"Sophisticated DSMC"

G.A. Bird
gab@gab.com.au

Notes prepared for a short course at the DSMC07 meeting
Santa Fe, September 30, 2007

DSMC is computationally intensive but, because many calculations are made with programs that employ the traditional "simple DSMC" procedures and/or inadequate geometry models, the difficulties are habitually overstated. The degree of overstatement in recent years has typically been by a factor of fifty, but has sometimes been by a much larger factor. The proponents of alternative approaches have been quick to take advantage of this situation. It is essential that the causes of such serious misunderstandings should be investigated and understood. An attempt can then be made to establish the optimal set of DSMC procedures for the best possible codes

CONTENTS

1. Introduction

We already have a DSMC code – why do we need another one?
(Typical manager)

A 2006 paper by Lofthouse, Boyd and Wright [1] has been the trigger for a great deal of work on comparative DSMC calculations. This paper compared DSMC and Navier-Stokes CFD calculations for a Mach ten flow of argon past a twelve inch diameter circular cylinder. The stream temperature was 200 K and the surface temperature was a uniform 500 K. The case with a number density of 4.247×10^{20} per cubic meter and a Knudsen number of 0.009 has been chosen as the subject for benchmark calculations with a number of DSMC codes.

The temperature flowfield from a calculation with Version 4.3 of the DS2V program is shown in Figure 1.1. This and the following figures were obtained by a "cut and paste" from the GUI of the DS2V program and the shapes of the sampling cells are evident in the details of the flow. However, it should be noted that the collision cells were a quarter the size of the sampling cells and the molecules within the collision cells carry physical information on the flow at a still higher resolution.

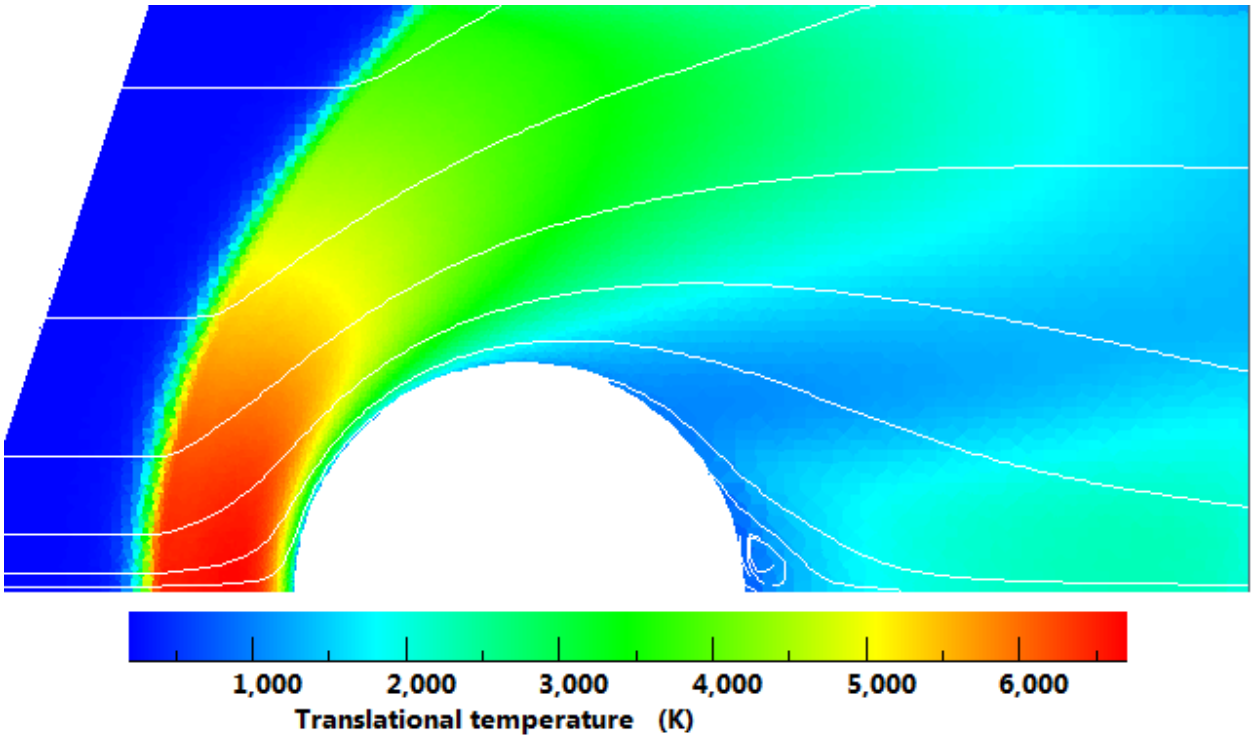


Figure 1.1 Continuous temperature contours in the benchmark flow.

The only notable feature of the flow is a small wake vortex, but it is instructive to look also at the contours of Mach number and number density in Figure 1.2.

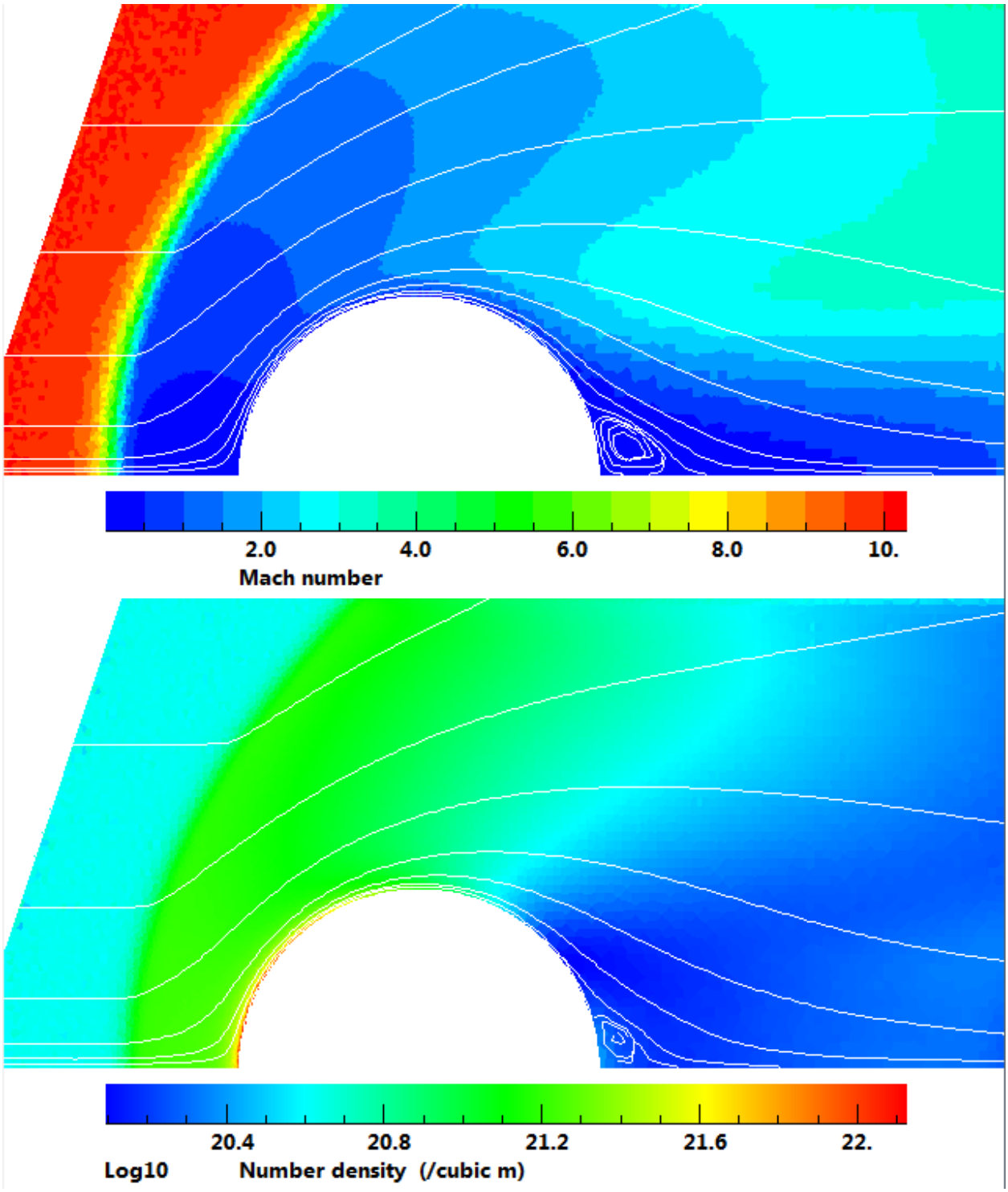


Figure 1.2 Banded Mach number and continuous number density contours.

The temperature and number density contours were generated by the short run that is the main benchmark case. The Mach number contour was generated from a run with a similar number of molecules, but its duration was 20 times longer and, after adaption, it employed an average of two, rather than six, molecules per collision cell. It will be shown that two molecules per collision cell is too few and that this was responsible for the apparent increase in the size of the vortex..

A comparison of the temperature and Mach number contours with the density contours clearly illustrates the differences between the effective widths of the temperature and density shock profiles. The edge of the temperature shock wave is marked by the onset of speckled contours. Even at this overall Knudsen number less than 0.01, the shock thickness measured from the initial temperature increase is about two thirds the shock stand-off distance. The width of strong shock waves in N-S CFD calculations is many times less than the real width. While this is of little consequence in a simple flowfield like this, it can lead to serious errors in continuum CFD studies of flows that involve complex shock wave interactions.

There is reference to this "benchmark case" throughout these notes. It is an "aerodynamic" calculation and, although the majority of practical DSMC applications now involve laboratory and manufacturing related vacuum systems, the aerodynamic applications are the more demanding. In addition, they can be compared with the corresponding continuum CFD calculations and the sometimes contentious relationship between the two approaches has been a continuing motivational factor behind the development of DSMC. The benchmark case is favorable for DSMC because it is a large disturbance flow, but the two order of magnitude change in density and the fact that much of this change occurs very close to the surface makes it a moderately challenging calculation. At the same time, the vortex occurs in a region where the average flow speed is around 10 m/s, so it also provides a test of the ability of DSMC to cope with low speed flows.

