Journal of Computational Physics **164,** 62–80 (2000)
doi:10.1006/jcph.2000.6568, available online at http://www.idealibrary.com on IDEAL®

A New Droplet Collision Algorithm

David P. Schmidt* and C. J. Rutland†

*Department of Mechanical and Industrial Engineering, University of Massachusetts—Amherst, Amherst, Massachusetts 01003; and †Engine Research Center, University of Wisconsin, Madison, Wisconsin 53706
E-mail: rutland@engr.wisc.edu

Received September 17, 1999; revised March 3, 2000

The droplet collision algorithm of O'Rourke is currently the standard approach to calculating collisions in Lagrangian spray simulations. This algorithm has a cost proportional to the square of the number of computational particles, or "parcels." To more efficiently calculate droplet collisions, a technique applied to gas dynamics simulations is extended for use in sprays. For this technique to work efficiently, it must be able to handle the general case where the number of droplets in each parcel varies. The present work shows how the no-time-counter (NTC) method can be extended for the general case of varying numbers of droplets per parcel. The basis of this improvement is analytically derived. The new algorithm is compared to closed-form solutions and to the algorithm of O'Rourke. The NTC method is several orders of magnitude faster and slightly more accurate than O'Rourke's method for several test cases. The second part of the paper concerns implementation of the collision algorithm into a multidimensional code that also models the gas phase behavior and spray breakup. The collision computations are performed on a special collision mesh that is optimized for both sample size and spatial resolution. The mesh is different every time step to further suppress the artifacts that are common in the method of O'Rourke. The parcels are then sorted into cells, so that a list of all the parcels in a given cell are readily available. Next, each cell is individually checked to see if it is so dense that a direct collision calculation is cheaper than the NTC method. The cheaper method is applied to that cell. The final result is a method of calculating spray droplet collisions that is both faster and more accurate than the current standard method of O'Rourke.

INTRODUCTION

Stochastic collision models are commonly used in Lagrangian simulations of particulate and spray flows. Numerical results suggest that collision processes in sprays have a great influence on the average drop size [1]. Unfortunately, droplet collisions can be very expensive to calculate. The direct simulation of every drop would result in N^2 collision computations

0021-9991/00 $35.00

every time step, where N is the number of drops in the spray. This is prohibitively expensive for most purposes. The approach of Dukowicz can reduce this cost by representing the spray with a reduced number of computational particles [2]. Each particle, called a "parcel" of drops, represents q_i physical droplets. This approach, when applied to rarefied gas dynamics simulations, is commonly known as the direct simulation Monte Carlo (DSMC) technique [3]. However, in gas dynamics calculations, the value of q_i is a constant. In spray modeling, the size of the droplets varies enormously. To apportion computational resources efficiently, many spray models let q_i range over several orders of magnitude [4].

To calculate collisions, almost all DSMC spray models currently use an algorithm similar to that of O'Rourke [5]. O'Rourke's collision algorithm has long been the preferred method of calculating spray collisions in Lagrangian spray simulations. This collision algorithm is a critical component of the KIVA suite of codes originating from Los Alamos Laboratories [4]. Additionally, this collision algorithm is widely used in commercial CFD codes. No other collision algorithm has been as useful or as popular as O'Rourke's. The current work presents the next generation of spray collision modeling that addresses most of the remaining shortcomings in O'Rourke's algorithm.

O'Rourke's collision algorithm is consistent with the DSMC method, using a stochastic algorithm to determine droplet collisions. O'Rourke's collision algorithm is a type of "direct" technique, because it considers all possible collision partners. As is commonly done in gas dynamics calculations, O'Rourke's algorithm only allows parcels within the same gas-phase cell to collide. O'Rourke assumes that there is a probability of any droplet colliding with any other droplet given by

$$p_{i,j} = \frac{\sigma_{i,j} v_{i,j} \Delta t}{\forall}. \tag{1}$$

The variable $v_{i,j}$ represents the relative velocity between the two droplets, and $\forall$ represents the cell volume. In this equation, $\sigma_{i,j}$ is the collision cross section of the two drops and is defined as

$$\sigma_{i,j} = \pi (r_i + r_j)^2. \tag{2}$$

Thus, the mean expected number of collisions between a droplet in parcel i and the drops in parcel j is given by

$$\bar{\mu} = q_j \frac{\sigma_{i,j} v_{i,j} \Delta t}{\forall}, \tag{3}$$

where the number of droplets in parcel j is q_j. O'Rourke then determined the number of collisions by sampling from a Poisson distribution with a mean of $\bar{\mu}$. A similar approach has also been used in rarefied gas dynamics calculations and is known as the "direct" method of Kac [3]. There are two differences between the schemes: (1) O'Rourke allows for the possibility of a droplet from parcel i striking more than one droplet in parcel j and samples the number of collisions from the Poisson distribution; (2) O'Rourke's algorithm is generalized for the case of $q_i \neq q_j$. Bird [3] points out that the direct approach of Kac incurs a cost proportional to N_{p}^2, where N_{p} is the number of parcels in a cell. This is also true of O'Rourke's method.

To allow for a sufficient number of computational particles, algorithms with a cost proportional to N_{p}^2 must be avoided. Such methods are too expensive for accurate statistical

representations of sprays. Lagrangian spray calculations must represent the spray population in two or three spatial dimensions, a temporal dimension, a multidimensional velocity space, and a drop-size dimension. In order to fully represent all of these characteristic dimensions, a model should use a very large number of computational particles. However, the cost of N_p^2 algorithms is prohibitive for large numbers of particles.

Kitron *et al.* [6] applied the time-counter method, which has a cost linearly proportional to N_p, to sprays. However, Kitron used a fixed number of drops in each parcel. Also, the time-counter method, though historically important, has been faulted for its susceptibility to statistical scatter. The creator of the time-counter method described the scheme as "obsolete" even before the work of Kitron was published [7]. In any case, it is not clear how the time-counter method could be extended to parcels with different numbers of droplets. Without such generalization, a spray collision algorithm has limited utility.

A new approach to modeling droplet collisions based on the NTC algorithm is proposed. The NTC algorithm has been used by Abe [8] and Alexander and Garcia [9] in gas dynamics simulations. This algorithm has a computational cost linearly proportional to the number of computational particles and reduced variance when compared to the time-counter method. The technique randomly chooses a subset of candidate particles from the cell. These particles are then considered possible collision partners and are accepted with a probability proportional to their relative velocity and cross section. In the current work, the NTC approach will be generalized for the case of differing values of q and applied to test cases. Finally, an implementation into a multidimensional spray model will be presented.

