HOKKAIDO UNIVERSITY

Title	Application of Algebraic Approximation to Three Dimensional Multibody Coulomb Problem : Implementation of GPGPU
Author(s)	Oikawa, Shun-ichi; Higashi, Koichiro; Matsumoto, Yutaka
Citation	Plasma and Fusion Research, 5, S2026 https://doi.org/10.1585/pfr.5.S2026
Issue Date	2010
Doc URL	https://hdl.handle.net/2115/46963
Type	journal article
File Information	PFR5_S2026.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Application of Algebraic Approximation to Three Dimensional Multibody Coulomb Problem: Implementation of GPGPU

Shun-ichi OIKAWA, Koichiro HIGASHI and Yutaka MATSUMOTO

Graduate School of Engineering, Hokkaido University, N-13, W-8, Sapporo 060-8628, Japan

(Received 7 December 2009 / Accepted 5 March 2010)

The algebraic model (ALG) proposed by the authors has sufficiently high accuracy in calculating the motion of a test particle with all the field particles at rest. When all the field particles are moving, however, the ALG has relatively poor prediction ability on the motion of the test particle initially at rest. Nonetheless, the ALG approximation gives a good results for the statistical quantities, such as variance of velocity changes or the scattering cross section, for a sufficiently large number of Monte Carlo trials. We have implemented a graphics processing unit (GPU) using NVIDIA's CUDA architecture into the ALG scheme for Coulomb multibody problems. For N=28-body problem, the ALG calculations on the GPU is several times faster than on a typical CPU. The achieved speedup ratios on an NVIDIA GTX-285 are 10.5 and 2500 against the ALG-CPU and the DIM-CPU, respectively both on an Intel Celeron @3.06 GHz.

© *2010 The Japan Society of Plasma Science and Nuclear Fusion Research*

Keywords: multibody problem, algebraic approximation, diffusion, General Purpose Graphics Processing Unit (GPGPU)

DOI: 10.1585/pfr.5.S2026

1. Introduction

Since it is difficult to rigorously deal with multibody Coulomb collisions, the current classical theory [1,2] considers them as a series of temporally-isolated binary Coulomb collisions within the Debye sphere. We have proposed the algebraic analysis approximation (ALG) for Coulomb multibody interactions [3–5]. When all the field particles are moving, the ALG has relatively poor prediction ability [4,5] on the two-dimensional motion of the test particle initially at rest. Nonetheless, the ALG approximation gives excellent results for the statistical quantities [5], such as variance of velocity changes or the scattering cross section, for a sufficiently large number of Monte Carlo trials.

In addition, the ALG scheme conserves the angular momentum, as well as linear momntum, of the entire system. The conventional Coulomb collision models, such as the Takizuka-Abe binary collision model [6], treats Coulomb interactions only in the velocity space. These models cannot be applied to problems in which the numerical conservation of total angular momentum is crucial. One of such problems includes the spontaneous plasma rotation [7,8].

In this paper, we will report implementation of the general purpose graphics processing unit (GPGPU) using CUDA [9] architecture into our ALG scheme. CUDA is a general purpose parallel computing architecture developed by NVIDIA Corporation on several NVIDIA graphics processing units (GPUs).

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

2. Algebraic Approximation

As shown in Fig. 1 the scattering angle, $\chi \equiv \pi - 2\theta_0$, for a binary interaction is given 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 the initial relative speed at $r = \infty$ and $\theta = -\theta_0$. The angular component of the equation motion gives the well-known invariant of

$$r^2 \frac{\mathrm{d}\theta}{\mathrm{d}t} = \text{const} = bg_0, \tag{1}$$

and the radial component is given by

$$\frac{\mathrm{d}^2 r}{\mathrm{d}t^2} = \frac{g_0^2 b_0}{r^2}\left(1 + \frac{b_0}{r}\tan^2\theta_0\right). \tag{2}$$

The first term in the parentheses on the right hand side of Eq. (2) stands for the Coulomb force $F_\mathrm{C} \propto r^{-2}$. This force is much smaller for small angle scattering, i.e., $\chi \ll 1$, than the second term. This latter force scales as $F_\mathrm{a} \propto r^{-3}$ and results from the conservation of angular momentum Eq. (1). At the closest point $r_\mathrm{min} = r(\theta = 0)$ shown in Fig. 1, we have

$$\frac{b_0 \tan^2\theta_0}{r_\mathrm{min}} \simeq \frac{2}{\chi} \gg 1. \tag{3}$$

