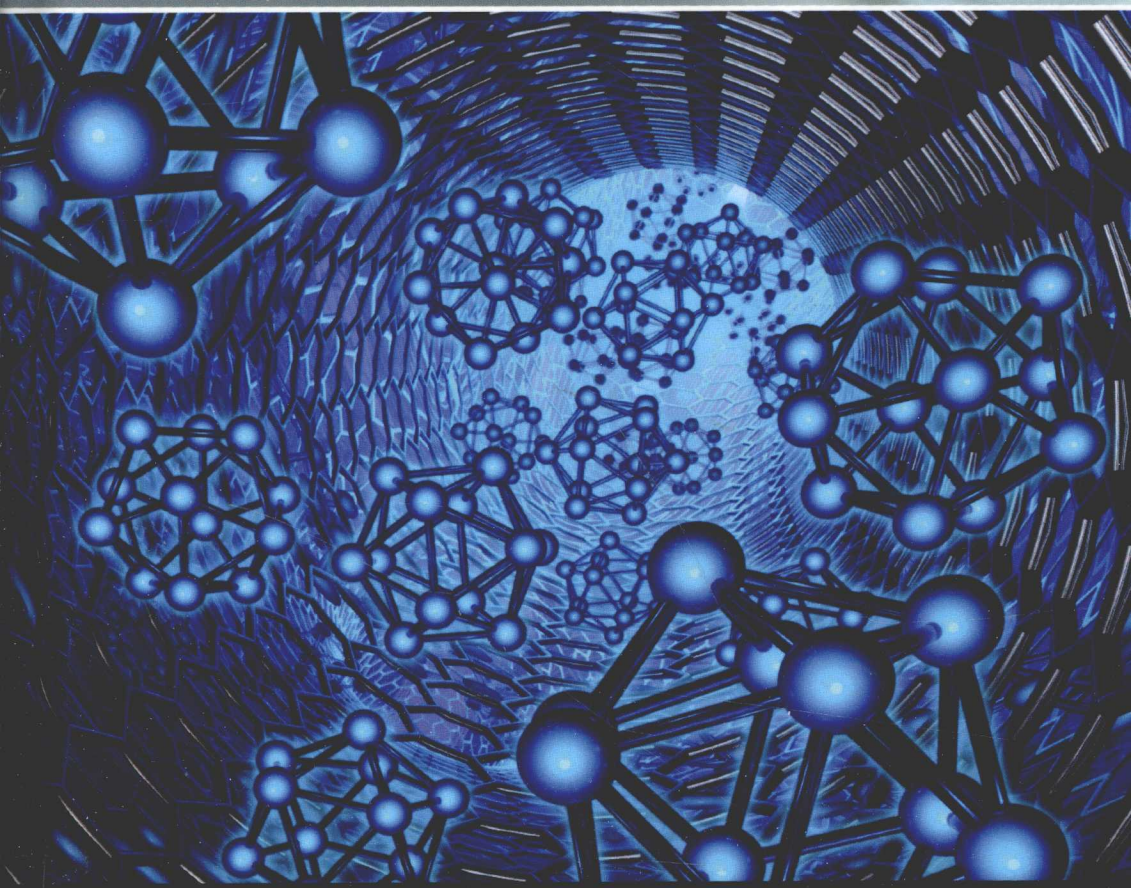
ELSEVIER INSIGHTS



INTRODUCTION TO PRACTICE OF MOLECULAR SIMULATION

MOLECULAR DYNAMICS, MONTE CARLO, BROWNIAN DYNAMICS, LATTICE BOLTZMANN AND DISSIPATIVE PARTICLE DYNAMICS

AKIRA SATOH

Introduction to Practice of Molecular Simulation

Molecular Dynamics, Monte Carlo, Brownian Dynamics, Lattice Boltzmann, Dissipative Particle Dynamics

Akira Satoh
Akita Prefectural University
Japan

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
32 Jamestown Road London NW1 7BY
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

First published 2011

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-385148-2

Introduction to Practice of Molecular Simulation

Preface

The control of internal structure during the fabrication of materials on the nanoscale may enable us to develop a new generation of materials. A deeper understanding of phenomena on the microscopic scale may lead to completely new fields of application. As a tool for microscopic analysis, molecular simulation methods—such as the molecular dynamics and the Monte Carlo methods—have currently been playing an extremely important role in numerous fields, ranging from pure science and engineering to the medical, pharmaceutical, and agricultural sciences. The importance of these methods is expected to increase significantly with the advance of science and technology.