THEORY

The following derivation is a new way of showing the mathematical basis of the NTC method. The derivation also gives a final result that is generalized for the case of varying numbers of drops per parcel. A final check shows that the new scheme also gives the correct number of collisions, on average, for each "class" of particle. The "class" refers to a region in the droplet velocity and radius space.

If a cell contains N droplets, which have a collision cross section given by Eq. (2), then the expected number of collisions in the cell over a time interval of Δt is given by summing the probability of all possible collisions:

$$M_{\text{coll}} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{v_{i,j}\sigma_{i,j}\Delta t}{\forall}. \tag{4}$$

The factor of one-half is a result of symmetry. If we group the individual droplets into parcels having identical properties, then the double summation becomes

$$M_{\text{coll}} = \frac{1}{2}\sum_{i=1}^{N_p} q_i \sum_{j=1}^{N_p} q_j \frac{v_{i,j}\sigma_{i,j}\Delta t}{\forall}, \tag{5}$$

where N_p is the number of parcels in the cell. Evaluating this summation directly would be as expensive as the Kac method, with the cost on the order of N_p^2. However, this summation

can be modified by pulling a constant factor outside the summation:

$$M_{\text{coll}} = \frac{(q v \sigma)_{\max} \Delta t}{2 \forall} \sum_{i=1}^{N_{\text{p}}} q_i \sum_{j=1}^{N_{\text{p}}} \frac{q_j v_{i,j} \sigma_{i,j}}{(q v \sigma)_{\max}}. \tag{6}$$

The value of $(q v \sigma)_{\max}$ is used for scaling the selection probability of a collision. The value chosen must be sufficiently large that the following restriction holds:

$$\frac{q_j v_{i,j} \sigma_{i,j}}{(q v \sigma)_{\max}} < 1. \tag{7}$$

Now it is assumed that a representative subsample of parcels may be randomly selected from the set of parcels in the cell, such that

$$\sum_{i=1}^{a N_{\text{p}}} x_i = a \sum_{i=1}^{N_{\text{p}}} x_i, \tag{8}$$

where $a \leq 1$ and x_i is a characteristic of each parcel. Hence, a subset of the parcels is used to represent the larger population. This statistical approximation allows a constant multiplier to reduce the limits of summation. Using this relationship, the limits of the summations in Eq. (6) are both reduced:

$$M_{\text{coll}} = \sum_{i=1}^{N_{\text{p}} \sqrt{\frac{(q v \sigma)_{\max} \Delta t}{2 \forall}}} q_i \sum_{j=1}^{N_p \sqrt{\frac{(q v \sigma)_{\max} \Delta t}{2 \forall}}} \frac{q_j v_{i,j} \sigma_{i,j}}{(q v \sigma)_{\max}}. \tag{9}$$

This equation is written more succinctly by defining the quantity M_{cand}:

$$M_{\text{cand}} = \frac{N_{\text{p}}^2 (q v \sigma)_{\max} \Delta t}{2 \forall}. \tag{10}$$

This definition is used in the limits of the summation of Eq. (9).

$$M_{\text{coll}} = \sum_{i=1}^{\sqrt{M_{\text{cand}}}} q_i \sum_{j=1}^{\sqrt{M_{\text{cand}}}} \frac{q_j v_{i,j} \sigma_{i,j}}{(q v \sigma)_{\max}}. \tag{11}$$

This equation is the final expression of the NTC method for application to parcels representing varying numbers of drops. Equation (11) is mathematically equivalent to the summation of Eq. (5). The difference is how one evaluates the summations. Equation (11) includes a summation over M_{cand} terms, while Eq. (5) includes a summation over N_{p}^2 terms. Both Eqs. (5) and (11) are statistical representations of Eq. (4).

In the limit of constant cross section and constant q, Eq. (11) reduces to the expression of Alexander and Garcia [9]. The overall cost will be proportional to the product of the limits of the summation, namely M_{cand}. The value of M_{cand} is linearly proportional to N_{p}, because q goes as $1/N_{\text{p}}$. The user of this model must make a sensible choice for $(q v \sigma)_{\max}$ for the algorithm to be efficient. If the spray is so dense that $M_{\text{cand}} > N_{\text{p}}^2/2$, then direct calculation of collisions may be more efficient than the NTC algorithm for this cell.

The double summation of Eq. (11) is evaluated using an acceptance–rejection scheme. The number of candidate pairs given by M_{cand} is selected with replacement from the cell

population. Because the parcels are selected with replacement, multiple collisions per time step may occur between parcels. O'Rourke has observed that the consideration of multiple collisions is required for accurate results with large time steps in dense sprays [5].

After a pair has been selected, a uniform deviate from [0, 1) is used to determine if the candidate pair actually collides. A collision takes place between parcels i and j if the deviate, r, satisfies the inequality

$$r < \frac{q_g v_{i,j} \sigma_{i,j}}{(q v \sigma)_{\max}}. \tag{12}$$

The variable q_g represents the greater number of droplets between q_i and q_j. If the collision is accepted, then q_l, the lesser number of droplets, actually participates in the collision. This distinction is important in the case of droplet coalescence, where one parcel of drops absorbs the other. The differentiation of the parcels by the larger and smaller values of q does not change the expected outcome of the scheme.

This differentiation is a subtle point that occurs only when the NTC method is applied to systems with varying values of q. The number of possible collisions is limited by the lesser number of droplets. However, the chance that collisions occur depends on the greater number of droplets. Thus, if the collision is accepted, only q_l collisions occur. Because of this distinction, Eq. (11) is written with one of the q terms outside the inner summation.

The NTC scheme is much more efficient than direct integration for large numbers of parcels in sparse sprays. The basic idea is shown in the Figs. 1–3. These figures represent the chance of droplet collisions in a cell containing seven parcels. Omitting the trivial cases of a parcel colliding with itself, there are 21 possible collision pairs, as shown in Fig. 1. A direct integration scheme would scan through all 21 possible pairs of parcels, checking each for collision. The NTC scheme estimates the maximum chance of collision, shown by the horizontal line in Fig. 2. This maximum is a quick and approximate calculation that impacts the speed of the code, but not the final answer.

The maximum probability is used to reduce the number of pairs considered for collision. The number of pairs actually considered for collision is M_{cand}. The number of pairs indicated

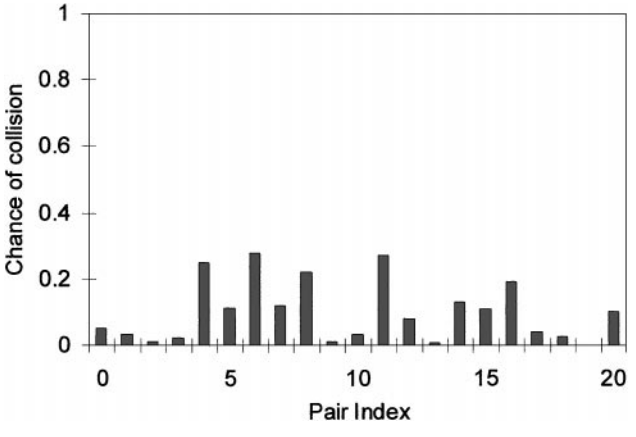

FIG. 1. Chance of collisions for 20 pairs of parcels.