The net heat transfer at the stagnation point has been the principal subject of the benchmark studies and the value from reference [1] was 39,319 W/m^2. This was based on a MONACO calculation with 26,800,000 simulated molecules that required 1,827.2 CPU hours.

Figure 1.3 shows the results from a calculation by Version 4.3.09 of the DS2V program that employed 327,000 simulated molecules with a computation time of 3335 seconds (0.93 hours) on a notebook with an Intel Core 2 Duo T72000 CPU (2.0 GHz). This run employed an executable program that was compiled by Compaq Visual Fortran Version 6.6 and was run under 32 bit Vista. During the course of writing these notes, a change was made the new Intel Visual Fortran Compiler 10.0 for Windows and to a desktop computer with an Intel Core 2 Quad Q6600 CPU (2.4 GHz) running 64 bit Vista. The speed of the executable more than doubled and the same calculation was made in 1600 seconds (0.44 hours). Compiler options were chosen such that only one CPU core was employed for the DSMC calculation. The DS2V interactive GUI program was run concurrently and would have used part of the capacity of the other cores. The GUI is a REALbasic application that is coupled to the DSMC calculation through binary stream files. The speed meter in the Vista sidebar indicated an average CPU utilization of 27%.

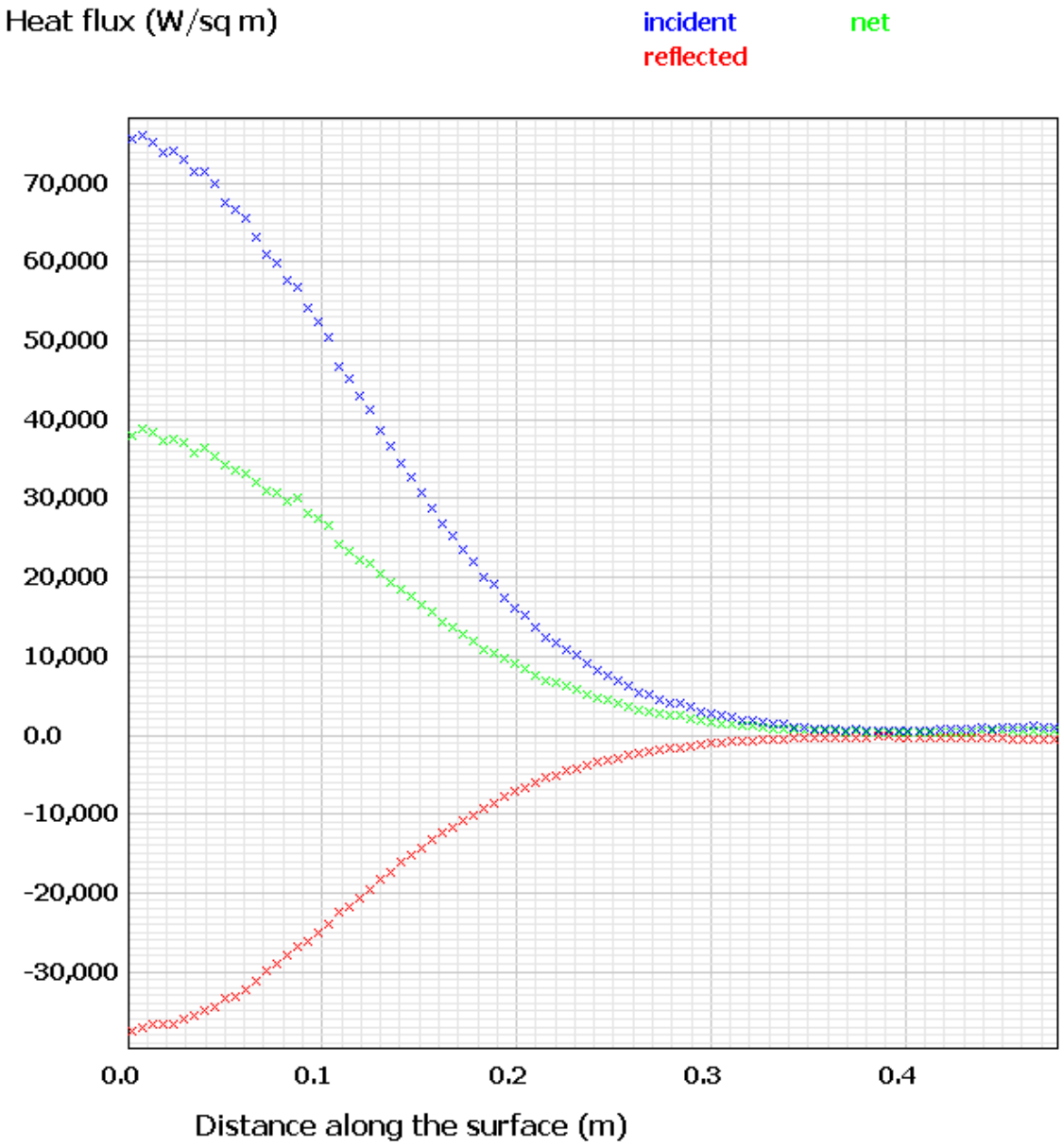


Figure 1.3 Heat transfer distribution in the DS2V benchmark calculation.

When steady flow had been attained, the collision cells in DS2V were adapted to 6 molecules per cell and the sampling cells to 24. These plots were also obtained by a "cut and paste" from the DS2V GUI and have not been subjected to smoothing. The net heat transfer at the stagnation point is 38,400±300 W/m^2. The benchmarking in the author's paper [2] at the 25th RGD meeting indicated that the converged value is 38,000±100 W/m^2. The possible effect of the representation of the curved surface by straight line segments was not taken into account in this process. This is investigated in the following section and the use of two degree straight line segments may have caused the stagnation point heat transfer to be about 1% too low. Even so, despite the short run with total duration of much less than one hour,

DS2V has produced a more converged result than MONACO. The DS2V overall drag per meter width from this calculation was 39.71 N, with 36.68 N being due to the pressure. The MONACO result was 40.0 N/m.

This calculation is at an overall Knudsen number less than 0.01 and the Navier-Stokes equations would be expected to provide a reasonably good solution. Reference [1] also included N-S CFD results from the NASA Ames DLPR code. The peak heating was quoted as 40,884 W/m^2 and the overall drag as 40.2 N/m. The peak surface slip quantities, as measured by the DS2V surface sampling, were 230 m/s for the velocity and 320 K for the temperature. The continuum calculation assumed zero slip and the errors in the Navier-Stokes model are surprisingly small.

The really interesting result is that the quoted computation time for the continuum CFD solution was 0.78 hours. This is almost double the 0.44 hours that was required for this DSMC calculation!

Furthermore, the DS2V version 4 program is readily available from gab.com.au as a free download and, because the grid and computational parameters are generated automatically by the program, even an inexperienced user should be able to set up the data for a simple calculation like this in just a few minutes. Any reader can therefore repeat the calculation for this benchmark DSMC case and verify all the claims that have been made with very little effort and at virtually no cost. After very many years of enquiries, the author is still waiting for someone to point to the existence, let alone the availability, of a Navier-Stokes CFD program that would enable him, even at some cost and with much greater effort, to similarly verify the continuum calculation!

A characteristic of DSMC codes appears to be that, in comparison with N-S CFD codes, there is less variability in the results but a greater variability in the computation time. It is worth reproducing a table from Reference [2] that compares results for this problem from five DSMC codes.

Program	Total Simulated Molecules	Total Cells	Time Step (μ sec)	Computation Time (hours)	Drag (N)	Peak Heat Transfer (W/sq m)
MONACO	26,800,000	34,770	0.02	1872 parallel CPU	40.00	39,319
DS2G (ver. 2)	2,900,000	129,600 (sub-)	0.2	20 on 3GHz PC	39.95	38,300
SMILE	24,000,000	4,000,000 (coll.)			39.76	39,000
DAC	1,300,000			15 on 3GHz PC	39.71	38,500
DS2V	330,000	41,000 (collision)	0.12 (min.)	10 on 3GHz PC	39.76	38,400

TABLE 1. .Results for the hypersonic cylinder test case from five DSMC codes.

The new calculation required less than one tenth the computation time of the original DS2V calculation in Table 1. That calculation was made with an earlier DS2V version on a slower computer and employed larger collision cells. However the main factor is that, because the statistical scatter is inversely proportional to the square root of the sample size, the smaller sample size in the new calculation has had little adverse effect on the results. This smaller sample is plotted in Figure 1.4.

