Application of Graphics Processing Units to Droplet Collision Simulation between Diesel Sprays

Daisuke Tsuru *1; Hiroshi Tajima *2

*1: Interdisciplinary Graduate School of Engineering Sciences (IGSES), Kyushu University, Kasuga-koen 6-1, Kasuga, Fukuoka 816-8580, JAPAN, E-mail dtsuru@ence.kyushu-u.ac.jp, Phone +81-92-583-7893
*2: Interdisciplinary Graduate School of Engineering Sciences (IGSES), Kyushu University, Kasuga-koen 6-1, Kasuga, Fukuoka 816-8580, JAPAN, E-mail tasima@ence.kyushu-u.ac.jp, Phone +81-92-583-7592

Simulation of the merging process of two diesel sprays which is a time-consuming task was accelerated using the so-called General-Purpose computing on a GPU (Graphics Processing Unit). The calculation bottleneck was identified as the deterministic judgment model of droplet collision, which seems imperative for a complicated injection system such as that of a Direct Water Injection (DWI) system. The calculation algorithm of the judgment model was carefully modified from that designed for a conventional CPU (Central Processing Unit), and the acceleration was so evident in certain cases that total calculation time was reduced down to nearly one-fourth of the equivalent CPU calculation.

1. Introduction

As is well known, IMO (International Maritime Organization) set the stringent target in their tier III regulations against NOx emissions from ships. They are required to reduce NOx emission as much as that of 75% from the present tier II level in the designated waters called ECA (Emission Control Area). This level of NOx reduction is usually considered that it is too difficult to achieve with any single potent reduction measure such as SCR (Selectice Catalytic Reduction) or EGR (Exhaust Gas Recirculation). Therefore the introduction of the secondary NOx abatement measure has been earnestly studied[1]. DWI (Direct Water Injection) is one of the so-called wet technologies which utilize water for lowering combustion temperature and is typified by FWE (Fuel Water Emulsion) and HAM (Humid Air Motor). The DWI system in practical use has an injector with two built-in needles that control the injection of fuel and water separately. These needles are located close to each other and both of the immiscible fluids are injected almost coaxially. These kinds of closely-aligned nozzle holes are also utilized in a colliding fan spray nozzle[2] and a group-hole nozzle[3] in order to promote spray dispersion and atomization. If the injections from such close-set holes are overlapped both temporally and spatially, injection amounts, periods, directions, locations, and the others. could be controllable factors.

It is easily expected that the factors and their degree of freedom become too large to be determined experimentally so that the numerical optimization of them should be absolutely necessary. In the preceding paper by the authors[4,5], it was indicated that the immiscible spray merging behavior in the DWI system can be numerically reproduced with reasonable accuracy through introduction of a four-criteria collision outcome model between the droplets of fuel and water. In the process of the model validation, it was revealed that the calculation cost is no ignorable for the droplet collision prediction when the numerical spray develops and the number of the droplet parcels get considerably larger in a DDM (Discrete Droplet Model) which is a standard spray expression for saving computation cost in the diesel CFD codes based on so-called LDEF (Lagrangian Droplet Eulerian Fluid) frame works. The situation could be fatal when applying a deterministic discriminant model to the droplet collision since the job time increases more than a hundred times of a conventional CFD run without the droplet collision discrimination and the reaction schemes after the four collision outcomes. This would be an obstruction of the practical DWI simulation in the case of the large marine diesels.

The present study deals with the GPU (Graphics Processing Unit) application for the first time to accelerate the deterministic collision discriminating with calculation space of larger dimensions in mind. A merging process of two identical fuel sprays was chosen to focus on the accelerating effects of GPU on the collision discriminant routines and the points of attention were also investigated for GPU application to the engine simulation.