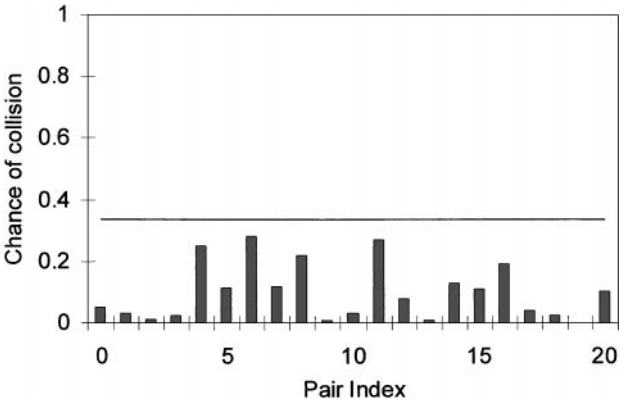

FIG. 2. The chance of collisions is bounded by an approximate estimate of the maximum, as shown by the horizontal bar.

by M_{cand} is selected at random, with replacement. Each of the pairs is tested for collision using the probability shown in Fig. 3. The probability is scaled up by the same factor used to reduce the number of collision pairs. The end result is, on average, the same as if the whole distribution shown in Fig. 1 were sampled directly.

The derivation of Eq. (9) guarantees that the average total number of collisions processed will be correct. It can further be shown that the average number of collisions processed for any two classes of drop is correct, where each class is identified by its velocity and drop size. For any two classes of particle represented by parcels i and j, the expected number of collisions between the two classes over a short time interval is

$$M_{12} = \frac{q_1 q_2 v_{12} \sigma_{12} \Delta t}{V}. \tag{13}$$

We now consider the prediction of the NTC scheme. The expected number of times that computational particles from class i and j will be selected is equal to the probability of the selection multiplied by the number of candidate pairs. The probability of selecting i and j from the candidates is $2/N_{\text{p}}^2$ because there are two ways to select these two particles

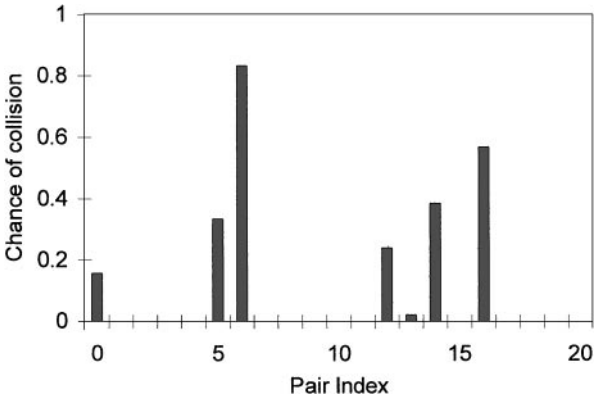

FIG. 3. The reduced sample space used by the NTC method. The number of pairs considered has been scaled down, and only seven possible pairs will be considered. Similarly, the chance of collision for these pairs has been scaled up, and each pair is more likely to be accepted.

from the N_p possible combinations. The number of candidate pairs is given by Eq. (10). The product of these two quantities is the expected number of times that i and j will be considered as collision partners, N_{sel}.

$$N_{sel} = \frac{(qv\sigma)_{max}\Delta t}{\forall}. \tag{14}$$

Given that i and j are selected candidates, the expected number of collisions to be processed is q_l multiplied by the probability of acceptance. The probability of acceptance is given by Eq. (12). So the expected number of collisions, given that i and j are selected candidates, is

$$\frac{N_{exp}}{N_{sel}} = q_l \frac{q_g v_{i,j}\sigma_{i,j}}{(qv\sigma)_{max}} = \frac{q_i q_j v_{i,j}\sigma_{i,j}}{(qv\sigma)_{max}}. \tag{15}$$

The product of Eqs. (14) and (15) recovers the expected number of collisions between i and j:

$$N_{exp} = \frac{q_i q_j v_{i.j}\sigma_{i,j}\Delta t}{\forall}. \tag{16}$$

Thus, this scheme produces the correct expected number of collisions, on average, for any two classes of particle as well as the correct total number of collisions. This derivation is sufficient to demonstrate the consistency of this usage of the NTC method. The derivation also reveals a subtle point that is often overlooked in applying the NTC method. The correct result given by the NTC method in Eq. (16) requires the probability of selecting both i and j to be $2/N_p^2$. The implication of this assumption is that both parcels are selected with replacement, allowing i and j to occasionally indicate the same parcel. In contrast, the implementations given in Garcia [10] and Bird [7] do not allow i to equal j. Fortunately, the significance of this error is small if the number of parcels in the cell is large.

VALIDATION

To test the cost of the NTC scheme and verify the correct result, the algorithm has been compared to analytical solutions and to O'Rourke's scheme. The analytical solutions were derived for this purpose and are simplifications of realistic problems. The test cases are used to check the overall prediction of the number of collisions, the computational cost, and the temporal and spatial order of accuracy of these schemes. The formal accuracy of O'Rourke's model has never been reported.

The first test case was spatially uniform, with a range of drop size, velocity, and number of drops per parcel. To test the total number of collision predictions, it was sufficient to calculate the number of collisions for a single time step. The effects of the collisions were not considered, only the number of collisions between various velocity and size classes. The consideration of realistic collision outcomes would make it much more difficult to obtain an analytical solution. Furthermore, the improvement of the present work is directed only at predicting the incidence of collision. O'Rourke's models of the outcomes of collision can still be used with the NTC method [5].

