SANDIA REPORT

SAND2006-6692
Unlimited Release
Printed October 2006

Accelerating DSMC Data Extraction

Edward S. Piekos and Michael A. Gallis

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2006-6692
Unlimited Release
Printed October 2006

Accelerating DSMC Data Extraction

E.S. Piekos and M.A. Gallis
Microscale Science and Technology
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0826

Abstract

In many direct simulation Monte Carlo (DSMC) simulations, the majority of computation time is consumed after the flowfield reaches a steady state. This situation occurs when the desired output quantities are small compared to the background fluctuations. For example, gas flows in many microelectromechanical systems (MEMS) have mean speeds more than two orders of magnitude smaller than the thermal speeds of the molecules themselves. The current solution to this problem is to collect sufficient samples to achieve the desired resolution. This can be an arduous process because the error is inversely proportional to the square root of the number of samples so we must, for example, quadruple the samples to cut the error in half. This work is intended to improve this situation by employing more advanced techniques, from fields other than solely statistics, for determining the output quantities. Our strategy centers on exploiting information neglected by current techniques, which collect moments in each cell without regard to one another, values in neighboring cells, nor their evolution in time. Unlike many previous acceleration techniques that modify the method itself, the techniques examined in this work strictly post-process so they may be applied to any DSMC code without affecting its fidelity or generality. Many potential methods are drawn from successful applications in a diverse range of areas, from ultrasound imaging to financial market analysis. The most promising methods exploit relationships between variables in space, which always exist in DSMC due to the absence of shocks. Disparate techniques were shown to produce similar error reductions, suggesting that the results shown in this report may be typical of what is possible using these methods. Sample count reduction factors of approximately three to five were found to be typical, although factors exceeding ten were shown on some variables under some techniques.

Acknowledgments

The authors gratefully acknowledge the support of the Laboratory Directed Research and Development (LDRD) program. We are also grateful for the technical reviews of this document performed by Sean Kearney, Chris Bourdon, and Dan Rader.

Contents

Figures

Nomenclature

c	collision fraction	
C	conductance function, cost function	
f	filter coefficient	
g	normalized coherence measure	
k_B	Boltzmann constant, J/K	$1.381 \times 10^{-23}\ J/K$
M	atomic mass, $a.m.u.$	
n	number density, $particles/m^3$	
N	number of samples, molecules, etc.	
P	pressure, Pa	
t	time, s	
T	temperature, K	
u	input quantity	
v	flow speed, m/s	
w	weighting constant	
x	position in domain, m	
y	output quantity	
κ	normalized curvature	
λ	mean free path, m	
ρ	correlation coefficient	

1 Introduction

In the four decades since Graeme Bird originated direct simulation Monte Carlo (DSMC) [1], it has evolved into a remarkably flexible and accurate means for treating a wide range of molecular and thermal transport problems, from reentry vehicle aerodynamics to phonon transport in silicon films. In fact, for many strongly non-equilibrium problems, DSMC is the only available tool because it is not subject to the local equilibrium assumptions underlying traditional fluid and thermal simulation techniques.

As is generally the case in engineering, this advantage comes at a cost. DSMC's flexibility stems from its modeling of the Boltzmann transport equation rather than the more common equations derived from it, such as the Navier-Stokes equations and Fourier's law. The Boltzmann equation, however, has not proven tractable for general problems using continuous variable techniques, such as finite elements. DSMC therefore relies on a stochastic scheme for representing the distribution function that is the Boltzmann equation's dependent variable. Due to this stochastic representation, moments of the distribution, which provide the quantities of interest in most problems, such as number density and temperature, must be recovered via statistical sampling techniques. From a numerical standpoint, this means that additional computational work must be performed to recover the desired quantities from a field that has reached the correct state; a point generally considered as "converged" in other simulation techniques. Indeed, it is not unusual for DSMC simulations to consume the majority of their computation time after the flow field reaches a steady-state.

The difficulty associated with distinguishing a quantity from background noise generally increases if its magnitude is small compared to that of the background. The emergence of microelectromechanical systems (MEMS) has made this a common occurrence in DSMC. In these systems, mean flow speeds are often more than two orders of magnitude smaller than the thermal speeds of the molecules themselves. Furthermore, they often operate at atmospheric pressure, where collisions between molecules are common, also increasing computational expense.

Traditionally, the problem of stochastic noise is overcome with either very long runs or with the use of an excessively large number of particle simulators. This can be an arduous process because the error is inversely proportional to the square root of the number of samples so we must, for example, quadruple the number of samples to cut the error in half. This problem is what has given DSMC the reputation of a computationally intensive method.

This problem is not unique to DSMC but a common problem of all Monte Carlo algorithms. As computers have become more powerful, the importance of Monte Carlo algorithms has increased as researchers increase the fidelity of their simulations. Applications of these techniques can be found in modeling stochastic processes in physics, biology and even finance. It is therefore not surprising that noise or variance

reduction techniques have been extensively investigated and developed. This work presents a critical review of various available techniques applied specifically to DSMC simulations of low speed flow.

2 Theory

2.1 Statistical Sampling in DSMC

In DSMC simulations, the molecule simulators usually represent a far larger (typically 10-15 orders of magnitude) population of real molecules. The distribution function at a given instant in time is therefore much more sparsely populated in the simulation, which has the unwelcome effect of introducing statistical scatter. This scatter, in most cases, is several orders of magnitude greater than that present in the real gas. The uncertainty associated with an instantaneous sample is therefore typically unacceptable.

In standard practice, there are two ways to overcome this difficulty:

1. Perform a time average for steady flows or an ensemble average for unsteady flows.
2. Use an extremely large number of simulators per cell.

In the past, memory constraints typically precluded the second approach in all but the smallest problems. Modern computational platforms can accommodate several hundred million, or even billions, of simulators so it is now a viable, albeit costly, option.

Time averaging is employed in steady state problems and involves sampling over a number of time steps. After every time step, some of the molecules in a cell have collided, some have left, and some new molecules have entered the cell from its neighbors. This results in a partial reconstruction of the velocity distribution function in each cell at each step. Successive samples of the same flow field therefore result in different snapshots of the same velocity distribution function. Averaging over these snapshots will reduce the variance in the sampled domain properties.

Ensemble averaging involves sampling properties at the same instant (typically instants) in time over many flow realizations. In the limit of infinite samples, the ensemble average approaches the ensemble mean. Ensemble averaging is utilized in DSMC simulations for improving statistics in the unsteady state or when a parallel implementation of the problem would be inefficient (mainly due to small requirements that do not allow efficient distribution and parallelization).

The statistical scatter is generally assumed to follow the Poisson distribution, giving a standard deviation on the order of the inverse square root of the number of independent samples. The undesirable effect of this property is that for the statistical scatter to be reduced by one order of magnitude, two orders of magnitude more samples must be taken. This has a very significant impact on the total execution time.

At this point, we should note that the scatter in DSMC simulations is not solely a consequence of the numerical scheme. In fact, it has been demonstrated [2] that, assuming a one-to-one representation of real to simulated molecules, the noise DSMC reproduces has the same characteristics as physical noise.

It may be argued that DSMC simulations in 1-D or 2-D are, in fact, one-to-one representations. In these simulations, we typically assume that the cross sectional area or the depth of the domain, respectively, is unity. For example, in a 1-D simulation of a domain of length L, we can assume a cross sectional area, A, of unity. If n is the real number density in the flow represented by N simulators with a simulation ratio of S, we have for the number density:

$$n = \frac{NS}{LA} \tag{1}$$

It therefore makes no mathematical difference if we assume the simulation ratio equals unity and the cross sectional area is modified accordingly to give us the correct number density. In this case, any fluctuation observed in the simulation will have physical meaning. The importance of these fluctuations can be demonstrated by considering the calculation of temperature. Temperature is typically calculated as a function of the deviation of the molecular velocities from the average value. A flow where all molecules have the average flow velocity would therefore have a temperature of zero (solid body motion).

Concluding, we expect any DSMC simulation to produce some level of noise because this is a feature of the flow. (A feature that the macroscopic description of continuum mechanics fails to address.) In the vast majority of work, however, we are interested in macroscopic quantities that are moments of the distribution function. In this context, the noise is manifested as uncertainty in a given moment, which we would like to minimize.

A systematic noise reduction study for DSMC had not been performed until fairly recently. This has led to a number of misconceptions about the accuracy of the method and its convergence speed. In their recent work, Rader *et al.* [3] studied the error in DSMC as a function of the spatial, temporal and velocity space discretization parameters. To isolate the discretization error from the statistical error, excessively long runs were performed.

The work described in this document takes the opposite approach. Herein, we focus on the statistical error and explore methods to reduce it in a strictly post-processing manner. The methods are therefore applicable to any DSMC code and they do not affect the accuracy of the DSMC algorithm itself.

2.2 Key Concepts from Statistics

Monte Carlo algorithms use sampling methods drawn from standard statistical techniques. It is therefore useful at this point to define a few terms from statistics.

2.2.1 Variance

Typically a set of measurements or estimations have the tendency to gather around some particular value, best known as the mean value. So, if $u_1, u_2, ...u_N$ are N collected values, the mean is given as

$$\bar{u} = \frac{\sum_{i=1}^{N} u_i}{N} \tag{2}$$

Having characterized the central value of a distribution of points we can calculate the "width" or the "variance" around that value. The most common definition of the variance is given as

$$Var[u] = \frac{\sum_{i=1}^{N} (u_i - \bar{u})^2}{N-1} \tag{3}$$

Another common measure is the standard deviation, which is the square root of the variance:

$$\sigma[u] = \sqrt{Var[u]} \tag{4}$$

The estimate of the variance of a measurement is a crucial piece of information. In many cases it is worth repeating an experiment many times so an accurate measure of the variance can be obtained. In numerical Monte Carlo simulations "repeating the experiment" is equivalent to running the problem again with a different random number generator or with a different seed. This technique has been extensively used in the Monte-Carlo literature for variance estimation [3, 4].

The quality of a Monte Carlo result can be assessed by computing its variance: the smaller the variance the better the simulation. Our preoccupation with variance reduction is therefore aimed at extracting the maximum physical information from the data on hand without having to repeat our simulations to obtain more data.

2.2.2 The Central Limit Theorem

In its simplest form, the central limit theorem states that the sum of a large number of independent observations from an arbitrary distribution, under certain general con-

ditions, is approximately normally distributed. Moreover, the approximation steadily improves as the number of terms in the sum increases. Dividing by the number of observations, this theorem is shown to apply to the average observed value as well.

The central limit theorem, perhaps more aptly named the "normal convergence theorem," is the foundation for many statistical procedures, including the control charts [5] popularized by Deming for ensuring quality in manufacturing processes. Its power is drawn from an ability to group diverse processes under a common mathematical theory because the distribution of the phenomenon under study does not have to be normal to ensure that its average will be.

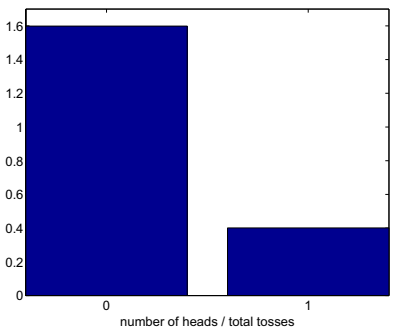
For example, suppose a coin is tossed a number of times and the number of heads is counted. This is equivalent to scoring 1 for a head and 0 for a tail and computing the total score for each trial. This score is thus is the sum of a number of independent samples from a distribution with a constant value of 0.5. By the central limit theorem, the observed number of heads will become increasingly normal as the number of tosses in each trial is increased.

The central limit theorem is illustrated graphically in Fig. 1. To create asymmetry in the distribution, an unfair coin with an 80% chance of landing on "tails" is employed. For a given number of tosses, the total score (the number of heads) is arranged along the horizontal axis, normalized by the number of tosses in the trial. The vertical axis shows the probability distribution function for the trial. With only one toss per trial, the distribution appears decidedly non-normal and asymmetric (due to the unfair coin). With only ten tosses per trial, the distribution is already becoming similar in form to the normal distribution, with a peak in the center and smaller values on either end, though it is still asymmetric. With only 100 tosses per trial, the shape strongly resembles the normal distribution, showing symmetry around the expected value, and a slope that increases then decreases as one moves away from the peak. With more tosses, the distribution simply becomes narrower, reflecting the reduction in standard deviation achieved with increased sampling.

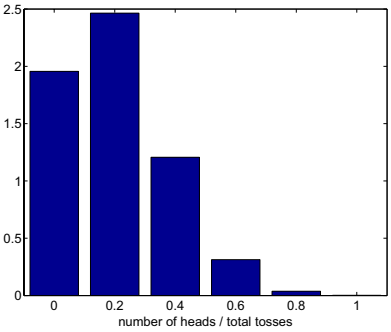
It has been empirically observed that various natural phenomena, such as the heights of individuals, follow an approximately normal distribution. A suggested explanation is that these phenomena are sums of a large number of independent random effects and hence are approximately normally distributed according to the central limit theorem.

The central limit theorem is relevant to sampling in DSMC because the mean of the normal distribution is simply the arithmetic mean. This implies that the best estimate of a quantity produced from independent random processes is the arithmetic mean of multiple samples of this quantity. As described in the previous section, this is the typical approach used in DSMC. Backed by such a well-accepted theory, one could reasonably wonder whether it is possible for such an estimate to be improved.