As shown above, the r-dependence of $F_\mathrm{a} \propto r^{-3}$ is steeper than that of $F_\mathrm{c} \propto r^{-2}$, the change in momentum $\boldsymbol{p} \equiv \mu\boldsymbol{g}$ is almost due solely to F_a near $r = r_\mathrm{min}$. As a consequence, the exact hyperbolic (or occasionally elliptic) trajectory for the particle can be approximated as a broken line with an impulse force of

$$\mu\Delta\boldsymbol{g} = 2\mu g_0 \cos\theta_0 \boldsymbol{e}_x = 2\mu g_0 \sin\frac{\chi}{2}\boldsymbol{e}_x, \tag{4}$$

© 2010 The Japan Society of Plasma Science and Nuclear Fusion Research

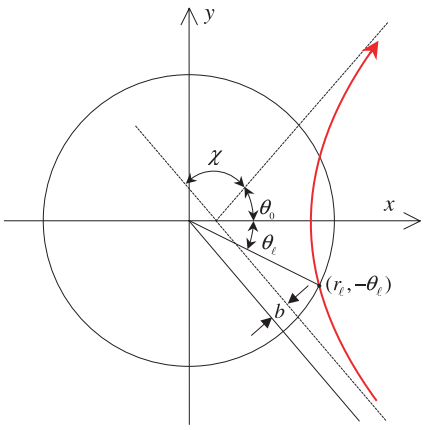

Fig. 1 Unperturbed relative trajectory $r = r(\theta)$ in an orbital plane for the repulsive force. The scattering center is at the origin. An impact parameter is $b = b_0 \tan\theta_0$.

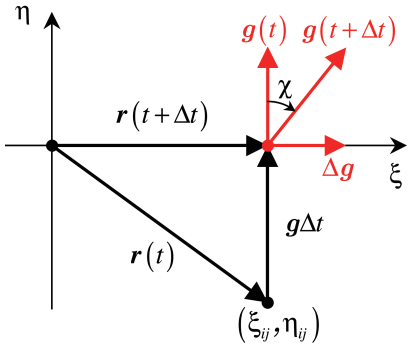

Fig. 2 Coordinate transform from the (x, y, z) to (ξ, η), in such a way that the third axis, say ζ, is set parallel to the angular momentum $\boldsymbol{L} = \boldsymbol{r} \times \mu \boldsymbol{g} = L\boldsymbol{e}_\zeta$, and $\boldsymbol{g} = g\boldsymbol{e}_\eta$. In this coordinate system, the scattering angle χ, i.e., the impact parameter b and the time of the interaction Δt are approximately given by ξ and η, respectively. The relative velocity at time t and $t+\Delta t$, are $\boldsymbol{g}(t)$, and $\boldsymbol{g}(t+\Delta t) = \boldsymbol{g}(t) + \Delta\boldsymbol{g}$, respectively.

near the closest point [5]. With this in mind, we have approximated a multibody problem to a series of binary deflections near their closest point.

The binary Coulomb and gravitational interaction conserves the relative angular momentum $\boldsymbol{L}$, and the motion is in the so-called orbital plane perpendicular to $\boldsymbol{L}$. The application of the ALG method, therefore, to three dimensional systems is straightforward.

2.1 Coordinate transformation

In order to apply the above binary interaction approximation (ALG) to three-dimensional multibody cases, first we seek for a field particle that gives the test particle an impulse force *at the earliest time*. For this purpose, it is convenient to transform the coordinate system from the original three dimensional $(x,\ y,\ z)$ coordinates to a two dimensional $(\xi,\ \eta)$ ones, in such a way that the third axis, say ζ, is set parallel to the angular momentum $\boldsymbol{L} = \boldsymbol{r} \times \mu\boldsymbol{g} \equiv L\boldsymbol{e}_\zeta$, and the relative velocity $\boldsymbol{g} = \left(g_\xi,\ g_\eta\right) = (0, g) = g\boldsymbol{e}_\eta$, as shown in Fig. 2.