Translated from Journal of the JIME Vol.49, No.6 ©2014 (Original Japanese)
2. General Outline of Parallel Computation Using GPU

2.1 Target in-cylinder phenomena for GPU application

GPGPU (General Purpose computing on GPU) has been recently introduced as an affordable and effective acceleration method for various kinds of scientific parallel computation on PCs (personal computers) since every PC surely includes one or more GPUs as standard. It is supposed that the computing speed of GPGPU may reach up to dozens of times the speed of the conventional CPUs (Central Processing Units)\(^{6-9}\).

However, it should be pointed out that the acceleration by GPGPU does not always come instantly. It is important to examine in advance how effective the GPU application is against a certain bottleneck of the computational traffic and how suitable the program flow is for the GPU oriented modification around the bottleneck.

In the present study, a merging process of two identical fuel sprays was chosen as a simulation target because the process is known to be CPU-power-consuming especially when a deterministic model is applied in discriminating the droplet collision outcomes. With optimized arithmetic procedures for GPUs, the effectiveness of GPGPU was examined through the acceleration of the discrimination routines.

The CFD code; KIVA-3V was adopted as a platform because it is one of the most popular simulation tools for in-cylinder phenomena relating diesel sprays. The collision discriminant routines are re-coded for GPU operation while the other parts are still executed in conventional CPU operation.

2.2 Internal architecture of GPU

Figure 1 shows a typical internal architecture of a GPU chip. Dozens of Streaming Multi-processors (SMs) are built in and each SM has several Streaming Processors (SPs or Cores) inside, which execute actual calculation. GPUs are slower than CPUs in respect of the clock frequency of these SPs but GPUs can outperform CPUs thanks to the parallel computation operated by some hundreds of SPs in total so that it means the accelerating effect will be easily impaired with branching after conditional statements or frequent access to external memories. Moreover, operations requiring double-precision floating point (DPFP) number may slow down the computing speed unless the GPU is compatible with such operations.

Coming back to memories that GPUs use, they are classified into “Shared” one which is shared only by the SPs in the same SM and “Global” one implemented in the outside of the GPU chip. Global memory corresponds to a video memory mounted on a graphics card and can be referred by all the SPs inevitably with longer access time than Shared memory. GPUs should communicate via a PCI bus with main memories under CPUs’ control because GPGPU calculations are not completed by GPUs alone. This channel often can be a bottleneck of the GPGPU acceleration.

Therefore, a series of the parallel calculation by a GPU should keep the number of operations and the amount of store data to the extent able to handle for the Shared memory. Specifically, the SPs in a certain SM cooperate as a unit of arithmetic operator and they execute an instruction group (Thread) containing \(10^3\)~\(10^4\) simple operations. Deceleration instead of
acceleration has been reported in the calculation execution with a too short Thread of $10^7$ simple operations. All in all, to maximize the acceleration effect by GPGPU, it is a must to rewrite a target operation process from CPU-oriented one to GPU-optimized one.

2.3 Specifications of tested GPU

Table 1 lists the main specifications of the tested GPU; core GT206 on the graphics board; GeForce GTX295 from NVIDIA Co., US. Two GT206s are mounted on the GeForce GTX295 but only one of the two GT206s is used for the GPGPU of the present study. Each SM in GT206 can execute one calculation of one DPFP number.

As previously mentioned, a subroutine for discriminating the droplet collisions inside two merging diesel sprays was converted into a GPU-oriented Thread to examine its acceleration effect. Coding of the Thread was done in the GPU development language; “CUDA” provided by NVIDIA Co. Although the reduction of calculation speed would be a trade-off, all the variables in the Thread were described as DPFP type considering the compatibility with all the other variables defined in the subroutines that CPU executes. One may point to another effective acceleration method based on the parallel CPU computation typified by MPI or Open MP, since the method is easier to apply than GPGPU. However, a single PC can offer at most a few dozen “threads” (in this case corresponding to the number of SPs of a GPU chip) so that GPGPU would be better especially in accelerating the routines in which the objects to calculate are huge in number and yet each computing process consists of simple operations such as the discrimination of the droplet collision. As a matter of fact, the authors once rewrote the discriminating routine to optimize it for the Open MP and recompiled the other KIVA routines using its auto-parallelizing option but the acceleration were too low to be noteworthy in simulating a developing process of fuel spray with the after-mentioned deterministic model for droplet collision discrimination.


