

Application of the MCMC Method for the Calibration of DSMC Parameters

James S. Strand and David B. Goldstein

Aerospace Eng. Dept., 1 University Station, C0600, The University of Texas at Austin, Austin, TX, 78712, USA

Abstract. A Markov Chain Monte Carlo (MCMC) algorithm was employed to obtain a calibrated distribution for the hard sphere diameter, the VHS reference diameter, and the temperature viscosity exponent of argon, for use in the Direct Simulation Monte Carlo (DSMC) method. Shock-tube experiments from Alsmeyer [1] were used to provide the necessary calibration data for use in the MCMC method. The DSMC method used in this work employs the algorithm of Bird [2], with modifications to allow for the efficient simulation of a 1D shock. When calibrating for the hard sphere diameter (the temperature viscosity exponent is set to 0.5 for the hard sphere method), the results of the MCMC method agree with a simple brute-force method, and a single value for the hard-sphere diameter is obtained. For the VHS method, however, when simultaneously calibrating the VHS reference diameter and the temperature viscosity exponent, we find that normalized density data alone does not provide sufficient information to obtain a single solution for both parameters. Instead we find a band in parameter space where acceptable solutions are obtained.

Keywords: DSMC, MCMC, Shock Structure

PACS: 51.10.+y, 51.20.+d

INTRODUCTION

The DSMC method employs many parameters whose values strongly affect the simulation results, and whose values are not always precisely known. For example, as discussed by Valentini and Schwartzentruber [3], there is still some debate as to what value of the temperature velocity exponent (ω) in the VHS molecular model should be used in a given regime. Even in simple cases without chemistry, values must still be supplied for collision cross-sections (or reference diameters used in the calculation of collision cross-sections) and for the dependence of these cross-sections on the relative velocity of the colliding particles. In mixtures, values must be supplied (or calculated from other supplied values) for collisions within each species, as well as for each potential set of colliding species. When chemistry is included, the number of parameters increases further.

In many cases, these parameters are not directly measurable, and the data used to obtain these parameters are often by necessity somewhat limited. For example, collision-cross section parameters might be obtained from viscosity measurements (using the Chapman-Enskog expansion). Thus, parameter values are frequently used in regimes far from the regime in which they were obtained.

In this work, we use the MCMC method (described in the next section) to obtain calibrated values for the VHS reference diameter and the temperature viscosity exponent for argon. For calibration we use a normalized density profile for a Mach 3.38 shock in argon from Alsmeyer [1], which was obtained by means of interferometry in a shock tube. In this work and that of Alsmeyer, normalized density is defined as

$$\rho_n = \frac{\rho - \rho_1}{\rho_1 - \rho_2} \quad (1)$$

where ρ_1 and ρ_2 are the upstream and post-shock densities, respectively.

NUMERICAL METHOD

DSMC Method

The DSMC code used in this work, which was written from scratch with the idea of compatibility with MCMC in mind, is based on the method described by Bird [2]. Two key modifications have been made to the method.

First, the entire DSMC code is written as a subroutine, to be called by a driver program. While the DSMC code does have an input file that is read on each call to this subroutine, certain parameters are passed to the DSMC code from the driver code. This allows for easy interaction between the MCMC driver and the DSMC subroutine. Second, although the work presented here concerns monatomic gases, and for these cases the Rankine-Hugoniot jump conditions provide an analytic solution for the post-shock conditions, in the future when the method is applied in cases with real-gas behavior, the post-shock conditions will not always be exactly known *a priori*. The method for simulating 1D shocks presented in Bird [2], however, requires that the post-shock conditions be known initially. Furthermore, if a steady 1D shock is desired, some form of artificial stabilization must be applied in order to keep the shock steady within the computational domain, since in a 1D flow there is nothing to hold the shock in place.

Due to these concerns, we have chosen instead to simulate the development of an unsteady 1D shock. The flow in the domain is initialized with a Maxwellian velocity distribution at the freestream (pre-shock) temperature, with a bulk velocity moving to the right. The right boundary of the domain is set as a specular wall (therefore no wall temperature must be specified). The left boundary of the domain represents a freestream; it generates molecules entering the domain with a Maxwellian velocity profile (offset by the freestream velocity) at the freestream temperature. With this initial setup, at the beginning of the simulation an unsteady normal shock forms and begins propagating to the left. After a short time, the shock becomes self-similar and moves at a constant speed, and we begin tracking the shock location. A set of sample cells follows the motion of the shock, with the set of sample cells moved each time sampling is performed, so that they are centered about the shock center. The shock center is defined as the location where, after suitable smoothing to account for instantaneous noise, the normalized density equals 0.5. The smoothing is used only when determining the location of the sample cells; no smoothing is used in the simulation itself, or in the final results. A schematic showing our shock-tracking method is shown in Figure 1a. This figure shows both the overall domain in which the unsteady shock propagates and also the sample region which propagates with the shock. In this sample region, the shock can be viewed as steady, and the normalized density profiles from the sample region are used for comparison with Alsmeyer's data.

