Binary Interaction Approximation to *N*-Body Problems

Shun-ichi OIKAWA and Hideo FUNASAKA

Graduate School of Engineering, Hokkaido University, Sapporo 060-8628, Japan

(Received 6 January 2009 / Accepted 16 May 2009)

The binary interaction approximation (BIA) to *N*-body problems is proposed. The BIA conserves total linear momenta in principle. Other invariants, such as the total angular momentum and total energy, are conserved to at least 12 effective digits for a two-dimensional hydrogen plasma of $T = 10\,\mathrm{keV}$ and $n = 10^{20}\,\mathrm{m}^{-3}$. For such a plasma, the total CPU time of the BIA is found to scale as approximately $N^{1.9}$, while the conventional direct integration method scales as approximately N^3.

Keywords: *N*-body problem, algebraic approximation, binary interaction approximation, variable step size, parallel computation.

DOI: 10.1585/pfr.5.S1051

1. Introduction

In an isolated *N*-body charged particle system, as shown in Fig. 1, the non-relativistic equation of motion for the *i*-th particle with an electric charge q_i and a mass m_i is

$$m_i \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = q_i \sum_{j\neq i}^{N} \frac{q_j}{4\pi\epsilon_0} \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j|^3}. \tag{1}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{v}_i$ stand for the position and the velocity of the *i*-th particle. Hereafter, the calculation using the above equation of motion, Eq. (1), will be referred to as the direct integration method (DIM).

When $N \geq 3$, it is well known that no exact/analytical solution can be obtained, and one should be content with solutions approximated by using a numerical integration method. In principle, to arbitrary error levels, the numerical solution can be found. However, it is practically impossible for the large number of particles, i.e. $N \ggg 1$, since the number of force calculations on the right-hand side of Eq. (1) is in proportion to N^2. Moreover, the number of time steps tends to increase with increasing N, so the total CPU time should scale as N^3.

The efficient, fast algorithms to calculate inter-particle forces include the tree method [1, 2], the fast multipole expansion method (FMM), and the particle-mesh Ewalt (PPPM) method [3]. Efforts have been made to use parallel computers and/or to develop special-purpose hardware to calculate interparticle forces, e.g., the GRAvity PipE (GRAPE) project [4]. The authors have recently developed an algebraic model for multibody problems [5] and have shown that the momentum transfer cross-section with our model is in good agreement with the exact one [5, 6]. Unfortunately, this model turns out to lack sufficient accuracy in predicting individual particle motions [6].

As shown in Fig. 2, which depicts the relative motion of the particle pair *i* and *j* in the center-of-mass coordinate system, the scattering angle, $\chi \equiv \pi - 2\theta_0$, is found by $b = b_0 \tan\theta_0$, where b is the impact parameter, $b_0 \equiv e^2/4\pi\varepsilon_0\mu g_0^2$ corresponds to $\chi = \pi/2$ scattering, and g_0 is the initial relative speed at $r = \infty$ and $theta = -\theta_0$. Here $\mu \equiv m_i m_j/(m_i + m_j)$ is the reduced mass. In the binary system with an impact parameter b, a typical velocity change Δg in the relative velocity is given by

$$\Delta g = 2g_0 \sin\frac{\chi}{2} \sim \epsilon g_0, \ \epsilon \equiv \frac{b_0}{\Delta\ell}. \tag{2}$$

where $\Delta\ell$ is the average interparticle separation.

In *N*-body systems with $\epsilon \ll 1$, such as the fusion plasmas, Eq. (2) suggests that three-or-more-body interaction is of the order of ϵ^2 and can be ignored. Note that the Debye length λ_D in fusion plasmas generally satisfies $\lambda_\mathrm{D} \gg \Delta\ell$, thus, typical binary interaction is characterized by the nondimensional parameter ϵ. This parameter is of the order of U/K, where U and K stand for the potential and kinetic energies, respectively.