3.1 Discrimination model of droplet collision

A collision discrimination model of spray droplets is naturally needed ahead of the prediction of the collision behavior of the droplets. The model gives the probability of the collision occurrence between the two focused droplets during the current time step of the CFD run. These two droplets should be representative ones from parcels in the case of a DDM system. The collision discrimination model is generally classified into a stochastic model of a similarity to a kinetic theory of molecules and a deterministic model of a similarity to ballistic analysis. O’Rourke model for the former model and Nordin model for the latter are both popular in diesel sprays.

O’Rourke model has been commonly used in CFDs for the engine simulation. It assumes a constant collision probability both in time and in space inside a control cell where two different droplets (of the two parcels) exist together. This means the droplet collision is a stochastic process following a Poisson distribution, and the probability; $p_{ok}$, with which one collision occurs during the current time step is given as follows.
\begin{equation}
    p_{ork} = \lambda \exp(-\lambda) \quad , \quad \lambda = \left| U_1 - U_2 \right| \pi \left( r_1 + r_2 \right)^2 \frac{n_2}{V_{cell}} \Delta t
\end{equation}

\( \lambda \) is a Poisson distribution parameter (expectation value), subscripts 1 and 2 are to distinguish the relevant two parcels from collision discrimination. \( U \) is for droplet speed, \( \Delta t \) for a time step, \( n \) for the number of droplets in the parcel (assuming here \( n_1 \geq n_2 \)), and \( V_{cell} \) for the volume of the cell including the two parcels, respectively. The fright direction of the droplet has no effect on the collision discrimination and resulting droplet direction after the collision is randomly given from uniform random numbers. Supposing the collision discrimination of O’Rourke model simply depends on whether the parcel on focus is in the calculated cell or not, which raises so-called mesh dependency of the model.

By contrast, Nordin model is free from the mesh dependency since it computes the collision probability; \( p_{nd} \) within the current time step purely based on the flight trajectories of any two droplets (or parcels) and it does not need the two droplets on focus to be in the same calculation cell. Its collision probability; \( p_{nd} \) is given as below.

\begin{equation}
    U_{a12} = (U_1 - U_2) \cdot \left( \frac{x_2 - x_1}{r_2} \right) \geq 0 \quad , \quad U_{a12} \Delta t > \left| x_2 - x_1 \right| - \left( r_1 + r_2 \right)
\end{equation}

Figure 2 helps understanding of the symbols in equation (2) and their meaning. The equation decides whether the collision happens or not first. \( x \) is for a position vector of the parcel, \( U_{a12} \) for an approaching velocity component between the droplets. Collision will happen under the condition that \( U_{a12} \) is positive and large enough for the droplets to reach to a contact point within a current time step.

If \( a_{12} \) can be introduced to represent the distance to the point where the droplets contact to each other along with \( U_{a12} \) vector. So, the collision probability; \( p_{nd} \) is given as equation (3) with a constant; \( C \).

\begin{equation}
    p_{nd} = \min \left( 1, \frac{C\left( r_1 + r_2 \right)}{L_{a12}} \right)
\end{equation}

In Nordin model, different from O’Rourke model, it is a single droplet that represents the whole parcel to which the droplet belongs. Therefore, the number of droplets in the parcel does not affect the collision probability. This leads the tendency that Nordin model predicts much smaller number of collision events than O’Rourke model. Taking into account of practicality of the simulation in the present study, Nordin model were modified as illustrated in Fig.2 and equation (4). Representing droplet is changed to a virtual sphere of radius \( r' \) which closely packs all the droplets of a certain parcel.