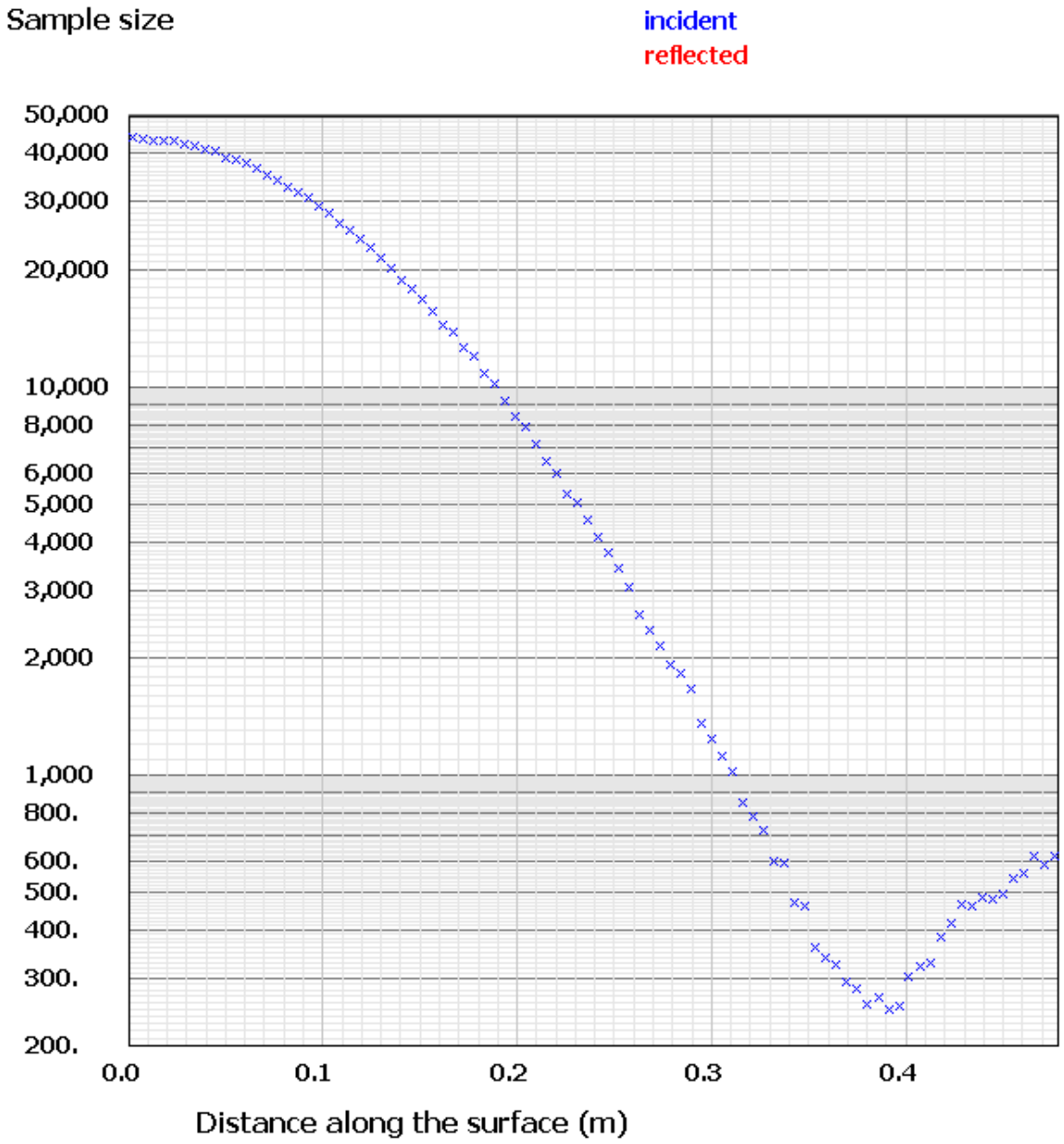


Figure 1.4. The sample of incident molecules at each surface element.

The sampling intervals on the cylinder were of equal area so that the number flux to the surface is proportional to the sample. The density varies by more than two orders of magnitude and the sample drops to the low hundreds at the rear of the cylinder. Even so, the number flux distribution is well defined in this region.

Because the samples of incident molecules at the front of the cylinder are over a hundred times larger than those at the rear, the scatter at the front would be expected to a tenth the magnitude of the scatter at the rear. Because the heat transfer at the rear of the cylinder is so small, it is not easy to estimate the level of scatter in Figure 1.3. This comparison is best made through the shear stress distribution that is shown in Figure 1.5.

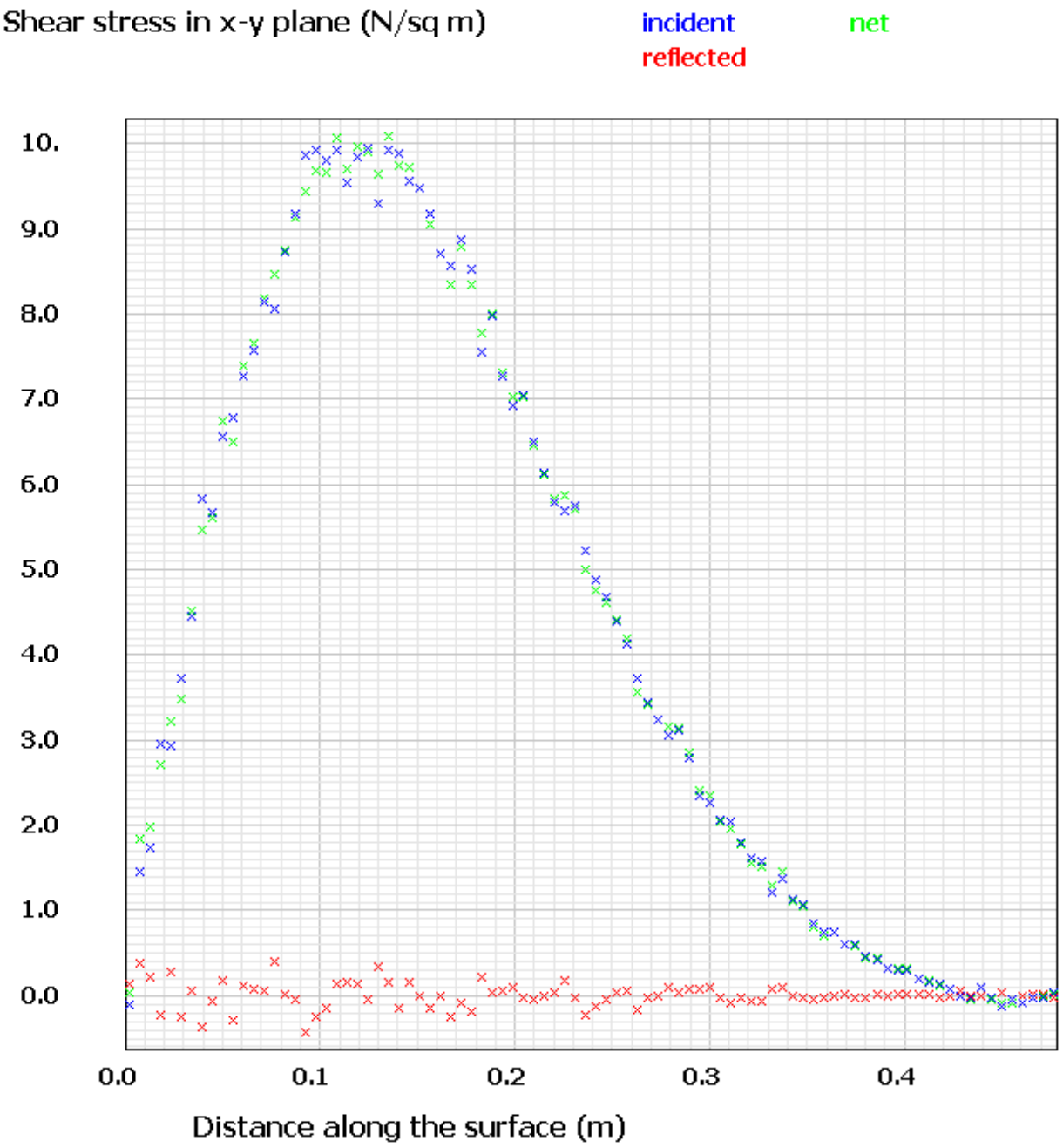


Figure 1.5. Shear stress distribution.

Contrary to the expectations based on sample size, the scatter is larger at the front of the cylinder than at the rear where even the region of negative shear stress under the vortex has been resolved. The apparent explanation is that the flow gradients are orders of magnitude larger at the front of the cylinder. These gradients have not been properly resolved by the grid that has been employed and this has led to an uneven distribution of the flow properties. While the current DS2V results compare well with those from other DSMC programs, there is clearly an opportunity for them to be a great deal better.

It is now clear that systematic error due to excessively large and/or irregular cells has habitually been regarded as statistical scatter.

This first became obvious to the author when the following figures that deal with the same flow as that in the benchmark calculation were prepared for Reference [2] .

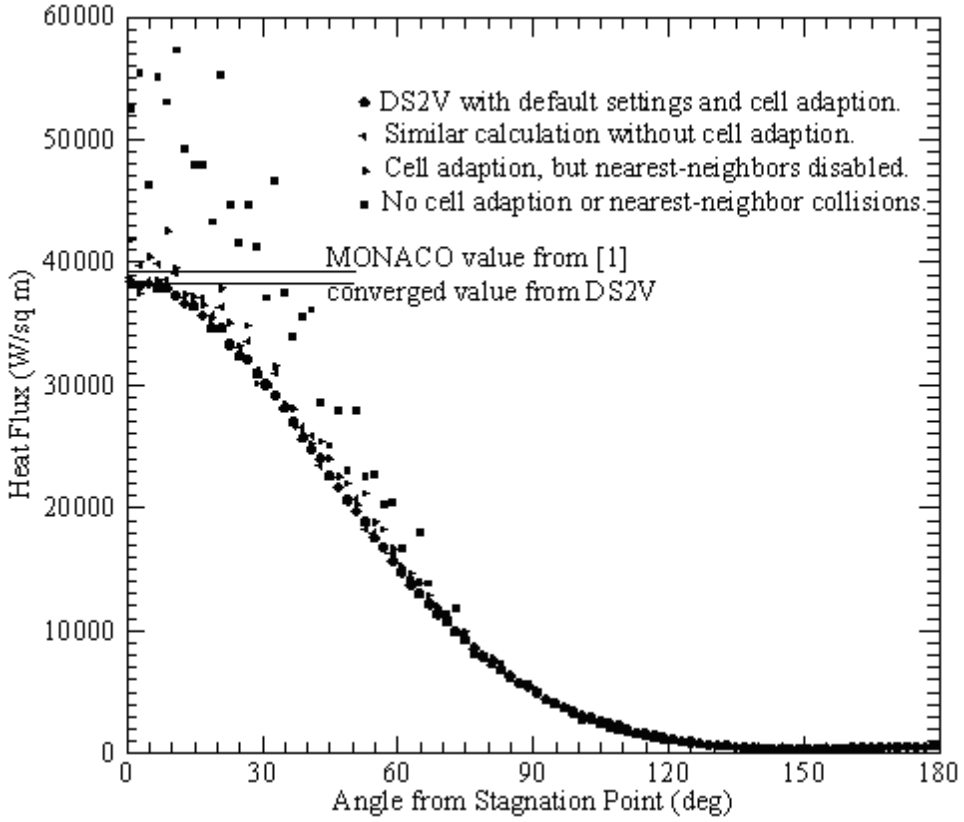


Figure 1.6 Results with equal sample sizes when sophisticated features are disabled.