In this study, we will propose the binary interaction approximation (BIA) to the *N*-body systems with $\epsilon \ll 1$, and compare it with the DIM, both using the six-stage fifth order Runge-Kutta-Fehlberg (RKF65) integrator [7,8] with

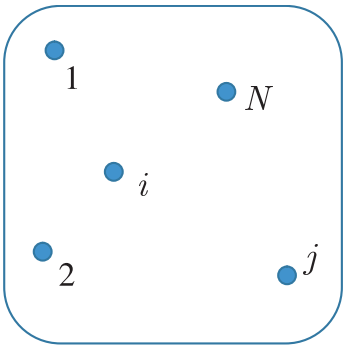

Fig. 1 An *N*-body system.

author's e-mail: oikawa@qe.eng.hokudai.ac.jp

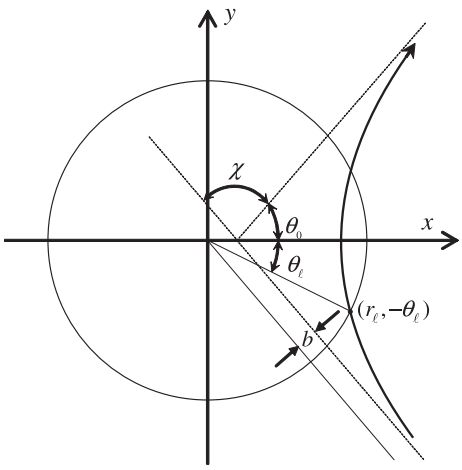

Fig. 2 Unperturbed relative trajectory $r = r(\theta)$ in an orbital plane. The scattering center is at the origin. Impact parameter is $b = b_0 \tan\theta_0$.

an absolute numerical error tolerance of 10^{-16}.

2. BIA: Binary interaction approximation to *N*-body problems

The equation of relative motion for the particle pair (i, j) in an N-body system used by the BIA is

$$\mu_{ij}\frac{\mathrm{d}\boldsymbol{g}_{ij}}{\mathrm{d}t} = \frac{q_i q_j}{4\pi\epsilon_0}\frac{\boldsymbol{r}_{ij}}{r_{ij}^3}. \tag{3}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ stands for the relative position, $\boldsymbol{g}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$ is the relative velocity, and $\mu_{ij} = m_i m_j/\left(m_i + m_j\right)$ is the reduced mass. In the BIA, the above equation is integrated, completely ignoring the other particles, from $t = 0$ to $t = \Delta t$ to yield $\Delta\boldsymbol{r}_{ij}$ and $\Delta\boldsymbol{g}_{ij}$. The total number of integrations is ${}_N\mathrm{C}_2 = N(N-1)/2$ for an N-body problem. The individual changes in position $\Delta\boldsymbol{r}_i$, and velocity $\Delta\boldsymbol{v}_i$, of the i-th particle are

$$m_i\Delta\boldsymbol{r}_i = m_i\boldsymbol{v}_i\Delta t + \sum_{j\neq i}^{N}\mu_{ij}\left(\Delta\boldsymbol{r}_{ij} - \boldsymbol{g}_{ij}\Delta t\right). \tag{4}$$

$$m_i\Delta\boldsymbol{v}_i = \sum_{j\neq i}^{N}\mu_{ij}\Delta\boldsymbol{g}_{ij}\,. \tag{5}$$

for $i = 1, 2, \cdots, N$. Note that the term within the parentheses, $\delta\boldsymbol{r}_{ij} \equiv \Delta\boldsymbol{r}_{ij} - \boldsymbol{g}_{ij}\Delta t$ as shown in Fig. 3, on the right-hand side of Eq. (4) vanishes when the interaction between pair (i, j) vanishes. In other words, the BIA scheme is exact for free particles. Note also that the total momentum $\boldsymbol{P} \equiv \sum_{i=1}^{N} m_i\boldsymbol{v}_i$ is kept constant with this approximation, since, from Eq. (4) and $\mu_{ij} = \mu_{ji}$,

$$\sum_{i=1}^{N} m_i\Delta\boldsymbol{v}_i = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\mu_{ij}\left(\Delta\boldsymbol{g}_{ij} + \Delta\boldsymbol{g}_{ji}\right) = 0\,. \tag{6}$$