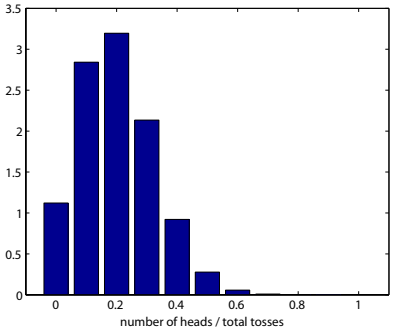
The answer to this question lies in the central limit theorem's implicit assumption



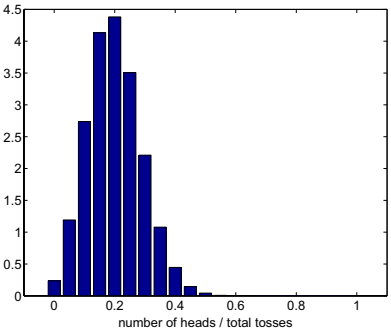
(a) 1 toss



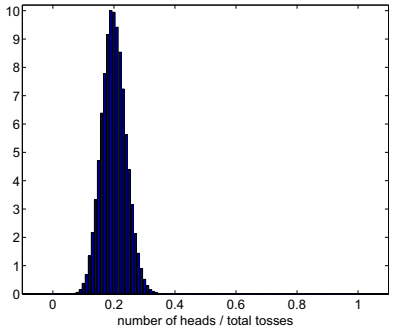
(b) 5 tosses



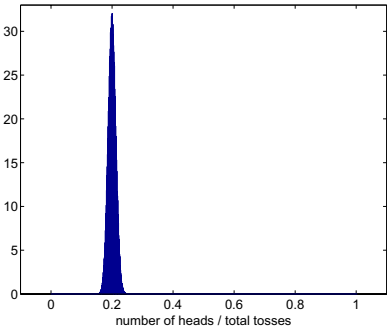
(c) 10 tosses



(d) 20 tosses



(e) 100 tosses



(f) 1000 tosses

Figure 1. Demonstration of the central limit theorem for the sum of a varying number of tosses of a coin with an 80% chance of landing on "tails." As the number of tosses increases, the distribution increasingly resembles the normal distribution.

that the samples are statistically independent. In the above example, the outcome of a coin toss is independent of previous and following tosses. However, in a fluid system, where computational cells are connected by fluxes and only a small fraction of the velocity distribution function changes in each time step, sequential samples are not independent. The central limit theorem is therefore not strictly applicable and some improvement is possible.

2.2.3 Correlated vs. Uncorrelated Samples

Let a and b be two random variables, the sum of which we want to calculate. Assume that somehow an accurate prediction of variable a is known. To calculate their sum, two options immediately spring to mind: 1) make an estimate of both a and b and sum them, 2) make an estimate of b and sum the accurate prediction of a to it. Clearly, the latter option sounds more appealing because it uses additional information to reduce the total variance. However, there is a third option. If there is a known relationship between the variables we can take advantage of it to improve our guess of the unknown variable. Assuming, for example, that there is a linear relationship between the two variables, and assuming we have an accurate prediction of a and its expected value $E[a]$, we can express b as

$$b = (a/E[a])E[b]. \tag{5}$$

In general, if variables a and b are correlated, this estimation of b will reduce its variance and therefore we can achieve a greater variance reduction for the sum than using the more intuitive scheme (2) from above.

2.3 Can DSMC Data Extraction be Accelerated?

In this work, we introduce a new way of looking into DSMC data based on the correlations between variables. The central limit theorem dictates that for a number of uncorrelated samples, their average is the best estimate for their mean. However, as demonstrated in the preceding section, if some variables are correlated, the estimate of their values can be improved based on the knowledge of other variables. It should be clear by this point that, in general, some kind of additional information about the problem must be introduced to improve upon results obtained by basic averaging. This additional information may come in the form of constitutive relations, known problem parameters, or simply knowledge of the flow field.

DSMC-calculated hydrodynamic variables are correlated through either molecular fluxes or conservation and constitutive relations. For example, we know from basic kinetic theory that the pressure, density, and temperature at a particular location in physical space are correlated through the ideal gas law:

$$P = nk_bT, \tag{6}$$

where n is the number density, T is the temperature and k_b is the Boltzmann constant.

Additionally, hydrodynamic variables in neighboring cells are correlated through molecular fluxes. If a deficit of mass occurs in a cell, molecules from the neighboring cells will diffuse towards it to eliminate the difference. Therefore, two samples or "snapshots" of the same flow field taken after successive time steps are expected to be correlated. It is only after a sufficient number of collisions that memory of the previous state is erased.

We therefore conclude that DSMC data are indeed correlated so some improvement over the standard practice of averaging variables independently should be possible. The stronger the correlation between variables, the greater the improvement will be. A more mathematical description of this concept will be given in the following sections.

2.4 Characteristics of an Ideal Method

Before beginning this work (and becoming encumbered by reality), we can compile a wish list of properties for the variance reduction method we seek.

An ideal method would be:

- **Physically realistic:** The method should be able to remove statistical noise without distorting the physical solution.
- **Numerically efficient:** The purpose of developing such a method is to reduce compute time, so the efficiency of the scheme is important.
- **General:** A method should be able to be applied blindly by non-experts without running the danger of degrading the results.
- **Local:** All operations of the method should be local, *i.e.* it shouldn't make any use of neighboring information in space or time.
- **Three-dimensional:** Although 1-D methods are very useful (because 1-D codes are often used for detailed physics simulations) an ideal method would be applicable to three dimensional flows, where the computational requirements are most daunting.

Relaxing one or more of these requirements may yield a more powerful method, although one that must be applied by a knowledgeable user to avoid negatively impacting the physical fidelity of the results.

3 Test Problem and Evaluation Methods

3.1 Test Problem

Particularly for methods that work on inter-cell relationships, a problem with spatial features of some sort is desirable. Otherwise, a simple average across the domain is the most effective method. The Fourier problem was therefore chosen to provide data for the methods examined in Sec. 5.

The Fourier problem consists of a quiescent flow between two flat plates of unequal temperature. The problem is mathematically one dimensional. Due to its geometrical simplicity, it has been extensively used as a benchmark case in kinetic theory and fluid mechanics. In accordance with most of these studies, the gas between the two flat plates is assumed in this work to be hard-sphere argon. The molecular properties are given in Table 1.

molecular diameter	$3.66 \times 10^{-10}\, m$
molecular mass	$6.63 \times 10^{-26}\, kg$
viscosity coefficient	0.5
temperature at cold wall	$223\, K$
temperature at hot wall	$323\, K$
domain length	$1\, mm$
pressure	$2\, Torr$

Table 1. Parameters for Fourier problem test case.

At these conditions, the Knudsen number based on channel height is $Kn = 0.024$. This value is high enough to imply that nonequilibrium Knudsen layers of significant thickness will form near the boundaries but low enough to indicate that an equilibrium region will be present in the domain center. This set of conditions therefore allows the methods to be tested against regions of both types.

Figure 2 presents well-converged results for this case. As expected, the temperature profile is almost linear between the cold (left) and hot (right) wall. Near the two walls, highly non-equilibrium Knudsen layer regions form. In these regions, the velocity distribution function gradually changes from the Chapman-Enskog found in the middle of the domain to the half-range Maxwellian found adjacent to the walls.

This problem provides a reasonable representation of conditions found in DSMC simulations of microdevices and it has been carefully characterized by Gallis *et al.*. [3, 6, 7].

The simulation was performed using the ASC/SIERRA DSMC code FAUST run-

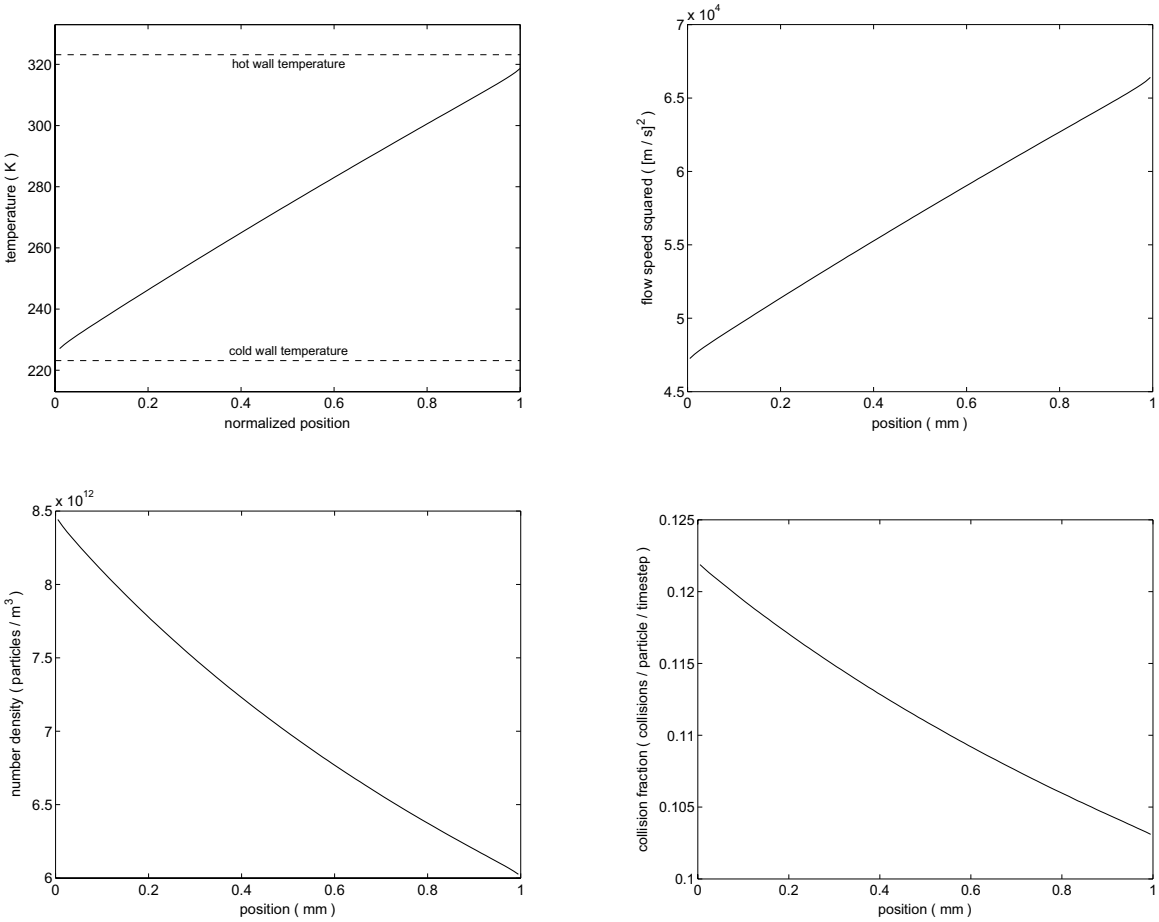


Figure 2. Well-converged results for the Fourier problem with parameters shown in Table 1.

ning in two dimensions, with a 1mm×1mm domain discretized into 100×100 cells. This calculation was carried out to 2.4 million samples with output every 10,000 samples to provide a large number of snapshots as well as convergence data for the traditional treatment.

When evaluations are performed in one dimension, the output is broken into 100 1-D results and the reported performance metrics for each method represent averages over independent applications to each result. Performance in all cases was measured in terms of an RMS error between the smoothed data and a result containing over a billion samples.

Extending the methods from one to multiple dimensions is not clear in many cases. This issue may be circumvented by acting along each dimension separately, treating, for example, a three-dimensional dataset as three one-dimensional datasets. The

striping may be applied sequentially or it may be applied independently, averaging the results to reach a final product.

3.2 Evaluation Methods

Error Ratio for a Given Number of Samples

The "standard DSMC" error is determined by computing a root-mean-squared (RMS) difference between the unmodified output from the model simulation and either an analytical solution (in the case of flow speed, which is zero), or a DSMC solution with otherwise identical parameters carried to over a billion samples. The reported "error ratio" is this error divided an error calculated using the same procedure, but using the output from the method under study instead of the standard DSMC output.

The advantage of this measure is that it allows a very compact representation of the performance of a method. In particular, all studied variables can be presented on one plot because the error in the post-processed result is normalized by the error in the unprocessed (standard DSMC) result at each sample count. Values above unity represent an improvement, while values below unity represent an increase in the RMS error over the standard technique.

One disadvantage of this method is that the error ratio can a be misleading measure for variables such as temperature, which tend to have a lower error at a given number of samples, because it is harder to improve upon the unmodified result.

Samples Required to Reach Specified Error

Plots of this form are given in the summary at the end of a section. They present the same data used in error ratio plots, but without computing the ratio. The unnormalized RMS error is therefore visible, as is its behavior as the number of samples is increased. By comparing two lines on this plot, a measure of the reduction in DSMC computation time achievable by a given method can be obtained. This format provides a more level playing field for comparing results on variables with differing convergence rates. It also provides a useful format for comparing the results of multiple methods. Only one variable can be reasonably evaluated on a single axis in this format, however.

Computation Time

While the computation time of the post processing methods studied in this work is generally small compared to that of the DSMC run, computation time remains a useful quantity for comparing methods to one another. When many datasets are to be examined, or where large datasets generated on massively parallel machines are post-processed on more modest hardware, computation time can be an important consideration when choosing a method.

4 Exploiting Intra-Cell Relationships

This chapter discusses methods for decreasing the variance of Monte Carlo simulations based on relationships between variables measured within the same cell. These relationships represent known features of the problem, for example the fact that a flow field may be isobaric or isothermal, or known from theory relationships between hydrodynamic variables.

4.1 Control Variate Method

The control variate method is among the most broadly applicable and potentially most effective techniques for improving the efficiency of Monte Carlo Simulations [8]. It is based on the existence of correlations between data and it uses the errors in a known quantity to reduce the error and estimate an unknown quantity. Besides its value as a variance reduction method, it offers a good mathematical framework for understanding the operation of variance reduction techniques in general.

We will first demonstrate the method with an example. Assume that we want to estimate the average temperature in a room from a series of temperature measurements. Assuming that these measurements are independent of each other (a very reasonable assumption) the obvious answer is to sum all the temperatures and divide the sum by the number of measurements, *i.e.*

$$\bar{T} = \frac{\sum_{i=1}^{N} T_i}{N} \tag{7}$$

Now suppose we have measurements of the relative humidity taken in the same room at the same time as the temperature measurements. Thermodynamics tells us that the higher the temperature of air the lower the relative humidity (absent a source of moisture). Humidity and temperature are therefore correlated.

Could we use this information about the humidity to improve our estimate of the temperature? The answer is *yes*. The method that allows us to do that is the control variate method. The method does not violate the central limit theorem because it adds information regarding the correlation between variables. Basically we add what we know about the system in question to improve our estimate.

In a more general case, let $Y_1, Y_2, ...Y_N$ be outputs of N replications of a simulation. We assume that the Y_i quantities are all independent from each other. Our objective is to obtain the average value $\bar{Y}$. The usual estimator is the average of all samples:

$$\bar{Y} = (Y_1 + Y_2 + ... + Y_N)/N. \tag{8}$$

This estimator converges to the correct value as $N \to \infty$.