\begin{equation}
    r_1' = n_1^{1/3} r_1 \quad , \quad r_2' = n_2^{1/3} r_2
\end{equation}

As would be expected, it is better to apply this “modified” Nordin model to predict the droplet collisions happening in the injection systems such as the DWI of two merging sprays. This is because the model is near free from mesh dependency and gives more reasonable collision probability. However, Nordin model has innate weakess in its ultra-high computation load regardless of the modification compared with O’Rourke model. Nordin model normally evaluates the collision probability between any couple of all the droplet parcels and it includes conditional sentences and vector calculations in the discrimination process. On the other hand, O’Rourke model only considers the group of parcels coexisting in the same computational cell and can give the collision probability by simple arithmetic.

![Figure 2](image_url)
3.2 GPGPU conversion of droplet collision

At the GPGPU conversion of the collision discrimination on modified Nordin model, it is necessary to condense the processing data and the operations into one thread properly. In this study, the attribute values of the representative droplet of a relevant parcel are passed on to each SP and the collision discrimination with the other parcel droplets is described so as to be completed within a single thread. The computational flow is as follows.

Computations in KIVA-3V are controlled by CPUs except for the droplet collision discrimination subroutine. Once the subroutine is called, the necessary information is transferred to the shared memory of GPUs. The information includes the velocity, position and radius of a representative droplet and the number of droplets in the parcel.

After finishing the discrimination for all the representative droplets, the colliding pairs are extracted and returned to CPUs’ memory and CPUs update each velocity of the relevant droplets based on a classification model for droplet collision outcomes.

Figure 3 exemplifies the discrimination flow in parallel computation for a GPU compared with the one for a CPU. For simplification, the system of eight representative droplets is picked up in the figure. The actual number of parcels (representative droplets) could reach to $10^4$ in total even within a framework of DDM.

The CPU case is as described below. At first, the representative droplet reference number “$i$” is set to its initial value; $i = 1$, then the collision occurrence is determined between all the other droplets referred “$j$” and $j \geq 2$ (or $j > i$). Next, the reference number is sequentially increased and the similar determination follows. When the number of parcels is set to $N_p$ and the reference droplet could have multiple collisions during the time step, these discrimination processes should be repeated $N_p \times (N_p - 1) / 2$ times to be completed.

On the other hand, in the GPU case, the SP number is equivalent of the droplet number. The calculation cycle starts after the required information of the corresponding droplet and the dedicated thread for the discrimination are assigned to each SP. Although the total number of the calculation is the same with the CPU case, all the SPs work in parallel so that the average number of reputations can be down to $(N_p - 1) / 2$ and the calculation time would be ideally reduced in inverse proportion to the number of SPs.

![Figure 3 Algorism comparison between CPU / GPU calculation for droplet collision discrimination](image)

Translated from Journal of the JIME Vol.49, No.6  ©2014(Original Japanese)
3.3 Classification model of binary droplet collision

Reproduction of the droplet behavior after the collision needs formulation of a binary drop collision system and a classification model of the outcomes of the binary collision.

The outcomes of the binary droplet collision is typically classified on a diagram between an impact parameter ($\chi$) and a Weber number ($We$) defined on the relative velocity between relevant two droplets. “$\chi$” is a non-dimensional distance between the droplet center lines, both of which are parallel to the relative velocity. Normalization is done based on the sum of the droplet diameters. The present study embraces the four category model by Qian and Law [13]. They conducted binary collision experiments between two fuel droplets of identical diameter under pressurized conditions, plotted the results on the $\chi$-We diagram, and classified them into four criteria; (I) bounce, (II) coalescence, (III) stretching separation, and (IV) reflexive separation respectively.

At the application of the model, borderlines on the $\chi$-We diagram separating the collision outcomes are properly formulated and the additional equations are newly implemented in KIVA-3V code to give the droplet momentum after the binary collision.