Many physics textbooks address the molecular simulation method for pure liquid or solid systems. In contrast, textbooks concerning the simulation method for suspensions or dispersions are less common; this fact provided the motivation for my previous textbook. Moreover, students or nonexperts needing to apply the molecular simulation method to a physical problem have few tools for cultivating the skill of developing a simulation program that do not require training under a supervisor with expertise in simulation techniques. It became clear that students and nonexpert researchers would find useful a textbook that taught the important concepts of the simulation technique and honed programming skills by tackling practical physical problems with guidance from sample simulation programs. This book would need to be written carefully; it would not simply explain a sample simulation program, but also explains the analysis procedures and include the essence of the theory, the specification of the basic equations, the method of nondimensionalization, and appropriate discussion of results. A brief explanation of the essence of the grammar of programming languages also would be useful.

In order to apply the simulation methods to more complex systems, such as carbon-nanotubes, polymeric liquids, and DNA/protein systems, the present book addresses a range of practical methods, including molecular dynamics and Monte Carlo, for simulations of practical systems such as the spherocylinder and the disk-like particle suspension. Moreover, this book discusses the dissipative particle dynamics method and the lattice Boltzmann method, both currently being developed as simulation techniques for taking into account the multibody hydrodynamic interaction among dispersed particles in a particle suspension or among polymers in a polymeric liquid.

The resulting characteristics of the present book are as follows. The important and essential background relating to the theory of each simulation technique is explained, avoiding complex mathematical manipulation as much as possible. The equations that are included herein are all important expressions; an understanding

of them is key to reading a specialized textbook that treats the more theoretical aspects of the simulation methods. Much of the methodology, such as the assignment of the initial position and velocity of particles, is explained in detail in order to be useful to the reader developing a practical simulation program.

In the chapters dedicated to advancing the reader's practical skill for developing a simulation program, the following methodology is adopted. First, the sample physical phenomenon is described in order to discuss the simulation method that will be addressed in the chapter. This is followed by a series of analyses (including the theoretical backgrounds) that are conducted mainly from the viewpoint of developing a simulation program. Then, the assignment of the important parameters and the assumptions that are required for conducting the simulation of the physical problem are described. Finally, results that have been obtained from the simulation are shown and discussed, with emphasis on the visualization of the results by snapshots. Each example is conducted with a sample copy of the simulation program from which the results were obtained, together with sufficient explanatory descriptions of the important features in the simulation program to aid to the reader's understanding.

Most of the sample simulation programs are written in the FORTRAN language, excepting the simulation program for the Brownian dynamics method. We take into account that some readers may be unfamiliar with programming languages, that is, the FORTRAN or the C language; therefore, an appendix explains the important features of these programming languages from the viewpoint of developing a scientific simulation program. These explanations are expected to significantly reduce the reader's effort of understanding the grammar of the programming languages when referring to a textbook of the FORTRAN or the C language.

The present book has been written in a self-learning mode as much as possible, and therefore readers are expected to derive the important expressions for themselves—that is the essence of each simulation demonstration. This approach should appeal to the reader who is more interested in the theoretical aspects of the simulation methods.

Finally, the author strongly hopes that this book will interest many students in molecular and microsimulation methods and direct them to the growing number of research fields in which these simulation methods are indispensable, and that one day they will be the preeminent researchers in those fields.

The author deeply acknowledges contribution of Dr. Geoff N. Coverdale, who volunteered valuable assistance during the development of the manuscript. The author also wishes to express his thanks to Ms. Aya Saitoh for her dedication and patience during the preparation of so many digital files derived from the handwritten manuscripts.

Akira Satoh
Kisarazu City, Chiba Prefecture, Japan
December 2010

Contents

1 Outline of Molecular Simulation and Microsimulation Methods

In the modern nanotechnology age, microscopic analysis methods are indispensable in order to generate new functional materials and investigate physical phenomena on a molecular level. These methods treat the constituent species of a system, such as molecules and fine particles. Macroscopic and microscopic quantities of interest are derived from analyzing the behavior of these species.