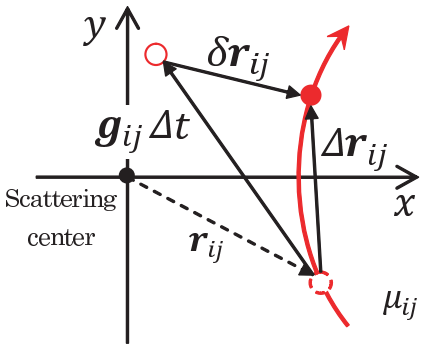

Fig. 3 Relative motion for particle pair (i, j). Scattering center is at the origin. The change in position of the particle with a mass μ_{ij} is $\Delta\boldsymbol{r}_{ij}$. If no interaction occurs, the change in position is $\boldsymbol{g}_{ij}\Delta t$ during a time interval of Δt.

which also guarantees the center of mass position $\boldsymbol{R}_{\mathrm{CM}}$ to be exact,

$$\boldsymbol{R}_{\mathrm{CM}}(\Delta t) = \boldsymbol{R}_{\mathrm{CM}}(0) + \boldsymbol{G}_{\mathrm{CM}}(0)\,\Delta t, \tag{7}$$

where $\boldsymbol{G}_{\mathrm{CM}}$ is the center of mass velocity.

3. Calculation

3.1 Initial condition for an $N = 122$-body problem

Figure 4 depicts the initial condition for a two-dimensional $N = 122$-problem, in which there are 61 protons and 61 electrons. Note that only 26 particles near the origin are depicted in the figure. In this and the following figures, positions are normalized by the interparticle separation $\Delta\ell \equiv n^{-1/3}$ and velocities by the relative thermal speed among electrons, $g_{\mathrm{th}}^{ee} = \sqrt{2}v_{\mathrm{th}}^{e}$. Squares with arrows in blue in Fig. 4 represent electron positions and velocities, and diamonds with arrows in red represent protons. Spatial distribution is uniform with an average particle distance being $\Delta\ell = n^{-1/3}$, and the velocity distribution is Maxwellian for both species with temperatures of $T = T_{\mathrm{electron}} = T_{\mathrm{proton}} = 10\,\mathrm{keV}$. A number density $n = 10^{20}\,\mathrm{m}^{-3}$ is assumed, which yields the parameter $\epsilon = 1.67\times10^{-7} \ll 1$.

Since we have assumed Maxwellian velocity distributions, no particle has a velocity of exactly zero, i.e., no particle at rest. In terms of numerical errors, however, we have found that the BIA method has given larger numerical errors, especially in position, for particles at rest than for those in motion. For this reason, we have intentionally assigned zero velocity to one proton at the origin $(X, Y) = (0, 0)$, as shown in Fig. 4. The trajectories of this proton as well as a typical electron will be shown in the next section.

3.2 Trajectories of a proton and an electron

The 122-body system is integrated for $\Delta t \equiv \Delta\ell/g_{\mathrm{th}}^{ee}$, i.e., the time for the electron with its thermal speed to travel