Suppose now that on each replication of the experiment we calculate another output X_i of known expectation $E[X]$. Then for a fixed value b and for the ith replication we can compute

$$Y_i(b) = Y_i - b(X_i - E[X]). \tag{9}$$

The average $\bar{Y}$ is therefore

$$\bar{Y}(b) = \bar{Y} - b(\bar{X} - E[X]) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - b(X_i - E[X])) \tag{10}$$

The quantity $\bar{X} - E[X]$ serves as a control in estimating $E[Y]$. As an estimator of $E[Y]$, the control variate estimator $\bar{Y}(b)$ is unbiased because:

$$E[\bar{Y}(b)] = E[\bar{Y} - b((\bar{X} - E[X])] = E[\bar{Y}] = E[Y], \tag{11}$$

i.e. the estimation of the average of Y equals that of $Y(b)$.

The variance of $Y_i(b)$ is

$$Var[Y_i(b)] = Var[Y_i - b(X_i - E[X])], \tag{12}$$

which is

$$Var[Y_i(b)] = \sigma_Y^2 - 2b\sigma_X\sigma_Y\rho_{XY} + b^2\sigma_X^2, \tag{13}$$

where $Var[X] = \sigma_X^2$, $Var[Y] = \sigma_Y^2$ and ρ_{XY} is the *covariance* or the *correlation* between X and Y.

The control variate estimator $\bar{Y}(b)$ has variance $\sigma^2(b)/N$ and the ordinary sample mean $\bar{Y}$ has variance σ_Y^2/N. Thus, the control variate estimator has smaller variance than the standard estimator if $b^2\sigma_X < 2b\sigma_Y\rho_{XY}$.

Parameter b can be selected so that it minimizes the variance of Y. By differentiating Eq. 13 we can calculate the value of b as

$$b = \frac{\sigma_Y}{\sigma_X}\rho_{XY} = \frac{Cov[X,Y]}{Var[X]}. \tag{14}$$

Substituting b in equation Eq. 13:

$$Var[\bar{Y}(b)] = Var[Y] - \frac{Cov(Y,X)^2}{Var[X]} = Var[\bar{Y}] - \frac{Cov(Y,X)^2}{Var[X]}. \tag{15}$$

Therefore, we must have $Cov(Y,X) \neq 0$ to achieve a variance reduction.

The problem with this relationship is that we often don't know the $Cov(Y,X)$. In such cases, we can perform N *pilot* simulations setting

$$Cov(X,Y) = \frac{\sum_{j=1}^{p}(Y_j - \bar{Y}_p)(X_j - E[X])}{N-1}. \tag{16}$$

If the variance of X is also unknown we can estimate it with

$$Var(X) = \frac{\sum_{j=1}^{p}(X_j - E[X])^2}{N-1}. \tag{17}$$

If no such simulations are available the same data to be analyzed can be used to estimate the covariance and the average X. When these parameters are not known and have to be estimated based on a small sample it is theoretically possible for the control variate method to increase the variance.

The ratio of the variance of the optimally controlled estimator to that of the uncontrolled estimator as

$$\frac{Var[\bar{Y} - b^*(\bar{X} - E[X])]}{Var[\bar{Y}]} = 1 - \rho_{XY}^2. \tag{18}$$

From this last equation we note that the variance reduction factor $\frac{1}{1-\rho_{XY}^2}$ increases rapidly as the correlation $|\rho_{XY}|$ approaches unity. For example, a correlation of 0.95 produces a ten-fold speed-up while a correlation of 0.980 produces a speed-up of twenty five. It is therefore necessary for a strong correlation to exist for the control variate to produce significant time savings.

4.1.1 Physical Meaning of the Control Variate Method

The physical meaning of the control variate method can be easily understood by examining Eq. 9 and Eq. 14. The b parameter is the slope of the least-squares linear regression that goes through points $(X_i, Y_i),\ i = 1, N$. The link between control variates and regression is useful in the statistical analysis of the control variate estimators and also permits a graphical interpretation of the method.

A linear correlation between the variables is not necessary. In the same way that non-linear regression (see Bevington and Robinson [9]) is defined we can define non-linear correlations (exponential for example). Naturally, the form of the correlation must be known a priori, which may not be possible.

4.1.2 Control Variate Method Application Example:Fourier Flow

Due to the Fourier problem's simplicity, there are properties of the flow field that are known a priori. For example the pressure in the domain is known because it remains constant. From the ideal gas law (Eq. 6), we can expect that pressure will be strongly correlated with number density and temperature. More explicitly, assuming an isobaric flow field, temperature and inverse density are linearly correlated.

Performing the statistical analysis and plotting the results in Fig. 3, we find that the correlation between pressure and temperature for this particular flow field, measured parallel to the gradient and averaged across the domain, increases with the number of samples, reaching an asymptote of 70% at the billion sample reference (not shown). For the sub-300k sample counts chosen as the benchmarks for this work, this correlation is extremely low, struggling to reach 15% at 300,000 samples. From Eq. 18, we would therefore expect a maximum reduction of the variance of only 2% for the test suite. Even worse, this variance reduction is even lower at the smaller sample counts, where we would like a greater reduction because the data is noisier.

Looking perpendicular to the gradient, a nearly constant correlation of approximately 30% is visible in Fig. 3, corresponding to a possible variance reduction of under 10%. This already unimpressive variance reduction is further reduced for one-dimensional problems by the need to collect multiple sets of samples to provide data in the cross-gradient direction.

We can perform the same analysis for any other hydrodynamic variable as long as it is significantly correlated with other known variables. In practice, this proves to be a more stringent restriction than anticipated. Hydrodynamic correlations exist only between a few variables while there are moments of the Boltzmann equation that may not be correlated to anything of known value. For example, it was observed that the flow speed is not correlated to any of the other four variables treated in this work - not even the flow speed squared.

4.2 Multiple Control Variate Method

In our analysis so far we have restricted ourselves to a single control variate. The analysis in the previous section can easily be extended to more than one variate with beneficial results to variance reduction, assuming significant correlations exist.

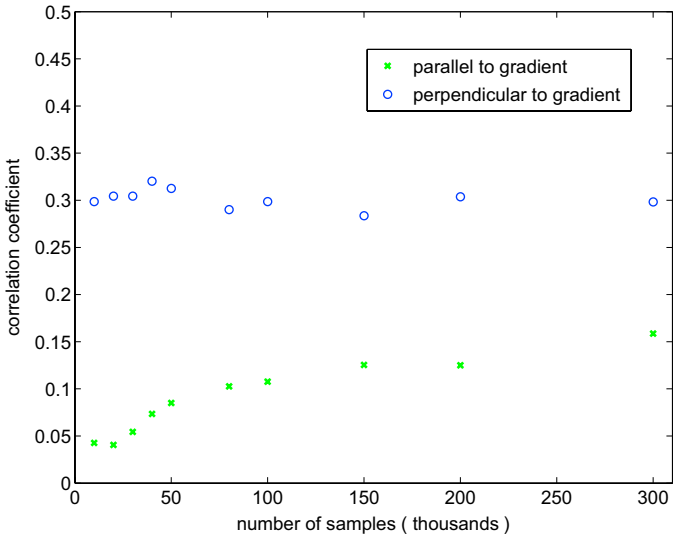


Figure 3. Correlation coefficient, measured parallel and perpendicular to the gradient, for the two dimensional Fourier problem test case described in Sec. 3.1.

In a similar fashion to the single variate theory, we assume a variate Y for which we wish to minimize the variation of its estimate. Also assume a series on M variates (also problem outputs) whose estimation is somehow known. An example for such a scenario in a DSMC simulation could be the simulation of a MEMS device when we know the pressure, the temperature, or the velocity of the gas.

In the case when the estimation of more than one variate is known we can calculate a series of b_is so the the estimation of the optimized Y has a smaller variance than the usual estimator.

As before, we can write the estimator $E[Y]$ as

$$\bar{Y}(b) = \bar{Y} + b_1(X_1 - E[X_1]) + b_2(X_2 - E[X_2]) + ... + b_m(X_m - E[X_m]). \qquad (19)$$

Our goal now is the selection of b_is that will minimize the variance of Y. To do so we can take partial derivatives with respect to b_i set them equal to 0 and solve the system of m linear equations with m unknowns. The optimization parameters b_i will involve the estimation of the correlation functions and variances of the parameters. These can be either known or calculated using a pilot simulation.

Finally the variance of the control variate can be written:

$$Var[\bar{Y}(b)] = Var[Y] + 2\sum_{i=1}^{m} b_i Cov(Y, X_i) + \sum_{i=1}^{m}\sum_{i=1}^{m} b_i b_j Cov(X_i, X_j). \qquad (20)$$

4.2.1 Multiple Variates for the Fourier Problem

As noted in the previous section, it is difficult to find the strong correlations between variables required to have a significant impact on the variance in the Fourier problem, as with most other fluid problems. It is therefore *much* more difficult to find correlations between multiple variables.

Looking toward the ideal gas law of Eq. 6, one possibility presents itself: the number density is correlated to the pressure and the temperature. In cases where expected values for both these quantities are known, an improvement should therefore be possible. Unfortunately, it is difficult to envision such a situation in a practical problem. For thermally-driven MEMS problems, the pressure is known, but the temperature is not. For damping problems, the reverse is true. Some knowledge of the general shape of the fields is often possible, but the uncertainty in these estimates will likely cancel the slight improvement possible with the weak correlations observed in the previous section.

4.3 Exploring Time and Other Relations

4.3.1 Common Variates

Quite often in DSMC simulations, the objective is not to estimate a parameter represented by a single moment of the velocity distribution function, such as the number density, but to estimate a parameter that may be a function of several other moments. If the velocity distribution function moments are positively correlated, it is possible that the estimated parameter may have a smaller variance than the constitutive moments. In fact, this happens quite often in DSMC. The temperature, heat flux, conductivity and viscosity, to name but a few, are all calculated as combinations of velocity distribution function moments.

This beneficial effect was observed by Rader *et al.* [3] when estimating the thermal conductivity of argon, which is calculated as the ratio of the heat flux to the temperature difference.

4.3.2 Stratified Sampling

Stratified and Importance Sampling are other methods that can be used in Monte Carlo simulations to reduce variance [10]. It easy to visualize how the methods work with a simple Monte Carlo integration example. Assuming that we have a function we want to integrate we can make sure that a larger fraction of samples are taken from important areas or specified regions ("strata"). This way, regions with more significant contributions (larger values) are sampled more heavily.

Despite the relative success of this method in some Monte Carlo methods, it is unlikely that its application will improve DSMC simulations. The physical accuracy of DSMC simulations depends on their ability to capture unlikely events, *i.e.* events that only take place in the tails of the distribution. Such events may be rare chemical reactions, or excitation of energy states, which, although rare, may contribute significantly to the formation of the flow field.

4.3.3 Temporal Correlations

Time is another independent coordinate of the Boltzmann equation. DSMC simulation data present non-vanishing correlations in time. These correlations are, in fact, much stronger than instantaneous spatial correlations (see, for example, Figs. 11.19 and 11.20 in Bird [1]).

DSMC steady state sampling can be seen as taking advantage of temporal correlations in steady state. Since we know that steady state has been reached, all averages (snapshots) are expected to have the same average value. Therefore, a horizontal line is expected to predict all data points. To realize this form, the correlation factor b in Eq. 9 should vanish, and the corrected expectation reduces to the mere average of the sampled values.

Even with this extremely simple sampling scheme, a parameter must still be chosen: the number of time steps between samples. Hadjiconstantinou *et al.* [11] studied temporal correlations and concluded that, in a typical DSMC simulation, it takes $O(10)$ moves for two sequential samples to be statistically independent. After numerous simulations performed for this study, this number was found to be closer to 20. Sampling more often than every 20 time steps therefore adds very little knowledge of the flow field while adding a non-negligible overhead to the simulation. Typically, a DSMC simulation sampling after every move spends about 20% of its time sampling. Reducing the sampling frequency to once every 20 moves can therefore provide a non-trivial performance increase.

4.4 Summary

The control variate method was applied to reduce the statistical noise in DSMC flow fields. This method takes advantage of correlations between variables and known quantities in the problem to achieve variance reductions. This method is one of the most popular in the general area of Monte Carlo simulations because it has a solid mathematical background and it is easy to understand and implement. This method also has the advantage of being local, *i.e.* only information coming from the same cell is considered. It can therefore be applied regardless of spatial dimension (velocity is always three-dimensional in DSMC) and there is no danger of smearing physical features.

The control variate method therefore satisfies nearly all the requirements listed for an ideal technique in Sec. 2.4. Unfortunately, it is also very weak for the test problem considered here, which is representative of flows found in MEMS. This weakness stems from a lack of strongly correlated data. In all but the most over-resolved (in terms of particles per cell), highly-collisional flows, the fluctuations in DSMC simulations are a consequence of an instantaneous local nonequilibrium that is not amenable to correlations between hydrodynamic variables. Furthermore, even in reasonably resolved flows, certain variables have weak or nonexistent correlations to the others, making these techniques useless for improving upon the estimates derived by standard DSMC sampling.

5 Exploiting Inter-Cell Relationships

The correlations we have examined so far are local, meaning that all necessary information comes from the same computational cell and instant in time as the variable in question. Such intra-cell correlations are not the only possibilities however: DSMC cells are connected to each other through molecular fluxes. Thus molecular properties in one cell are to some degree correlated to those in the neighboring cells. Keeping in mind that the DSMC cells are typically much smaller than a mean free path, and that it takes several mean free paths for a molecular property to be modified through molecular collisions, we can establish correlations between neighboring cells.

Spatial correlations appear as a particular shape of the spatial distribution of a hydrodynamic variable. These correlations are always present because they represent the features of the flow field. In more concrete terms, we can expect all variables in the flowfield to be continuous (even across a shock in DSMC), thus the values in neighboring cells are correlated to some degree. Sudden changes can therefore be attributed to statistical noise and targeted for removal.

In terms of the control variate method presented in the previous section, a spatial filter applied to a one-dimensional flow can be seen as single variate method, correlating position, x, with a hydrodynamic variable. In two or three dimensional flows the spatial filters can be seen as multiple control variate methods correlating x, y, z with a hydrodynamic variable.

Naturally, correlations between spatial and hydrodynamic variables are not always linear. There are two options in such a case: (1) employ a linear filter over a small area of the domain to minimize the departure from linearity or (2) employ a different functional form. Both options are examined in this chapter.

5.1 Single-Pass Methods

The methods described in this section are distinguished by the fact that they do not iterate, *i.e.* the DSMC data pass through the algorithms only once. These techniques therefore generally require fewer computational resources than the iterative methods. Their non-iterative nature also obviates the need to define a convergence criterion. For most of these methods, the footprint is limited and easily defined, though this is not true of the Fourier methods, whose footprint is the entire domain.