MCMC Method

The Metropolis-Hastings MCMC algorithm [4] is used to solve the statistical inverse problem in order to perform the desired calibrations. In our MCMC approach a sequence of DSMC simulations is run with different input parameters. The MCMC process proceeds as follows:

1.) Prior knowledge/educated guesswork is used to establish a prior distribution for the parameters to be calibrated. The range of this prior distribution should be large enough so that a reasonably large section of the parameter space is explored, but small enough that the problem remains tractable. In this work, the range used for both the hard sphere and VHS reference diameters is 3×10^{-10} m through 6×10^{-10} m. The limits for the diameter are somewhat arbitrary; they are chosen so that the value of 4.17×10^{-10} m, which is given by Bird [2] for the VHS reference diameter for argon, is near the middle. Bird's VHS value was not put at the center of the range because the same range was used for both hard-sphere and VHS diameters, and it was assumed that the calibrated hard-sphere diameter would be lower than the VHS reference diameter from Bird. This assumption was correct, as will be shown later. The range for the temperature viscosity exponent is 0.5 through 1.0. This range is not arbitrary, because $\omega = 0.5$ corresponds to the hard-sphere limit and $\omega = 1.0$ corresponds to Maxwell molecules, so this range includes all physically reasonable values of ω .

2.) Initial values for the parameters are randomly selected from within their prior distributions. The DSMC simulation is run with this set of parameters.

3.) The DSMC results are compared to the experimental data, and an error term is calculated from the equation

$$Error = \frac{1}{N} \sum_{i=1}^N (\rho_{n,i,Als} - \rho_{n,i,DSMC})^2 \quad (2)$$

where $\rho_{n,i,DSMC}$ is the normalized density for the i th sample cell, $\rho_{n,i,Als}$ is the normalized density from Alsmeyer at the x -location corresponding to the i th sample cell, and N is the total number of sample cells. Alsmeyer's results in [1] are presented as continuous curves, so they are treated as continuous for the purposes of this work. In the limit as the number of sample cells becomes very large, this equation would become

$$Error = \int (\rho_{n,i,Als} - \rho_{n,i,DSMC})^2 dx \quad (3)$$

where the integral is carried out over the sample region. This integral is shown as the blue shaded area in the upper-left image of the schematic in Fig 1a.

4.) A probability for this set of parameters is calculated by plugging the error term into the likelihood equation

$$Prob = e^{-\frac{1}{2}\beta(Error)} \quad (4)$$

where β is a constant chosen so that the likelihood equation produces a reasonable range of probabilities (i.e., not all near 0 or all near 1). In the results presented here, $\beta = 5000$.

5.) A new set of parameters is selected by moving a certain distance in the parameter space from the previous location. This distance is randomly chosen based on a Gaussian centered at the old parameter location. The width of the Gaussian is set to 10% of the range of the parameter.

6.) The DSMC simulation is run with the new set of parameters and a probability is determined.

7.) If the probability of the new set is higher than that of the old set, the new set is accepted. If the probability of the new set is lower, it is accepted or rejected based on a random number draw, $Prob_{accept} = Prob_{new}/Prob_{old}$.

8.) Steps 4-6 are repeated until the Markov Chain has reached the desired length. A histogram is then created for each parameter based on the set of chain positions. This histogram represents an estimate of the PDF of that parameter, based on calibration with the given set of experimental data.

An example of a single MCMC chain is shown in Figure 1b. This chain is one of the chains used during the calibration for the VHS case. The chain starts at a random location, and then moves through parameter space. It moves quickly into a band region in parameter space where the probability is high. This band will be discussed more in the Results section.

Although it is not done in this work, more than one set of experimental data can be used at the same time. In steps 2 and 5, the DSMC code can be run (with the same set of physical parameters) for multiple scenarios (i.e., a normal shock, a 0D relaxation, a boundary layer, etc.). A single error term which encompasses the full set of scenarios would need to be calculated, and this error term would then be used in the likelihood equation. The advantage of using multiple scenarios is that the parameters can then be calibrated to obtain the greatest possible accuracy over a wide range of applications.