These approaches, called "molecular simulation methods," are represented by the Monte Carlo (MC) and molecular dynamics (MD) methods [1–3]. MC methods exhibit a powerful ability to analyze thermodynamic equilibrium, but are unsuitable for investigating dynamic phenomena. MD methods are useful for thermodynamic equilibrium but are more advantageous for investigating the dynamic properties of a system in a nonequilibrium situation. This book examines MD and MC methods of a nonspherical particle dispersion in a three-dimensional system, which may be directly applicable to such complicated dispersions as DNA and polymeric liquids. This book also addresses Brownian dynamics (BD) methods [1,4], which can simulate the Brownian motion of dispersed particles; dissipative particle dynamics (DPD) [5–8]; and lattice Boltzmann methods [9–12], in which a liquid system is regarded as composed of virtual fluid particles. Simulation methods using the concept of virtual fluid particles are generally used for pure liquid systems, but are useful for simulating particle dispersions.

1.1 Molecular Dynamics Method

A spherical particle dispersion can be treated straightforwardly in simulations because only the translational motion of particles is important, and the treatment of the rotational motion is basically unnecessary. In contrast, since the translational and rotational motion has to be simulated for an axisymmetric particle dispersion, MD simulations become much more complicated in comparison with the spherical particle system. Simulation techniques for a dispersion composed of nonspherical particles with a general shape may be obtained by generalizing the methods employed to an axisymmetric particle dispersion. It is, therefore, very important to understand the MD method for the axisymmetric particle system.

Introduction to Practice of Molecular Simulation. DOI: 10.1016/B978-0-12-385148-2.00001-X

1.1.1 Spherical Particle Systems

The concept of the MD method is rather straightforward and logical. The motion of molecules is generally governed by Newton's equations of motion in classical theory. In MD simulations, particle motion is simulated on a computer according to the equations of motion. If one molecule moves solely on a classical mechanics level, a computer is unnecessary because mathematical calculation with pencil and paper is sufficient to solve the motion of the molecule. However, since molecules in a real system are numerous and interact with each other, such mathematical analysis is impracticable. In this situation, therefore, computer simulations become a powerful tool for a microscopic analysis.

If the mass of molecule i is denoted by m_i, and the force acting on molecule i by the ambient molecules and an external field denoted by $\mathbf{f}_i$, then the motion of a particle is described by Newton's equation of motion:

$$m_i \frac{\mathrm{d}^2 \mathbf{r}_i}{\mathrm{d}t^2} = \mathbf{f}_i \tag{1.1}$$

If a system is composed of N molecules, there are N sets of similar equations, and the motion of N molecules interacts through forces acting among the molecules.

Differential equations such as Eq. (1.1) are unsuitable for solving the set of N equations of motion on a computer. Computers readily solve simple equations, such as algebraic ones, but are quite poor at intuitive solving procedures such as a trial-and-error approach to find solutions. Hence, Eq. (1.1) will be transformed into an algebraic equation. To do so, the second-order differential term in Eq. (1.1) must be expressed as an algebraic expression, using the following Taylor series expansion:

$$x(t+h) = x(t) + h\frac{\mathrm{d}x(t)}{\mathrm{d}t} + \frac{1}{2!}h^2\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2} + \frac{1}{3!}h^3\frac{\mathrm{d}^3x(t)}{\mathrm{d}t^3} + \cdots \tag{1.2}$$

Equation (1.2) implies that x at time $(t+h)$ can be expressed as the sum of x itself, the first-order differential, the second-order differential, and so on, multiplied by a constant for each term. If x does not significantly change with time, the higher-order differential terms can be neglected for a sufficiently small value of the time interval h. In order to approximate the second-order differential term in Eq. (1.1) as an algebraic expression, another form of the Taylor series expansion is necessary:

$$x(t-h) = x(t) - h\frac{\mathrm{d}x(t)}{\mathrm{d}t} + \frac{1}{2!}h^2\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2} - \frac{1}{3!}h^3\frac{\mathrm{d}^3x(t)}{\mathrm{d}t^3} + \cdots \tag{1.3}$$

If the first-order differential term is eliminated from Eqs. (1.2) and (1.3), the second-order differential term can be solved as

$$\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2} = \frac{x(t+h) - 2x(t) + x(t-h)}{h^2} + \mathrm{O}(h^2) \tag{1.4}$$