Then the relative position $\boldsymbol{r} \equiv \boldsymbol{r}_i - \boldsymbol{r}_j$ between particle-i and particle-j has a ξ-, an η-, and a ζ-coordinates of

$$\xi_{ij} = \boldsymbol{r} \cdot \boldsymbol{e}_\xi, \quad \eta_{ij} = \frac{\boldsymbol{r} \cdot \boldsymbol{g}}{g}, \quad \zeta_{ij} = 0, \tag{5}$$

where the unit vector $\boldsymbol{e}_\xi$ is defined as $\boldsymbol{e}_\xi \equiv \boldsymbol{e}_\zeta \times \boldsymbol{e}_\eta = \boldsymbol{L}/L \times \boldsymbol{g}/g$. When $\eta_{ij} < 0$ the two particles are approaching, while they are moving away for $\eta_{ij} > 0$.

The particle moves in the direction parallel to the η-axis with a constant velocity of g, and is to interact at $\left(\xi_{ij}, 0\right)$ with this field particle in a time interval of

$$\Delta t_{ij} = -\frac{\eta_{ij}}{g}. \tag{6}$$

Accordingly, the field particle that the test particle is given an impulse force at the earliest time has the smallest positive Δt_{ij}, i.e.,

$$\Delta t_{\min} \equiv \min\left(\max\left(0, -\frac{\eta_{ij}}{g}\right)\right), \text{ for } 1 \le i, j \le N. \tag{7}$$

We have ignored the effect of field particles with $\eta_{ij} > 0$, since the interaction is completed at $\eta = 0$ in our approximation. In other words, such field particles have already interacted with the test particle in the past.

When the test particle moves to the position of $\left(\xi_{ij}, 0\right)$, it changes the relative velocity by $\Delta\boldsymbol{g}_{ij}$ as

$$\Delta\boldsymbol{g}_{ij} = 2\, g \sin\frac{\chi_{ij}}{2}\, \boldsymbol{e}_\xi, \tag{8}$$

$$\frac{\chi_{ij}}{2} \simeq \arctan\frac{b_0}{\xi_{ij}}, \tag{9}$$

where the pair i and j satisfies Eq. (7), and we have approximated that the impact parameter is given by $b = \xi_{ij}$ in Eq. (4) as shown in Fig. 2. Thus, in the (ξ, η) coordinate system, the field particle position ξ_{ij} and η_{ij} correspond to the velocity change $\Delta\boldsymbol{g}_{ij}$ and the time of the interaction Δt_{ij}, respectively.

2.2 Changes in position and momentum

In an N-body problem, changes in the position of each particle-I, $(1 \le I \le N)$, during the time interval Δt_{ij} for a interacting particle pair (i, j), is given by

$$\Delta\boldsymbol{r}_I = \boldsymbol{v}_I \Delta t_{ij}, \tag{10}$$

irrespective of whether they are *interacting*, $I \in \{i, j\}$, or not, since they are assumed to be moving straight in the ALG approximation.

On the other hand, changes in linear momenta of the interacting particle pair (i, j), during $\Delta t_{ij} = \Delta t_{\min}$, are given by

$$\Delta\boldsymbol{p}_i \equiv m_i \Delta\boldsymbol{v}_i = +\mu_{ij} \Delta\boldsymbol{g}_{ij}, \tag{11}$$

$$\Delta\boldsymbol{p}_j \equiv m_j \Delta\boldsymbol{v}_j = -\mu_{ij} \Delta\boldsymbol{g}_{ij}, \tag{12}$$

since $\mu_{ji} = \mu_{ij}$ and $\Delta\boldsymbol{g}_{ji} = -\Delta\boldsymbol{g}_{ij}$ by definition. Therefore the ALG approximation conserves the binary momentum,

Table 1 Relative errors in invariants with the ALG for the number of particles N = 28, 1332, and 9292.

# of particles	Relative errors in		
N	$\boldsymbol{P}$	$\boldsymbol{L}$	E
28	3×10^{-16}	4×10^{-16}	2×10^{-7}
1,332	4×10^{-16}	3×10^{-14}	7×10^{-7}
9,262	3×10^{-15}	2×10^{-13}	4×10^{-7}

$\boldsymbol{p}_{ij} \equiv \boldsymbol{p}_i + \boldsymbol{p}_j$, since, from Eqs. (11) and (12),

$$\Delta\boldsymbol{p}_{ij} = \Delta\boldsymbol{p}_i + \Delta\boldsymbol{p}_j = \boldsymbol{0}, \tag{13}$$

in the system in principle. Furthermore, the interaction between i and j conserves the binary angular momentum, $\boldsymbol{L}_{ij} \equiv \boldsymbol{r}_{ij} \times \boldsymbol{p}_{ij}$, as

$$\Delta\boldsymbol{L}_{ij} \equiv \Delta\boldsymbol{r}_{ij} \times \boldsymbol{p}_{ij} + \left(\boldsymbol{r}_{ij} + \Delta\boldsymbol{r}_{ij}\right) \times \Delta\boldsymbol{p}_{ij} = \boldsymbol{0}, \tag{14}$$

since both cross products are those between two vectors with the same direction, as shown in Fig. 2.