The test case was a distribution of N drops over a cylindrical volume $\forall$. The drops were given sizes from $[0, r_{max})$ and velocities in the axial direction from $[0, u_{max})$. The size and

velocity were chosen independently based on a uniform deviate. The expected number of collisions was calculated from the integral of the collision probability over the size and velocity space of each possible partner. The integral is

$$M_{\text{coll}} = \frac{N^2 \Delta t}{2\forall} \int_0^{r_{\max}} \int_0^{r_{\max}} \int_0^{u_{\max}} \int_0^{u_{\max}} f_u(u_1)\, f_u(u_2)\, f_r(r_1)\, f_r(r_2) |u_1 - u_2| \pi (r_1 + r_2)^2 \, du_1 \, du_2 \, dr_1 \, dr_2. \quad (17)$$

The functions f represent the probability distribution functions, which in this case are simply uniform distributions. The evaluation of this integral gives the following expected number of collisions:

$$M_{\text{coll}} = \frac{7\pi \Delta t u_{\max} r_{\max}^2 N^2}{36\forall}. \quad (18)$$

A simulation was run with a volume of unity represented by a single cell. Because the test problem was spatially uniform, there were no concerns about spatial resolution. The parcels were randomly assigned a radius from zero to 5×10^{-5} and velocities from zero to 10^2. The number of droplets per parcel was also randomly assigned, based on how many parcels were used for the calculation. The parcels represented a total of 10^8 droplets. The simulation consisted of a single time step of 10^{-4}.

The most important difference between the two approaches may be seen in Fig. 4. This figure shows a comparison of the cost of each scheme using a single-processor 100-MHz computer. For the numbers of parcels used in this test case, the NTC method is orders of magnitude faster than the O'Rourke method. The cost of O'Rourke's method shows a quadratic dependence on the number of parcels, and the NTC method shows a linear dependence. For clarity, an enlarged view of the cost of the NTC method is shown in Fig. 5.

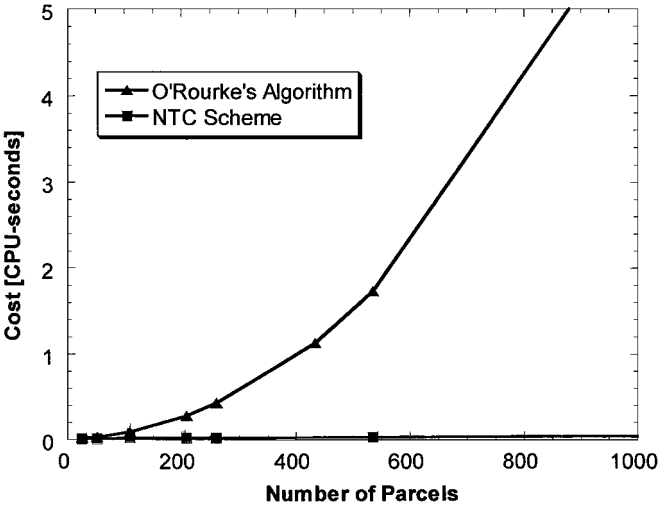

FIG. 4. The computational cost of each scheme is plotted versus the number of parcels for comparison. The cost of O'Rourke's scheme increases quadratically with the number of parcels. See Fig. 5 for a clearer presentation of the cost of the NTC scheme.

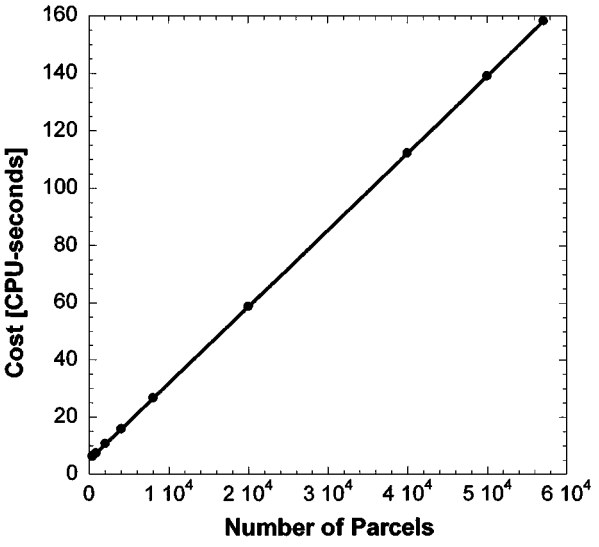

FIG. 5. The computational cost of the NTC scheme is plotted versus the number of parcels. The cost increases linearly with the number of parcels.

Figure 6 shows the average error in the predicted number of collisions for the new NTC method and for the O'Rourke method. The error, as defined by

$$\varepsilon = \frac{|M_{\text{theo}} - M_{\text{sim}}|}{M_{\text{theo}}}, \tag{19}$$

was averaged over 50 independent runs for each data point. M_{theo} is the prediction of Eq. (18), and M_{sim} is the prediction of the numerical scheme. Both schemes are Monte

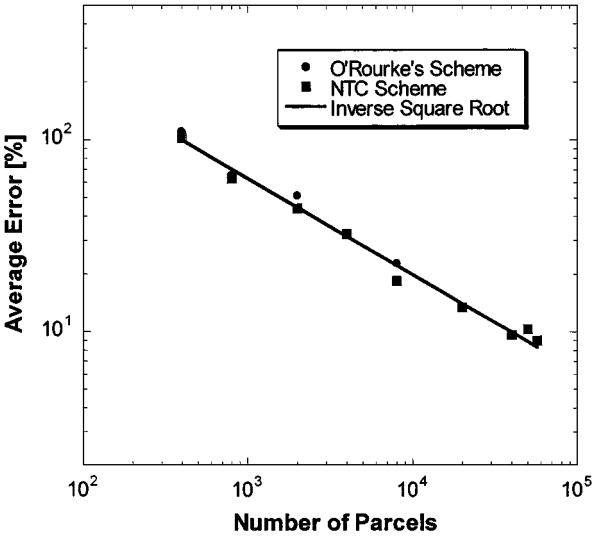

FIG. 6. The average error of both schemes is plotted versus the number of parcels. The error is defined in Eq. (19) and is averaged over 50 simulations for each data point. The average error decreases with the inverse square root of the number of parcels.

Carlo methods and show the characteristic convergence based on the inverse of the square root of the number of parcels. The results show that both schemes are converging to the theoretical answer as the number of parcels increases. The random variance of the two schemes is comparable.

Both schemes were tested against a transient analytical solution to determine the order of temporal accuracy. Note that neither this test nor the spatial test below measures the accuracy of the Lagrangian particle tracking scheme. These tests only indicate the accuracy of the predicted number of collisions. A single cell was used, but this time the outcome of the collision was considered so that a transient response could be observed. The droplets in the initial population were all of one size, with a radius of r_0. The velocity of the droplets was bimodal, given by the following probability density function:

$$f_u(u) = \frac{1}{2}\delta(u - u_0) + \frac{1}{2}\delta(u + u_0). \tag{20}$$

The δ in Eq. (20) represents the Dirac delta function. The probability density function was only applied to one component of velocity; all others were set to zero. The value of q was constant for all drops.