The last term on the right-hand side of this equation implies the accuracy of the approximation, and, in this case, terms higher than h^2 are neglected. If the second-order differential is approximated as

$$\frac{d^2x(t)}{dt^2} = \frac{x(t+h) - 2x(t) + x(t-h)}{h^2} \tag{1.5}$$

This expression is called the "central difference approximation." With this approximation and the notation $\mathbf{r}_i = (x_i, y_i, z_i)$ for the molecular position and $\mathbf{f}_i = (f_{xi}, f_{yi}, f_{zi})$ for the force acting on particle i, the equation of the x-component of Newton's equation of motion can be written as

$$x_i(t+h) = 2x_i(t) - x_i(t-h) + \frac{h^2}{m_i} f_{xi}(t) \tag{1.6}$$

Similar equations are satisfied for the other components. Since Eq. (1.6) is a simple algebraic equation, the molecular position at the next time step can be evaluated using the present and previous positions and the present force. If a system is composed of N molecules, there are $3N$ algebraic equations for specifying the motion of molecules; these numerous equations are solved on a computer, where the motion of the molecules in a system can be pursued with the time variable. Eq. (1.6) does not require the velocity terms for determining the molecular position at the next time step. This scheme is called the "Verlet method" [13]. The velocity, if required, can be evaluated from the central difference approximation as

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t+h) - \mathbf{r}_i(t-h)}{2h} \tag{1.7}$$

This approximation can be derived by eliminating the second-order differential terms in Eqs. (1.2) and (1.3). It has already been noted that the velocities are unnecessary for evaluating the position at the next time step; however, a scheme using the positions and velocities simultaneously may be more desirable in order to keep the system temperature constant. We show such a method in the following paragraphs.

If we take into account that the first- and second-order differentials of the position are equal to the velocity and acceleration, respectively, the neglect of differential terms equal to or higher than third-order in Eq. (1.2) leads to the following equation:

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{v}_i(t) + \frac{h^2}{2m_i}\mathbf{f}_i(t) \tag{1.8}$$

This equation determines the position of the molecules, but the velocity term arises on the right-hand side, so that another equation is necessary for specifying

the velocity. The first-order differential of the velocity is equal to the acceleration:

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \frac{h}{m_i}\mathbf{f}_i(t) \tag{1.9}$$

In order to improve accuracy, the force term in Eq. (1.9) is slightly modified and the following equation obtained:

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \frac{h}{2m_i}(\mathbf{f}_i(t) + \mathbf{f}_i(t+h)) \tag{1.10}$$

The scheme of using Eqs. (1.8) and (1.10) for determining the motion of molecules is called the "velocity Verlet method" [14]. It is well known that the velocity Verlet method is significantly superior in regard to the stability and accuracy of a simulation.

Consider another representative scheme. Noting that the first-order differential of the position is the velocity and that of the velocity is the acceleration, the application of the central difference approximation to these first-order differentials leads to the following equations:

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{v}_i(t+h/2) \tag{1.11}$$

$$\mathbf{v}_i(t+h/2) = \mathbf{v}_i(t-h/2) + \frac{h}{m_i}\mathbf{f}_i(t) \tag{1.12}$$

The scheme of pursuing the positions and velocities of the molecules with Eqs. (1.11) and (1.12) is called the "leapfrog method" [15]. This name arises from the evaluation of the positions and forces, and then the velocities, by using time steps in a leapfrog manner. This method is also a significantly superior scheme in regard to stability and accuracy, comparable to the velocity Verlet method.

The MD method is applicable to both equilibrium and nonequilibrium physical phenomena, which makes it a powerful computational tool that can be used to simulate many physical phenomena (if computing power is sufficient).

We show the main procedure for conducting the MD simulation using the velocity Verlet method in the following steps:

1. Specify the initial position and velocity of all molecules.
2. Calculate the forces acting on molecules.
3. Evaluate the positions of all molecules at the next time step from Eq. (1.8).
4. Evaluate the velocities of all molecules at the next time step from Eq. (1.10).
5. Repeat the procedures from step 2.

In the above procedure, the positions and velocities will be evaluated at every time interval h in the MD simulation. The method of specifying the initial positions and velocities will be shown in Chapter 2.

Finally, we show the method of evaluating the system averages, which are necessary to make a comparison with experimental or theoretical values. Since

microscopic quantities such as positions and velocities are evaluated at every time interval in MD simulations, a quantity evaluated from such microscopic values—for example, the pressure—will differ from that measured experimentally. In order to compare with experimental data, instant pressure is sampled at each time step, and these values are averaged during a short sampling time to yield a macroscopic pressure. This average can be expressed as

$$\overline{A} = \sum_{n=1}^{N} A_n / N \tag{1.13}$$

in which A_n is the nth sampled value of an arbitrary physical quantity A, and $\overline{A}$, called the "time average," is the mathematical average of N sampling data.

