Volume 36, issue 8, September 2007
ISSN 0045-7930
ELSEVIER
An International Journal
computers
&
fluids
Editor-in-Chief:
STANLEY G. RUBIN
Associate Editor:
MICHEL DEVILLE

Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 36 (2007) 1319–1326
computers & fluids
www.elsevier.com/locate/compfluid

Fast vortex method calculation using a special-purpose computer

T.K. Sheel *, K. Yasuoka, S. Obi

Department of Mechanical Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-Ku, Yokohama 223-8522, Japan

Received 1 April 2006; received in revised form 7 November 2006; accepted 20 January 2007
Available online 14 April 2007

Abstract

A mathematical formulation for the 3D vortex method has been developed for calculation using a special-purpose computer MDGRAPE-2 that was originally designed for molecular dynamics simulations. We made an assessment of this hardware for a few representative problems and compared the results with and without it. It is found that the generation of appropriate function tables, which are used to call libraries, embedded in MDGRAPE-2 is of primary importance in order to retain accuracy. The error arising from the approximation is evaluated by calculating a pair of vortex rings impinging to themselves. Consequently, acceleration about 50 times greater is achieved by MDGRAPE-2 while the error in the statistical quantities such as kinetic energy and enstrophy remain negligible.
© 2007 Elsevier Ltd. All rights reserved.

1. Introduction

N-Body simulations were devised in the 1950s and have been widely used since the 1970s when digital computers became powerful enough and affordable. Today it is considered to be an orthodox method for studying particle systems. The classical *N*-body problem simulates the evolution of a system of *N* bodies, where the force exerted on each body arises due to its interaction with all other bodies in the system. *N*-Body algorithms have numerous applications in areas such as astrophysics, molecular dynamics, plasma physics and computational fluid dynamics using the vortex method. For each of these computational problems the calculation takes on a slightly different form but each share common features. The simulation proceeds over time steps, with each step computing the net force on every body and thereby updating its position and other attributes. The cost of a direct summation algorithm for force calculation is $O(N^2)$, therefore calculation time grows rapidly as the number of bodies *N* increases.

There are two ways to reduce the force calculation cost of an *N*-body simulation. One is to use fast algorithms such as the tree code developed by Barnes and Hut [1] or the fast multipole method (FMM) by Greengard and Rokhlin [2]. The tree code is an $O(N\log N)$ algorithm based on a hierarchical octree representation of space in three dimensions. It computes interactions between distant particles and reduces the number of operations by means of a first-order approximation. Many existing implementations of tree code algorithms only use up to quadropole moments and calculation costs rise quickly when high accuracy is required. In the FMM, the long-range forces are approximated by multipole expansion truncated at a certain degree, while the contributions from particles within nearby regions are calculated directly in a usual manner without approximation. Including higher order terms in multipole approximations and/or increasing the size of a nearby region can improve the computational accuracy. However, either effort substantially increases the computation time. In particular, the computation of a high-order term is very expensive.

The other way is to execute the *N*-body simulation with special-purpose hardware such as MDGRAPE-2 developed by Susukita et al. [3]. MDGRAPE-2, one of the GRAPE (Gravity Pipe, developed by Sugimoto et al. [5]) series machines, is a special-purpose computer designed for force calculations between point-charge or point-mass particles. Its performance is much higher than ordinary computers. It can speed up force calculations about

[*] Corresponding author. Fax: +81 45 566 1495.
E-mail address: y12469@educ.cc.keio.ac.jp (T.K. Sheel).

0045-7930/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2007.01.006

10–100 times when compared to general-purpose (defined as ’host’ hereafter) computers of the same cost.

The vortex method solves time-dependent incompressible flow problems by discretizing the vorticity into vortex elements and following these elements in time. This results in a volume mesh-free algorithm and saves significant time in preprocessing when compared to the conventional Navier–Stokes approach where grids need to be generated. The vortex methods have been developed and applied for the analysis of complicated, unsteady and vortical flows related to a wide range of problems found in industry, as it consists of a simple algorithm based on the physics of flow. For details see [6,7].

The main difficulty with vortex methods as originally formulated is that the cost of the evaluation of the velocity field induced by N vortices is $O(N^2)$. This is expensive, particularly in three dimensions where a large number of elements are computed simultaneously. In the calculation of vortex methods, the largest computational load occurs in the routine that calculates the Biot–Savart law and the stretching term in the vorticity equation. In regions of high strain the spatial resolution becomes worse because the distance of each element becomes large. It is required to split vortex elements to keep the spatial resolution in the direction of the vorticity vector of the element. The result is a N-body interaction calculation for millions of particles having calculation cost of $O(N^2)$ with growing N. Nevertheless, these calculations have the same mathematical architecture as a multibody problem, thus permitting the use of special-purpose computers in multibody problems, e.g., MDGRAPE-2.

Our long-term objective is to solve high Reynolds number turbulent flows for engineering problems via a reasonable computational effort. The use of the 3D vortex method is attractive because of its simple formulation and flexibility in moving and/or deforming boundary problems. The long computation time due to the above-mentioned $O(N^2)$ problem may be reduced when we apply a special-purpose computer.

The purpose of the present paper is, therefore, to address a few issues that hindrance the application of MDGRAPE-2 to the vortex method. First, because of the simple architecture, it is required to generate an optimum function table when the embedded libraries are called from the main routine. Second, the cross-product calculation which is not considered in the original command set must be handled in a proper manner, which is treated in some previous works, e.g. [8–11]. These points are discussed one after another in the subsequent sections after an introduction to the basic mathematical formulae.