Any other particle-k ($k \neq i, j$) does not change its velocity, i.e.,

$$\Delta\boldsymbol{p}_k = m_k\Delta\boldsymbol{v}_k = \boldsymbol{0}, \tag{15}$$

$$\Delta\boldsymbol{L}_k = \boldsymbol{v}_k\Delta t_{ij} \times \boldsymbol{p}_k = \boldsymbol{0}, \tag{16}$$

since they are not interacting during the interval of Δt_{ij}.

We have, thus far, shown above that the *binary* interaction between particles i and j conserves linear and angular momenta, i.e., $\boldsymbol{p}_{ij}\left(t + \Delta t_{ij}\right) = \boldsymbol{p}_{ij}(t)$ and $\boldsymbol{L}_{ij}\left(t + \Delta t_{ij}\right) = \boldsymbol{L}_{ij}(t)$. It is easy to show the total liner momentum and total angular momentum of the entire system are conserved with our model. Table. 1 lists the relative errors in invariants with the ALG for the number of particles N = 28, 1332, and 9292. The invariants include the linear momentum $\boldsymbol{P} = \sum_{i=1}^{N} m_i\boldsymbol{v}_i$, the angular momentum $\boldsymbol{L} = \sum_{i=1}^{N} \boldsymbol{r}_i \times m_i\boldsymbol{v}_i$, and the total energy $E = \sum_{i=1}^{N} m_i\boldsymbol{v}_i^2/2 + \sum_i\sum_j e^2/4\pi\varepsilon_0 r_{ij}$ of the entire system. The relative errors in the total momenta $\boldsymbol{P}$ and $\boldsymbol{L}$ are due to the finite-bit calculation on the computer. The relative errors in total energy E, of order of 10^{-6} are large compared with those of the total momenta, since most particles move freely ignoring the interparticle forces in our model.

3. Test Calculation

In the following test calculations, protons in a hydrogen plasma with temperature of T = 10 keV and number density of $n = 10^{20}\,\mathrm{m}^{-3}$ will be analyzed.

3.1 Convergence with N

Figure 3 shows the N-dependence on changes in position $\Delta\boldsymbol{r}$ and in velocity $\Delta\boldsymbol{v}$ of a test particle. From the figure, changes in position converge at around $N \sim 700$, and those in velocity are almost independent from N. Therefore, in order to make statistical calculations concerning velocity changes, such as the scattering cross section, the number of particles N needs not be large; N = 28 will be adopted in the next section.

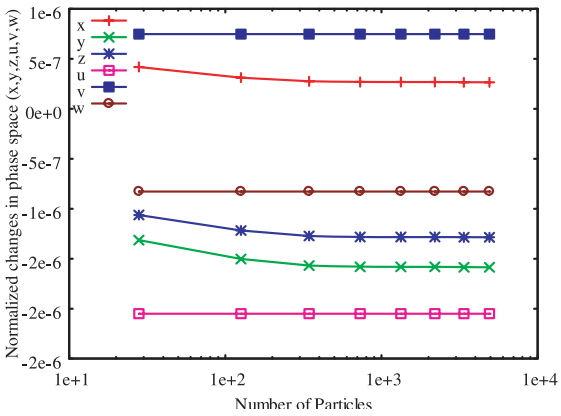

Fig. 3 Convergence of the ALG with the number of particles N: Changes in position and in velocity of a test particle.