1.1.2 Nonspherical Particle Systems

1.1.2.1 Case of Taking into Account the Inertia Terms

For the case of nonspherical particles, we need to consider the translational motion of the center of mass of a particle and also the rotational motion about an axis through the center of mass. Axisymmetric particles are very useful as a particle model for simulations, so we will focus on the axisymmetric particle model in this section. As shown in Figure 1.1, the important rotational motion is to be treated about the short axis line. If the particle mass is denoted by m, the inertia moment by I, the position and velocity vectors of the center of mass of particle i by $\mathbf{r}_i$ and $\mathbf{v}_i$, respectively, the angular velocity vector about the short axis by ω_i, and the force and torque acting on the particle by $\mathbf{f}_i$ and $\mathbf{T}_i$, respectively, then the equations of motion concerning the translational and rotational motion can be written as

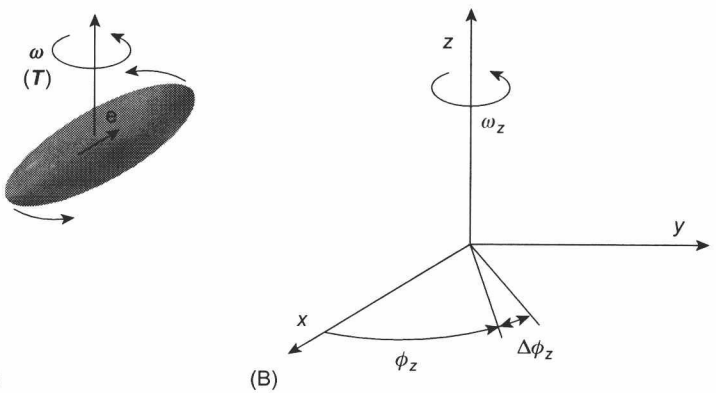


Figure 1.1 Linear particle and angular velocity: (A) the axisymmetric particle and (B) the coordinate system.

$$m\frac{\mathrm{d}^2\mathbf{r}_i}{\mathrm{d}t^2} = \mathbf{f}_i \tag{1.14}$$

$$I\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \mathbf{T}_i \tag{1.15}$$

Since the translational velocity $\mathbf{v}_i$ is related to the position vector $\mathbf{r}_i$ as $\mathbf{v}_i = \mathrm{d}\mathbf{r}_i/\mathrm{d}t$, we now consider the meaning of a quantity $\boldsymbol{\phi}_i$, which is related to the angular velocity $\boldsymbol{\omega}_i$ as $\boldsymbol{\omega}_i = \mathrm{d}\boldsymbol{\phi}_i/\mathrm{d}t$. It is assumed that during a short time interval Δt, $\boldsymbol{\phi}_i$ changes into $(\boldsymbol{\phi}_i + \Delta\boldsymbol{\phi}_i)$ where $\Delta\boldsymbol{\phi}_i$ is expressed as $\Delta\boldsymbol{\phi}_i = (\Delta\phi_{ix}, \Delta\phi_{iy}, \Delta\phi_{iz})$. As shown in Figure 1.1B, ω_z is related to the rotational angle in the xy-plane about the z-axis, $\Delta\phi_z$. The other components have the same meanings, so that $\boldsymbol{\phi}_i$ and $\boldsymbol{\omega}_i$ for particle i can be related in the following expression:

$$\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}_i(t + \Delta t) - \boldsymbol{\phi}_i(t) = \Delta t\boldsymbol{\omega}_i(t) \tag{1.16}$$

Is the use of the quantity $\boldsymbol{\phi}_i$, corresponding to $\mathbf{r}_i$, general? It seems to be more direct and more intuitive to use the unit vector $\mathbf{e}_i$ denoting the particle direction rather than the quantity $\boldsymbol{\phi}_i$. The change in $\mathbf{e}_i$ during an infinitesimal time interval, $\Delta\mathbf{e}_i$, can be written using the angular velocity $\boldsymbol{\omega}_i$ as

$$\Delta\mathbf{e}_i(t) = \mathbf{e}_i(t + \Delta t) - \mathbf{e}_i(t) = \Delta t\boldsymbol{\omega}_i(t) \times \mathbf{e}_i(t) \tag{1.17}$$

From Eqs. (1.16) and (1.17), $\mathbf{e}_i$ can be related to $\boldsymbol{\phi}_i$ as

$$\Delta\mathbf{e}_i(t) = \Delta\boldsymbol{\phi}_i(t) \times \mathbf{e}_i(t) \tag{1.18}$$

Equation (1.17) leads to the governing equation specifying the change of the particle direction:

$$\frac{\mathrm{d}\mathbf{e}_i(t)}{\mathrm{d}t} = \boldsymbol{\omega}_i(t) \times \mathbf{e}_i(t) \tag{1.19}$$

Hence, Eq. (1.15) for the angular velocity and Eq. (1.19) for the particle direction govern the rotational motion of an axisymmetric particle.