2. Basic equations

We are studying the three-dimensional incompressible flow of a viscous fluid. The evolution equation for vorticity is

$$\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu\nabla^2\omega \tag{1}$$

where ω is the vorticity defined as $\omega = \nabla \times \mathbf{u}$, $\mathbf{u}$ is the velocity of the vortex element, $(\omega\cdot\nabla)\mathbf{u}$ is called the stretching term and represents the rate of change of vorticity by deformation of vortex lines and the term $\nu\nabla^2\omega$ represents the change of vorticity by viscous diffusion. The velocity field in a three-dimensional problem is,

$$\mathbf{u}(\mathbf{x}) = -\frac{1}{4\pi}\int \frac{(\mathbf{x}-\mathbf{x}')\times\omega(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^3}\,dV(\mathbf{x}') \tag{2}$$

where $\mathbf{x}$, and $\mathbf{x}'$ are the positions of vortex elements and dV is the volume of the element. Using the model [12] as a cut-off function, the Biot–Savart law is formulated as follows:

$$\mathbf{u}_i = -\frac{1}{4\pi}\sum_{j=1}^{N}\frac{\mathbf{r}_{ij}^2 + (5/2)\sigma_j^2}{\left(\mathbf{r}_{ij}^2+\sigma_j^2\right)^{5/2}}\mathbf{r}_{ij}\times\gamma_j \tag{3}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, σ_j and γ_j are the distances of the position vector, core radius and strength of element. The stretching term of Eq. (1) can be discretized as follows:

$$\frac{d\omega}{dt} = (\omega\cdot\nabla)\mathbf{u} \tag{4}$$

If we put vortex strength $\gamma_i = \omega_i d^3\mathbf{x_i}$ in Eq. (4), then it becomes

$$\frac{d\gamma_i}{dt} = (\gamma_i\cdot\nabla)u_i \tag{5}$$

Hence, the vortex strength of an individual element is expressed by Eq. (3) in a discretized formulation as

$$\frac{d\gamma_i}{dt} = \frac{1}{4\pi}\sum_{j=1}\left\{-\frac{|\mathbf{r}_{ij}|^2+(5/2)\sigma_j^2}{(|\mathbf{r}_{ij}|^2+\sigma_j^2)^{5/2}}\gamma_i\times\gamma_j + 3\frac{|\mathbf{r}_{ij}|^2+(7/2)\sigma_j^2}{(|\mathbf{r}_{ij}|^2+\sigma_j^2)^{7/2}}(\gamma_i\cdot\mathbf{r}_{ij})(\mathbf{r}_{ij}\times\gamma_j)\right\} \tag{6}$$

where all notations carry the same meaning as in Eq. (3).

3. Mathematical formulations

MDGRAPE-2 is a special-purpose hardware for the calculation of force or potential between point-mass or point-charge particles that was originally designed for molecular dynamics simulation. The calculation of interactions between particles as represented by potential and force are carried out in MDGRAPE-2. In case of calculating the potential,

$$\Phi_i = \sum_{j=1}^{N} b_{ij}g(w) = \sum_{j=1}^{N} b_{ij}g(a_{ij}(|\mathbf{r}_{ij}|^2+\epsilon_{ij}^2)) \tag{7}$$

and the force calculations

$$\mathbf{f}_i = \sum_{j=1}^{N} b_{ij}g(w)\mathbf{r}_{ij} = \sum_{j=1}^{N} b_{ij}g(a_{ij}(|\mathbf{r}_{ij}|^2+\epsilon_{ij}^2))\mathbf{r}_{ij} \tag{8}$$

are treated similarly, where $g(w)$ is an arbitrary function equivalent to an intermolecular force, and a_{ij}, b_{ij}, ϵ_{ij} are arbitrary coefficients which are settled down for every model. To apply these libraries to the calculation of a vortex method, Biot–Savart law in Eq. (2) is expressed as follows:

$$\mathbf{u}_i = \sum_{j=1}^{N} B_j g(A_j(|\mathbf{r}_{ij}|^2 + \epsilon_{ij}^2))\mathbf{r}_{ij} \tag{9}$$

where A_j, B_j are arbitrary constants. To implement Eqs. (3) and (4), it is required to take special treatment to the cross-product calculation in MDGRAPE-2 which is similar to [8–11] as follows.

The calculation of the cross product $\mathbf{r}_{ij} \times \boldsymbol{\gamma}_j$ is considered.

$$\mathbf{r}_{ij} = (x_{ij}, y_{ij}, z_{ij}); \quad \boldsymbol{\gamma}_j = (\gamma_j^x, \gamma_j^y, \gamma_j^z) \tag{10}$$

The total moment is

$$\sum_j \mathbf{r}_{ij} \times \boldsymbol{\gamma}_j = \sum_j (y_{ij}\gamma_j^z - z_{ij}\gamma_j^y, z_{ij}\gamma_j^x - x_{ij}\gamma_j^z, x_{ij}\gamma_j^y - y_{ij}\gamma_j^x) \tag{11}$$

The cross product is considered with regard to the sum of the following three tensor products.

$$\sum_j \mathbf{r}_{ij} \otimes \gamma_j^x = \sum_j (x_{ij}\gamma_j^x, y_{ij}\gamma_j^x, z_{ij}\gamma_j^x) \tag{12}$$

$$\sum_j \mathbf{r}_{ij} \otimes \gamma_j^y = \sum_j (x_{ij}\gamma_j^y, y_{ij}\gamma_j^y, z_{ij}\gamma_j^y) \tag{13}$$

$$\sum_j \mathbf{r}_{ij} \otimes \gamma_j^z = \sum_j (x_{ij}\gamma_j^z, y_{ij}\gamma_j^z, z_{ij}\gamma_j^z) \tag{14}$$

It should be noted that only non-diagonal components of Eqs. (12)–(14) are required for the calculation of a moment of Eq. (11).