Parallelization

There are two levels of parallelism within the overall MCMC-DSMC code. Any given chain must be run in sequence, since the parameters for the next chain position cannot be determined until the DSMC simulation at the current chain position has finished and a probability has been calculated. However, the MCMC method can easily be parallelized by starting multiple chains, which can then be run simultaneously. At the beginning of the overall MCMC run, the available processors are divided into groups, and each of these groups is assigned one or more chains to run. Depending on the parameters at a given chain location there may be far more or fewer collisions per time step, and thus chains with locations in regions with high number of collisions will take longer to complete. When a group completes its assigned chain or chains, the processors in that group are then added to another group with work left to do, in order to speed up the slower groups and so that all processors have work to do as much of the time as possible. When all chains have finished, the final parameter PDFs are then based on the combined histogram from all the chains.

In addition to the MCMC driver code, the DSMC code itself is MPI parallel, allowing it to be run on an arbitrary number of processors (provided the total number of processors is less than the total number of cells). The goal of the parallelism in this DSMC code is not primarily to allow for larger problem sizes to be simulated, but rather to allow for extremely fast simulation of relatively small problems, as is necessary for integration with MCMC. Load balancing is performed within the code, in which the DSMC cells are redistributed to processors to account for changes in workload. Due to the small problem sizes, better cache usage when running on multiple processors actually leads to super-linear speedup on up to 64 processors when running the simulations presented here. This allows runs that formerly took 20 minutes of wall-time on a single processor to be run in less than 20 seconds, which in turn allows reasonably long MCMC chains to be run within a practical amount of computational time.

RESULTS

Hard Sphere Case

As a simple example, we first calibrated the hard-sphere diameter (d_{HS}) for argon using Alsmeyer's Mach 3.38 shock tube results as the calibration data. The post-calibration PDF for the hard sphere diameter is plotted in Figure 2a. This figure shows that, for a hard sphere DSMC model, the probability is greatest that $\sim 3.2 \times 10^{-10}$ m is the argon diameter which leads to DSMC simulation results which best reproduce the experimental results of Alsmeyer.

Figure 2b. shows a plot of the error term as a function of d_{HS} , based on a uniform sample from the assumed prior distribution. In this simple case, simple uniform sampling of d_{HS} could be used to calibrate the parameters, but as the number of parameters grows uniform sampling becomes impractical. Note that the lowest error in Fig. 2b. corresponds to the peak of the PDF in Fig. 2a., as expected.