In order to solve Eqs. (1.15) and (1.19) for the rotational motion on a computer, these equations have to be translated into finite difference equations. To do so, as already explained, the first- and second-order differentials have to be expressed as algebraic expressions using the finite difference approximations based on Taylor series expansions. General finite difference expressions are as follows:

$$\left.\begin{aligned} &\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{x(t + \Delta t) - x(t)}{\Delta t} + \mathrm{O}(\Delta t), \quad \frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{x(t) - x(t - \Delta t)}{\Delta t} + \mathrm{O}(\Delta t) \\ &\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} + \mathrm{O}((\Delta t)^2) \end{aligned}\right\} \tag{1.20}$$

$$\frac{d^2x(t)}{dt^2} = \frac{x(t+\Delta t) - 2x(t) + x(t-\Delta t)}{(\Delta t)^2} + O((\Delta t)^2) \tag{1.21}$$

The simplest algorithm can be obtained using the forward finite difference approximation in Eq. (1.20) as

$$\left.\begin{aligned} \mathbf{e}_i(t+\Delta t) &= \mathbf{e}_i(t) + \Delta t \boldsymbol{\omega}_i(t) \times \mathbf{e}_i(t) \\ \boldsymbol{\omega}_i(t+\Delta t) &= \boldsymbol{\omega}_i(t) + \Delta t \frac{\mathbf{T}_i(t)}{I} \end{aligned}\right\} \tag{1.22}$$

This algorithm is quite straightforward and understandable, but in practice does not have sufficient accuracy, since the error of the forward finite difference approximation is of the order of Δt. In order to improve the accuracy, the following algorithm has already been presented.

If the new vector function $\mathbf{u}_i(t)$ such as $\mathbf{u}_i\ (t) = \boldsymbol{\omega}_i\ (t) \times \mathbf{e}_i\ (t)$ is introduced, Eq. (1.19) can be written as

$$\frac{d\mathbf{e}_i(t)}{dt} = \mathbf{u}_i(t) \tag{1.23}$$

By conducting the operator $\times \mathbf{e}$ from the right side on the both sides of Eq. (1.15), the following equation is obtained:

$$\frac{d\boldsymbol{\omega}_i(t)}{dt} \times \mathbf{e}_i(t) = \frac{1}{I}\mathbf{T}_i(t) \times \mathbf{e}_i(t) \tag{1.24}$$

The left-hand side of this equation leads to

$$\frac{d\boldsymbol{\omega}_i}{dt} \times \mathbf{e}_i = \frac{d(\boldsymbol{\omega}_i \times \mathbf{e}_i)}{dt} - \boldsymbol{\omega}_i \times \frac{d\mathbf{e}_i}{dt} = \frac{d\mathbf{u}_i}{dt} - \boldsymbol{\omega}_i \times \mathbf{u}_i \tag{1.25}$$

By substituting this equation into Eq. (1.24), the following equation can be obtained:

$$\begin{aligned} \frac{d\mathbf{u}_i(t)}{dt} &= \frac{1}{I}\mathbf{T}_i(t) \times \mathbf{e}_i(t) + \boldsymbol{\omega}_i(t) \times \mathbf{u}_i(t) = \frac{1}{I}\mathbf{T}_i(t) \times \mathbf{e}_i(t) - \left|\boldsymbol{\omega}_i(t)\right|^2 \mathbf{e}_i(t) \\ &= \frac{1}{I}\mathbf{T}_i(t) \times \mathbf{e}_i(t) + \lambda_i(t)\mathbf{e}_i(t) \end{aligned} \tag{1.26}$$

In the transformation from the first to the second expressions on the right-hand side, we have used the identity $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$ in evaluating $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{e})$. The quantity $\lambda_i\ (t)$ in the third expression has been introduced in order to satisfy the following relationship:

$$\mathbf{e}_i \cdot \mathbf{u}_i = \mathbf{e}_i \cdot (\boldsymbol{\omega}_i \times \mathbf{e}_i) = 0 \tag{1.27}$$