The collision outcome was contrived to resemble droplet coalescence and to allow a closed-form solution. All collisions resulted in the elimination of one parcel, chosen at random from the pair. For a discussion of realistic collision outcomes, see O'Rourke [5]. Though the particles had nonzero velocities, their positions were not updated between time steps because of the spatial homogeneity of the problem. This situation produces a decay in the number of droplets due to collision. Because one colliding parcel is removed from the calculation and the other parcel is unchanged, the probability distribution functions for size and velocity are constant with respect to time. The solution for the number of droplets remaining at time t can be found analytically to be

$$N(t) = \frac{N_0}{1 + t/t^*}. \tag{21}$$

The time scale, t^*, is defined as a combination of physical problem parameters as

$$t^* \equiv \frac{V}{2\pi r_0^2 N_0 u_0}. \tag{22}$$

The quantity N_0 is the number of droplets at the beginning of the simulation. The error for this test case was based on the absolute value of the difference between the numerical and analytical predictions at time t. Random variation was not anticipated to be an issue, because the calculation is a multistep integration. Ten thousand parcels were used to further suppress statistical fluctuations. The results, showing first-order temporal accuracy for the NTC method, are shown in Fig. 7. First-order accuracy is the best that can be expected for most stochastic collision routines, since complicated collision outcomes preclude the use of implicit methods. The method of O'Rourke shows less than first-order temporal accuracy for small time steps. This lower accuracy may be evidence of slightly higher statistical scatter in transient problems. Unlike in the previous steady-state test problems, collisions can change a parcel in this case. Since O'Rourke's method loops through parcels in the same order every time step, some parcels will consistently come first in the loop, and others will consistently come later.

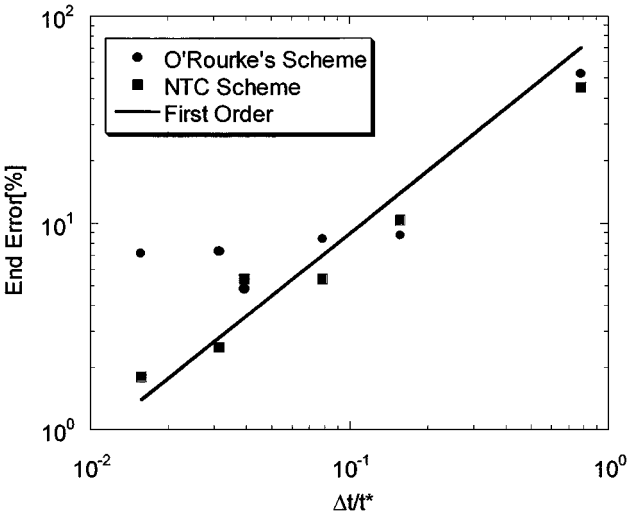

FIG. 7. This figure shows the error in a temporal calculation over a time interval of $t/t^* = 0.785$. The error is defined as the difference in the analytical and numerical prediction in the number of droplets at the end of the time interval.

The parcels that come later in the loop will always have fewer possible collision partners, due to the results of the previous collisions. The NTC method demonstrates first-order accuracy for the complete range of time steps. Because collision partners are chosen at random in the NTC method, it is unlikely that the order of storage could effect the outcome.

As a test for spatial accuracy, another test case was constructed. This problem domain was a three-dimensional cube with a volume of unity. The size, velocity, and number of droplets for 20,000 parcels were selected from uniform probability density functions, as in the problem described by Eq. (17). However, the y position of the parcels was chosen to produce an exponential distribution of number density, as given by the function in the equation

$$n(y) = n_{\max} \exp(-\alpha y). \tag{23}$$

and shown in Fig. 8. The parameter α controls the degree of spatial nonuniformity in the problem. For the current work a value of 10.0 was used, producing a highly nonuniform distribution. The value of $n_{\max}$ is calculated from the total number of droplets in the domain. The x and z locations of parcels were chosen from a uniform distribution. The analytical prediction for the number of collisions over the domain was derived by integrating the number of collisions predicted by Eq. (17) for a differential volume. Integrating over volume for the number density given by Eq. (23) gives the predicted number of collisions over a given time interval for a cube with one corner at the origin and the diagonally opposite corner at $(\Delta x, \Delta y, \Delta z)$ as

$$M_{\text{coll}} = \frac{7\pi u_{\max} r_{\max}^2 n_{\max}^2 \Delta t \Delta x \Delta z}{72\alpha} [1 - \exp(-2\alpha \Delta y)]. \tag{24}$$

Five computational cells were used in the x direction, one cell was used in the z direction, and varying numbers of cells were used in the y direction. Twenty thousand parcels were

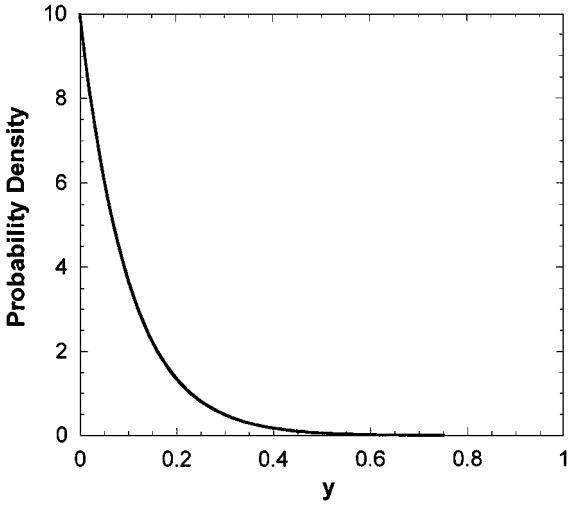

FIG. 8. The probability density function for the distribution of drops is plotted versus the y coordinate. The value of α is 10.0, producing a highly nonuniform distribution.

used for each calculation. The error was defined to be the average number of collisions predicted by 50 simulations compared to the analytical prediction.

Because the droplets are considered to be uniformly spread throughout a computational cell, underresolved simulations tend to underestimate the number density of the droplets. This causes underresolved simulations to underpredict the number of collisions. When spatial variations are sufficiently resolved, the NTC scheme is second-order accurate in space. This result may be seen in Fig. 9. Due to the very high cost of running O'Rourke's scheme with a large number of parcels, only the NTC scheme was tested for spatial accuracy.