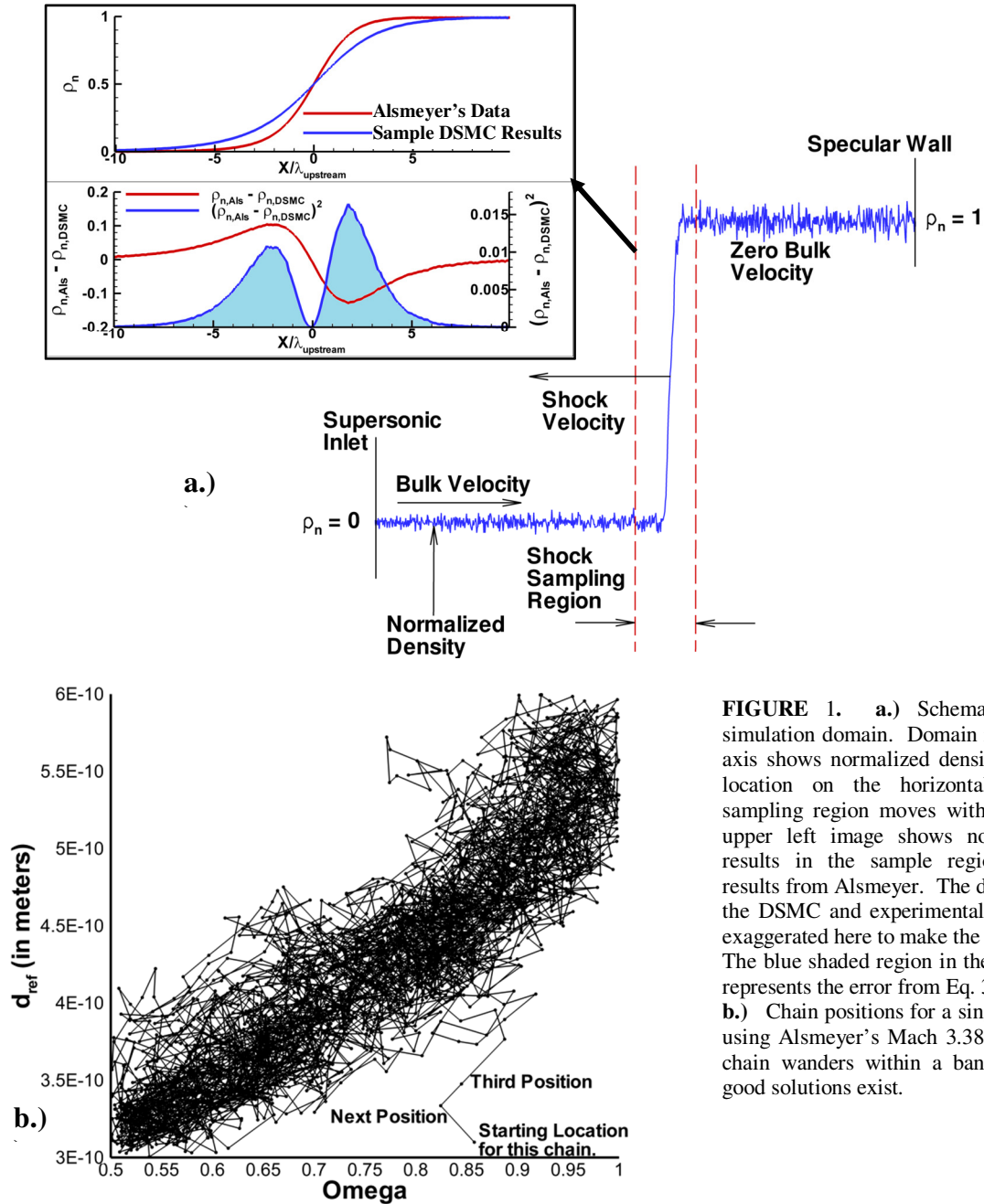


FIGURE 1. a.) Schematic showing the simulation domain. Domain is 1D, the vertical axis shows normalized density vs. streamwise location on the horizontal axis. Shock sampling region moves with the shock. The upper left image shows normalized density results in the sample region, compared to results from Alsmeyer. The difference between the DSMC and experimental results is greatly exaggerated here to make the schematic clearer. The blue shaded region in the upper left image represents the error from Eq. 3. **b.)** Chain positions for a single MCMC chain, using Alsmeyer’s Mach 3.38 shock data. The chain wanders within a band where multiple good solutions exist.

Variable Hard Sphere Case

As discussed previously, the calibration for the VHS method involves two parameters, the VHS reference diameter (d_{ref}) and the temperature viscosity exponent (ω). The calibration data from Alsmeyer [1] are the same as that used above for the hard-sphere model. With only two parameters, it is still suitable to uniformly sample the parameter space. Figure 3a shows a contour plot of probability as a function of d_{ref} and ω . To obtain the results shown in Fig. 3a, the DSMC code is run at a total of 10,000 locations on a 100×100 mesh covering the parameter

space. At each location, the error term is calculated from Eq. 2, and then the probability is calculated from Eq. 4. From Fig. 3a, it is clear that the normalized density data alone does not provide enough information to uniquely calibrate both parameters. Instead, a band exists in parameter space, and combinations of d_{ref} and ω within this band can reproduce the density profiles of Alsmeyer to high accuracy.

We have also used MCMC to calibrate the VHS parameters. Due to the band structure mentioned above, a large number of MCMC-chain positions are required in order to provide a smooth histogram. Even though Fig. 3a demonstrates which areas of the parameter space will yield good solutions (and thus high MCMC probabilities), in cases with greater number of parameters it will not be possible to generate plots by the method used to create Fig. 3a. For example, to generate a mesh with only 100 grid points for each parameter would require 10^4 DSMC runs for a three-parameter case, 10^6 runs for a four-parameter case, and 10^8 runs for a five parameter case. For this reason, we have run the MCMC as if we did not have *a priori* knowledge of the regions of parameter space where the best solutions will be found. In general, it is important to run more than one MCMC chain because a chain can quickly move from its starting location into a region of high probability and some regions of the parameter space may never be explored. Thus, for our work, we have used 64 chains. Each chain contained 4000 positions, leading to a total of 256,000 DSMC simulations. This is of course far more than was required for the “brute force” method of Fig. 3a described above, but that is not unexpected, as using MCMC for a two-parameter system is overkill. Each chain was started with 64 processors (when a chain finishes, it gives its processors up to other chains), and the overall MCMC calibration required ~18 hours on 4096 processors. Note, however, that just one chain gives a reasonably good approximation of the band structure seen in Fig 3a, as shown by the chain in Fig 1b.