Unlike the deterministic binary collision system, DDM takes the view that the collision numerically takes place among the two groups of droplets of the relevant parcels and neither droplet position nor droplet direction at the impact moment is knowable. According with this DDM concept, the impact parameter at the collision instant is randomly given from uniform random number. In spite of its deterministic feature, Nordin model adopted in this study is no exception on this point.

4. Measurement and calculation conditions

4.1 Observation apparatuses for merging process of sprays

Figure 4 depicts an outline of a constant-volume visual combustion chamber (VCC) and optics for merging spray visualization. The chamber has the inside dimension of $\phi$150 mm in diameter and 270 mm in height and two opposing viewing field of 120 mm in height and 60 mm in width. Two single-hole injectors are closely mounted on the vessel’s top lid so as to enable their injection direction to intersect with each other. By overlapping their injection periods, spray merging and inter-spray droplet collision are induced. Sprays were visualized based on the back-diffused illumination technique using a scattering plate as a surface light source.

Table 2 lists up the injection conditions and the in-cylinder gas conditions. The nozzle tips are 7 mm apart from each other horizontally and the intersecting angle of the two sprays are set to 50°. The chamber is filled with pure nitrogen at room temperature, which means the measurements were done under the conditions without combustion and with negligible evaporation of spray droplets.

![Figure 4 Outline of VCC and optics for visualization of spray merging process](image-url)
4.2 Computational space and mesh layout

Figure 5 illustrates the computational space and the mesh layout dividing it to simulate the sprays merging in the VCC by GPGPU. As for the computational space, the space is given a cubic shape of 200 mm on each side and results in rectangular meshing all around, although the VCC chamber has a cylindrical shape actually. The reasons are as follows. A computational space and meshing based on a cylindrical coordinate is likely to bring heavy mesh dependency because the merged trajectory of the two sprays will be close to the center line of the cylindrical mesh. In case of larger intersecting angle between the sprays and less inter-spray droplet collision probability, fuel droplets could impact on the cylinder wall, which may complicate the situation. Finally, it is not important in reproducing the in-cylinder pressure of the VCC to keep the real chamber volume by the aforementioned non-combustion conditions.

As for the mesh layout, its horizontal pitch is adjusted to be proportional to the distance from the vertical center line of the computational space and the vertical pitch extends proportionally to the depth from the upper surface. These are intended to compromise the prediction accuracy with the calculation time by applying finer meshes in the vicinity of the nozzle holes and the route of developing sprays. The finest mesh pitch around the centerline was about 1 mm and the total number of the computational cells reached near 70,000 respectively in the present study.

The numerical injection conditions follow the VCC’s ones and the number of droplet parcels for the injection mass are initially set to 3,000 which is rather normal in KIVA codes. The number of parcels varied from 4,000 to 8,000 when its influence on the calculation load is investigated.

<table>
<thead>
<tr>
<th>Injection conditions</th>
<th>Main</th>
<th>Sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hole diameter×no.</td>
<td>φ0.23 × 1</td>
<td>φ0.23 × 1</td>
</tr>
<tr>
<td>Depression angle</td>
<td>245</td>
<td>-65</td>
</tr>
<tr>
<td>Injection pressure</td>
<td>80</td>
<td>40</td>
</tr>
<tr>
<td>Injection period</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Injection timing</td>
<td>Sub inj. precedes by 0.2 ms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VCC in-cylinder conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient gas</td>
</tr>
<tr>
<td>Pressure [MPa]</td>
</tr>
<tr>
<td>Temperature [K]</td>
</tr>
</tbody>
</table>

Table 2 Injection conditions and in-cylinder conditions of VCC

Figure 5 Computational space and mesh layout
4.3 Calculation circumstances for droplet collision process

Table 3 summaries the numerical models for expressing the necessary in-cylinder phenomena. These models are not directly connected to the droplet collision and selected from a general list in the field of combustion simulations of marine diesels. The volume and the shape of a computational cell could affect the droplet behavior in its inside. To control the influence, velocity interpolation proposed by Nordin\(^1\) was adopted and the model constants were optimized in advance based on the spray observation results in the VCC.