3.2 Variance in velocity changes

Figure 4 shows the variance of change in velocity,

$$\left\langle\Delta\boldsymbol{v}^2\right\rangle \equiv \frac{1}{N_{\mathrm{MC}}}\sum_{I_{\mathrm{MC}}=1}^{N_{\mathrm{MC}}}\frac{1}{N}\sum_{i=1}^{N}(\Delta\boldsymbol{v}_i)^2\,, \tag{17}$$

of *all* the particles for three-dimensional Coulomb N =28-body problem, where N_{MC} stands for the number of Monte Carlo trials. With the direct integration method, DIM, calculation ends up to the number of Monte Carlo trials of $N_{\mathrm{MC}} = 10^8$, since it needs much more CPU times than the ALG. The agreement between the ALG and the DIM is excellent. As the DIM in this study, we have used the six-stage fifth-order Runge-Kutta-Fehlberg method known as the RKF65 [10,11] with the absolute numerical error tolerance of 10^{-16}. Several jumps seen in Fig. 4, such as those at $N_{\mathrm{MC}} \sim 7\times10^6$, 9×10^7, and 3×10^8 are due to close encounters, i.e., large angle scattering with $\Delta v_{\mathrm{i}}/g_{\mathrm{th}} \sim \mathrm{O}(1)$. If a head-on collision of $\Delta g_{ij} = 2\,g \sim 2g_{\mathrm{th}}$ between the pair (i, j) occurred during the next, i.e., the $5\times10^{10}+1$st, Monte Carlo trial in this 28-body system, it alone could raise the dimensional variance by

$$\frac{2\times g_{\mathrm{th}}^2}{28\times(5\times10^{10}+1)} \sim 0.14\times10^{-11}\,g_{\mathrm{th}}^2. \tag{18}$$

Thus the variance by using the ALG, at the number of Monte Carlo trials $N_{\mathrm{MC}} = 5\times10^{10}$, can be expressed as

$$\left\langle\left(\Delta\boldsymbol{v}_{\mathrm{ALG}}\right)^2\right\rangle \sim \left(1.88\pm0.14\right)\times10^{-11}g_{\mathrm{th}}^2. \tag{19}$$

This calculation for variance in velocity changes corresponds to a situation, in which the motion of a test proton in the plasma is analyzed during a time interval of $N\times N_{\mathrm{MC}}\times\Delta t = N\times N_{\mathrm{MC}}\times n^{-1/3}/g_{\mathrm{th}} \sim 140\,\mathrm{ms}$.

The CPU time taken for one Monte Carlo trial of the DIM is 31 ms and that of the ALG is 0.13 ms on a computer with an Intel® Celeron® D 346@3.06 GHz. That is, the ALG achieves 263 times speed up for N = 28 body problems.

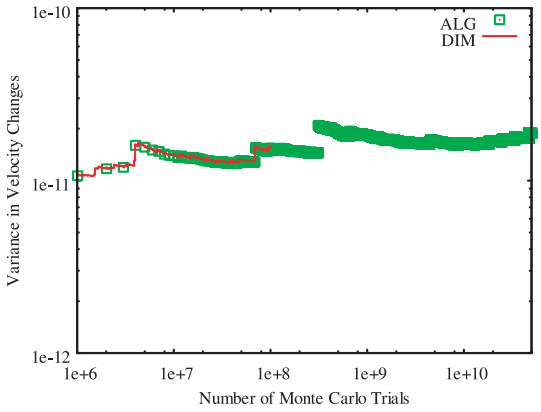

Fig. 4 Normalized variance of velocity changes for $N = 28$ by using the ALG with $N_{\rm MC} = 5 \times 10^{10}$, while up to $N_{\rm MC} = 10^8$ for the DIM.

Table 2 CPU times for $N = 28$ with and without GPU.

method	device	CPU time [ms]	ratio
DIM	CPU	31.0	2500
ALG	CPU	0.13	10.5
ALG	GPU	0.0124	1

The most CPU-time-consuming part of the ALG scheme is finding $\Delta t_{\rm min}$ in Eq. (7), which can be executed in parallel on GPUs. As shown in Table 2, CPU/GPU time for $N = 28$ by using the ALG with $N_{\rm MC} = 5 \times 10^{10}$ is 180 hr (6.4×10^5 sec) on a GPU; NVIDIA GTX-285 with the RAM of 1 GB, the number of cores of 240 at the clock rate of 1.48 GHz. The average calculation time on the GPU is around 0.0124 ms per one ALG calculation, so that the speedup ratios of the GPU calculation are 10.5 and 2500 against the ALG-CPU and the DIM-CPU, respectively.