The run with the DS2V default settings and cell adaption is the case listed in Table 1. It differs from the benchmark case in that the sample at the stagnation point is 600,000 rather than 45,000 and the collision cells were adapted to 8 rather than 6 molecules per collision cell. An identical run in which the sampling and collision cells were not adapted to a uniform number of simulated molecules per cell led to maximum heat transfer value similar to that from the long MONACO calculation. As well as the less converged result, the scatter in the distribution was greatly increased even though the sample size was unchanged. It is this latter point that was unexpected.

There is an option in the DS2V program to disable the procedures that lead to near nearest-neighbor collisions. With cell adaption, the deterioration in the results was about twice that due to not adapting the cells with nearest-neighbors enabled. The deterioration when neither nearest-neighbors nor cell adaption was employed was markedly greater than the sum of the individual deteriorations.

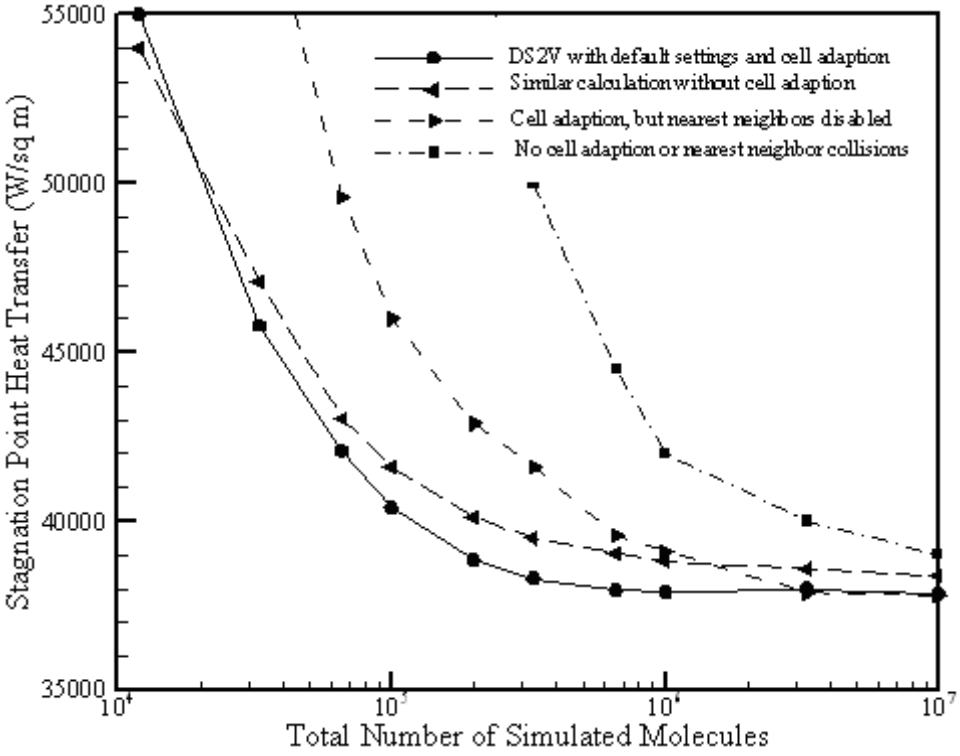


Figure 1.7 Convergence with molecule number.

The most compelling result was obtained when the four classes of run were repeated with different numbers of simulated molecules. The four convergence curves are compared in Figure 1.7. The physical size of the collision and sampling cells varies with the total number of molecules and this variation was determined by the internal logic of the DS2V program. The user controls the total number of simulated molecules through the specification of the number of megabytes of memory that are to be used by the program at the start of the calculation. The value of the time step is set by the internal logic to satisfy the requirements for a good DSMC calculation. There are options for advanced users to alter the computational variables, but all the calculations that contributed to the results in Figure 1.7 employed the default values. The adaption in these cases was to eight molecules per collision cell and 27 molecules per sampling cell.

There is an advantage in adapting the cells to a uniform number of molecules per cell until the total number of molecules becomes so small that adaption leads to fewer cells. This has occurred in these calculations when the total number fell to about twenty thousand. Selecting for collisions between the nearest neighbors has the greatest beneficial effect when the total number of molecules is small and, in these calculations, there was no significant advantage when the total number reached about three million. For a given level of accuracy, the difference in the number of molecules that are involved in the best and worst cases is relatively constant at about a factor of fifty until the heat transfer decreases to three or four percent above the converged value. We can define an "adequate engineering

calculation" as one that produces results within one or two percent of the converged value. The number ratio for such a calculation is difficult to judge from the convergence curves in Figure 1.7 because ten million molecules proved insufficient to achieve adequate engineering accuracy in all cases. However, it appears that the ratio is above one hundred and could reach one thousand.

The benchmark case achieves engineering accuracy with about 300,000 simulated molecules. However, we will find the current DS2V geometry model is not optimal and the required total number of molecules could fall to as low as 100,000. The required number for the geometry models that correspond to the unadapted near rectangular cells and with no nearest-neighbor procedures would be between 10,000,000 and 100,000,000 simulated molecules.

These figures explain why DSMC workers who are competent and who operate with goodwill towards the method, but who employ whatever DSMC codes happen to be at hand, have come up with such different estimates of the computational effort that is required for the production of useful engineering results.

The initial results for the benchmark case were obtained with a mid-range notebook computer (Acer Travelmate 4230) that is the modern equivalent of the proverbial "pencil and paper". The desktop computer was only slightly more expensive and Intel have since halved the CPU price. If the same program was run on a single core of the fastest CPU in the Intel catalog, the 26 minutes would be below 20 minutes. The potential gains from an improved geometry model and internal code speed-ups are less certain, but it is considered that an improvement by a factor of two is a conservative estimate and the benchmark computation time would fall below ten minutes. The production of multi-threaded code that took advantage of current quad-core CPU's would hopefully lead to a two or three minute run time for the benchmark case on a current top of the line PC. Version 10 of the Intel Fortran compiler appears to now provide the necessary tools.

For two-dimensional and axially symmetric flows, the current programs permit an extensive overlap of the DSMC results with those from N-S CFD calculations. Overlap was not achieved for the benchmark case because the N-S CFD result for the stagnation point heat transfer was 5% too high and this is outside the limits for a good engineering calculation. However, DSMC calculations can readily be made at Knudsen numbers that are sufficiently low for continuum calculations to achieve the necessary level of accuracy. The situation is more problematic for three-dimensional flows. A three-dimensional calculation in the Knudsen number range that is accessible to both DSMC and N-S CFD requires computing resources that are up to a thousand times greater than those for the corresponding two-dimensional calculation. This is possible with the current programs, but the computation time becomes large. DSMC runs of less than a day for difficult cases in the "continuum overlap" regime will become possible only when the 3-D codes reach the degree of refinement that corresponds to a three minute runtime for the benchmark case.

The gross over-estimation of DSMC computational requirements that is all too prevalent in the literature appears to have led to a misallocation of resources to alternative methods that are unlikely to prove useful for engineering studies. The recent disinterment of the notoriously unrealistic BGK model is just one example.

2. Geometry models for DSMC

They can be ugly, but ugliness is best minimized.

Ideally, a geometry model would involve only the physical boundaries and there would be no grid. This is easily implemented in one dimension (molecules ordered according to their position), but is difficult in two and three dimensions. It is possible to construct "cells" from the molecules that are closest to a set of points (or cell nodes). However, the computation time of the procedures tends to be proportional to the square of the number of molecules and the effective area of the cell is unacceptably uncertain. An explicit grid or mesh appears to be necessary.

The simplest mesh is a single set of uniform rectangular cells that cannot conform to curved boundaries. These are employed in the mathematical models of the DSMC method, but can lead to very poor results in a real program. Rectangular cells were employed in the demonstration codes in the author's 1994 monograph [3]. The example programs were restricted to geometries that had flat surfaces that lay along cell boundaries. It was not intended that the programs should be adapted to deal with curved surfaces and the serious programs of the day (e.g. the DS2G code that appears in Table 1) employed far more complex geometry models. However, a number of workers have simply inserted curves surfaces into these programs and the results from these calculations appear to be as bad as would be expected.

The ideal grid would conform to the shape of surfaces, have a cell size inversely proportional to the density and have cell aspect ratios such that the rate of cell change along any line drawn in the flow is proportional to the flow gradients in the direction of that line. The latter condition has only been approached for special problems such as the boundary layer on a flat plate or with manually constructed grids for hypersonic blunt-body flows (e.g. the DS2G code). Manually specified grids that conform to the flow gradients as well as the density make significant demands on the user for 2-D flows and the effort becomes unacceptably large for 3-D flows.

DSMC has to keep track of the simulated molecules as they move through the mesh. With rectangular grids, molecules may be "indexed" to the cells by simple analytical expressions but, with irregular cells, a "ray-tracing" procedure has to be employed. This involves calculating, at every molecule move, the potential collisions between the molecule and the boundaries of the current cell. There is a very significant computational time penalty associated with ray tracing.

An enormous amount of effort has gone into grid generation for continuum CFD. These grids may be divided into "finite difference" grids and "finite element" grids. The former necessarily conform to surfaces, are as near orthogonal as possible and are generally adapted to the flow gradients. Finite element grids are generally triangular in 2-D flows and tetrahedral in 3-D flows. Some DSMC programs employ continuum CFD grids and, as long as steps are taken to achieve nearest neighbor collisions, the results can be satisfactory. However, ray-tracing procedures appear

to be necessary. Current thinking calls for separate sampling and collision cells with less than eight molecules per collision cell. The computational effort associated with ray-tracing is then prohibitive.

