = a_{ij} \quad (\text{VII-101})$$

$$i = j : \alpha_{i1}\beta_{1j} + \alpha_{i2}\beta_{2j} + \cdots + \alpha_{ii}\beta_{jj} = a_{ij} \quad (\text{VII-102})$$

$$i > j : \alpha_{i1}\beta_{1j} + \alpha_{i2}\beta_{2j} + \cdots + \alpha_{ij}\beta_{jj} = a_{ij} \quad (\text{VII-103})$$

- c) Eqs. (VII-101)–(VII-103) total N^2 equations for the $N^2 + N$ unknown α 's and β 's (the diagonal being represented twice).
- d) Since the number of unknowns is greater than the number of equations, we are invited to specify N of the unknowns arbitrarily and then try to solve for the others. In fact, it is always possible to take

$$\alpha_{ii} \equiv 1 \quad i = 1, \dots, N. \quad (\text{VII-104})$$

- e) Often, the **Crout algorithm** is used to solve the set of $N^2 + N$ equations (*e.g.*, Eqs. VII-101:103) for all the α 's and β 's. This is done by just arranging the equations in a certain order.
- i) Set $\alpha_{ii} = 1, i = 1, \dots, N$ in Eq. (VII-104).

- ii) For each $j = 1, 2, 3, \dots, N$ do these 2 procedures:
 First, for $i = 1, 2, \dots, j$, use Eqs. (VII-101), (VII-102), and (VII-103) to solve for β_{ij} , namely

$$\beta_{ij} = a_{ij} - \sum_{k=1}^{i-1} \alpha_{ik} \beta_{kj}. \quad (\text{VII-105})$$

(When $i = 1$ in Eq. (VII-28), the summation term is taken to mean zero.)

- iii) Second, for $i = j + 1, j + 2, \dots, N$, use Eq. (VII-103) to solve for α_{ij} , namely,

$$\alpha_{ij} = \frac{1}{\beta_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \alpha_{ik} \beta_{kj} \right). \quad (\text{VII-106})$$

Be sure to do both procedures before going on to the next j .

- iv) In brief, Crout's method fills in the combined matrix of α 's and β 's,

$$\begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \alpha_{21} & \beta_{22} & \beta_{23} \\ \alpha_{31} & \alpha_{32} & \beta_{33} \end{bmatrix}$$

by columns from left to right, and within each column from top to bottom.

- f) Pivoting is absolutely essential for the stability of Crout's method. *Partial pivoting* (interchange of rows) can be implemented efficiently, and this is enough to make the method stable. The *Numerical Recipe's* subroutine **LUDCMP** is an LU decomposition routine using Crout's method with partial pivoting. I recommend its use whenever you need to solve a linear set of equations.

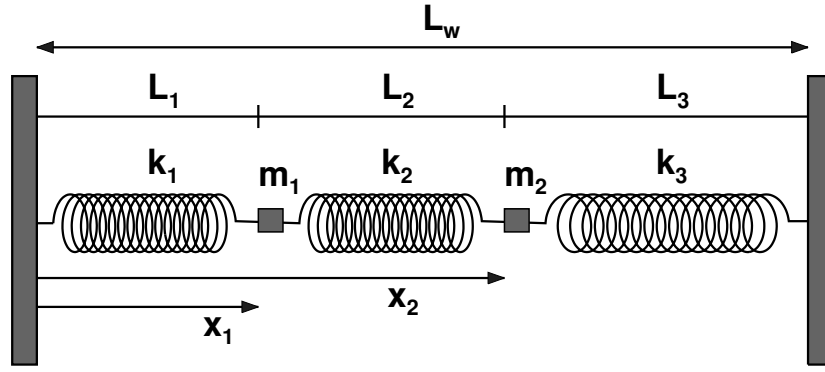


Figure VII-1: A two-mass coupled harmonic oscillator with the origin set at the position of the left wall (*i.e.*, Case A).