5.1.1 Digital Filtering

Theory

At their core, digital filters are a means for creating an output quantity y from an input u sampled at discrete points via convolution with one or more sets of coefficients.

Denoting the sample taken at the n^{th} point as u_n, one can define two major types [12] of digital filters: nonrecursive filters, where

$$y_n = \sum_{k=-N}^{N} f_k u_{n-k} \tag{21}$$

and recursive filters, where

$$y_n = \sum_{0}^{N} f_k u_{n-k} + \sum_{1}^{M} g_k y_{n-k}. \tag{22}$$

It should be noted that the above expressions implicitly assume a uniform data spacing. From a DSMC standpoint this implies that they are applicable only to structured grids, which is not overly restrictive because many popular codes are indeed structured, at least in a block sense. While it is possible to relax this constraint, variable coefficients and significant additional complexity would be required. Such schemes are not treated in this work. Alternatively, unstructured data could be interpolated to a structured grid before filtering.

The nonrecursive digital filters may be viewed as "moving window averages," where a data point is replaced with a weighted average of itself and a select group of its neighbors. For example, in one dimension, if $k = 3$ and $c_k = 1/3$ for all k, we have a simple 3-point average, where the value at a point is replaced by the average of itself and its two nearest neighbors. The differences between the filters are simply the size of the group and the weighting scheme.

The weighting factors (the c_k) can be determined in a number of ways. From a signal processing standpoint, we wish to choose weights that produce a transfer function in the form of an ideal low pass filter, passing low spatial frequencies without modification and removing high frequencies. In practice, we try to attenuate high frequencies and minimize the modification of low frequencies.

The filters evaluated in this section were drawn from Hamming [12] and include filters with weights derived by a least squares fit using both linear and quadratic trial functions, and modified versions of all these filters proposed by Hamming that improve their performance.

As the filter approaches the domain edges, its window increasingly includes areas outside the domain. The handling of these data points can affect overall filter performance, so a careful decision must be made. Possible methods include fixing the affected points, modifying the filter, and extrapolating the data.

Fixing the data points near the boundary is the most general option. Any point for which a complete dataset cannot be constructed is simply left unmodified. This choice avoids the danger of biasing the boundary in a nonphysical way. It does, however, eliminate the possibility of improving the values at these points. For the

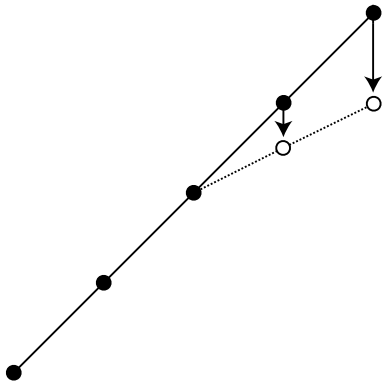


Figure 4. Demonstration of distortion of a unity-slope line if a five point average filter is re-weighted for missing points at the end.

large-footprint filters this can leave a significant number of points out of the process, lowering the overall achievable error reduction.

Modifying the filter near the boundary is another option. One possibility is to remove any part of the filter window that is outside the domain and redistribute the associated weight to the remaining points. An equal distribution of weights is generally non-optimal. For example, consider a simple five point average acting on linear data with unit spacing and unity slope. In this case, when the filter is acting on the last point in the domain, this point will be displaced downward by

$$\begin{aligned} y_n &= \frac{1}{3}[u_{n-2} + u_{n-1} + u_n] \\ &= \frac{1}{3}[(u_n - 2) + (u_n - 1) + u_n] \\ &= u_n - 1 \end{aligned} \tag{23}$$

Similarly, the point just inside the last point will be displaced downward by 1/2 unit. This case is illustrated graphically in Fig. 4.

Extrapolating the data at the boundary in some way provides a means for applying any filter to all points in the domain without modification. The method of extrapolation chosen for analysis in this section is a double reflection. Under this scheme, interior data is reflected across both a horizontal and a vertical line (1-D data) to extend the domain. The effect of this scheme is demonstrated for noisy collision fraction data in Fig. 5. The extension of this method to multiple dimensions was not investigated. As noted earlier, however, multidimensional data can be treated with multiple perpendicular applications of one-dimensional filters.

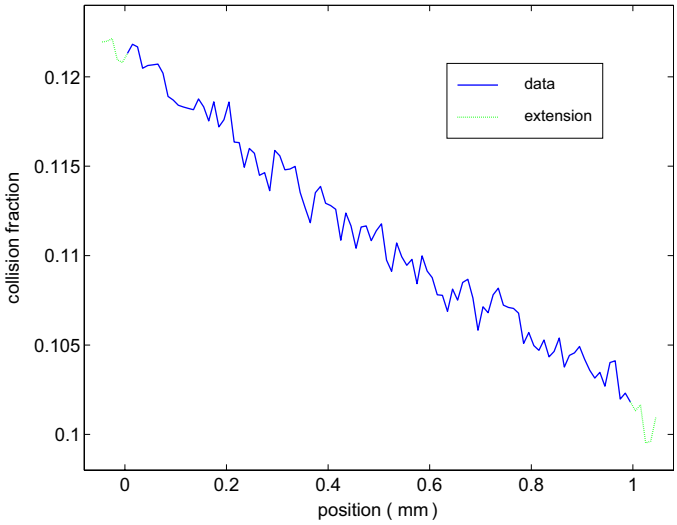


Figure 5. Demonstration of reflection boundary condition for collision fraction with five points added on each end.

Results

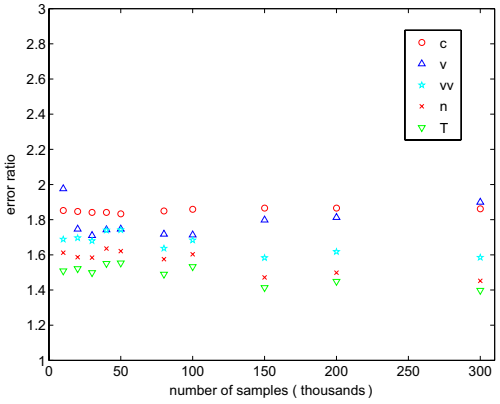
The Fourier problem, described in Sec. 3.1, provides the test case for the filters evaluated in this section. The statistical improvement was evaluated for the five most common macroscale variables: collision fraction (c), flow speed (v), flow speed squared (vv), number density (n), and temperature (T).

The simplest class of filters can be derived by fitting a series of data points with a straight line. In this case, the filter's window size corresponds to the length of the line segment used in the fit. A mathematical derivation may be found in Hamming [12], but in this simple case it can be seen that we are only interested in the intercept of the line. Because we center the line on the point of interest, this intercept is simply the average of this value and the surrounding values included in the fit. The filter coefficients are therefore all equal to one over the number of points. For example, a five-point simple linear filter has coefficients:

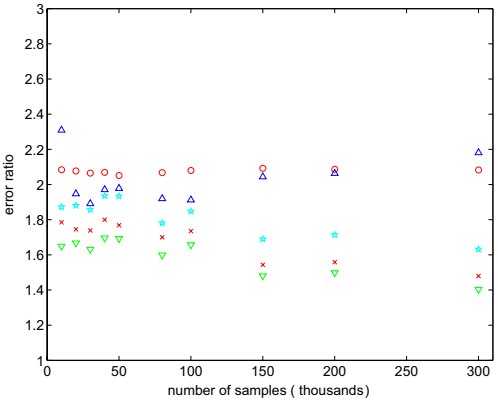
$$f = \frac{1}{5}[1\ 1\ 1\ 1\ 1] \tag{24}$$

In effect, these filters provide a "moving window average" between the point of interest and its nearest neighbors. The error ratio results for this filter with varying window sizes are shown in Fig. 6.

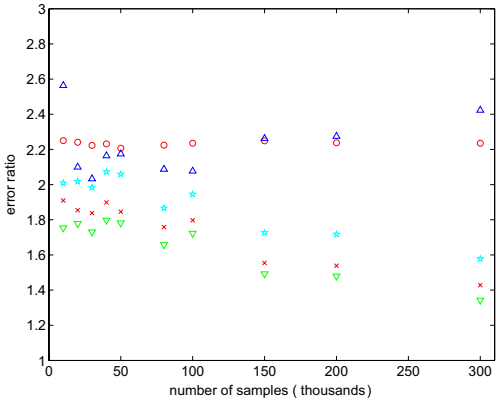
From this figure, it can be concluded that increasing the window size improves its performance - up to a point. As the window becomes very large, it begins to smooth physical features, which introduces error. This effect is most easily seen for the variables vv, n, and T. Increasing the window from five to seven points improves the error ratio for all variables at all sample counts. Increasing the window size from



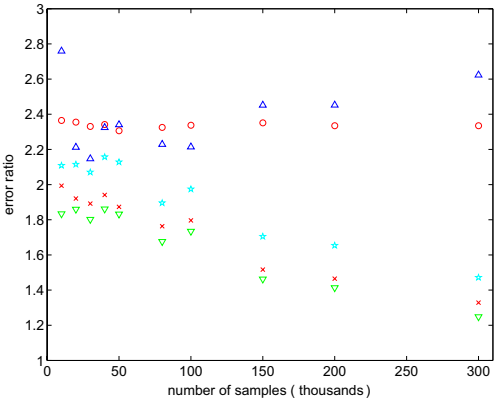
(a) 5 point



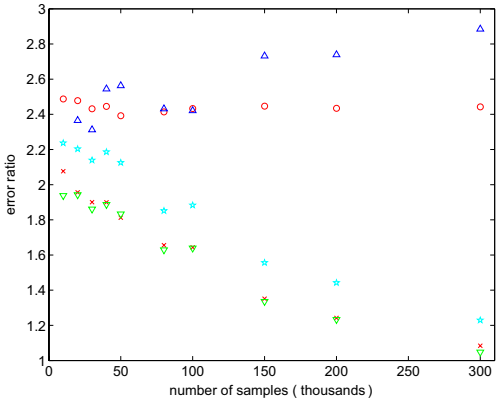
(b) 7 point



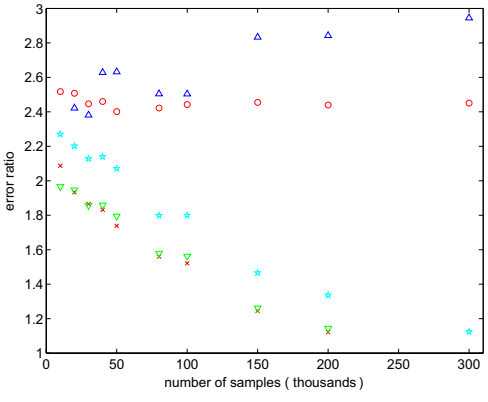
(c) 9 point



(d) 11 point



(e) 15 point



(f) 17 point

Figure 6. Error ratio for simple linear filters of various window size for five hydrodynamic variables.

seven to nine points, however, improves the error ratio at low sample counts for the three variables (because it is better able to eliminate noise), but degrades the error ratio at high sample counts (because it drives the solution away from the physical case). This behavior is not observed for the flow speed v because this quantity is constant (zero) in the Fourier problem. A bare average over the entire domain would therefore be the optimal solution for this variable.

It may be surmised that using a more complex trial function will enable the filter to better represent physical features in the data, which will mitigate the problem seen at large window sizes and high sample counts. Using the same technique for developing the coefficients described above, but switching to quadratic trial functions, we arrive at the coefficients:

$$f = \frac{1}{35}[-3 \quad 12 \quad 17 \quad 12 \quad -3] \tag{25}$$

for a five-point filter and:

$$f = \frac{1}{231}[-21 \quad 14 \quad 39 \quad 54 \quad 59 \quad 54 \quad 39 \quad 14 \quad -21] \tag{26}$$

for a nine-point filter [12].

The error ratios for these filters are shown in Fig. 7. Comparing the results of the quadratic filters to their linear counterparts in Fig. 6, two primary observations can be made:

1. As the sample count increases, the error ratio decreases much more slowly for the quadratic filters.

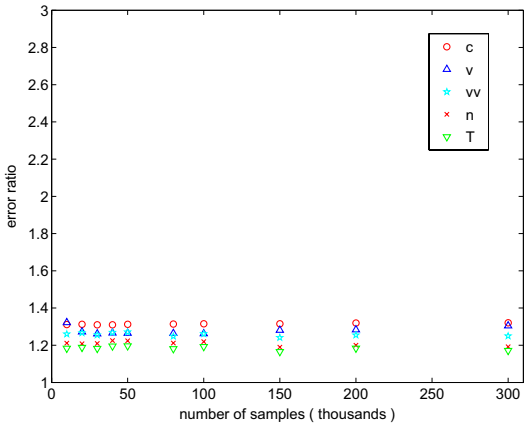
2. For a given window size, the low sample count error ratio is lower for the quadratic filter than for the linear filter.

The first observation closely follows our expectation that the quadratic trial function will be better able to represent the physical flow features. The second observation is related to the first: the more flexible trial function is also able to represent higher frequency components in the noise, thus they are not eliminated and the error ratio is reduced. The quadratic filters are thus better suited to application at higher sample counts where it is more difficult to distinguish between the noise and the flow features.

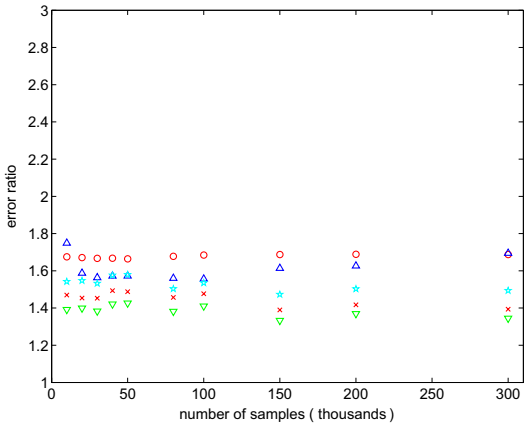
Hamming [12] suggests modifications to the filters derived using the least-squares technique. For the linear filters, the weight assigned to the outermost values is halved. This results in a modest improvement of the high sample count performance at the expense of low sample count performance (see Fig. 8).