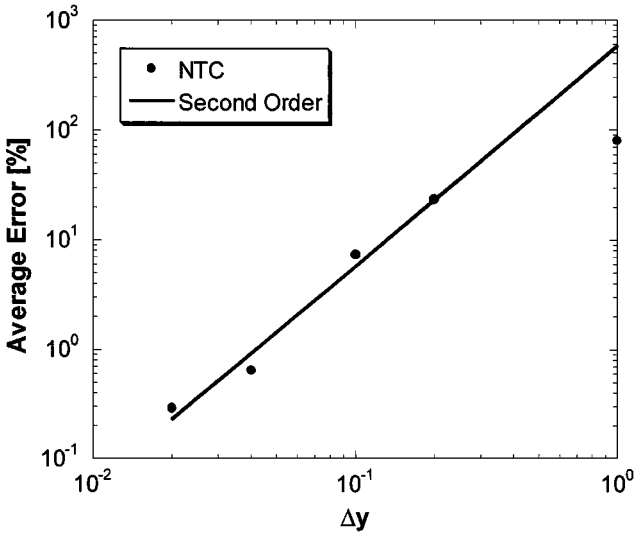

FIG. 9. The spatial error of the NTC scheme is plotted versus the cell size in the y direction. For sufficiently resolved cases, the scheme is second-order accurate in space.

IMPLEMENTATION INTO A MULTIDIMENSIONAL CODE

Spray simulations require modeling of the gas phase and its interactions with the spray. Correct implementation is quite difficult and requires special attention. A discussion of how the previously described NTC method should be applied follows. This new implementation is divided into five major portions:

1. Creation of a collision mesh.
2. Grouping of the parcels by their mesh location.
3. Cell-by-cell identification of the optimal method of calculating collisions.
4. Calculation of the incidence of collision.
5. Calculation of the outcome of each collision.

These five portions will each be described in detail. The entire algorithm will then be applied to some test cases, and the results will be compared to O'Rourke's algorithm.

1. *Creation of a Collision Mesh*

In O'Rourke's algorithm, only parcels within the same computational cell are allowed to collide. This approach is second-order in space, but suffers from the fact that gas-phase cells are usually much too large for sufficient spatial resolution. The results tend to be severely grid-dependent. Figure 10 shows a calculated hollow-cone spray in a dense gas environment. The dense ambient gas retards the expansion of the spray, increasing the number density and the significance of droplet collisions. The same spray is modeled on a Cartesian mesh and a polar mesh. Note that the entire shape of the spray is changed by the grid. The Cartesian mesh turns the hollow-cone spray into a "clover-leaf" shape.

The clover-leaf artifact is a result of the correlation between the droplets' velocity and position. Droplets that have trajectories of almost 90° apart can be located in the same cell of a Cartesian mesh near the spray origin. The relative velocities between these parcels are very large, and so collisions are very likely. The postcollisional velocities tend toward a mean velocity halfway between the original velocities because of coalescence and inelastic bouncing. The polar mesh provides better resolution in the azimuthal direction and does not permit this artifact to form. In comparison, the Cartesian mesh only resolves the azimuthal direction in 90° increments.

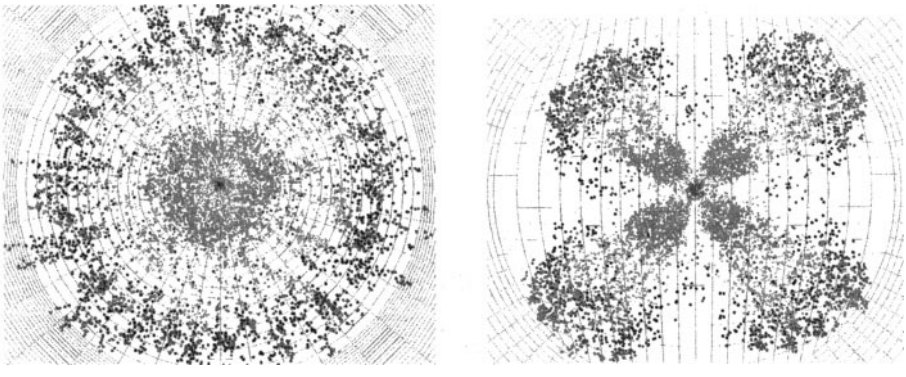

FIG. 10. A hollow-cone spray calculated using O'Rourke's method on a polar mesh (left) and a Cartesian mesh (right). The spray is directed toward the viewer. The physical situation is the same for both cases; only the mesh differs.

According to current modeling methodology, droplet collisions have no connection to the gas phase. Hence, there is no reason to rely on the gas-phase mesh. The new implementation creates a polar collision mesh around the spray for optimal accuracy. By making the mesh polar, the azimuthal direction is better resolved, and parcels are grouped with more appropriate collision partners. The use of a cylindrical collision mesh assumes that only a single spray is present. In the case of multiple sprays in the calculation, it would be possible to have several independent collision meshes if the sprays do not interact. However, in the most general case of multiple sprays that can interact, another grid strategy is required.

To further suppress numerical artifacts, the orientation of the collisional mesh is randomly rotated each time step around the axis of the injector by an angle from zero to 2π. This guarantees that azimuthal cell boundaries will change each time step. Due to the change of parcel locations, the radial and axial cell boundaries will also change each time step.

The extent of the collision mesh is dictated only by the location of the parcels. The mesh need not include parts of the gas-phase domain that do not contain droplets. Additionally, boundaries are of no concern because they do not directly affect droplet collisions. Because the collision mesh is polar, the collisional domain is naturally shaped like a cylinder. The axis of the cylinder is the axis of the atomizer. Thus, the position of each parcel is measured using a polar coordinate system determined by the location and orientation of the injector.

The size of the collision mesh cells can be optimized so that the cells are sufficiently small to capture important spatial information, yet large enough to have a statistically adequate number of parcels in each cell. After the number of parcels is counted and the furthest extent of the parcels noted, the mesh resolution is set so that the average number of parcels in each cell is about 5 to 10. This algorithm will then produce cells with a large number of parcels in cells in the dense regions of the spray where collisions are more important. In dense regions of the spray, it is desirable to have at least 20 parcels per cell. The sparse regions of the spray will have few parcels per cell, but will also have extremely low collision rates. A view of a collision mesh is shown in Fig. 11.

This linkage of the number of parcels to the collisional mesh means that it becomes imperative that the calculation contains a sufficient number of parcels. If the user chooses a large number of parcels, then the calculation will demonstrate good statistical representation of the spray as well as good spatial resolution. If the user fails to use enough parcels, then the calculation will fail in both respects. Fortunately, the current algorithm as a whole is much faster than O'Rourke's collision model, and so the calculation can use a large number of parcels without significant penalty.

Once the collisional mesh is established, it is inexpensive to identify which cell a parcel resides in, due to the regularity of the mesh. The axial, radial, and azimuthal location of the parcel can easily be translated into i, j, and k indices or into a cell identification number. This information is used for grouping the parcels into cells, as described below.

2. *Grouping of the Cells by the Mesh Location*

In the original implementation of O'Rourke's collision model, the algorithm looped over all possible droplet pairs and performed a test to see if the two parcels were in the same cell. Only when the two parcels were located in the same cell were they considered collision partners. The cost of this check is proportional to the number of parcels squared.