F. Coupled Harmonic Oscillators.

1. A canonical example of a system of linear equations is the case of a coupled harmonic oscillator as shown in Figure VII-1. Each spring has an unstretched length of L_1 , L_2 , and L_3 in this example and a spring constant of k_1 , k_2 , and k_3 . In between each spring is an object of mass m_1 and m_2 . Finally, the distance between the non-moving wall is L_w .
2. The equation of motion for block i is

$$\frac{dx_i}{dt} = v_i; \quad \frac{dv_i}{dt} = \frac{F_i}{m_i}, \quad (\text{VII-107})$$

where F_i is the net force on block i .

3. At the steady state, the velocities v_i , are zero and the net forces, F_i , are zero \implies *static equilibrium*.
4. When working with coupled oscillators, one must define a *frame of reference* from which the measurements are made. For instance, one could define the reference point to be the left wall of the system (Case A) as shown in Figure (VII-1), then the net

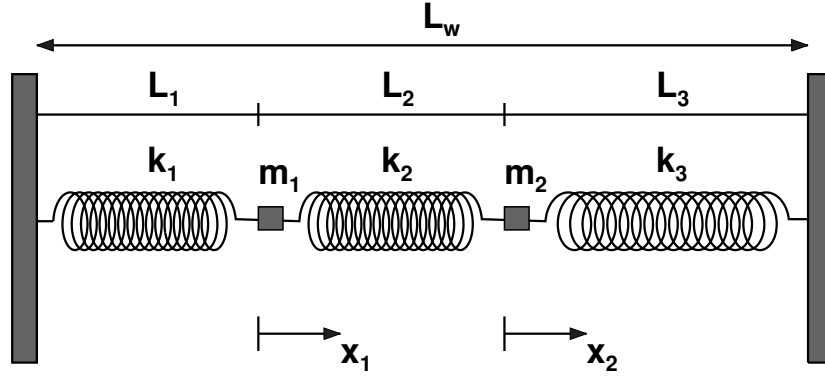


Figure VII-2: A two-mass coupled harmonic oscillator with the origins for each coordinate set at the mass's equilibrium positions (*i.e.*, Case B).

force equations become

$$F_1 = m_1 \ddot{x}_1 = -k_1(x_1 - L_1) + k_2(x_2 - x_1 - L_2) \quad (\text{VII-108})$$

$$F_2 = m_2 \ddot{x}_2 = -k_2(x_2 - x_1 - L_2) + k_3(L_w - x_2 - L_3) . \quad (\text{VII-109})$$

5. To solve these equations, we can use the matrix techniques that have been described earlier in this section (*e.g.*, Gaussian elimination or LU decomposition) by writing these equations in matrix form:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} -k_1 - k_2 & k_2 \\ k_2 & -k_2 - k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -k_1 L_1 + k_2 L_2 \\ -k_2 L_2 + k_3 (L_3 - L_w) \end{bmatrix} \quad (\text{VII-110})$$

or

$$\mathbf{F} = \mathbf{K}_A \cdot \mathbf{x} - \mathbf{b} \quad (\text{VII-111})$$

6. One could also choose the equilibrium positions of each block as the reference (Case B — see Figure VII-2), and write the net force equations as

$$F_1 = m_1 \ddot{x}_1 = -k_1 x_1 - k_2(x_1 - x_2) \quad (\text{VII-112})$$

$$F_2 = m_2 \ddot{x}_2 = -k_3 x_2 - k_2(x_2 - x_1) . \quad (\text{VII-113})$$

7. In matrix form, Case B takes the form of

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} -k_1 - k_2 & k_2 \\ k_2 & -k_2 - k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (\text{VII-114})$$

or in shorthand notation

$$\mathbf{F} = \mathbf{K}_B \cdot \mathbf{x} . \quad (\text{VII-115})$$

As can be seen, the unstretched lengths of the springs do not enter into the second case since measurements are being made from equilibrium positions.