Table 3 Numerical models for expressing necessary in-cylinder phenomena

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Sub model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary breakup</td>
<td>Blob method</td>
</tr>
<tr>
<td>Secondary breakup</td>
<td>KH-RT model</td>
</tr>
<tr>
<td>Collision possibility</td>
<td>O’Rourke model (reference)</td>
</tr>
<tr>
<td></td>
<td>Modified &amp; normal Nordin model</td>
</tr>
<tr>
<td>Collision outcome</td>
<td>Four outcomes and criteria</td>
</tr>
<tr>
<td></td>
<td>bounce, coalescence, stretching separation,</td>
</tr>
<tr>
<td></td>
<td>reflexive separation</td>
</tr>
<tr>
<td>Evaporation</td>
<td>Model of Amsden (not activated)</td>
</tr>
<tr>
<td>Turbulence</td>
<td>RNG k-ε model</td>
</tr>
</tbody>
</table>

5. Measurement / calculation results and discussion

5.1 Influence evaluation of collision discrimination models and parallel computation

Prior to the experiments for the spray merging and to the validating calculations for the GPGPU acceleration, it is necessary to investigate a couple of things, for example, differences in the level of the mesh dependency between the collision discriminating models and the effects of parallel computation on calculation results. For that purpose, the computational space in Figure 5 was scaled down by half and uniformly re-divided into 12 × 12 × 30 cells, then the tentative research was done to check out interaction of two crossing sprays in the space with the primary and the secondary breakups on Table 3 deactivated. The injection conditions were also altered as follows. At first, the horizontal distance between the nozzle holes was expanded to 17.6 mm. The intersecting angle between the spray trajectories was increased to 120° and the cone angle of each spray was decreased from usual value of 16° to 1°. Finally the same injection pressure of 40 MPa on both injectors was applied. These modifications serve to lighten the influence of droplet diffusion to radial direction of a spray and the influence of air motion around droplets, which leads to drastic increase in the geometrical probability of droplet collision. The results of the simulation are discussed below.

5.1.1 Effects of discrimination model on droplet collision probability

Figure 6 intercompares the spatial distributions of spray droplets predicted with O’Rourke model, Nordin model and modified Nordin model at 1 ms after start of injection (ASOI). As the central direction of the two nozzle holes intersects with each other on the longitudinal center line of the computational space, the droplets from the nozzles are expected to reach the intersecting point near at the same instant under the above-mentioned conditions. Hence higher droplet collision probability can be expected geometrically. As is evident in the figure, however, O’Rourke model case shows little collisions and little momentum exchanges between the droplets, so its results can be said to be illogical. This is due to the fact that it requires the two relevant droplets to exist in the same computational cell as necessary condition of collision discrimination. In the Nordin model case by its much less mesh dependency, the droplet probability increases by a large degree and the spatial distributions of spray droplets at 1 ms ASOI takes on a sector shape with its center on the intersecting point of the central lines of the two sprays. But the droplet distribution is biased toward the sector’s radiuses on both ends instead of toward the central line of the sector. The latter tendency could be anticipated under the above-explained injection conditions which enhances the collision probability than in reality. Which means that Nordin model also tends to underestimate the collision probability. It should be also pointed out that droplets tend to accumulate near the edge of the
sector circle. There the subsequent droplets can collide with their precedent ones because of deceleration by air drag so that the droplets distribution widens and thickens consequently. In the modified Nordin model case, droplet collisions occur most often and substantial spray merging is predicted for the first time. Hence, its droplet distribution assumes more axisymmetric feature densest around the vertical center line of the computational space in contrast to Nordin model.