In the present algorithm, the parcels are sorted by their collision cell before any consideration of collision. This step is required for the NTC algorithm, which requires knowledge

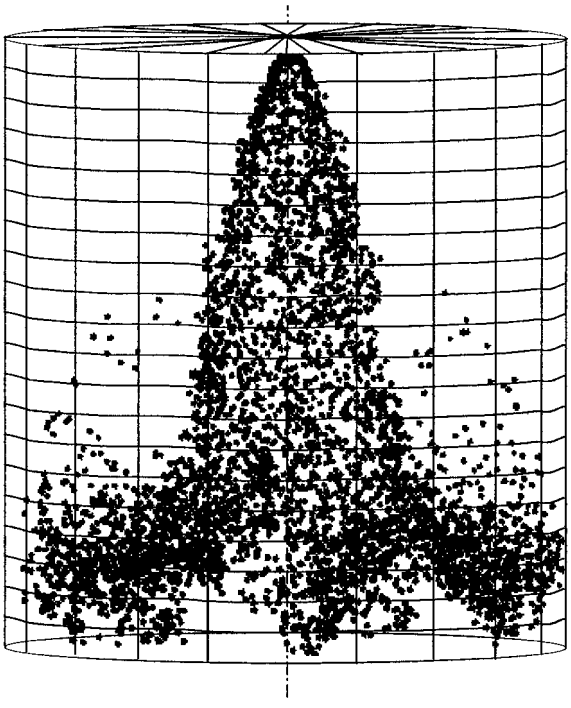

FIG. 11. A collision spray mesh. The lines of constant radius are omitted for clarity.

of the parcels that reside in a given cell. Grouping the parcels together by their cellular location has another advantage as well: looping over all possible collision partners becomes much cheaper because the loop limits only span the parcels of one cell. Looping over collision partners is required for the direct calculation of collisions. The direct consideration of collision calculations using the grouping information will be referred to as direct single-cell collision (DSCC) calculation and is discussed in the next section.

The grouping process can be achieved very quickly with two loops through the number of parcels [10]. A pointer array that indicates which parcels are in which cell is created. During the first loop, the number of parcels in each cell is counted. An indexing array is filled with pointers to a parcel identification array. The indexing array is packed with the location within the parcel identification array of the first parcel in each collisional cell. Then another sweep is made and the parcel identification array is filled with pointers to the individual parcels. This storage structure is illustrated in Fig. 12.

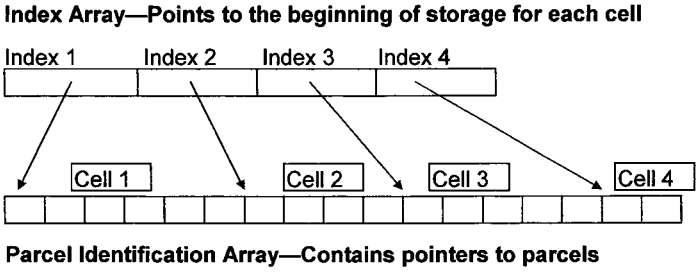

FIG. 12. Storage and sorting of parcel and cell information.

With this storage system, the code can access the list of parcels contained in any given cell. The code looks in the indexing array to find where a cell's storage begins in the particle identification array. The beginning of the cell's storage is explicitly stored in the indexing array, and the end of the storage can be found by looking where the next cell's storage begins. Then the pointers to parcels are used to access parcel information with indirect addressing.

3. *Cell-by-Cell Identification of the Optimal Collision Algorithm*

The current algorithm chooses between two methods of calculating the incidence of droplet collision. The NTC algorithm is usually cheaper for sparse cells, and the DSCC method may be cheaper for dense cells. DSCC integration refers to the consideration of collisions between every possible parcel within a cell. This differs from O'Rourke's method because the code does not have to scan through all parcels. The DSCC integration method is the same as O'Rourke's method but takes advantage of the cell grouping information.

To see the effect of the grouping on the computational cost of direct collision calculation, consider the following example. Let N parcels be located in a domain with N_c collision cells. To loop over all possible collision partners without grouping requires $N^2/2$ iterations. If the parcels are grouped by cell, then each cell will have, on average, N/N_c parcels. A loop over all the possible pairs within the cell would require N^2/N_c^2 iterations. If this is repeated over all the cells, then the cost is roughly N^2/N_c. However, in the creation of the collisional mesh, the mesh size was set so that the average number of parcels per cell was equal to a constant from 5 to 10. Thus, N/N_c is a constant, and the computational cost of looping over all possible collision pairs is linearly proportional to N.

Using the DSCC method, the code must consider collisions between N_p parcels in a cell for a cost proportional to $N_p^2/2$. The cost of the NTC method is proportional to the quantity M_{cand} defined in Eq. (10). The constants of proportionality are believed to be similar. So on a cell-by-cell basis, the code scans through collision cells, estimating the cost of each algorithm. The cheaper method is identified by comparing $N_p^2/2$ to M_{cand}. There is an alternative interpretation of the criterion that clarifies the meaning. The criterion for using the DSCC method can be written as

$$M_{cand} > \frac{N_p^2}{2}. \tag{25}$$

Using Eq. (10) and some algebraic manipulation, this is rewritten as

$$(q v \sigma)_{max} \Delta t > \forall. \tag{26}$$

This alternative form shows that the decision is actually a measure of how well the spray is being resolved. The inequality compares the swept volume of one parcel to the volume of the cell. Thus, the DSCC method may be used when the spray is poorly resolved by the number of parcels (q is large), when the time step Δt is greater than a collision time, or when a poor choice of $(q v \sigma)_{max}$ is made.

The DSCC method is cheaper when the spray is very dense or is underrepresented by the number of parcels. In such an extreme case, the fundamental assumptions of the DSMC treatment are in doubt [3]. The expected number of parcels participating in collisions is very large and may be a result of very high numbers of droplet collisions per time step.

Alternatively, the number of droplets per parcel may be too large, causing M_{cand} to be greater than the number of parcels in the cell. In the latter case, the user should increase the number of parcels used in the collision calculation. Finally, the DSCC method may be cheaper due to a poor estimate of M_{cand}. In any case, the DSCC method should be cheaper than the NTC scheme only for a few cells in the densest part of the spray. Otherwise, the simulation should use a smaller time step or a larger number of parcels.

4. *Calculation of the Incidence of Collision*

The algorithm scans through each cell calculating collisions between parcels. Depending on which is faster, the incidence of collision is calculated with either the NTC algorithm or the DSCC integration. If a large portion of the cells use the DSCC method, then the fundamental assumptions of sparse sprays probably have been violated as noted above.