Figure 3b shows a scatterplot of all the MCMC chain positions for the calibration, and the band structure is clearly visible here as well. Figure 3c shows a contour plot of the number of MCMC chain positions in a given region of parameter space, based on a 40×40 mesh of histogram bins. Again, the band structure is clearly visible, and this figure matches quite well with Fig. 3a.

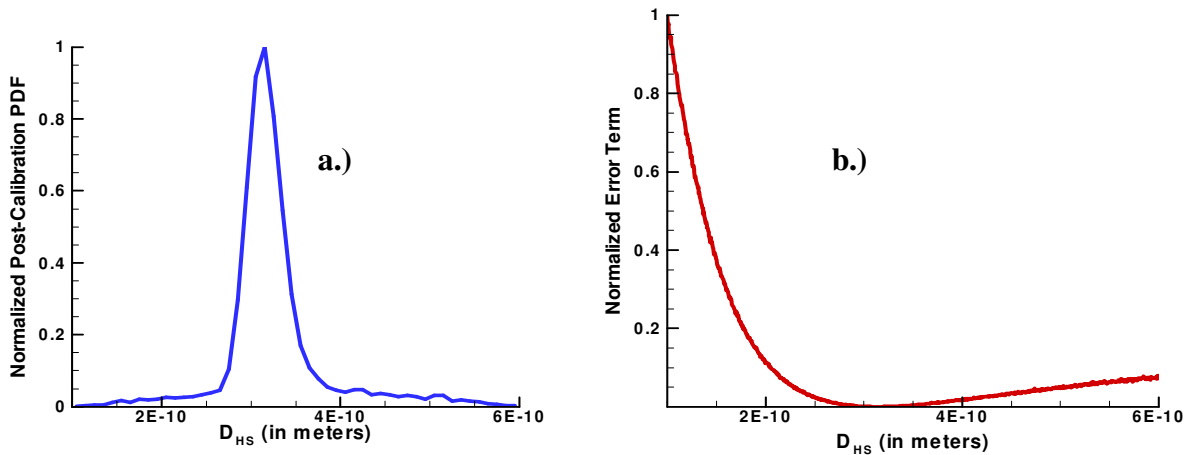


FIGURE 2. a.) Normalized post-calibration PDF for the hard sphere diameter of argon. Alsmeyer’s data for a Mach 3.38 shock was used in the calibration. b.) Error term (see Eq. 2) as a function of hard sphere diameter. For this figure, the error term has been normalized by its maximum value within the parameter range tested.

CONCLUSIONS

From the work shown here, we conclude that the MCMC method, while inefficient for systems with only one or two parameters, does provide a viable method for calibration of DSMC parameters, and that MCMC reproduces well the results of a uniform sampling of the parameter space.

We also conclude that the normalized density profile for one shock wave does not provide sufficient information to obtain unique values for both the VHS reference diameter and the viscosity temperature exponent simultaneously. This is true both for the uniform sampling and for the MCMC calibration. This is in agreement with the following statement by Valentini and Schwartzentruber [3]: “Thus, we argue that a comparison solely based on the density profiles, currently the most reliable experimental data that exist, may not be sufficient to assess the physical validity of collision models used to describe the shock wave structure, as is commonly done in previous studies.”

Assuming that there is a unique set of these two parameters which best represents the collision dynamics, it could most likely be obtained using other data which contain higher moments of the velocity distribution function, most likely temperature or even measurements of the velocity distribution function itself, such as those of Holtz and Muntz [5]. Future work will make use of these types of data.