In conclusion, one can say a collision discrimination model would affect the spray simulation results considerably. This is particularly true of injection systems of an inter-spray merging process. The DWI system in some marine diesels corresponds to such injection systems because of its simultaneous injection of fuel oil and water. So combining a proper collision discrimination model with meshing of a computational space would be necessary. In particular, O’Rourke model, which is standard in a single spray system, is heavily mesh-dependent and its simulation results are greatly influenced by the cell allocation of the two nozzle holes even though the distance between the nozzle holes is kept constant.

5.1.2 Effects of GPGPU on simulation of spray merging process

Figure 7 shows the similar spatial droplet distributions at 1 ms ASOI based on Nordin model and modified Nordin model but with the parallel computing algorithm optimized for GPU as explained in Fig.3. As compared with the equivalent distributions in the previous figure where the discrimination conditions are the same, one can argue that the algorithm modification for parallel computation operates the good reproducibility of the prediction results and no essential difference. Although it can be pointed that the level of droplet scattering after collision seems slightly lower than the level of the CPU cases, it could be attributed to the aforementioned parallel computation process which may change the pair of colliding droplets and the timing of the collision.

Through these investigations, the prospect is finally obtained for applying Nordin model more easily on the simulation of sophisticated injection systems including DWI where the deterministic collision discrimination would have been desirable in spite of its high computational load and its underestimation of collision probability.

5.2 Comparison of observation results with prediction results in spray merging process

Figure 8 exemplifies the spray merging process photographed in the VCC. In this particular case, the main injector delayed by 0.2 ms ASOI on the basis of the sub injector’s injection start. So at 0.4 ms ASOI, the spray body seen on the left half of the viewing field is rooted in the sub injector before spray merging and the shorter spray on the right half is a newborn merged one. At 3 ms ASOI, spray merging progressed and developed into a single spray body while the preceding sub injector spray was

Translated from Journal of the JIME Vol.49, No.6  ©2014(Original Japanese)
dimming away. Moreover, as clear in the image at 8 ms ASOI, the traveling direction of the merged spray was coincident with a vector sum of the spray momentums from both injectors. Accordingly the developing direction of the merged spray was biased toward a natural spray trajectory from the main injector.

Figure 9 presents the predicted droplet distributions on KIVA-3V to demonstrate the spray merging process in the computational space prepared for the VCC. The upper half of the figure corresponds to the results on O’Rourke model for collision discrimination and calculated only by CPUs. The lower half corresponds to those on the modified Nordin model with the optimized algorism for GPU. Calculation conditions and the spatial segmentation in these simulations were setup so as to simulate practical diesel sprays. O’Rourke model, however, underestimated the occurrence frequency of the droplet collision and held the two spray cores near in their original direction, so the droplet distribution was completely different from the observation results. By contrast, the modified Nordin model succeeded to increase the collision frequency and the momentum exchange between the two spray droplets considerably. The overall profile of the merged spray on the model provided much better agreement with the observation results, although the droplet density in the vector sum direction still seemed somewhat insufficient. As for the spray penetration length, both collision discrimination models well agreed with the experimental results.
5.3 Validation of acceleration effects of GPGPU on calculation speed

Figure 10 graphically shows computation time of the various simulations for the spray merging process in the previous section with different numbers of the initial parcels changing and different computing periods from the injection start. The figure adjacently compares the cases in which only CPUs ran all the way (dark shaded columns) with the cases in which the discrimination process was GPGPU (hatched columns). Both cases adopted the modified Nordin model for the droplet collision discrimination. As shown in the figure, acceleration effect of GPGPU was also evaluated based on an acceleration factor which is defined as a ratio of CPU computation time to GPU time (CPU/GPU). The columns at the right edge of the figure demonstrates how heavy a computing burden is in doing the droplet collision simulation based on the modified Nordin model even if it is customized for GPGPU. The rightmost column (near invisible) shows computation time in the case of skipping the droplet-collision relating routines.