For the quadratic filters, Hamming casts the outermost constant as a parameter and adds a constraint such that the transfer function is zero at the highest frequency. The resulting filter coefficients are given by

$$f = \frac{1}{96}[7 \quad 24 \quad 34 \quad 24 \quad 7] \tag{27}$$

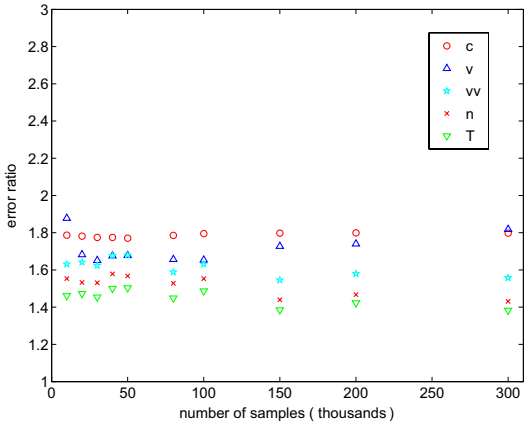


(a) 5 point

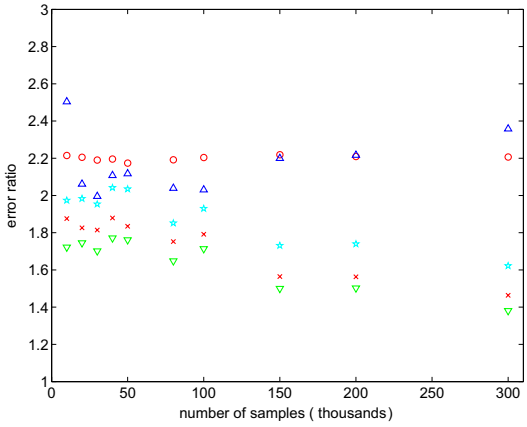


(b) 9 point

Figure 7. Error ratio for quadratic filters with five and nine point windows

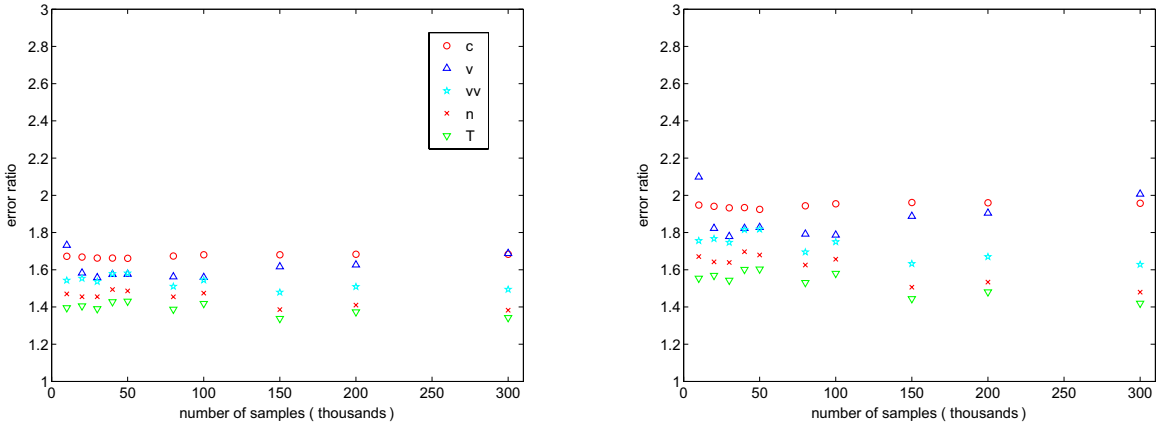


(a) 5 point



(b) 9 point

Figure 8. Error ratio for modified linear filters with five and nine point windows



(a) 5 point

(b) 9 point

Figure 9. Error ratio for modified quadratic filters with five and nine point windows

for a five-point filter and

$$f = \frac{1}{548}[-1 \quad 28 \quad 78 \quad 108 \quad 118 \quad 108 \quad 78 \quad 28 \quad -1] \tag{28}$$

for a nine-point filter [12].

The error ratios for these filters are shown in Fig. 9. Comparing this figure to its predecessors, it may be seen that the modification produces a significant increase in the error ratio from the unmodified quadratic result. At low sample counts, the error ratio is still lower than that produced by a linear filter with the same window size. At higher sample counts, the modified quadratic filter approaches or surpasses the performance of the comparable linear filter. Due to the decreased sensitivity of error ratio to sample count, it is likely that the modified quadratic filters will significantly outperform the linear filters at higher sample counts than those studied herein.

5.1.2 Fourier Filtering

Theory

In Fourier filtering, a discrete Fourier transform is computed using the DSMC data points. Because each successive term in a Fourier series represents an increasingly high frequency, the low-pass filter we seek can be constructed by keeping only the lowest terms. A reverse Fourier transform is then computed using the truncated series to yield the filtered result. This technique can be efficiently applied and easily extended to multiple dimensions by using commonly available multidimensional fast

Fourier transform (FFT) techniques [13, 14]. The truncation can also be made more gracefully by the use of windowing [12], which should improve the reverse transform. Because the effect is likely slight, windowing was not investigated in this work.

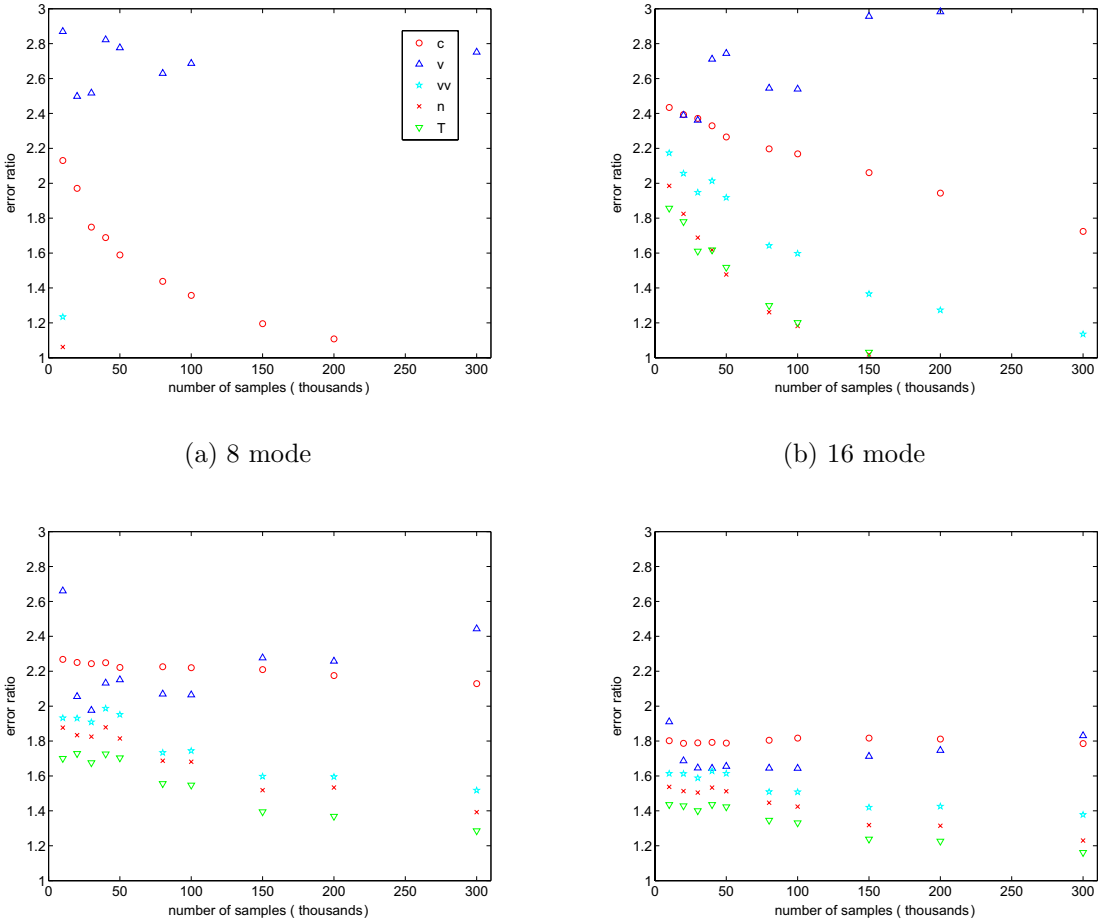
While each data point had a limited domain of influence under the nonrecursive digital filters, the Fourier filter essentially creates a global fit. Choosing the truncation point therefore involves a balance between accurate representation of the data and rejection of noise. For example, retaining only the first mode produces a constant (the average across the domain) and retaining all modes reproduces the unfiltered result.

Because Fourier series assume periodic functions, reflections are performed across the data endpoints, as was done for the nonrecursive filters, to avoid contaminating the domain edges with the Gibbs phenomenon [12]. At this point, the data more often resemble a half wave, yet full wave transforms were utilized, so a symmetry reflection is then performed to provide a periodic dataset. It was observed that the results generally improved as more points were added to the reflected condition, so results in this section were computed with a full reflection of the available data. Despite the greatly expanded number of points, compute times using MATLAB's FFT routine were still less than a minute for the test suite (100 1D slices for five variables at 10 sampling levels).

Results

The results of Fourier filtering, retaining 8, 16, 32, and 64 modes, are shown in Fig. 10. It is evident from this figure that more than 16 modes are required to describe the true solution for all the studied variables. With fewer modes, the error ratio drops below unity, indicating that the filter increased the overall RMS error. With more modes, the behavior is opposite that observed with the digital filters: increasing the number of modes decreases the low sample count error ratio but decreases the rate at which the error ratio decreases with increasing sample count. This observation is reasonable because adding more modes to a Fourier series allows it to describe increasingly complex functional behavior. It is therefore more able to represent (and thus retain) noise, as well as better represent the true value. This translates to a decrease in the error ratio at low sample counts, but an increase at high sample counts. Conversely, with the digital filters, adding points to the window increased the region around a point in question that is expected to conform to the trial function. It is thus better able to remove noise, but tends to move the filtered solution away from the true value, which is visible as a decrease in the error ratio at high sample counts. Interestingly, the Fourier filter with 32 modes produces results quite similar to those produced by the unmodified linear filter with 9 points.

The Fourier filter output for collision fraction is compared to the 'true' (high sample) solution and the nonrecursive filter output in Fig. 11 for 10,000 samples. It may be observed that the Fourier filter output is generally smoother than the nonrecursive filter output. Because it is a global fit, the Fourier result is also somewhat



(a) 8 mode

(b) 16 mode

(c) 32 mode

(d) 64 mode

Figure 10. Error ratio for Fourier filters with various retained mode counts.

less sensitive to localized peaks in the data that, in this case, are nonphysical.

5.2 Iterative Methods

Methods in this section make more than one pass through the data. In general, they are more sophisticated than the techniques investigated in the previous section. Consequently, they are generally more difficult to implement and more computationally demanding. In exchange, they promise, in theory at least, to be more powerful.

Because they make multiple passes, some sort of convergence criterion must be defined for iterative methods. In general, sufficient iterations are performed enable

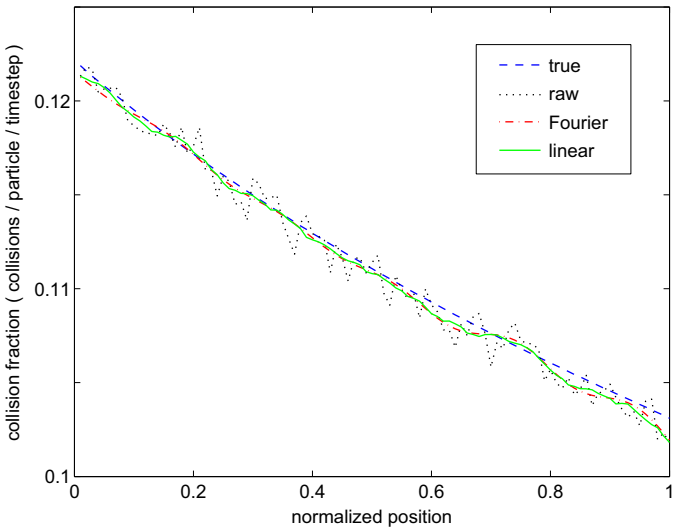


Figure 11. Comparison of 32 mode Fourier filtered representation of collision fraction after 10,000 samples to the true solution and the output from a 9 point linear filter.

information to spread through the entire domain. The mechanism of this information movement, as well as the convergence criterion, must therefore be carefully chosen because, otherwise, the solution can be driven far from the physical case.

5.2.1 Iterative Reduction of Noise

Theory

Fippel and Nüsslin developed the "iterative reduction of noise" (IRON) technique [15] to smooth Monte Carlo calculated dose distributions for radiation therapy. Smoothing is particularly important in this application because these results are often used as inputs to optimization algorithms for treatment planning. These algorithms can be driven to instability by noise in the interim solutions.

In the IRON technique, the data points are shifted by an optimization algorithm. Unlike the filters in the previous section, which were described in terms of frequency content, the optimization routine in the IRON technique is based on the idea of minimizing local second derivatives as a means for identifying and removing noise. For gas systems, this is a reasonable method because, like the dose distributions that inspired IRON, local second derivatives tend to be small in the physical case.

To produce a measure of the curvature, Fippel and Nüsslin use

$$\kappa(x) = \frac{|u''|}{[1 + u'^2]^{3/2}} \tag{29}$$

where κ is the normalized curvature and u is the quantity to be smoothed. For the equispaced, discrete data assumed in this work, this equation becomes

$$\kappa_i = \frac{|u_{i+1} - 2u_i + u_{i+1}|}{[1 + \frac{1}{4}(u_{i+1} - u_{i-1})^2]^{3/2}} \tag{30}$$

using second-order differences for both the first and second derivatives and setting Δx to unity. It may be noted that this definition of curvature only makes sense for unitless quantities. The quantity to be smoothed, u is therefore normalized by its maximum before applying Eq. 30.

The authors propose to use the sum of the curvature measures at each point as a positive contribution to the cost function, C. The optimizer will then seek a set of modified values for u that minimizes C. To simplify the optimization, the authors note that the denominator in Eq. 30 changes slowly and primarily serves as a normalization for the curvature. They therefore propose holding the values in the denominator at their unmodified state during the optimization procedure.

In order to prevent the optimizer from producing a strictly linear distribution, Fippel and Nüsslin add a restoration term to the cost function that is proportional to the displacement of a point from its original position. To allow greater displacements in regions with greater noise, this term is normalized by the local variance. Weighting factors, w, are then added in front of this term and the curvature term to set their relative strength.