5. *Calculation of the Outcome of Each Collision*

When droplets collide several outcomes are possible, such as bouncing, coalescence, and shattering. Recent work by Georjon investigated how to model the outcomes of collision [11]. For the present work, the outcomes of O'Rourke are used [5]. Based on the Weber number and a stochastically chosen "offset parameter," the parcels can coalesce or bounce. The offset parameter is a measure of whether the collision is head-on or relatively oblique. O'Rourke's model for inelastic bouncing was used in the current work.

One change has been made from O'Rourke's collisional outcomes. O'Rourke considered collisions from the point of view of the larger drop. In coalescence, the larger drop would absorb numerous smaller drops. However, this approach had a difficulty. Sometimes the parcel with the larger drops was more populous than the parcel with the smaller drops. The larger drops would lack a sufficient number of "mates" from the other parcel.

The current implementation notes which parcel is more populous. For coalescence, the number of drops of the less populous parcel is now subtracted from the more populous parcel. This method guarantees that there are sufficient mates for executing the coalescence model.

SPRAY CALCULATIONS

The new collision algorithm has been applied to the same hollow-cone spray shown in Fig. 10. The results are shown in Fig. 13 for Cartesian and polar meshes. The sprays are no longer severely mesh dependent. They do not display the "clover-leaf" artifact found in Fig. 10. As an additional benefit of the new algorithm, the calculations with the new collision algorithm required about 31% less CPU time for a spray and gas-phase calculation with 12,000 parcels. The average number of parcels per cell was set to 5.0. This resulted in cells in the densest part of the spray with over 100 parcels per collision cell. This number of computational particles far exceeds the minimum of 20 suggested by Alexander and Garcia [9].

Even with the improved collision model, there is a slight loss of spray axisymmetry on the Cartesian mesh due to the resolution of the gas-phase velocity and drag on the drops. The gas-phase flow, which should be axisymmetric, cannot be perfectly rendered on a Cartesian mesh. The gas-phase mesh would have to be extremely fine, to an impractical level, to achieve perfect axisymmetry. However, the new implementation is much better than the conventional approach of O'Rourke.

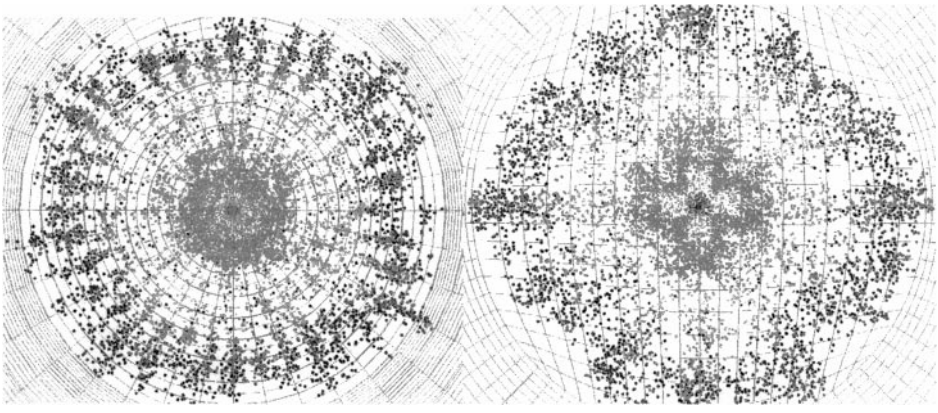

FIG. 13. A hollow-cone spray calculated using the new collision algorithm on a polar mesh (left) and a Cartesian mesh (right). The spray is directed toward the viewer. The physical situation is the same for both cases; only the mesh differs.

CONCLUSIONS

The NTC collision scheme has been extended for spray calculations, where the number of drops represented by a computational parcel varies significantly. This extension of the NTC scheme was derived from the basic equation for the probability of collision of two drops in a fixed volume. Comparison of the NTC scheme to an analytical solution showed that the average error decreases with the inverse square root of the number of parcels. The NTC scheme was also shown to be first-order accurate in time and second-order accurate in space. A comparison of computational cost showed that the NTC scheme is much faster than the standard approach of O'Rourke. The cost of the NTC scheme was shown to increase linearly with the number of parcels, while the cost of O'Rourke's method increases quadratically. Because of the reduced cost, the NTC method will allow computational spray models to use far larger numbers of parcels and achieve a superior sampling of the droplet characteristics.

Implementation into a multidimensional code has also been described. A special collision mesh is used to achieve a compromise between spatial resolution and sample size in each cell. By grouping the parcels into cells, the code can choose the fastest algorithm on a cell-by-cell basis. The grouping allows the application of either the NTC method or a cellular form of O'Rourke's method called the direct single-cell collision scheme. For very dense cells, the DSCC calculation is faster, and for all other cells the NTC method is faster. The new implementation is considerably faster than the method of O'Rourke and does not demonstrate severe grid-dependent artifacts.

REFERENCES

1. M. Gavaises, A. Theodorakakos, G. Bergeles, and G. Brenn, Evaluation of the effect of droplet collisions on spray mixing, *Proc. Inst. Mech. Eng*. **210**, 465.
2. J. K. Dukowicz, A particle-fluid numerical model for liquid spays, *J. Comput. Phys.* **35**, 229 (1980).
3. G. A. Bird, Perception of numerical methods in rarefied gasdyanmics, *Prog. Astronaut. Aeronaut.* **118**, 211 (1989).
4. A. A. Amsden, KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays (Los Alamos Report LA-11560-MS, May 1989).

5. P. J. O'Rourke, *Collective Drop Effects on Vaporizing Liquid Sprays* (Department of Mechanical and Aerospace Engineering, Princeton University, 1981).
6. A. Kitron, T. Elperin, and A. Tamir, Stochastic modelling of the effects of liquid droplet collisions in impinging streams absorbers and combustors, *Int. J. Multiphase Flow* **17**(2), 247 (1991).
7. G. A. Bird, *Molecular Gas Dynamics* (Clarendon Press, Oxford, 1994).
8. A. Takashi, Generalized scheme of the no-time-counter scheme for the DSMC in rarefied gas flow analysis, *Comput. Fluids* **22**(2/3), 253 (1993).
9. F. J. Alexander and A. J. Garcia, The direct simulation Monte Carlo method, *Comput. Phys.* **11**(6), (1997).
10. A. L. Garcia, *Numerical Methods for Physics* (Prentice-Hall, Englewood Cliffs, NJ, 1994).
11. T. L. Georjon and R. D. Reitz, A drop shattering collision model for multidimensional spray computations, *Atomization and Sprays* **9**(3), (1999).