Programs that employ rectangular grids often employ a subdivision of selected elements of the rectangular grid into smaller rectangles. There are many ways in which this can be done. Some employ a "tree-like" structure in which cells can be repeatedly divided into two in each physical dimension. A disadvantage is that the step in cell size is by a factor of four in two-dimensional grids and a by factor of eight in three-dimensional grids. This can lead to an undesirable level of non-uniformity in the number of simulated molecules per cell.

Alternatively (e.g. in the DS2V/3V codes), there are multiple levels of rectangular grid with the number of elements in the smallest grid being larger than the number of simulated molecules. The cells are then irregular aggregations of rectangular elements. They retain the computational efficiency of rectangular systems and permit very rapid adaption to the flow density. At present, these codes "grow" the cells from the elements that are closer to a given cell "node" than to any other node.

The "bounding rectangle" of a two-dimensional flow or the "bounding rectangular parallelepiped" of a three-dimensional flow constitutes the largest rectangle. This does not require the flowfield to be rectangular because parts of the space may be cut off by "open surfaces" that must terminate on the outer boundaries. There are also "closed surfaces" that are completely immersed in the flow. This single rectangle is divided into "divisions" that would approximate the initial cell system if the flow was a uniform stream or reference gas. In addition to providing one level of spatial reference, the segments of the surfaces (which are straight lines in two-dimensional systems and triangles in three-dimensional systems) are indexed to the divisions. The divisions are marked as "level" -1 if they are completely outside the flowfield or inside a surface, level 0 if they contain surface segments, or positive levels that are in layers around the level 0 divisions with the value of the level indicating the number of divisions between that division and the surface. As noted above, the surface segments are indexed to the level 0 divisions and, when moving the molecules, possible collisions with surfaces need not be considered unless a molecule is in a low level division. Finally, the divisions are subdivided into rectangular elements. The current defaults are 10×10 elements in DS2V and 5×5×5 elements in DS3V. The only information that must be carried for all elements is the number of the collision and sampling cells in which it lies, or 0 if it is not within the flow. This system permits a very rapid calculation of the collision and sampling cells that are associated with the position coordinates of each molecule.

The elements that are intersected by surface segments must have the intersections marked and their effective volumes recorded. The whole of these elements are shown in the figures, but the surface segments are also plotted and the portion of the element that is inside the surface should be disregarded. A solid surface must not be thinner than an element because there would then be a possibility of collisions between molecules that are on opposite sides of the surface.

These rectangular reference systems are the antithesis of the elegant grids that are employed in finite difference CFD calculations, but are computationally very efficient in handling the irregular motion of millions of simulated molecules.

To date, the DS2V/3V programs employ 32 bit arithmetic. The most difficult part of the whole programming exercise is then the marking of the elements as outside the body and within the flow, intersected by the surface, or inside the body. Because there are millions of elements and only about six significant digits of accuracy, round-off error is a major problem. In the case of the DS3V program, resort has been made to empirical pattern recognition algorithms to distinguish between valid and invalid intersections of multiple triangles with the sides of a rectangular parallelepiped. All the higher performance CPU's are now 64 bit and the single precision geometry routines in the DS2V/3V programs are being replaced with double precision versions. The complex code that copes with round-off will no longer be needed and the program speed should increase.

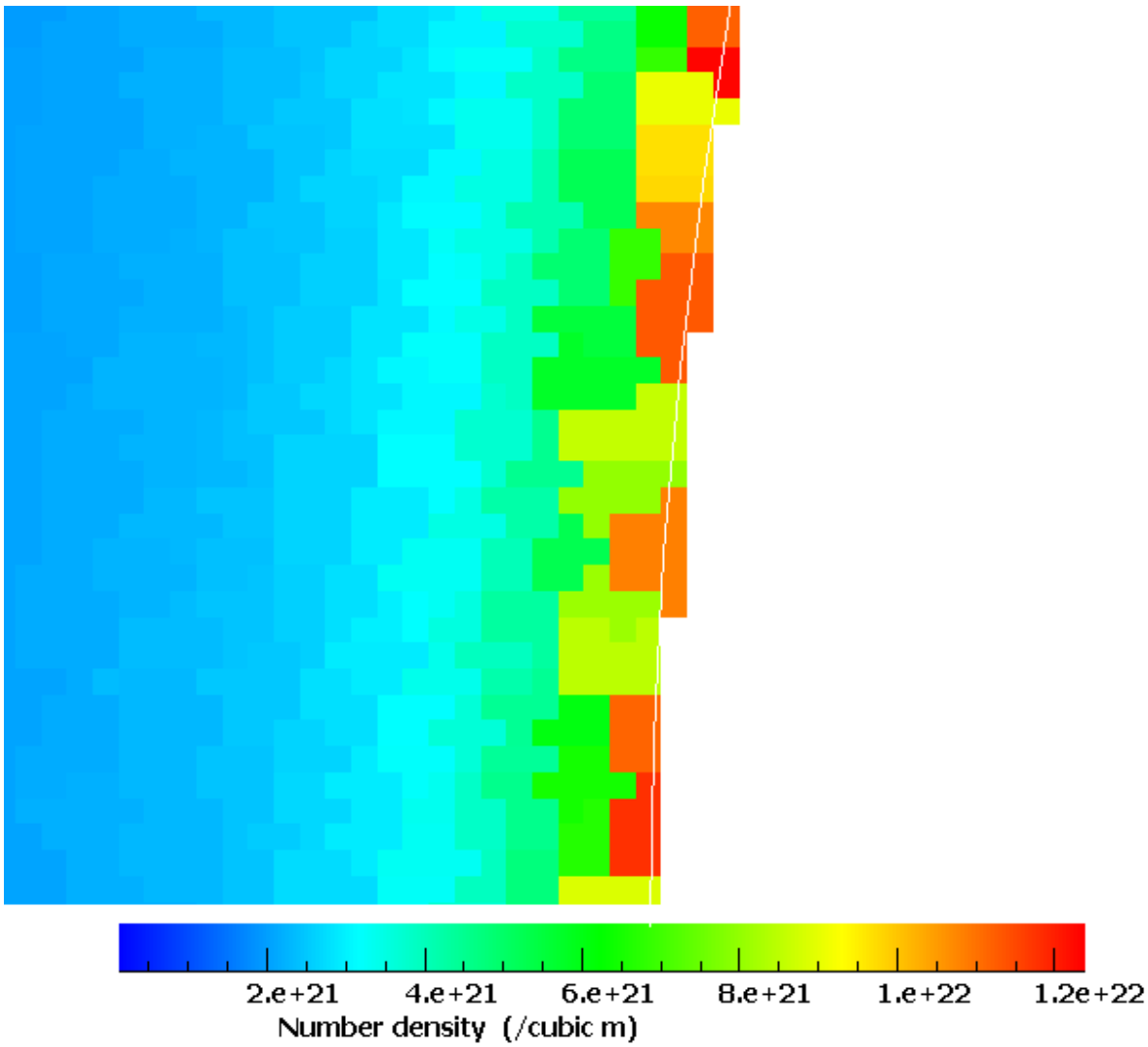


Figure 2.1 A highly magnified view of the DS2V sampling cells near the stagnation point of the benchmark flow.

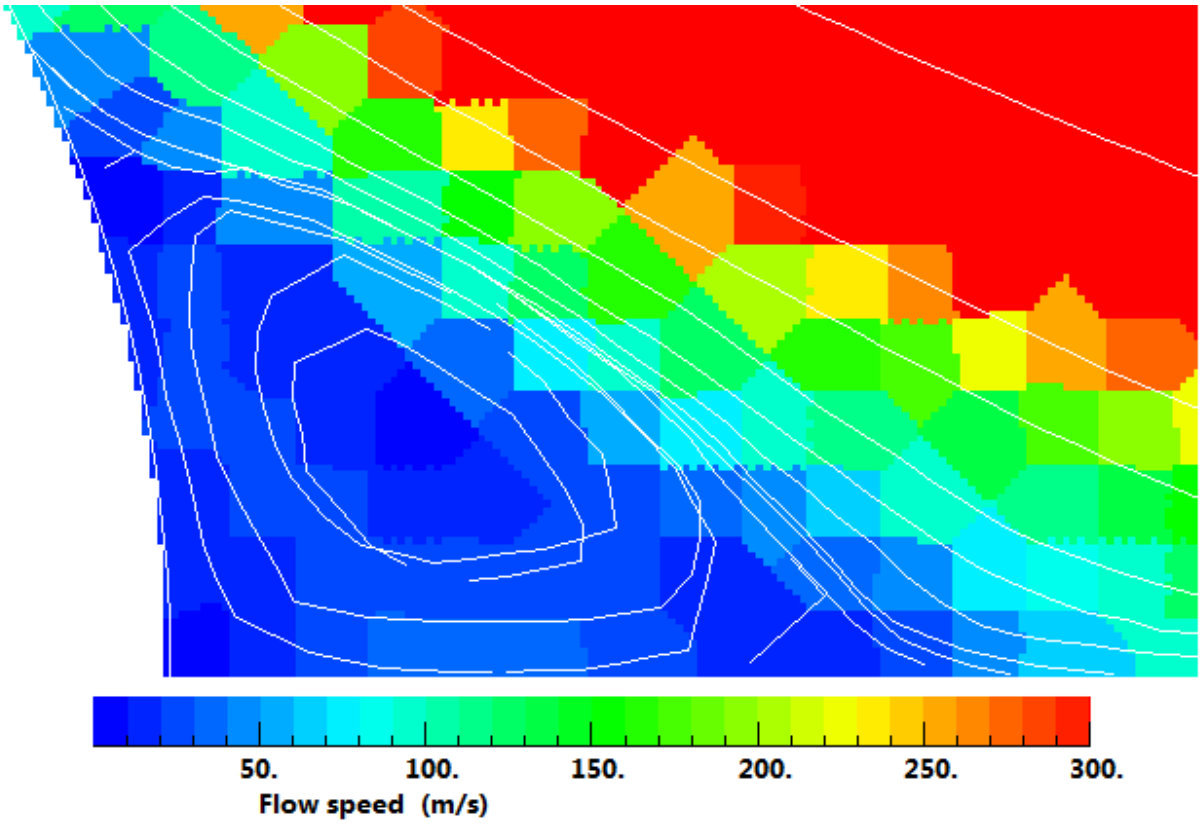


Figure 2.2. DS2V sampling cells in the vicinity of the vortex in the benchmark flow calculated with two molecules per collision cell (as in Figure 1.2).