From Eq. (3), the Biot–Savart law reduces according to Eq. (9) as

$$\mathbf{u}_i = -\frac{1}{4\pi} \sum_j \frac{1}{\sigma_j^3} g_1(w)(\mathbf{r}_{ij} \times \gamma_j) \tag{15}$$

where $g_1(w)$ is a function for velocity calculation defined as

$$g_1(w) = \frac{w + 5/2}{(w+1)^{5/2}}; \quad w = (|\mathbf{r}_{ij}|/\sigma_j)^2 \tag{16}$$

The stretching term appearing in Eq. (6) can be divided into two parts. We define here the first and second terms of the right hand side of Eq. (6) as **stx** and **tx**, respectively, as follows.

First part:

$$\begin{aligned}\mathbf{stx} &= -\frac{1}{4\pi} \sum_j \frac{|\mathbf{r}_{ij}|^2 + (5/2)\sigma_j^2}{(|\mathbf{r}_{ij}|^2 + \sigma_j^2)^{5/2}} \gamma_i \times \gamma_j \\ &= -\frac{1}{4\pi} \sum_j g_1(w)(\gamma_i \times \gamma_j)\frac{1}{\sigma_j^3} \\ &= -\frac{1}{4\pi} \sum_j g_1(w)(\gamma_i^y\gamma_j^z - \gamma_i^z\gamma_j^y, \gamma_i^z\gamma_j^x - \gamma_i^x\gamma_j^z, \gamma_i^x\gamma_j^y - \gamma_i^y\gamma_j^x)\frac{1}{\sigma_j^3}\end{aligned} \tag{17}$$

where $g_1(w)$ is defined as above Eq. (16) which is summarized in Table 1.

Table 1
Function and coefficients

$g(w)[w = (r_{ij}/\sigma_j)^2]$	A_j	B_j	ϵ_{ij}
$g_1(w) = \frac{w+5/2}{(w+1)^{5/2}}$	$\frac{1}{\sigma_j^2}$	$\frac{\gamma_j}{\sigma_j^3}$	0
$g_2(w) = \frac{w+7/2}{(w+1)^{7/2}}$	$\frac{1}{\sigma_j^2}$	$\frac{\gamma_j}{\sigma_j^5}$	0

Second part:

$$\begin{aligned}\mathbf{tx} &= \frac{3}{4\pi} \sum_j \frac{|\mathbf{r}_{ij}|^2 + (7/2)\sigma_j^2}{(|\mathbf{r}_{ij}|^2 + \sigma_j^2)^{7/2}} (\gamma_i \cdot \mathbf{r}_{ij})(\mathbf{r}_{ij} \times \gamma_j) \\ &= \frac{3}{4\pi} \sum_j g_2(w)(\gamma_i \cdot \mathbf{r}_{ij})(\mathbf{r}_{ij} \times \gamma_j)\frac{1}{\sigma_j^5} \\ &= \frac{3}{4\pi} \sum_j g_2(w)(\gamma_i \cdot \mathbf{r}_{ij})(y_{ij}\gamma_j^z - z_{ij}\gamma_j^y, z_{ij}\gamma_j^x - x_{ij}\gamma_j^z, x_{ij}\gamma_j^y - y_{ij}\gamma_j^x)\frac{1}{\sigma_j^5}\end{aligned} \tag{18}$$

where $g_2(w)$ is a function for the stretching term calculation summarized in Table 1 and defined as

$$g_2(w) = \frac{w + 7/2}{(w+1)^{7/2}} \tag{19}$$

From Eq. (18), we can write for γ_j^x as follows:

$$\begin{aligned}\mathbf{I}_i &= \sum_j g_2(w)(\gamma_i \cdot \mathbf{r}_{ij})(\gamma_j^x/\sigma_j^5) \cdot \mathbf{r}_{ij} \\ &= (\gamma_i \cdot \mathbf{r}_i) \sum_j g_2(w)(\gamma_j^x/\sigma_j^5) \cdot \mathbf{r}_{ij} - \Bigg\{ \gamma_i^x \sum_j g_2(w)(x_j\gamma_j^x/\sigma_j^5) \cdot \mathbf{r}_{ij} \\ &\quad + \gamma_i^y \sum_j g_2(w)(y_j\gamma_j^x/\sigma_j^5) \cdot \mathbf{r}_{ij} + \gamma_i^z \sum_j g_2(w)(z_j\gamma_j^x/\sigma_j^5) \cdot \mathbf{r}_{ij} \Bigg\} \\ &= (\gamma_i \cdot \mathbf{r}_i)\mathbf{S} - (\gamma_i^x\mathbf{T1} + \gamma_i^y\mathbf{T2} + \gamma_i^z\mathbf{T3})\end{aligned} \tag{20}$$

with

$$\mathbf{S} = \sum_j g_2(w)\frac{\gamma_j^x}{\sigma_j^5} \cdot \mathbf{r}_{ij}$$

$$\mathbf{T1} = \sum_j g_2(w)\frac{x_j\gamma_j^x}{\sigma_j^5} \cdot \mathbf{r}_{ij}$$

$$\mathbf{T2} = \sum_j g_2(w)\frac{y_j\gamma_j^x}{\sigma_j^5} \cdot \mathbf{r}_{ij}$$

$$\mathbf{T3} = \sum_j g_2(w)\frac{z_j\gamma_j^x}{\sigma_j^5} \cdot \mathbf{r}_{ij}$$

Similar formulations can be readily obtained for the other two components, γ_j^y and γ_j^z.