The cost function therefore becomes

$$C = w_s \sum_{i=1}^{n} \frac{|u_{i+1} - 2u_i + u_{i+1}|}{[1 + \frac{1}{4}(\tilde{u}_{i+1} - \tilde{u}_{i-1})^2]^{3/2}} + w_r \sum_{i=1}^{n} \left[\frac{(u_i - \tilde{u_i})}{\sigma_i}\right]^2 \tag{31}$$

where tildes indicate unmodified quantities and

$$w_s + w_r = 1 \tag{32}$$

The authors note that w_r should be on the order of 1% and that the variance can be set to a constant value if a local estimate is unavailable (as it would be in a typical case where only a single dataset is available). They suggest setting $\sigma = 0.02$.

In this case, the reflection boundary conditions used in previous methods are unnecessary because we simply calculate derivatives near the boundary using one-sided differencing. If we use the same number of points in the stencils, we drop to first order accuracy, but this is likely sufficient as it only affects the boundary values.

Results

For the purposes of this work, the IRON method was implemented in MATLAB using the function `lsqnonlin` with the default optimization parameters. Total compute

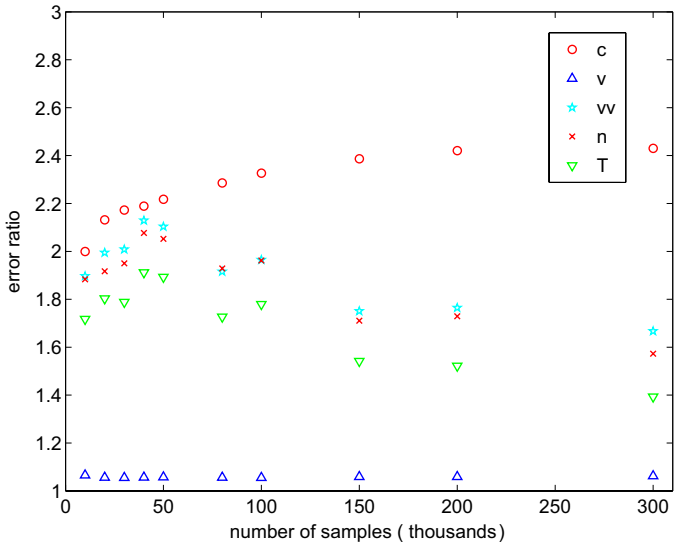


Figure 12. Error ratio for the IRON method with $w_r = 0.01$ and $\sigma = 0.02$.

time for the test suite was approximately 30 minutes on a 3.6GHz Xeon workstation. This is a significant wait compared to the noniterative filters, which all ran in less than one minute. It is, however, an insignificant quantity compared to the run times of most DSMC simulations, which are often measured in days.

The error ratio for the IRON technique on the test suite is shown in Fig. 12. For high sample counts, the collision fraction improvement is comparable to the very large window linear filters. The error ratio at low sample counts is significantly lower, however. This type of trend has not been observed for any technique investigated thus far.

The results for T, n, and vv are comparable to those of the nine point modified linear filter of Fig. 8. The results for v are notably worse than those of the previous techniques. Checking the filtered result in Fig. 13, it appears that this lackluster performance is related to the restoring terms because the technique appears to be doing the right thing, but weakly.

Testing this hypothesis, as well as Fippel and Nüsslin's argument that the restoring function is required to keep the optimizer from producing a straight line, the analysis was repeated with $w_r = 0$ and the results are shown in Fig. 14. Far from converging on the trivial solution, the results show a stunning improvement, particularly for the flow speed. The improvement was so great for this variable that the y-axis was extended from the value used thus far in this work to capture the data. Similar striking gains were made in the error ratio at all sample counts for the collision fraction. For the remaining three variables, significant gains were made at sample counts less than 100,000, with no significant loss at higher counts.

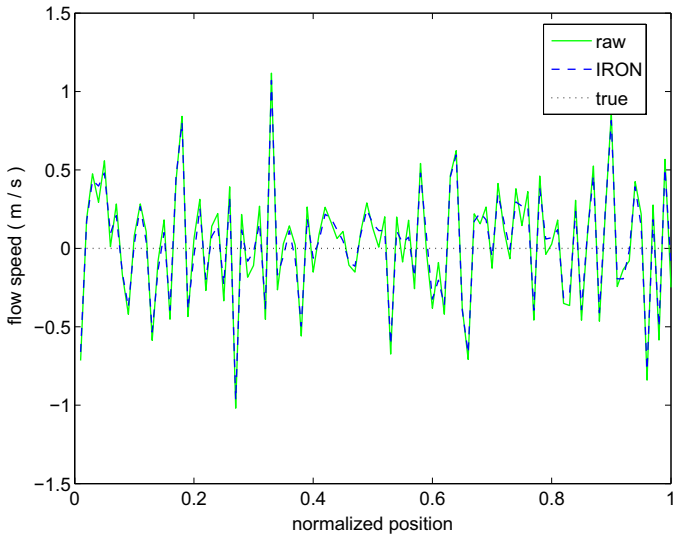


Figure 13. Flow speed with 10,000 samples processed with the IRON method with $w_r = 0.01$ and $\sigma = 0.02$ compared to raw result.

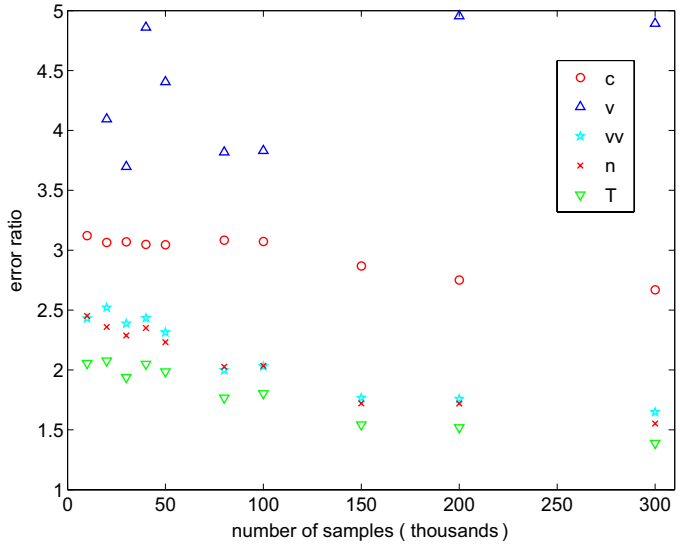


Figure 14. Error ratio for the IRON method with $w_r = 0$.

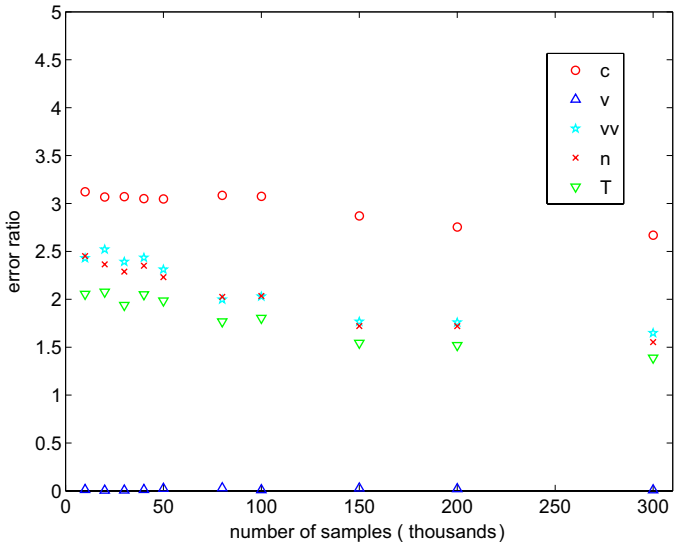
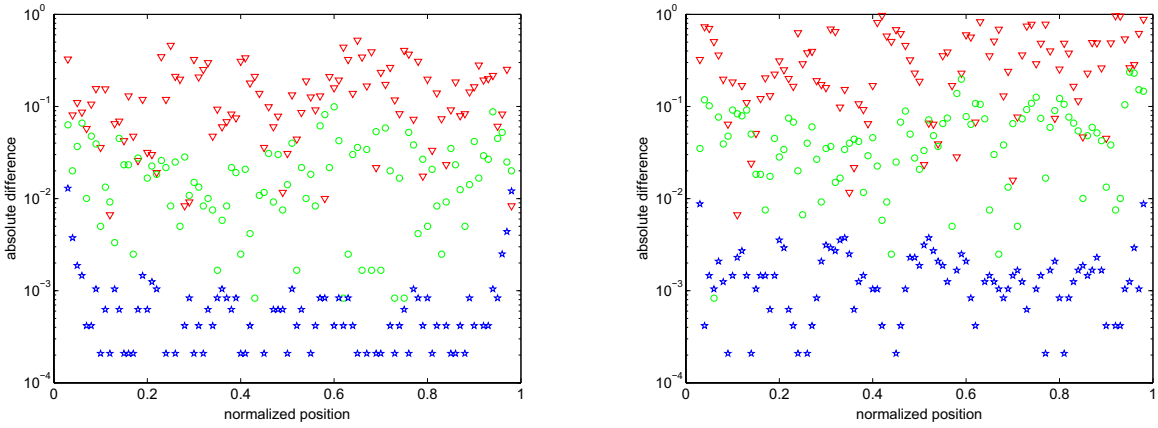


Figure 15. Error ratio for the IRON method with $w_r = 0$ and nonconstant values in the denominator of Eq. 30.

The effect of replacing $\tilde{u}$ in Eq. 31 with the variable under optimization was also investigated. Because the optimization utilized herein uses numerical differentiation, the only operation effect of this change is a very small increase in CPU time. As shown in Fig. 15, this change has a minimal influence on the results, except in the case of the flow speed. For this variable, the error ratio becomes extremely small. In fact, the lower limit on the y axis was reduced to zero to make it visible. Investigation showed that this behavior is due to the almost unrestrained movement of the boundary points, pushing them far from the true solution. Examination of Eq. 31 shows why this can happen: the boundary point can be moved somewhat independently of the others, thus it can be moved to a large value to drive the cost function to zero by inflating the denominator. This happened only occasionally, but because the test suite averages over 100 realizations, it only had to happen once in 100 times to reduce the error ratio to a very small value. It is unknown why this flaw only affected the flow speed, but because little is apparently gained from using modified values in the denominator, it was not investigated further.

Looking for a measure that is not biased against flowfields that show large curvature, such as the cross-channel direction in Poiseuille flow [16], it is surmised that noise will create differences between derivative estimates that use differing stencils. Investigating this hypothesis, the difference between second order first derivative estimate at point i:

$$\frac{\partial u}{\partial x}_i = \frac{u_{i+1} - u_{i-1}}{2} \tag{33}$$



(a) first derivative

(b) second derivative

Figure 16. Difference between second and fourth order derivative estimators as a function of position for the temperature field with various sample counts. Red triangles are 10,000 samples, green circles are 300,000 samples, and blue stars are over a billion samples.

and its fourth-order counterpart [17]:

$$\frac{\partial u}{\partial x}_i = \frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12} \tag{34}$$

assuming unit spacing is plotted in Fig. 16. The difference for second derivatives of second order:

$$\frac{\partial u}{\partial x}_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{2} \tag{35}$$

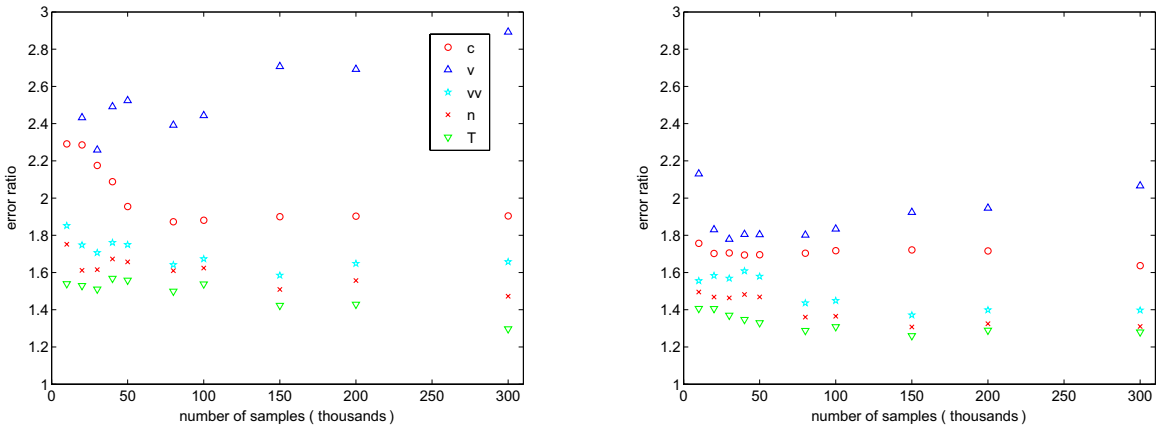
and fourth order:

$$\frac{\partial u}{\partial x}_i = \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12} \tag{36}$$

are also plotted. In both cases, a distinct reduction in the difference between these derivative estimates is visible as the sample count increases. As surmised, this difference is therefore increased by the presence of noise.

The results of the IRON technique using derivative differences as a cost function, and no restoring function, are presented in Fig. 17.[1] From this figure, it may be observed that the technique produces a generally lower error ratio with derivative

[1]It should be noted that the script used to generate Fig. 17 set the cost function to zero for the first and last two points because equivalent one-sided differences were not implemented. Correcting this deficiency would likely improve the result somewhat.



(a) first derivative

(b) second derivative

Figure 17. IRON technique with a cost function based on the absolute difference between second and fourth-order derivative estimates.

differences in the cost function than with curvature. The drop in error ratio with increasing sample count is somewhat slower, however, and the technique is more general, so this formulation may be attractive for certain classes of problems.

5.2.2 Flux-Corrected Transport Filtering

Theory

Kaplan and Oran [18] have presented the only technique examined in this work that was developed specifically for DSMC simulations of low-speed flows. These authors propose using the flux-corrected transport (FCT) algorithm as a filter to eliminate noise from the simulations.

Flux-corrected transport was developed as a means for solving the Navier-Stokes equations, which contain both convective and diffusive terms. In this application, only the diffusive steps are included. The authors note that the resulting technique has the following desirable properties:

- global conservation
- does not accentuate existing extrema

- does not introduce new extrema

It may be argued that the purely diffusive techniques, such as the digital filters examined in the previous section, also have these properties. Their flaw is that, if applied successively, or with excessively large windows, they will drive the solution to a constant value. This behavior was seen as a reduction in the error ratio, especially for large sample counts. Kaplan and Oran's method, however, is intended to be selectively diffuse; spreading the effect of noise, but retaining flow features.