Typical shapes of the cells that are grown from elements are shown in Figure 2.1 for the front stagnation region and 2.2 for the rear stagnation region that includes the vortex. The total number of elements in the x and y direction in the flowfield was 1150 and 540, respectively. The degree of zoom in Figure 2.1 relative to Figure 1.1 is more than 20:1. The density is uniform over a cell and its value depends on the distance that the cell extends from the surface. The density gradient normal to the surface can be estimated from this raw data, but these cells do not lead to good contours if processed through a program like Tecplot. Note that these are sampling cells adapted nominally to 24 molecules per cell and the sampling cell values do not affect the quality of the simulation. The collision cells were adapted to 6 molecules per cell and, in the region of highest density, would typically be comprised of 1 to 3 elements. The "growing" process can lead to isolated cells that contain a much smaller or larger number of molecules than the specified value. Also, for flows with large variations in density, if the number of elements per division is sufficiently large for a single-element collision cell in the highest density region to contain no more than six molecules, there are far more elements than are necessary in the regions with comparatively low densities.

It appears desirable to retain rectangular elements, but to vary the number of elements per division to cope with flows with very large density gradients. A further refinement would be to assemble cells from elements in a more organized manner. The divisions are already set to levels relative to the surface. One possibility is that elements in low level divisions also be assigned levels relative to the nearest surface and that the cells be assembled from elements at the same level. They would then conform well to the surfaces and to the large flow gradients normal. For example, if the cells in Figure 2.1 were collision cells, it would be far better if the lower right cell was aligned along the surface rather than normal to it. While rectangular reference systems are ugly but effective, the current implementation is too ugly.

With regard to scatter with the current geometry model, the results in the Introduction suggested that the representation of the flow in the vicinity of the rear stagnation point was surprisingly good. The benchmark calculation was run for 40 hours (on the notebook) and the sample was then one hundred times larger than that in Figure 1.4. The scatter was reduced by an order of magnitude and Figure 2.3 shows the shear stress in the region just upstream of and under the vortex.

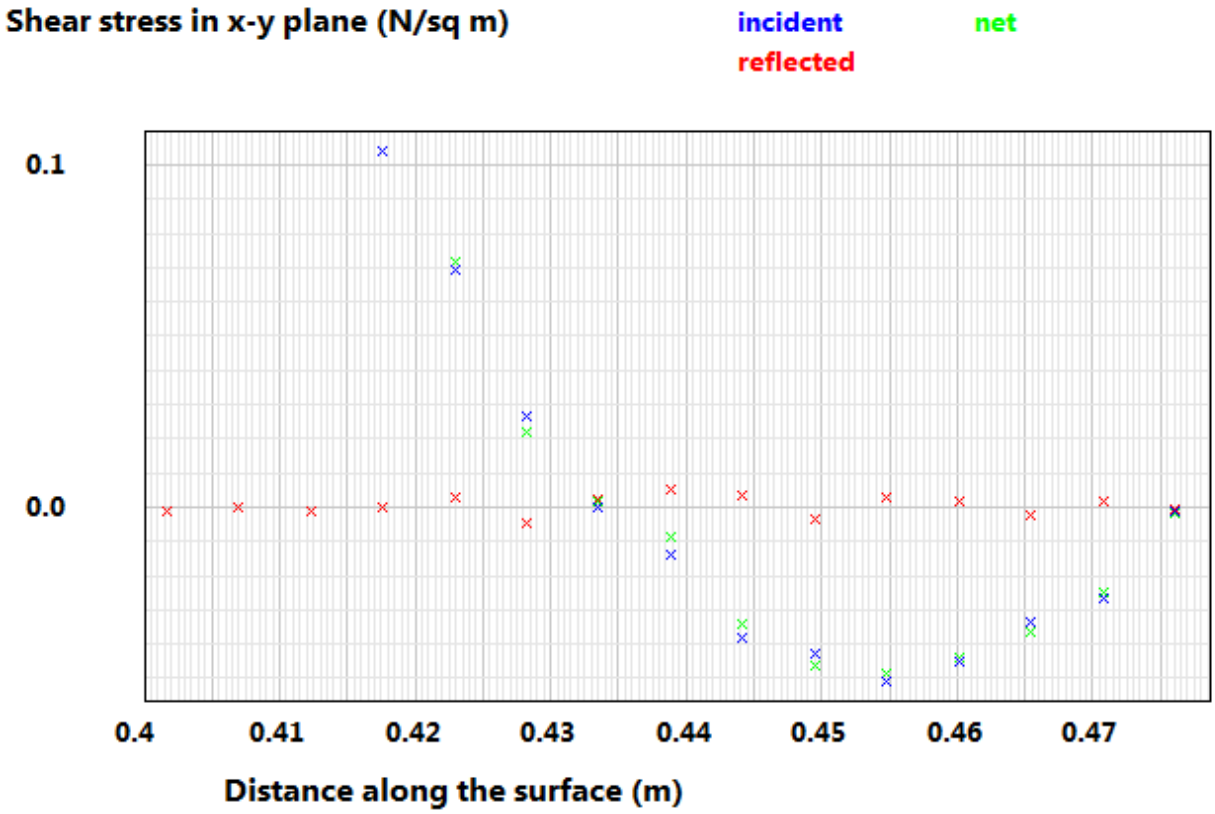


Figure 2.3. The shear stress in the vicinity of the flow separation point.

With an average sample size of 50,000, the separation point is well defined and useful results are obtained for the shear stress. One of the myths that surround DSMC is that it can be used only for high speed flows. The average flow velocity outside the boundary layer above the surface segments in Figure 2.3 is less than 10

m/s. This is less than is indicated by Figure 2.2, but that is dealt with in Section 5. It is also worth showing the effect of the increase sample on the shear stress distribution over the full surface. The scatter has been reduced by the expected factor of ten, but remains larger than would be expected in the vicinity of the maximum shear stress.

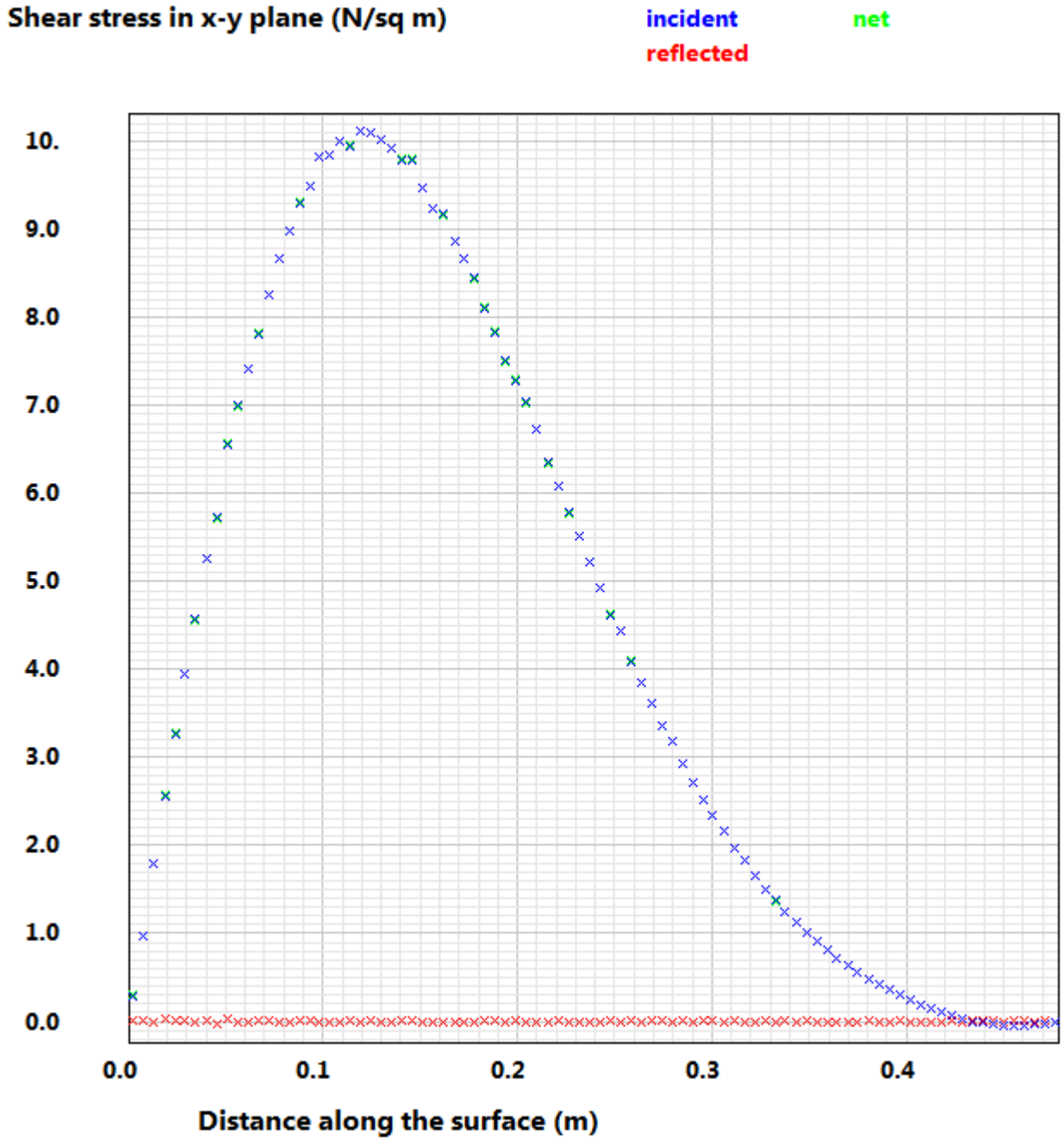


Figure 2.4. Shear stress distribution of Figure 1.5, but with 100 times the sample.