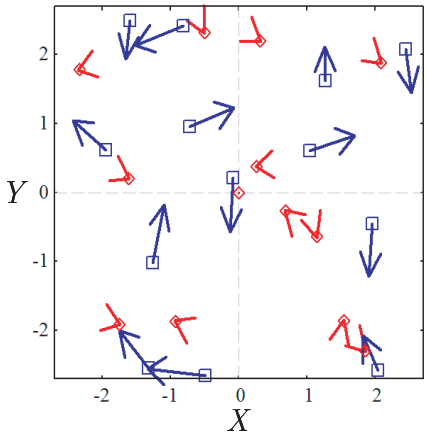

Fig. 4 Initial positions normalized by the interparticle separation $\Delta\ell \equiv n^{-1/3}$ of an $N = 122$-body problem. Only 26 particles near the origin are depicted. Squares with arrows in blue represent the electron positions and velocities, and diamonds with arrows in red represent protons. Spatial distribution is uniform for both protons and electrons. The Maxwellian velocity distribution is adopted for a given temperature $T = T_{\text{electron}} = T_{\text{proton}}$.

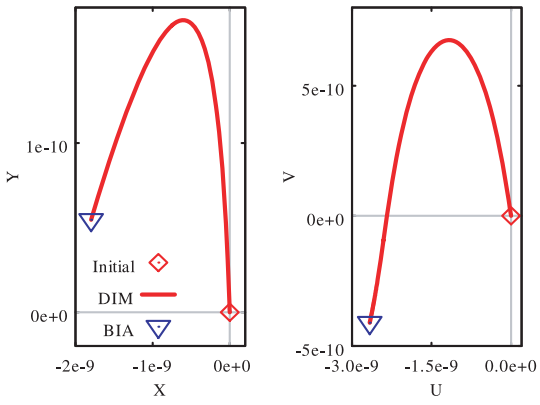

Fig. 5 Motion of a proton initially at rest in the configuration space (left), and in the velocity space (right) for the N =122-body system. Symbols represent the initial and final position calculated with the BIA. The particle starts at the diamonds and moves along lines which are calculated with the full N-body integration, i.e., the DIM.

the average interparticle separation $\Delta\ell \equiv n^{-1/3}$. Figures 5 and 6 show the trajectories in the configuration space (X, Y) on the left and velocity space (U, V) on the right for a proton initially at rest and a moving electron, respectively. In both figures, the diamonds labeled 'initial' are initial points at $t = 0$. The lines are trajectories obtained by using the DIM. Triangles indicate the final points at $t = \Delta t$ with the BIA. The agreement between the BIA and the DIM is excellent.

As shown on the right in Fig. 6, the complicated change in velocity with time, or the acceleration, is typically reproduced well with the BIA, in which three-or-more-body interactions are ignored.

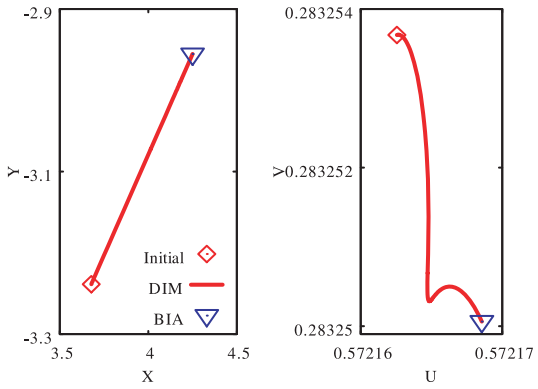

Fig. 6 Motion of an electron for the N =122-body system. Legends are the same as in Fig. 5.

Table 1 Effective digits for calculated invariants of motion and CPU time for $N = 122$. P_X and P_Y are the total linear momenta, L_Z is the total angular momentum, E is the total energy of the system.

method	P_X	P_Y	L_Z	E	CPU time
DIM	16	15	16	16	3.4
BIA	16	15	15	12	0.2
unit	digit				sec