The proposed method is a three-step process:

1. A value of the variable in question, u, based on plain diffusion, u^d, from diffusive fluxes f^d:
$$f^d_{i+\frac{1}{2}}(u) = \nu_{i+\frac{1}{2}}(u_{i+1} - u_i) \tag{37}$$
is computed according to
$$u^d_i = u_i + f^d_{i+\frac{1}{2}} - f^d_{i-\frac{1}{2}}. \tag{38}$$

2. The antidiffused fluxes are computed from this value:
$$f^{da}_{i+\frac{1}{2}}(u^d) = \mu_{i+\frac{1}{2}}(u^d_{i+1} - u^d_i) \tag{39}$$
and limited by applying
$$f^{dal}_{i+\frac{1}{2}} = S \cdot max\left\{0,\ \ min\left[S \cdot (u^d_{i+2} - u^d_{i+1}),\ \ \left|f^{da}_{1+\frac{1}{2}}\right|,\ \ S \cdot (u^d_i - u^d_{i-1})\right]\right\}, \tag{40}$$
where $S = sign(u^d_{i+1} - u^d_i)$ and $sign()$ a has magnitude of unity and a sign equal to the sign of its argument.

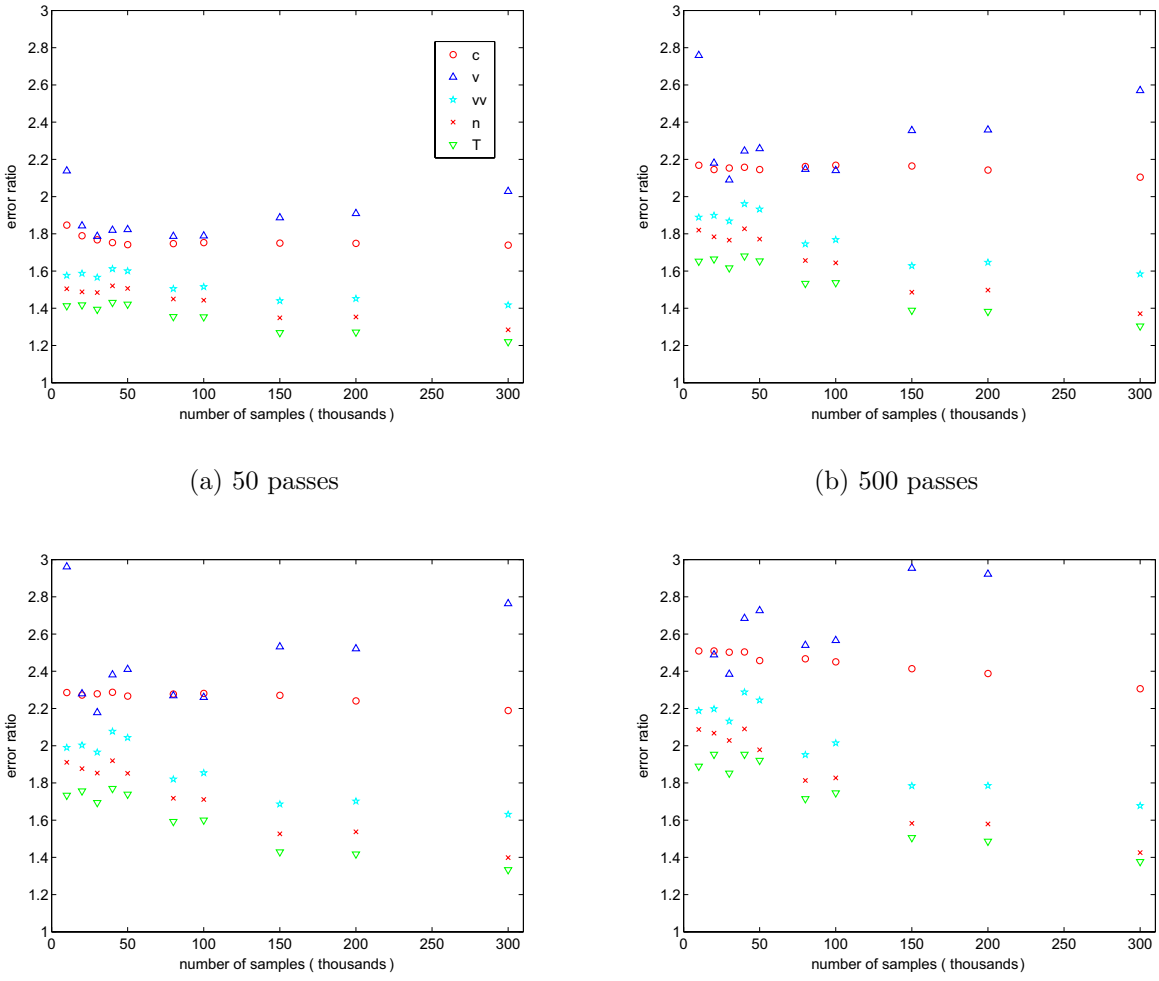
3. A final value is computed from
$$u^{new}_i = u^d_i - f^{dal}_{i+\frac{1}{2}} + f^{dal}_{i-\frac{1}{2}}. \tag{41}$$

The diffusion coefficient, ν, and antidiffusion coefficient, μ, are adjustable parameters. Kaplan and Oran suggest a value of 1/6 for both quantities, which was adopted in this work.

Boundary conditions for this filter were not mentioned in the article. In this work, the reflection condition employed for the noniterative methods was implemented. The length of the reflected region was eight points.

Results

The results for Kaplan and Oran's method with for a range of pass counts are shown in Fig. 18. Using the same workstation as for the IRON technique, this method proved considerably more expensive to reach a comparable error ratio. The run times for the 50, 500, 1000, and 5000 pass trials were 9, 81, 165, and 832 minutes, respectively. In the end, the behavior is similar to that seen in the linear filters: increasing the number of passes, like increasing the window size, tends to improve the error ratio at low sample counts, but it also increases the negative slope of the error ratio with sample count. Also, there is a point of diminishing returns where the error reduction from aggressively removing noise is canceled by the method's distortion of the true solution.



(a) 50 passes

(b) 500 passes

(c) 1000 passes

(d) 5000 passes

Figure 18. Error ratio for Kaplan and Oran's method with a varying number of passes.

5.2.3 Anisotropic Diffusion

Theory

Anisotropic diffusion was proposed by McCool [19] as a means for reducing noise in synthetic images (scene renderings) in which Monte Carlo algorithms were employed for determining lighting effects. Variants have also been proposed for reducing speckle in ultrasound images [20]. The thinking behind anisotropic diffusion is similar to that underlying Kaplan and Oran's flux-corrected transport filter: use diffusion to eliminate the noise, but use a means for controlling the diffusion to avoid affecting the physical solution. In Kaplan and Oran's method, the diffusive fluxes were controlled by antidiffusive fluxes of varying strength. In anisotropic diffusion, the diffusion is controlled by allowing the diffusion coefficient to vary in space. This formulation allows great flexibility in how the diffusion is controlled and is much easier to tailor than Kaplan and Oran's scheme.

In McCool's scheme, each pass advances a diffusion function given by

$$\frac{\partial u}{\partial t} = \nabla \cdot (C(u) \nabla u) \tag{42}$$

where $C(u)$ is a conductance function that varies between zero and unity in space and time as a function of some measure of coherence found in the distribution of u.

The behavior of this scheme is strongly dependent on the choices made in constructing $C(u)$. In particular, a measure of coherence and a means for converting this coherence measure to a conductance value must be determined. McCool provides suggestions for both these functions.

For the estimate of coherence, McCool suggests a function for color images that reduces to the magnitude of the gradient. The variables of interest in this work, with the exception of the velocity, are equivalent to monochrome data, so the magnitude of the gradient was used.

For the transfer function between the coherence measure (which varies from zero to infinity) and the conductance (which varies from unity to zero), McCool provides five possibilities, each with differing properties. In particular, some functions possess an inflected profile, which will tend to enhance sharp features with small pass counts (all edges eventually soften). For these profiles, the position of the inflection point must be carefully chosen to avoid enhancing noise (if set too low) or destroying features (if set too high). Because DSMC cannot produce sharp features on the sub-mean free path cellular scale, a non-inflected profile was chosen for this investigation:

$$C = e^{-g} \tag{43}$$

where g is the normalized coherence measure.

Results

Using the local gradient normalized by the global mean for the normalized gradient

g, the results of McCool's method for 50-500 passes are shown in Fig. 19. Normalizing by the global gradient is reasonable for the problem studied here, but could be problematic for flows with strongly varying gradients. In this case, a local average would be a better normalizing factor.

The results for collision fraction show the same anomalous behavior observed for IRON with $w_r = 0.01$; the error ratio is higher at high sample counts. This behavior is normalized as the pass count increases because the low sample count error ratio increases but little improvement is obtained at high sample count. The flow speed shows a similar behavior.

The error ratio behavior for the other variables is closer to expectation, showing an improvement with number of passes at low sample counts while exhibiting a plateau, and even a reduction, in error ratio at high sample counts as the number of passes increases. This, as in other techniques, is a consequence of the method driving the solution away from the true value. The effect is particularly striking for the number density.

For the sake of comparison, the results of applying isotropic diffusion are presented in Fig. 20. In this case, the diffusion constant was held at the base level used above. In some sense, this scheme could be viewed as having the same end goal as IRON when working to minimize second derivatives because this is exactly what diffusion does.

It is clear in Fig. 20 that, as seen for Kaplan and Oran's method, increasing the number of passes has a similar effect as increasing the window size on the digital filters: the error ratio is increased until an optimal value is reached, after which the error ratio begins to decline because the post-processing is moving the results away from the true solution. This optimal point is again smaller for larger sample counts.

5.3 Summary

The methods described in this section share a dependence on inter-cell relationships for reducing statistical noise. Otherwise, these methods appear quite different, ranging from simple windowed averaging to optimization schemes. Interestingly, despite their disparate approaches, the noise reductions achieved by an (admittedly roughly) optimized version of each technique are not strikingly different.

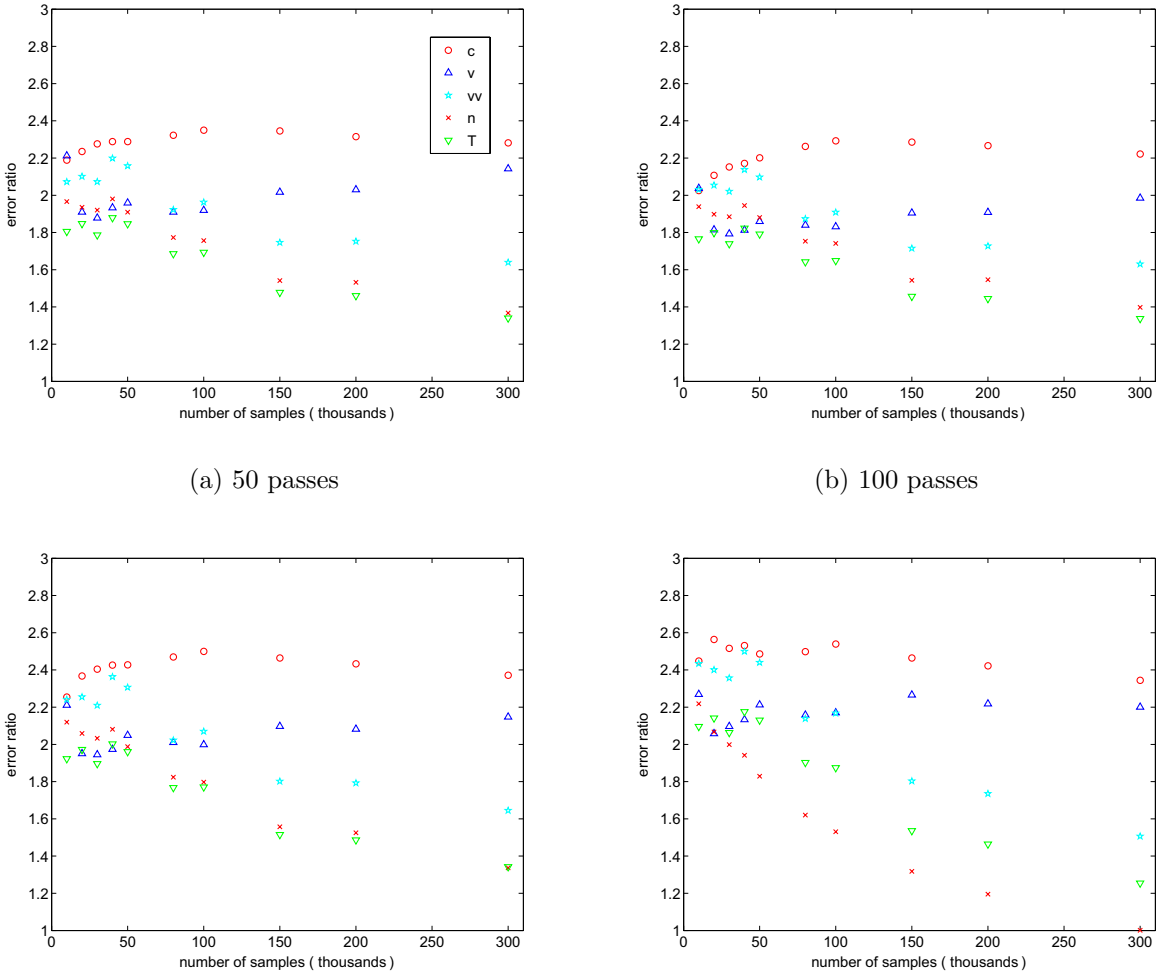
In order to compare these results in a compact fashion, the progression of RMS error with sample count is presented separately for each variable in Fig. 21. Each plot contains results for standard DSMC sampling, as well as a cross-section of the techniques discussed in this section. All data shown in Fig. 21 was previously shown in this section, but in the form of error ratio plots.