Another factor that affects the quality of the results is whether curved surfaces are represented by the exact shape or by straight line segments. The author is indebted to Mr. Hadley Cave of Christchurch University for raising this issue.

All calculations for the cylinder flow have been made with the cylinder represented by two degree straight line segments. The benchmark case was repeated for half degree segments. The DS2V program currently has a one-to-one correspondence between the size of the segments and the intervals for sampling the surface properties. This should be changed to permit the amalgamation of definition segments so that the number sampling intervals is not increased by the better resolution. In any case, there is a computational time penalty associated with a finer resolution.

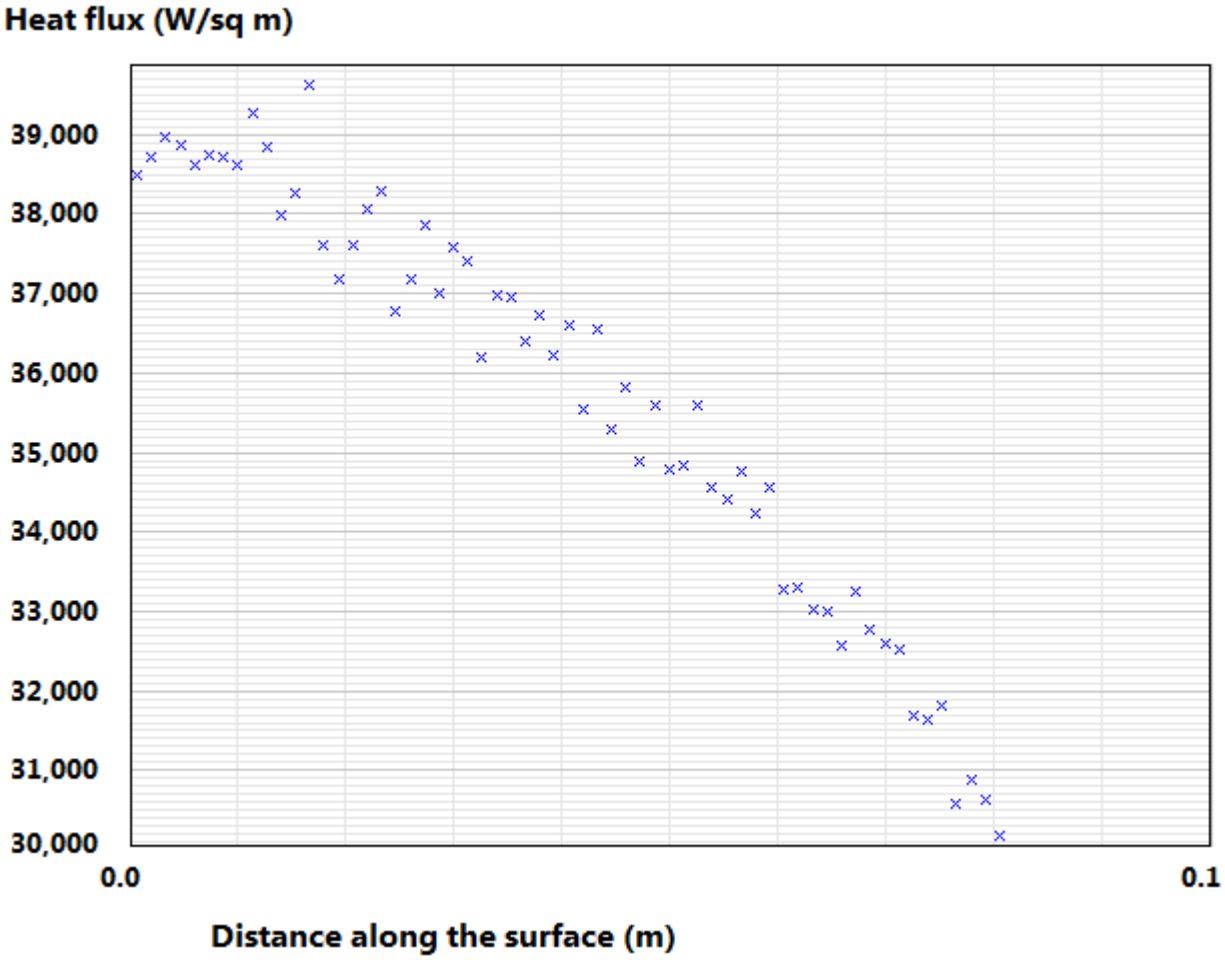


Figure 2.5 Effect of a finer surface definition by half degree straight line segments.

The unacceptable level of systematic scatter due to the poor cell structure with the current DS2V geometry model became more evident. The value for the stagnation point heat transfer is about 38,800 ± 200 W/sq m and the converged value of this quantity in the benchmark study should be adjusted up by about 1%.

The reason why DSMC cells can be highly irregular, while finite difference calculations demand a carefully crafted mesh, is that the DSMC procedures do not require the determination of the gradients in the flow properties. Gradients may be required as an output quantity from a DSMC calculation and the DS2V program includes the evaluation of the Chapman-Enskog perturbation terms as one of five options for the indication of the degree of non-equilibrium in the flow. It is instructive to compare the results before and after the adaption of the cells.

Figure 2.5 shows that, before the adaption, the cell shapes very nearly conform to the rectangular divisions and this leads to a pixellation of the contours that follows the divisions. The sample size in these calculations was four times that in the benchmark calculations.

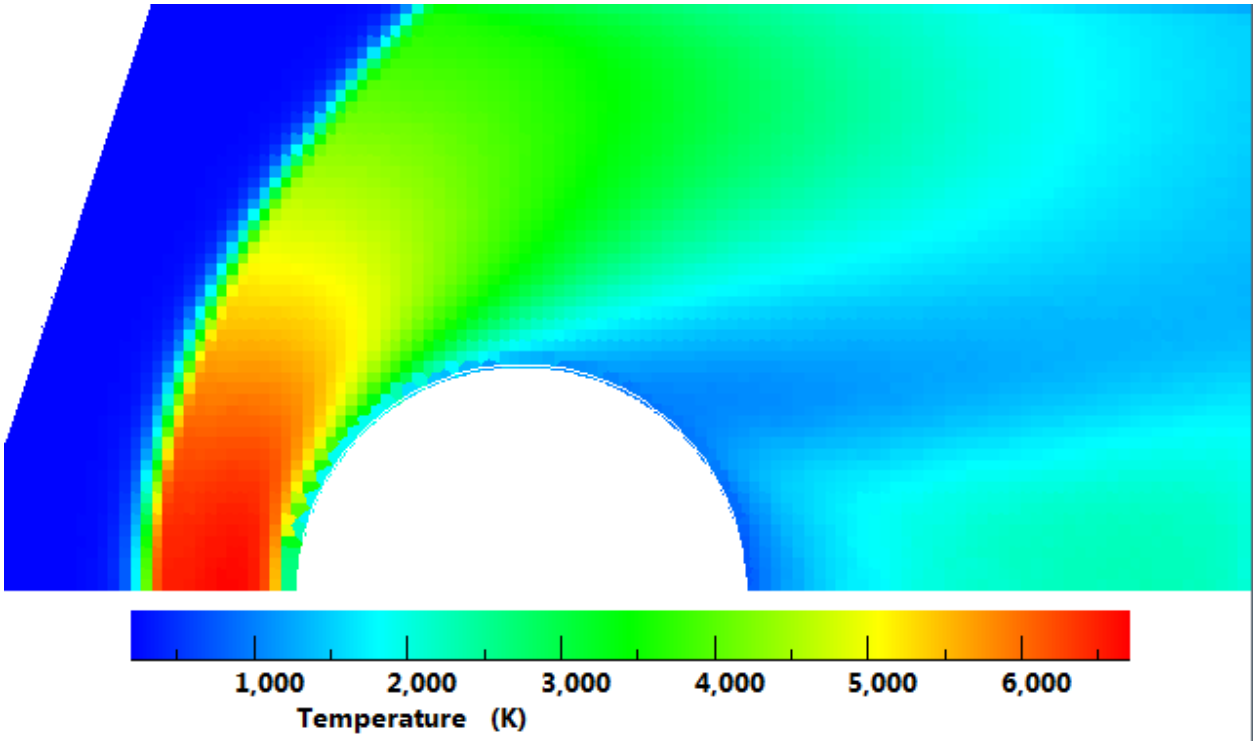


Figure 2.5. Temperature of the unadapted flow that corresponds to Figure 1.1.

Figure 2.6 shows contours of the largest of the perturbation terms based on either the velocity gradients or the temperature gradients in the unadapted benchmark flow. While the contours are again pixellated by the divisions, the results are of a quality that provides the desired information.

The Navier-Stokes equations are based on the Chapman-Enskog theory for the transport properties. This theory is generally considered to be valid as long as the perturbation terms are less than 0.1 and to be unusable when they exceed 0.2. For the benchmark flow, the Navier-Stokes equations are clearly invalid in a large region at the rear of the cylinder as well as within the shock wave. The boundary layer at the front of the cylinder has not been well resolved.

Because the adapted cells are highly irregular, the gradients were not obtained with sufficient accuracy for the results in Figure 2.7 to be useful. The irregularities are such that the maximum contour value in the legend is more than 50% too high.