3.3 Errors and effective digits of invariants

There are four invariants of motion in an isolated two-dimensional system: the total linear momenta $\boldsymbol{P} = (P_X, P_Y)$, the total angular momentum L_Z, and the total energy E. Effective digits for the calculated invariants of motion and the CPU time for $N = 122$ are listed in Table 1. In the table with $\boldsymbol{r}_i = (X_i, Y_i)$, and $\boldsymbol{v}_i = (U_i, V_i)$,

$$P_X = \sum_{i=1}^{N} m_i U_i \,. \tag{8}$$

$$P_Y = \sum_{i=1}^{N} m_i V_i \,. \tag{9}$$

are the total linear momenta,

$$L_Z = \sum_{i=1}^{N} m_i \left(X_i V_i - Y_i U_i\right). \tag{10}$$

is the total angular momentum, and

$$E = \frac{1}{2}\sum_{i=1}^{N} m_i \boldsymbol{v}_i^2 + \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1} q_i \sum_{j=i+1}^{N} \frac{q_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \,. \tag{11}$$

is the total energy of the system.

Note that 15–16 effective digits is the maximum for 64-bit calculation on the computer used in this study. In the case of the DIM, the effective digits for all invariants reach this maximum, whereas the total energy conservation for the BIA is 12 digits, worse than the DIM which is generally the case. The angular momentum conservation for the BIA happens to be 15 digits for this particular

initial conditions for $N = 122$-body problem. The conservation in L_Z, however, is generally close to that of E for different initial conditions and the numbers of particles N.

As for the CPU time, the BIA is 17 times faster than the conventional DIM for $N = 122$. Since the speed-up ratio depends essentially on the number of particles N, calculations for different N will be examined in the following subsection.

3.4 CPU time dependence on N

We made a calculation similar to the foregoing section while varying the number of particles N. The CPU time dependence on N is depicted with fitting lines in Fig. 7, in which CPU time inversions are found for the DIM, i.e., longer CPU time $\tau^{\rm CPU}_{\rm DIM}(N) > \tau^{\rm CPU}_{\rm DIM}(N')$ for fewer particles $N < N'$ at around $N \sim 700$ and 1600. Such inversions can occur because the integrator used here, RKF65, controls the time-step size during the calculation according to the given error tolerance. The CPU time for the DIM scales as

$$\tau^{\rm CPU}_{\rm DIM} \sim 1.2 \times 10^{-5} \times N^{2.7} \text{ sec.} \tag{12}$$

and that for the BIA as

$$\tau^{\rm CPU}_{\rm BIA} \sim 3.4 \times 10^{-5} \times N^{1.9} \text{ sec.} \tag{13}$$

both using the RKF65 with the same absolute error tolerance of 10^{-16}. Also, BIA(1) is the CPU time to calculate only one particle, which scales as

$$\tau^{\rm CPU}_{\rm BIA(1)} \sim 1.1 \times 10^{-5} \times N^{1.0} \text{ sec.} \tag{14}$$

If we are interested in the motion of only one test particle-i at a time $t = \Delta t$ from initial conditions at $t = 0$, it is possible with the BIA to calculate $\boldsymbol{r}_i(\Delta t)$ and $\boldsymbol{v}_i(\Delta t)$ *completely in parallel*, since it is based on the principle of superposition of $\Delta \boldsymbol{r}_{ij}$ and $\Delta \boldsymbol{v}_{ij}$ using Eqs. (4) and (5). For example, for an 10^8-body problem, which corresponds to full three-dimensional Coulomb interactions in a fusion plasma within the Debye sphere, the DIM would need a CPU time of 3×10^8 years, while the BIA would require less than an hour to calculate a test particle motion.

As was shown on the right in Fig. 6, the temporal electron acceleration is complicated due to its small mass. For a given numerical error tolerance, this tends to make the common time-step smaller and consequently make the total CPU time $\tau^{\rm CPU}$ longer especially in the DIM. On the other hand, the BIA with the same error tolerance as the DIM is a *pair wise* variable time-step scheme, since the time step for the pair (i, j) is independent of that for any other pairs (i, j').