To solve Eq. (15), it is necessary to call the library in Eq. (8), embedded in MDGRAPE-2, three times to compute the cross-product in a three-dimensional problem. Similarly, to solve Eq. (17), it is required to call the library in Eq. (7) three times. Finally, for the Eq. (18), it is required to call library in Eq. (8) for $(\gamma_j^x, \gamma_j^y, \gamma_j^z) \times 4$, i.e., 12 times according to Eq. (20). The values shown in Table 1 were substituted for function $g(w)$ and constants A_j, B_j, ϵ_{ij} are defined for each calculation.

4. Function table

The hardware of MDGRAPE-2 is a board, which is mounted on a PCI-slot of a PC. The main program runs on the host PC, while the force calculation is done on the board via subroutine calls. Due to hardware specifications the subroutine runs partly with single precision, hence special care is necessary for floating-point operations. The important issue here is to rewrite the function table in MDGRAPE-2 that determines the formula of Eqs. (3) and (6) so that the range of the function table contains all elements inside the computational domain. This is considered to be the primary importance of this paper.

A function evaluator (FE) unit, a unit of pipeline of MDGRAPE-2 chip, evaluates $b_j g(w)$. It is composed of two 32-bit floating-point multipliers, one 32-bit function evaluator, and one 32-bit floating-point comparator. The FE evaluates $g(w)$ by 4th order polynomial interpolation as

$$g(w) = c_0 + w(c_1 + w(c_2 + w(c_3 + wc_4))) \tag{21}$$

where c_0, c_1, c_2, c_3, c_4 are arbitrary constants stored in the RAM on the board.

The basic concept on calculation of an arbitrary function by the FE unit is as follows:

A whole range $[x_{\min}, x_{\max})$ is divided into 2^{10} segments. Assuming that an input x belongs to the $(k+1)$th segment, i.e., $x \in [x_k, x_{k+1})$, the function $f(x)$ is approximated by a polynomial of degree 4;

$$f(x) \simeq \sum_{i=0}^{4} c_i^{(k)} (\Delta x)^i \tag{22}$$

where Δx is defined as the difference of x from the center of the segment x_c;

$$\Delta x = x - x_c \tag{23}$$

$$x_c = \frac{x_k + x_{k+1}}{2} \tag{24}$$

The coefficients $c_i^{(k)}$ are constants in each segment. For details see in Susukita et al. [3] and Narumi [4].

The cut-off function is used in the vortex method calculation. It has the same shape between all particle interactions. It is important to produce an optimum function table in order to calculate the cut-off function considering the computational domain where the vortex elements are distributed. The function $g(w)$ is created prior to calculation and read during calculation. The domain of the function $g(w)$ is set to $w_{\min} \leqslant w \leqslant w_{\max}$ where $(w_{\max}/w_{\min}) \leqslant 2^{32}$ according to hardware specifications. In the vortex method calculation, this domain tends to become large when compared to the case of MD simulations. To secure accurate calculations, it is important to know the domain correctly prior to calculation. The net relative accuracy in the MDGRAPE-2 chip is set to 10^{-7}, since this accuracy is usually satisfactory in MD simulations. In the calculation of the Biot–Savart law, it is checked by the accuracy of an arbitrary function $g(w)$ and it is 10^{-7} which has been evaluated by 4th order polynomial interpolation as of Eq. (21).

The table range is different for different problems. It is necessary to generate a new function table for a new problem. The preparation of the function table operation takes only a few minutes, which does not affect the performance of the entire calculations. To simulate high Reynolds number flows using vortex method, it is required to incorporate large number of vortex elements for small-scale structures. There is no connection between the number of elements and the range of function table. The range is the key factor in maintaining the accuracy and the single precision calculation of MDGRAPE-2 board, which does not have any influence on the calculation of high Reynolds number flows.

In order to examine the validity and the applicability of our scheme, an inclined collision of two identical vortex rings is simulated; the details of the computation are given in a subsequent section. Fig. 1 shows the histogram of the distance between vortex elements scattering over the com-