From Fig. 21, it may be observed that a significant savings is possible by applying

these methods. Comparing the sample count required to reach a given error level using a method to the sample count required to reach the same error level with the standard technique, the speedup of that method can be obtained. These values are listed in Table 2 using the error level reached by the standard method at 300,000 samples as a target. Values of three are typical for most variables using the most successful version of each technique. For every variable, the IRON technique with the curvature formulation in the original paper, but with the restoring function disabled, performs the best. Factors near ten are obtained by this technique for the flow speed and the collision fraction, factors near five are obtained for the flow speed squared and the number density, and a factor of over three is obtained for the temperature.

	T	c	v	v^2	n
Modified 9 point linear filter	2.8	4.9	3.8	3.7	3.1
Fourier filter with 32 modes	2.4	5.0	3.8	3.2	2.9
IRON technique with $w_r = 0$	3.3	9.5	13.3	4.9	4.8
Kaplan and Oran's method with 5,000 passes	3.1	6.2	6.9	4.3	3.5
Anisotropic diffusion with 200 passes	3.1	5.8	3.7	4.5	3.4

Table 2. Reduction factor for the number of samples required to reach an error level equivalent to that of the standard technique at 300,000 samples.



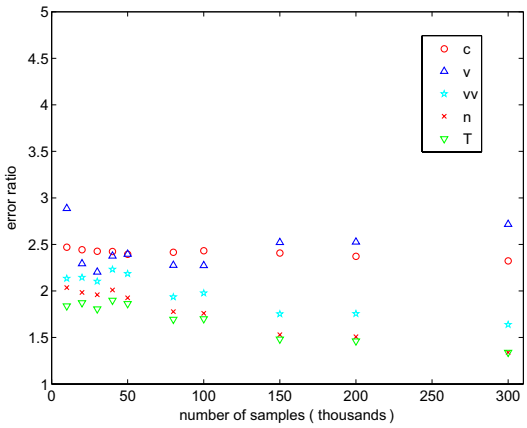
(a) 50 passes

(b) 100 passes

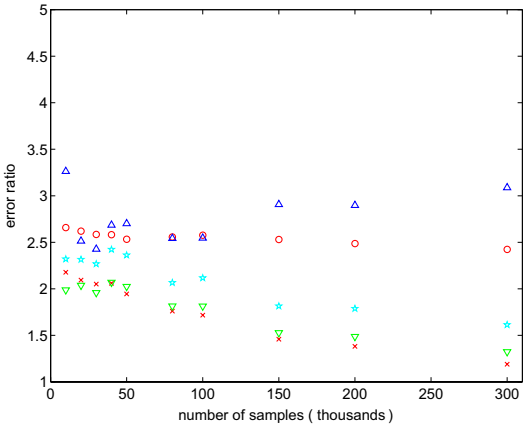
(c) 200 passes

(d) 500 passes

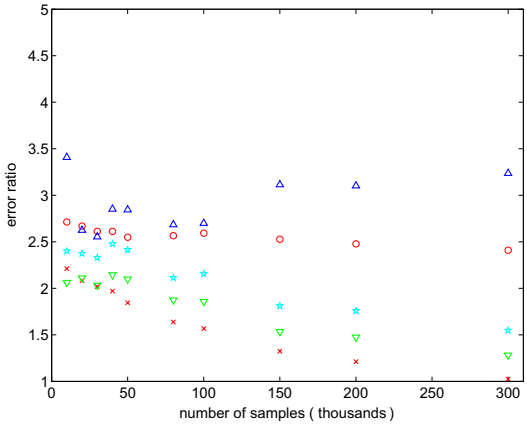
Figure 19. Error ratio for anisotropic diffusion with gradient-based diffusion coefficient and a varying number of passes.



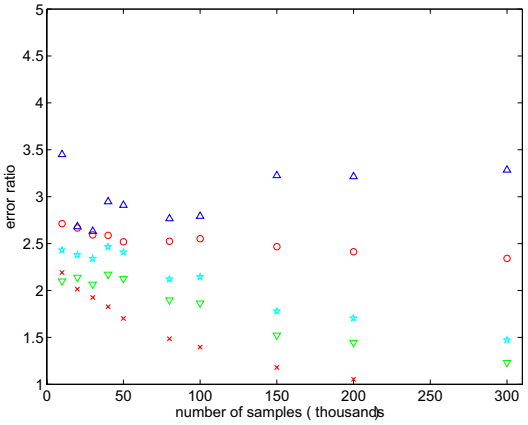
(a) 50 passes



(b) 100 passes

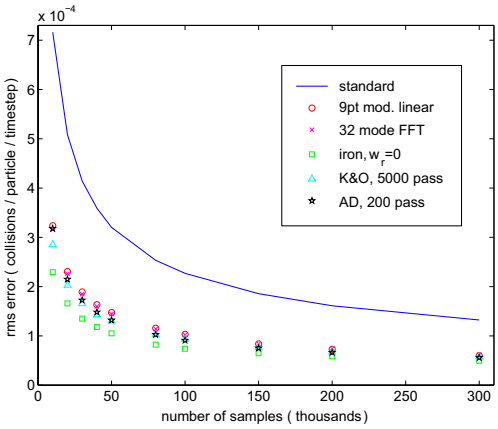


(c) 150 passes

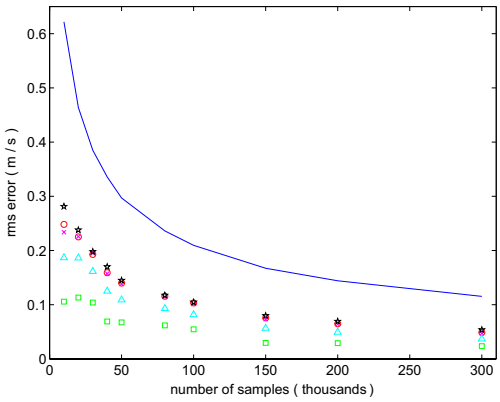


(d) 200 passes

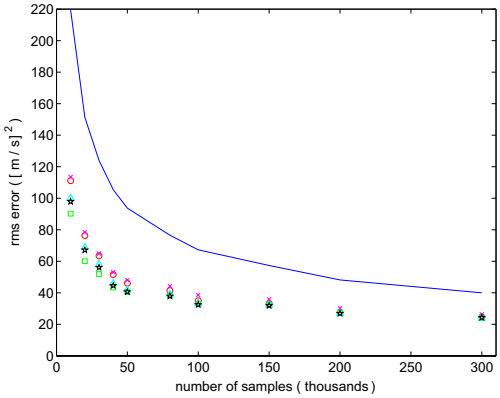
Figure 20. Error ratio for isotropic diffusion with a varying number of passes.



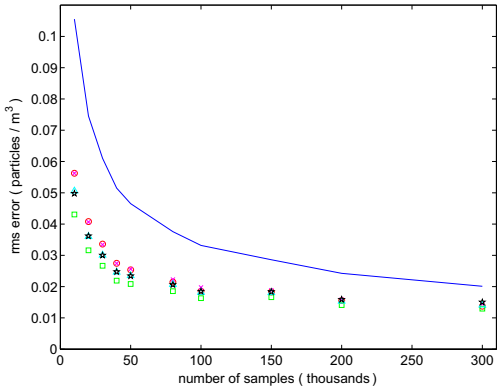
(a) collision fraction



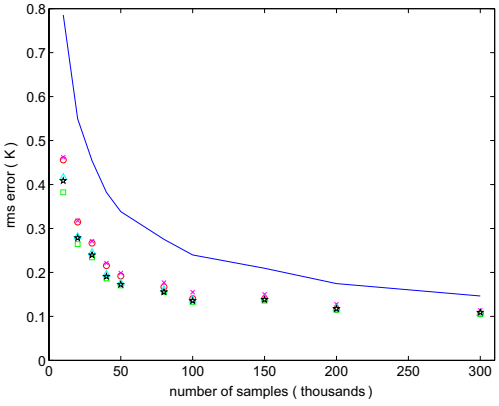
(b) flow speed



(c) flow speed squared



(d) number density



(e) temperature

Figure 21. Root mean squared error as a function of sample count for standard DSMC sampling compared to output from techniques studied in this section .

6 Conclusions

Several techniques were investigated in this work with the goal of reducing the statistical noise in estimates of the hydrodynamic variables computed from DSMC results. Techniques that rely on both intra-cell and inter-cell relationships were evaluated using the Fourier problem to provide data typical of MEMS flows, which are very demanding from a sampling standpoint due to their extremely low Mach numbers.

In general, the intra-cell methods were attractive due to their generality and solid mathematical footing. Unfortunately, they are also weak for the hydrodynamic problems of interest. This weakness stems from a low correlation coefficient between the quantities measured in a cell, such as number density and temperature. This finding is somewhat surprising in view of the ideal gas law (Eq. 6), but is a consequence of instantaneous local nonequilibrium in DSMC calculations with typical discretization parameters. Even worse, the flow speed, which is often a quantity of interest, was found to be uncorrelated with the other hydrodynamic variables. The control variate technique is therefore useless for reducing error in this quantity.

The spatial methods performed much better, yielding as much as an order of magnitude reduction in the number of samples required to reach a given level of error when compared to standard DSMC sampling. Viewed in terms of the control variate method, this result can be explained in terms of the extremely strong correlation between the hydrodynamic and the spatial variables. This correlation could be broken only in the presence of a discontinuity, such as a shock in a continuum method. In properly posed DSMC simulation, however, such discontinuities are impossible. Spatial correlations are therefore guaranteed to exist. Furthermore, the statistical noise present in the hydrodynamic variables is absent in the spatial variables, so a steady measure is available.

Perhaps the most interesting finding of this work is that, despite the wide range of increasingly sophisticated (and computationally expensive) techniques investigated in the spatial domain, the moving window averages commonly implemented for smoothing in plotting packages showed comparatively good results. Also, because the cells in DSMC simulations are typically less than a mean free path in extent, and significant changes cannot happen over such a length scale, a theoretical justification exists for averaging over nearest neighbors. This technique is already in use, in fact, in DSMC codes that utilize sub-cells for collisions; the super-cell serves as an averaging filter over its sub-cells.

While more possibilities certainly exist in this area, such as the techniques compared in [21], the reasonably small difference in error reduction between the wildly disparate techniques treated here implies that the results shown herein may be typical. Improvements in their generality in terms of treated variable, spatial dimension, number of samples, and other problem parameters are likely possible, but large increases in the error ratio may not be.

References

[1] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford Engineering Science. Oxford University Press, New York, NY, 1994.

[2] A. L. Garcia. "Nonequilibrium fluctuations studies by a rarefied gas simulation," *Phys. Rev. A*, 34(2):1454–1457, 1986.

[3] D. J. Rader, M. A. Gallis, and J. R. Torczynski. "Direct simulation Monte Carlo convergence behavior of the hard-sphere-gas thermal conductivity for Fourier heat flow," *Physics of Fluids*, 18:077102, 2006.

[4] A. L. Garcia. "Estimating hydrodynamic properties in the presence of microscopic fluctuations," *Communications in Applied Mathematics and Computational Science*, 1:53–78, 2006.

[5] W. A. Shewhart. *Economic control of quality of manufactured product*. D. Van Nostrand Company, New York, NY, 1931.

[6] M.A. Gallis, J.R. Torczynski, and D.J. Rader. "Molecular gas dynamics observations of Chapman-Enskog behavior and departures therefrom in nonequilibrium gases," *Physical Review E*, 69(4):paper 042201, 2004.

[7] M. A. Gallis, J. R. Torczynski, D. J. Rader, M. Tij, and A. Santos. "Normal solutions of the Boltzmann equation for highly nonequilibrium Fourier and Couette flow," *Physics of Fluids*, 18:017104, 2006.

[8] P. Glasserman. *Financial Engineering*. Springer-Verlag, New York, NY, 2004.

[9] P. R. Bevington and D. K. Robinson. *Data Reduction and Error Analysis*. Physical Series. McGraw Hill Higher Education, New York, NY, 2003.

[10] L.L. Baker and N.G. Hadjiconstantinou. "Variance reduction for Monte Carlo solutions of the Boltzmann equation," *Physics of Fluids*, 17(5):051703, 2005.

[11] N.G. Hadjiconstantinou, A.L. Garcia, M.Z. Bazant, and G. He. "Statistical error in particle simulations of hydrodynamic phenomena," *Journal of Computational Physics*, 187:274–297, 2003.

[12] R.W. Hamming. *Digital Filters*. Dover Books on Engineering. Dover Publications, New York, 1998.

[13] M. Frigo and S.G. Johnson. "The design and implementation of FFTW3," *Proceedings of the IEEE*, 93(2):216–231, 2005.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2nd edition, 1992.

[15] M. Fippel and F.B. Nüsslin. "Smoothed Monte Carlo calculated dose distributions by iterative reduction of noise," *Physics in Medicine and Biology*, 48:1289–1304, 2003.

[16] F.M. White. *Viscous Fluid Flow*. McGraw-Hill, New York, NY, 2 edition, 1991.

[17] D.A. Anderson, J.C. Tannehill, and R.H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Series in Computational Methods in Mechanics and Thermal Sciences. Taylor and Francis, Philadelphia, PA, 1984.

[18] C.R. Kaplan and E.S.Oran. "Nonlinear filtering for low-velocity gaseous microflows," *AIAA Journal*, 40(1):82–90, 2002.

[19] M.D. McCool. "Anisotropic diffusion for Monte Carlo noise reduction," *ACM Transactions on Graphics*, 18(2):171–194, 1999.

[20] Q. Sun, J.A. Hossack, J. Tang, and S.T. Acton. "Speckle reducing anisotropic diffusion for 3D ultrasound images," *Computerized Medical Imaging and Graphics*, 28:461–470, 2004.

[21] I. El Naqa, I Kawrakow, M Fippel, J.V. Siebers, P.E. Lindsay, M.V. Wickerhauser, M. Vicic, K Zakarian, N Kauffmann, and J.O. Deasy. "A comparison of Monte Carlo dose calculation denoising techniques," *Physics in Medicine and Biology*, 50:909–922, 2005.

Distribution

1	MS 0384	A. C. Ratzel, 1500
1	MS 0825	W. L. Hermina, 1510
1	MS 0826	D. J. Rader, 1513
2	MS 0826	E. S. Piekos, 1513
2	MS 0826	M. Gallis, 1513
1	MS 0826	J. R. Torczynski, 1513
1	MS 0836	R. J. Buss, 1514
1	MS 0836	M. M. Hopkins, 1514
1	MS 1243	S. N. Kempka, 5535
2	MS 9018	Central Technical Files, 8944
2	MS 0899	Technical Library, 4536
1	MS 0188	D. L. Chavez, LDRD Office, 1011