The effects of the number of initial droplet parcels are discussed at first. The computing period was set to 3 ms ASOI and the number of initial parcels varied in the range of 4,000 ~ 8,000 identified as (a) ~ (c) in the figure. The acceleration by GPGPU was so obvious that the calculation time was cut down to 1/3 on average for simulating the whole spray merging process. The number of the initial parcels is directly linked to the number of the representative droplets as discussed in the section 3.2, hence its increase is supposed to emphasize the effect of GPGPU, but the case of 8,000 parcels showed decrease of the acceleration factor, instead. It can be explained as follows. Increasing number of the initial parcels also leads the computing load required for droplet breakup model, the other and the collision discrimination model does not have large share of the overall calculation time.

In (d) of the figure, the calculation time and the acceleration factor were measured setting the number of the initial parcels and the calculation period to 3,000 and 8 ms respectively. The number of initial parcels specifies only the number of droplets created along with the injection amount during the injection period, which means the much more parcels are quickly generated during the actual calculation because of the droplet breakups. This is a reason why the calculations (d) took more calculation time than it was expected from the ratio of two calculation periods (8/3) and the acceleration factor reached the highest value exceeding 4. In addition, an injection period of 3 ms can be considered as the maximum in the case of smaller engines while an injection period over 10 ms should be assumed in low-speed marine engines and GPGPU is more significant for them.

In (e) in the figure as mentioned above, computation time is compared between the diesel spray prediction method proposed in the present study and the equivalent prediction method just skipping the droplet-collision relating routines in order to demonstrate the ratio of the deterministic droplet collision discrimination like the modified Nordin model to total prediction of developing behavior of a diesel spray. A rate of computation time increase by the collision discrimination was more than 130 times, so it is revealed that the discrimination routines occupy the greatest part of the computational time under the conditions of neither evaporation nor combustion. It is additionally pointed out that the rate of increase got down to 10 ~ 20 times in the case of applying O'Rourke model instead of the modified Nordin model, although the rate varies with mesh parting lines and locations of nozzle holes.

![Figure 10 Accelerated computing of spray polymerization by GPGPU](image)

Translated from Journal of the JIME Vol.49, No.6 ©2014(Original Japanese)
6. Conclusions

In the present paper, Graphics Processing Units (GPUs) was applied for the first time to numerical prediction of merging process of two sprays and coupled with a deterministic collision discrimination model. Needed modifications and focal points for GPUs were examined carefully and their acceleration effect on calculation speed was evaluated quantitatively. The following conclusions were shown as follows.

- A deterministic discrimination model for droplet collision is necessary to simulate the injection systems allowing inter-spray droplet collisions like a spray merging process but it turns out to extend the total computation time by a factor of near one hundred in some cases. This can be an obstacle to the practical usage of numerical simulations of such systems.
- The deterministic collision discrimination can be accelerated by applying an algorithm which is optimized for parallel computation by GPUs and keeps the compatibility of the calculation results with CPUs.
- Simulation of the spray merging process was actually accelerated through the parallel computation by GPU and an acceleration was about a factor of 4 under the conditions equivalent to those of low-speed marine engines.
- The accelerating effect by GPU basically increases with a number of droplets which depends on a number of initial droplet parcels and a required calculation period but the rate of acceleration declines because of the increased calculation load on CPUs.

For the future, it is naturally supposed to apply the parallel computation on GPUs to the whole spray simulation including droplet breakup and droplet deceleration from air drag. It also comes in sight to extend GPU application to combustion simulation based on detailed chemical dynamics. In the application, each arithmetic thread will be assigned to each computation cell and differential equations will be solved explicitly.

Finally, this work was supported by Grant-in-Aid for Scientific Research (A) Number 21360435.

7. References

4) Tajima et al 3 persons, JIME scholarly journal (Japanese), 44-3 (2009-05), 73-78
10) Mimata, Kusaka, Daisho (Japanese), JSAE Lecture Before printing collection, 20106089, 149-10, 9-12, (2010)
11) O'Rourke, P.J., PhD thesis, Princeton University, USA, 1981