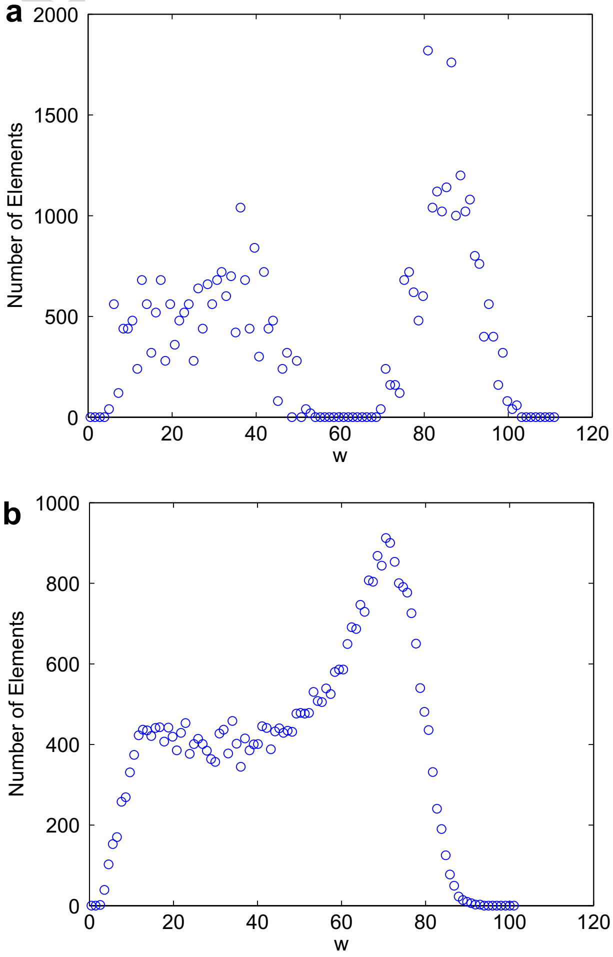

Fig. 1. Typical distributions of vortex elements at (a) $t\Gamma/R^2 = 1$ and (b) $t\Gamma/R^2 = 15$.

putational domain. The results are extracted from the data at non-dimensional time $t^* = 1$ and 15. In the figures, the abscissa w carries the same meaning as that defined in the function table, cf. also in Eq. (16). It is seen from Fig. 1a that the minimum and maximum distance in computational domain at time $t^* = 1$ are 4.38 and 101.39, respectively. These values vary as a function of time. Consequently, it is followed from Fig. 1b that they are 2.17 and 82.91 at $t^* = 15$. It is important that the range of the function table is adjusted to this varying distance between elements.

Based on the above observations, it appears the table ranges have been defined carefully. We generated various different function tables with different finite input ranges in the host calculation. We then checked the output ranges of those function tables by using MDGRAPE-2 and found that all ranges are not satisfied with the computational domain, which was determined in previous paragraph. Fig. 2a and b represent the ranges and scaling errors of two different types of function tables where the x-axis stands for the range of the function $g(w)$ and the y-axis for induced velocity $\mathbf{u}$. The ranges of these figures are $2^{-12} \leqslant w \leqslant 2^{20}$ and $2^{-22} \leqslant w \leqslant 2^{10}$, respectively. Finally, in our calculation, the table has been produced within the finite range of $2^{-12} \leqslant w \leqslant 2^{20}$ which satisfies the computational domain of original calculation to obtain significant accuracy. These figures also represent the error of the induced velocity, which affects the convection of vortex elements and generates discrepancies within the calculation without MDGRAPE-2.

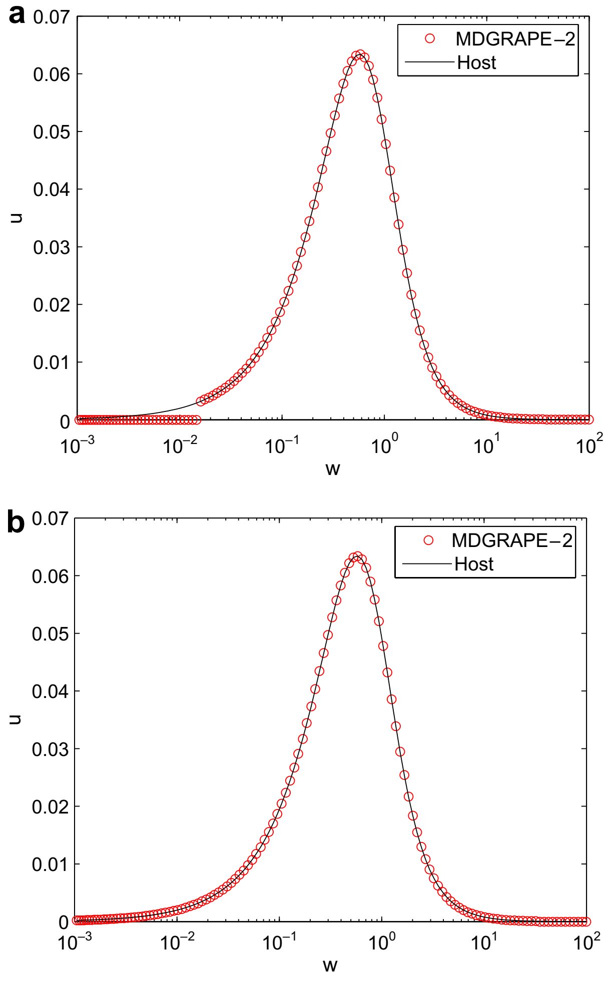

Fig. 2. Scaling error for function table in two different ranges: (a) $2^{-12} \leqslant w \leqslant 2^{20}$ and (b) $2^{-22} \leqslant w \leqslant 2^{10}$. ○ with MDGRAPE-2; – without MDGRAPE-2.

5. Application

5.1. Computational algorithm

We considered inclined collisions according to [12]. Here, we assumed that the initial radius of the vortex rings is $R = 1$ while the cross-section radius $r = 0.05$, see Fig. 3. The Reynolds number based on the ring circulation is $Re_\Gamma = 400$, and the core radius $\sigma = 0.065$. The rings are inclined at an angle $\theta = 15°$ relative to the z-axis. The total number of elements used for the preliminary calculation was $N = 6 \times 10^4$, with the number of cross sections in the circumference direction being 502, while 61 elements were distributed in each cross-section. All elements were evenly distributed.