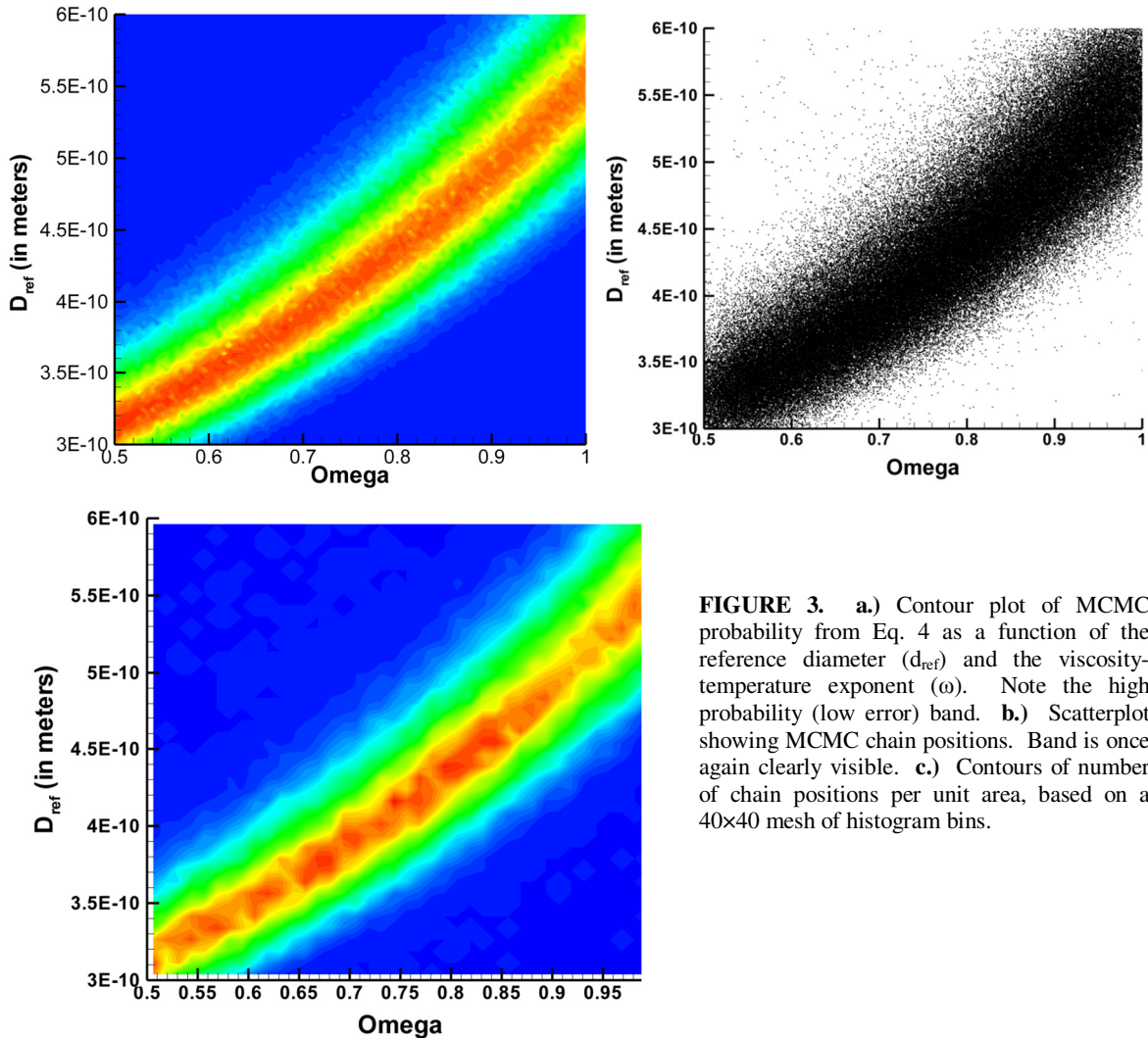


FIGURE 3. a.) Contour plot of MCMC probability from Eq. 4 as a function of the reference diameter (d_{ref}) and the viscosity-temperature exponent (ω). Note the high probability (low error) band. b.) Scatterplot showing MCMC chain positions. Band is once again clearly visible. c.) Contours of number of chain positions per unit area, based on a 40×40 mesh of histogram bins.

ACKNOWLEDGMENTS

Computing resources and funding for this work were provided by the DOE through the PSAAP program.

REFERENCES

- ¹H. Alsmeyer, *Journal of Fluid Mechanics*, **74**(3), 497-513 (1976).
- ²G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Univ. Press. Oxford, 1994.
- ³P. Valentini, T. E. Schwartzentruber, *Physics of Fluids*, **21**, (2009).
- ⁴S. Chib and E. Greenberg, *Journal of the American Statistical Association.*, **49**(4), 327-335 (1995).
- ⁵T. Holtz and E. P. Muntz, *Physics of Fluids*, **26**, pg. 2425-2436 (1983).