4. Summary and Discussion

The binary interaction approximation (BIA) to N-body problems is proposed. The BIA conserves total linear momenta in principle, and is a *pair wise* variable time-step scheme when used with an integrator using the embedded

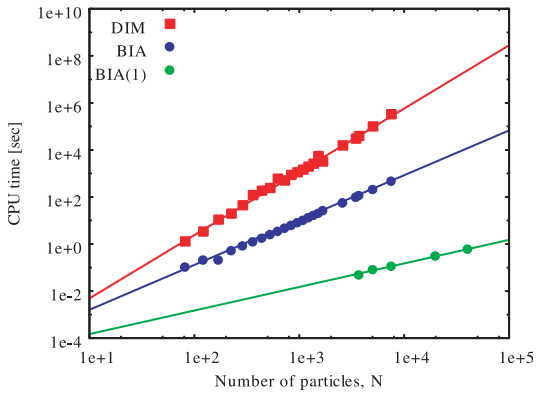

Fig. 7 CPU time $\tau^{\rm CPU}$ dependence on the number of particles N on a typical PC. Red squares represent the CPU time for the DIM with a fitting line in red, $\tau^{\rm CPU}_{\rm DIM} \propto N^{2.7}$. Blue circles represent the CPU time for the BIA with a fitting line in blue, $\tau^{\rm CPU}_{\rm BIA} \propto N^{1.9}$. Also, BIA(1) is the CPU time to calculate only one particle, which scale as $\tau^{\rm CPU}_{\rm BIA(1)} \propto N^{1.0}$.

formula, such as the Runge-Kutta-Fehlberg scheme [7, 8]. Other invariants, such as the total angular momentum and total energy, are conserved to at least 12 effective digits for a two-dimensional hydrogen plasma of $T = 10\,\text{keV}$ and $n = 10^{20}\,\text{m}^{-3}$, in which $\epsilon \sim 1.67 \times 10^{-7}$. The CPU time of the BIA scales as $\tau^{\rm CPU}_{\rm BIA} \propto N^{1.9}$ for such a plasma. Note that with the BIA, it is possible to calculate only one particle's motion [6] with the CPU time proportional to N.

The numerical results presented here are for two-dimensional systems with low density and high temperature, i.e., $\epsilon \ll 1$, which is the most appropriate for the BIA. We will soon apply the BIA to three-dimensional cases and/or to systems with $\epsilon \sim 1$, such as gravitational N-body systems, in the near future.

Acknowledgement

The author thanks Dr. A. Wakasa, Prof. Y. Matsumoto, and Prof. M. Itagaki for fruitful discussions on the subject. The author also acknowledges the continuous encouragement of the late Prof. T. Yamashina. This research was partially supported by a Grant-in-Aid for Scientific Research (C), 21560061.

[1] A.W. Appel, An Efficient Program for Many-Body Simulation, SIAM J. Sci. Stat. Comput. **6**, 85 (1985).
[2] J.E. Barnes and P. Hutt, A hierarchical $O(N\log N)$ force-calculation algorithm, Nature **324**, 446 (1986).
[3] P.P. Brieu, F.J. Summers and J.P. Ostriker, Cosmological Simulations Using Special Purpose Computers: Implementing P^3M on GRAPE, APJ **453**, 566 (1995).
[4] J. Makino, M. Taiji, T. Ebisuzaki, and D. Sugimoto, GRAPE-4: A Massively Parallel Special-Purpose Computer for Collisional N-Body Simulations, APJ **480**, 432 (1997).
[5] S. Oikawa and H. Funasaka, Algebraic analysis approach for multibody problems, Plasma Fusion Res. **36**, S1073 (2008).
[6] S. Oikawa, K. Higashi, H. Funasaka and Y. Kitagawa, Algebraic analysis approach for multibody problems II: Variance

of velocity changes, ITC-18, P2-58(2008); also submitted to PFR.

[7] E. Fehlberg, Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems, NASA Technical Report 315 (1969).

[8] E. Fehlberg, Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Warmeleitungsprobleme, *Computing* (Arch. Elektron. Rechnen), vol. **6**, 61 (1970).