In this calculation, the viscous diffusion was calculated using the core-spreading method developed by Leonard [13]. For convection of the particles, the second order accurate Adams–Bashforth method was used in the calculation of time advances.

The convection error is defined as the difference in the position of the same particles between the host and MDGRAPE-2 for the same time steps. Here error defined as the distance δ is as follows:

$$\delta(\text{difference}) = \sqrt{(x_{\text{host}} - x_{\text{md}})^2 + (y_{\text{host}} - y_{\text{md}})^2 + (z_{\text{host}} - z_{\text{md}})^2} \tag{25}$$

where the suffices md and host represent with and without the use of MDGRAPE-2, respectively.

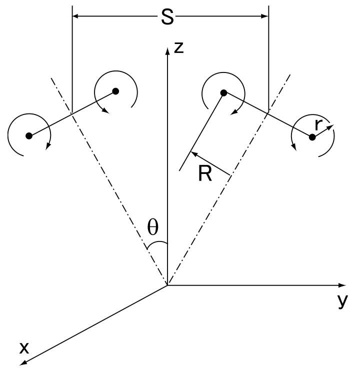

Fig. 3. Initial condition for the computation of the collision of two vortex rings. Here, R – radius of ring, $\mathbf{r}$ – radius of cross-section, S – distance between two rings, θ – inclined angle.

The kinetic energy E and enstrophy Ω are evaluated from the particle positions and strengths according to [12], are defined as follows:

$$E = \frac{1}{16\pi}\sum_{i,j}\left[\frac{2(\gamma_i\cdot\gamma_j)}{(\mathbf{r}_{ij}^2+\sigma_j^2)^{1/2}} + \frac{(\mathbf{r}_{ij}\cdot\gamma_i)(\mathbf{r}_{ij}\cdot\gamma_j)-\mathbf{r}_{ij}^2(\gamma_i\cdot\gamma_j)}{(\mathbf{r}_{ij}^2+\sigma_j^2)^{3/2}}\right] \tag{26}$$

and

$$\Omega = \frac{1}{4\pi}\sum_{i,j}\left[\frac{5\sigma_j^4-\mathbf{r}_{ij}^2(\mathbf{r}_{ij}^2+3.5\sigma_j^2)}{(\mathbf{r}_{ij}^2+\sigma_j^2)^{7/2}}(\gamma_i\cdot\gamma_j)\right.$$
$$\left. +\,3\frac{(\mathbf{r}_{ij}^2(\mathbf{r}_{ij}^2+4.5\sigma_j^2)+3.5\sigma_j^4 i)\sigma_j^2}{(\mathbf{r}_{ij}^2+\sigma_j^2)^{9/2}}(\mathbf{r}_{ij}\cdot\gamma_i)(\mathbf{r}_{ij}\cdot\gamma_j)\right] \tag{27}$$

5.2. Numerical results

Fig. 4 shows the snapshots of vortex elements of two colliding vortex rings in various time stages. The initial setup in colliding ring simulations consists of two identical vortex rings initially inclined at an angle $\theta = 15°$. In Fig. 4a, the rings are initially placed at a non-dimensional distance of $s = 2.7$ in the z-direction. Each vortex ring approaches by self-induced velocity from this initial stage. At $t^* = 3$ in Fig. 4b, where $t^* = t\Gamma/R^2$ with Γ being the initial circulation, the first impact occurs and the two vortex rings are stretched and deformed. As time progresses, considerable differences appear in each stage. At $t^* = 8$, the arced-shape structure is formed and the downward stretch is strong, cf. Fig. 4c.

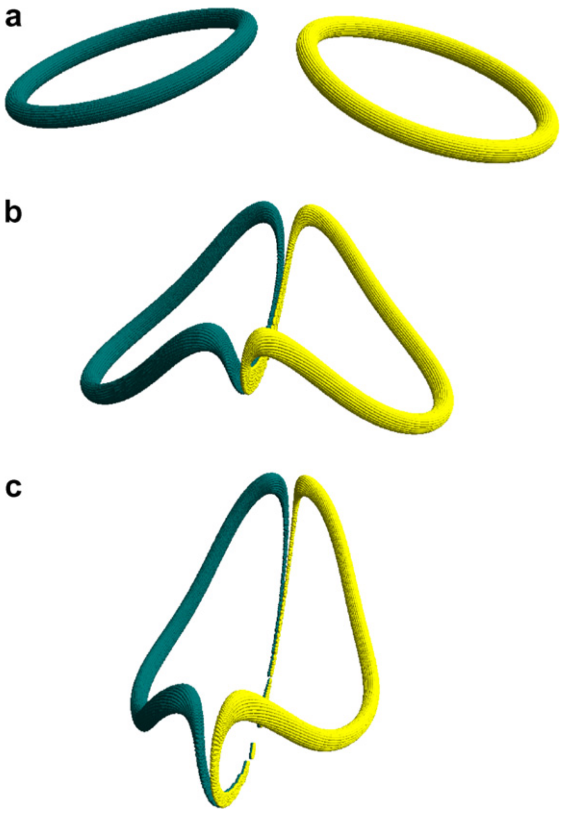

Fig. 4. Snapshots of vortex elements at: (a) $t\Gamma/R^2 = 1$, (b) $t\Gamma/R^2 = 3$ and (c) $t\Gamma/R^2 = 8$.