3.3 CPU time with and without GPU

Figure 5 shows the CPU time for the ALG and the DIM. The number of particles dependence of the CPU time for the ALG approximation, $\tau^{\rm CPU}_{\rm ALG}$, is in proportion to $N^{3.02}$, as

$$\tau^{\rm CPU}_{\rm ALG} \sim 5.56 \times 10^{-9} N^{3.02}. \quad (20)$$

Similarly, $\tau^{\rm CPU}_{\rm DIM}$ for the DIM is:

$$\tau^{\rm CPU}_{\rm DIM} \sim 3.06 \times 10^{-5} N^{2.34}, \quad (21)$$

from which the ALG is suited for multibody problems for moderate number of particles of $N \lesssim 3 \times 10^5$. Also depicted in Fig. 5 is the calculation time, $\tau^{\rm GPU}_{\rm ALG}$, on a GPU, specifically NVIDIA GTX-295, which scales as

$$\tau^{\rm GPU}_{\rm ALG} \sim 1.88 \times 10^{-9} N^{2.75}. \quad (22)$$

The ALG on the GPU is suited for multibody problems for huge numbers of particles of $N \lesssim 2 \times 10^{10}$ which is several orders of magnitudes larger than the total number of particles in the Debye sphere in typical fusion plasmas.

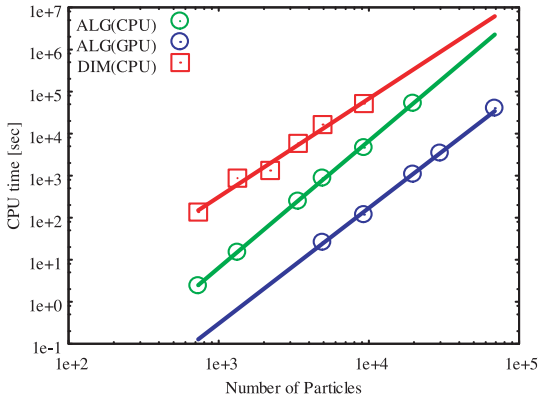

Fig. 5 CPU time vs the number of particles N with fitting lines, $\tau^{\rm CPU}_{\rm ALG} \sim 5.56 \times 10^{-9} N^{3.02}$ for the ALG on a CPU in green, and $\tau^{\rm CPU}_{\rm DIM} \sim 3.06 \times 10^{-5} N^{2.34}$ for the DIM on the CPU in red. Also depicted are CPU times for ALG-GPU calculations with a fitting line $\tau^{\rm GPU}_{\rm ALG} \sim 1.88 \times 10^{-9} N^{2.75}$ in bule.

4. Conclusion

We have implemented a graphics processing unit (GPU) using NVIDIA's CUDA architecture into the ALG scheme for Coulomb multibody problems. For N=28-body problem, the ALG calculations on a GPU is several times faster than on a typical CPU. The achieved speedup ratios on NVIDIA GTX-285 are 10.5 and 2500 against the ALG-on-CPU and the DIM-on-CPU, respectively both on an Intel Celeron @3.06 GHz.

Acknowledgement

The authors would like to thank Dr. A. Wakasa and Prof. M. Itagaki for their fruitful discussions on the subject. The author would also acknowledge the continuous encouragement of the late Prof. T. Yamashina. Part of the GPU/CUDA coding was done by Mr. R. Ueda. This research was partially supported by a Grant-in-Aid for Scientific Research (C), 21560061.

[1] L. Spitzer, Jr., *Physics of Fully Ionized Gases* (Interscience Publishers, New York, 1959) p. 120.
[2] P. Helander and D.J. Sigmar, *Collisional transport in Magnetized Plasmas.*, (Cambridge U. P., Cambridge, 2002).
[3] S. Oikawa and H. Funasaka, Plasma Fusion Res. **3**, S1073 (2008).
[4] K. Higashi et al., in *Proc. 18th Int. Toki Conf.* (ITC18), P2-58, Toki (2008).
[5] S. Oikawa, K. Higashi, and H. Funasaka, Plasma Fusion Res. **5**, S1048 (2010).
[6] T. Takizuka and H. Abe, J. Comput. Phys. **25**, 205 (1977).
[7] K. Ida *et al.*, Phys. Rev. Lett. **65**, 1364 (1990).
[8] W. D. Lee *et al.*, Phys. Rev. Lett. **91**, 205003 (2003).
[9] `http://www.nvidia.com`.
[10] 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).
[11] E. Fehlberg, "Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Warmeleitungsprobleme", *Computing* (Arch. Elektron. Rechnen), **6**, 61 (1970).