Fig. 5 shows the calculation time against the number of vortex elements with and without the use of MDGRAPE-2. It is observed that the calculation time is reduced by a factor of 50 for $N\sim10^5$. This acceleration rate is below our expectations, but it can be improved by reducing the number of calls to the MDGRAPE-2 library for cross-product calculations. However, it is recommended that high-speed algorithms such as tree code seek further high-speed operations. The combined use of these algorithms together with MDGRAPE-2 has already been accomplished in molecular dynamics simulations, see [14–17].

The present issue is to evaluate the overall error caused by the use of MDGRAPE-2. We compared the final result of the calculation in terms of the position of the particles for the present test case. Fig. 6 represents the convection error, cf. Eq. (19). In this figure, $\delta_{\min}$, δ_{mean} and $\delta_{\max}$ stand for the minimum, average and maximum values of δ evaluated from all of the particles in the domain. It is observed that δ slightly increases for a larger number of elements. This means that MDGRAPE-2 induces certain errors for calculations with a large number of elements but the *error* stays within a finite range. This error is caused by the influence of self-induced velocities from the initial position of the particles.

The evolution of the kinetic energy of host and MDGRAPE-2 compared with Winckelmans work are shown in Fig. 7a. In the present calculations the flow is incompressible and unbounded, so there are physically no kinetic energy sources. The kinetic energy can be dissipated by both viscosity and numerical errors. From the comparison between the results obtained with the various time steps, it has been observed that there is no significant differ-

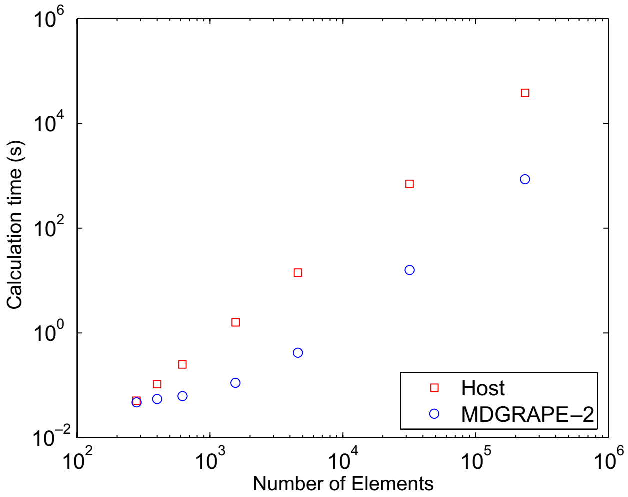

Fig. 5. Calculation time against the vortex elements. □ – without the use of MDGRAPE-2 ; ○ – with the use of MDGRAPE-2.

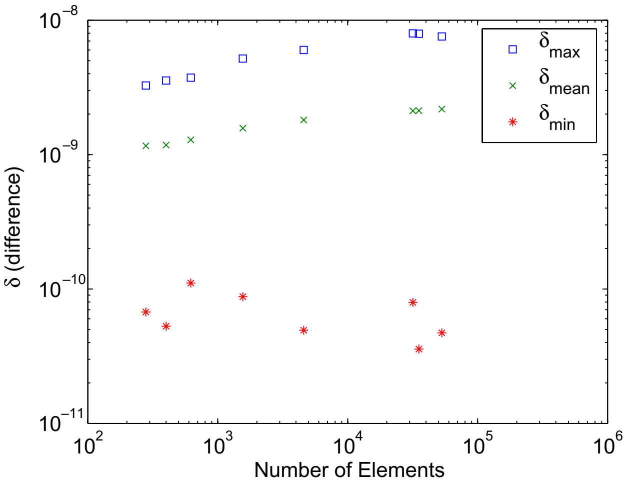

Fig. 6. Convection error for MDGRAPE-2 for different numbers of elements at the same time step. Here, □ – maximum value of all δ, × – average value of all δ, ∗ – minimum value of all δ.

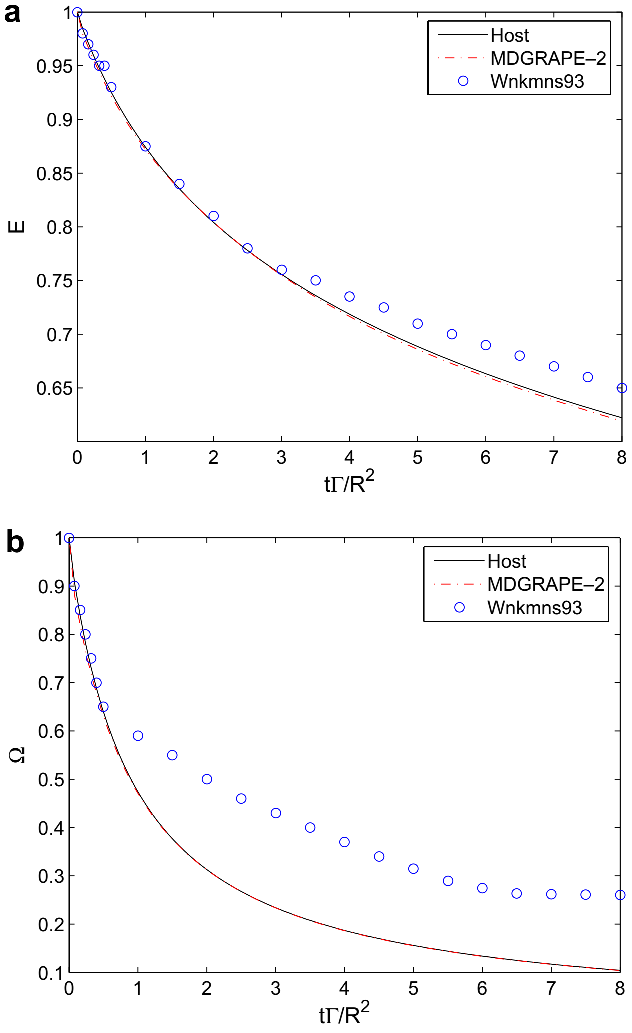

Fig. 7. Time series of kinetic energy (top) and enstrophy (bottom) compared with Winckelmans work; 'Host' – without MDGRAPE-2, 'MDGRAPE-2' – with MDGRAPE-2 and 'Wnkmns93' – Winckelmans work. (a) E – Kinetic energy and (b) Ω – enstrophy.

ence between host calculation and MDGRAPE-2. It is shown in Fig. 7a for the time step $\Delta t = 0.08$, it is easily observed that the agreement with the existing data of Winckelmans and Leonard [12] is also satisfactory.

On the other hand, the slight difference in the decay of enstrophy, Fig. 7b, is observed between the present computation and that by Winckelmans and Leonard [12], though this is due to the difference in the treatment of viscous diffusion schemes and has nothing to do with the accuracy of MDGRAPE-2. Totsuka and Obi [18] have also observed a similar tendency in the computation of two-dimensional homogeneous isotropic turbulence where the decay of enstrophy is subject to the choice of diffusion approximation. Nevertheless, the main target of this article is to discuss the issues related to the use of MDGRAPE-2 in combination with vortex method calculation and accuracy of different viscous diffusion schemes does not have any influence on the assessment and the accuracy of MDGRAPE-2. It is highly considered to use different viscous diffusion schemes for the realistic flow fields in subsequent work as a continuation of present study.

6. Conclusions

A special-purpose computer MDGRAPE-2 for N-body simulations was applied to the calculation of the vortex method. The definition of the function table range plays an essential role to achieve satisfactory accuracy in MDGRAPE-2. The improvement in speed was 50 times when compared with the calculation of a conventional PC. Although there are still certain areas that can be improved, further acceleration should be achievable with the combination of a fast algorithm.

References

[1] Barnes J, Hut P. A hierarchical $O(N\log N)$ force calculation algorithm. Nature 1986;324:446–9.

[2] Greengard L, Rokhlin V. A fast algorithm for particle simulations. J Comput Phys 1987;73:325–48.

[3] Susukita R, Ebisuzaki T, Elmegreen BG, Furusawa H, Kato K, Kawai A, et al. Hardware accelerator for molecular dynamics: MDGRAPE-2. Comput Phys Commun 2003;155:115–31.

[4] Narumi T. Special-purpose computer for molecular dynamics simulations, Doctoral Thesis, College of Arts and Sciences, University of Tokyo; 1997.

[5] Sugimoto D, Chikada Y, Makino J, Ito T, Ebisuzaki T, Umemura M. A special-purpose computer for gravitational many-body problems. Nature 1990;345:33–5.

[6] Shankar S. A new mesh-free vortex method. Ph.D. Thesis, The Florida State University; 1996, unpublished.

[7] Mansfield JR, Knio MO, Meneveau C. Dynamic LES of colliding vortex rings using a 3D vortex method. J Comput Phys 1999;152:305–45.

[8] Yatsuyanagi Y, Ebisuzaki T, Hatori T, Kato T. Filamentary magnetohydrodynamic simulation model, current vortex method. Phys Plasmas 2003;10:3181–7.

[9] Yatsuyanagi Y, Kiwamoto Y, Ebisuzaki T, Hatori T, Kato T. Simulation of diocotron instability using a special-purpose computer, MDGRAPE-2. Phys Plasmas 2003;10:3188–95.

[10] Elmegreen BG, Koch R, Yasuoka K, Furusawa H, Narumi T, Susukita R, et al. Numerical simulations of magnetic materials with MD-GRAPE: curvature induced anisotrophy. J Magn Magn Mater 2002;250:39–48.

[11] Elmegreen BG, Koch R, Schabes ME, Crawford T, Ebisuzaki T, Furusawa H, et al. Simulations of magnetic materials with MDGRAPE-2. IBM J Res Dev 2004;48:199–207.

[12] Winckelmans GS, Leonard A. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. J Comput Phys 1993;109:247–73.

[13] Leonard A. Vortex methods for flow simulations. J Comput Phys 1980;37:289–335.

[14] Makino J. Treecode with a special-purpose processor. Publ Astron Soc Jpn 1991;43:621–38.

[15] Makino J. Yet another fast multipole method without multipoles-pseudo-particle multipole method. J Comput Phys 1999;151:910–20.

[16] Kawai A, Makino J, Ebisuzaki T. Performance analysis of high-accuracy tree code based on the pseudoparticle multipole method. The Astrophys J Suppl Ser 2004;151:13–33.

[17] Chau NH, Kawai A, Ebisuzaki T. Implementation of fast multipole algorithm on special-purpose computer MDGRAPE-2. Proc 6th World Multiconf Systematics, Cybernetics Informatics SCI 2002,Orlando 2002;XVI:477–81.

[18] Totsuka Y, Obi S. Viscous dissipation model for fast vortex method in simulation of decaying turbulence. Trans Jpn Soc Mech Eng 2005;71(701):23–9 [in Japanese].