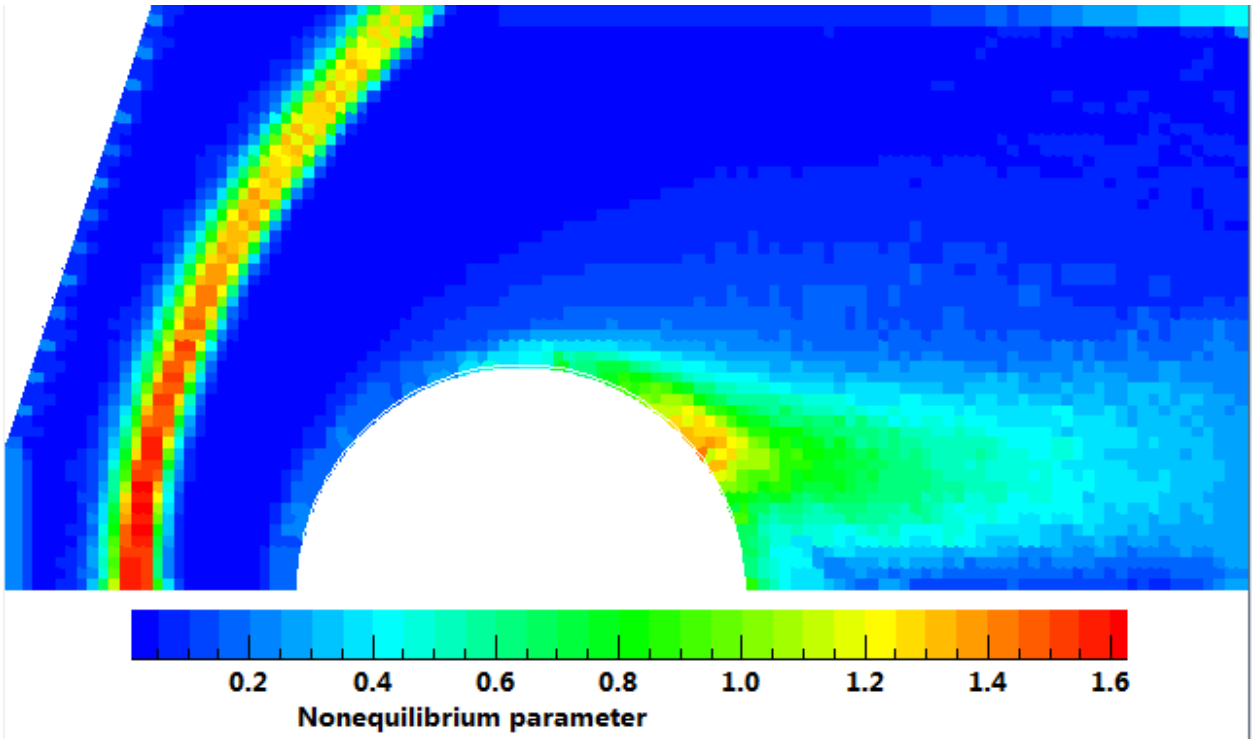


Figure 2.6. The maximum Chapman-Enskog perturbation terms before adaption.

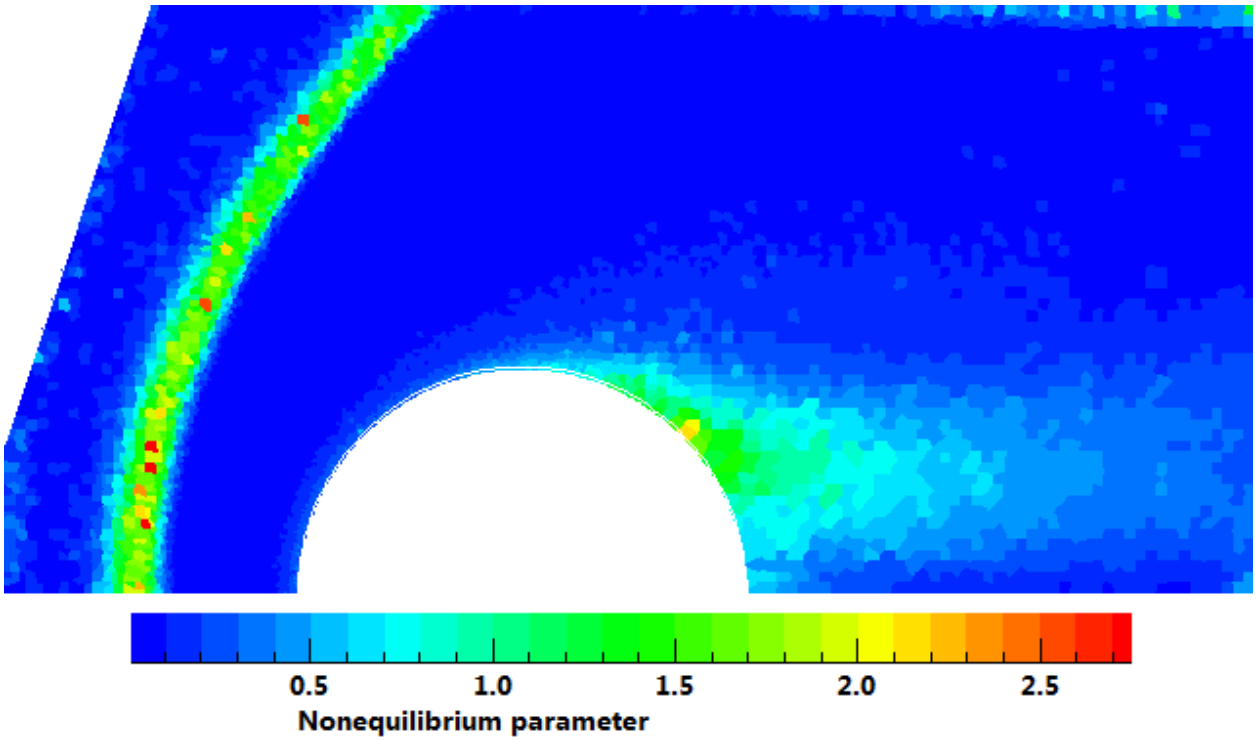


Figure 2.7. The maximum Chapman-Enskog perturbation terms after adaption.

The current version of the DS2V program evaluates gradients only for this optional output quantity and the irregular cells that are produced by the flow adaption do not affect the quality of the simulation. However, the program does allow a small degree of ionization on the assumption that the gas is electrically neutral and the electron temperature is equal to the ion temperature. The next step would be to include ambipolar diffusion and this would require the evaluation of electron density gradients. There are other situations in which gradients may be required and it appears desirable to move to adaptive rectangular cells such as the tree-like structure referred to earlier.

The default nonequilibrium parameter is the root mean square value over the three spatial components of the asymmetry in the pressure tensor. The results for the benchmark case are shown in Figure 2.8. The asymmetry is measured by the ratio of the temperature components to the overall temperature. The x component of temperature is greater than the y component over most of the flowfield but the opposite applies in the wake. This means that there is a region in which the two components are near equal and the parameter is then very small. This shows in Figure 2.8 as the dark blue band above the wake. The flow in not in equilibrium in that region and this effect must be kept in mind when interpreting these plots.

The main motivation behind the quantitative measurement of nonequilibrium is to identify near isentropic flow regions that would allow collision limiting to be applied in DSMC calculations. Shock waves are clearly distinguished, but boundary layers tend to be lost in the scatter.

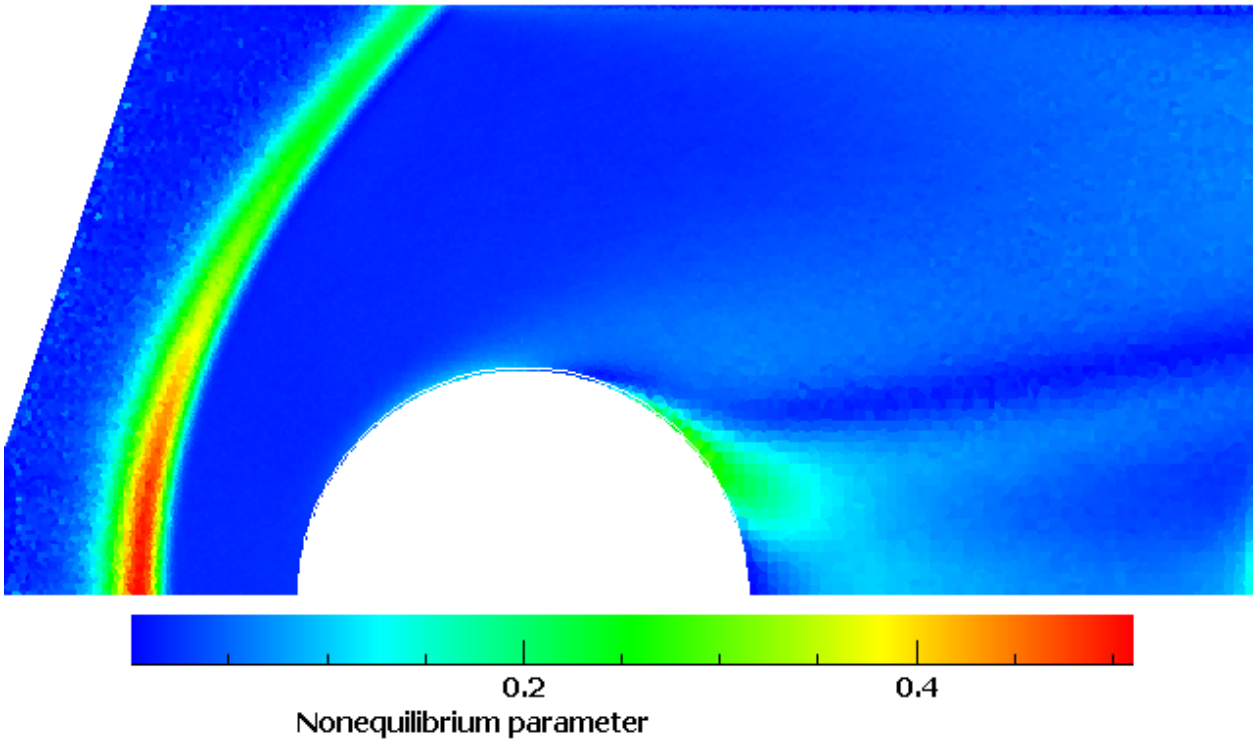


Figure 2.8 The root mean square asymmetry of the pressure tensor.

Almost all the flow plots for the benchmark case have employed the least possible number of simulated molecules and the cell structure is apparent in the results. These computational artifacts disappear when a very large number of molecules is employed. The following figure is from a DS2V calculation with 36 million molecules made by Dr Wolfgang Christen of the Humboldt University of Berlin. This employed a PC with 64 bit Windows that allows more than 3.2 GB of memory.

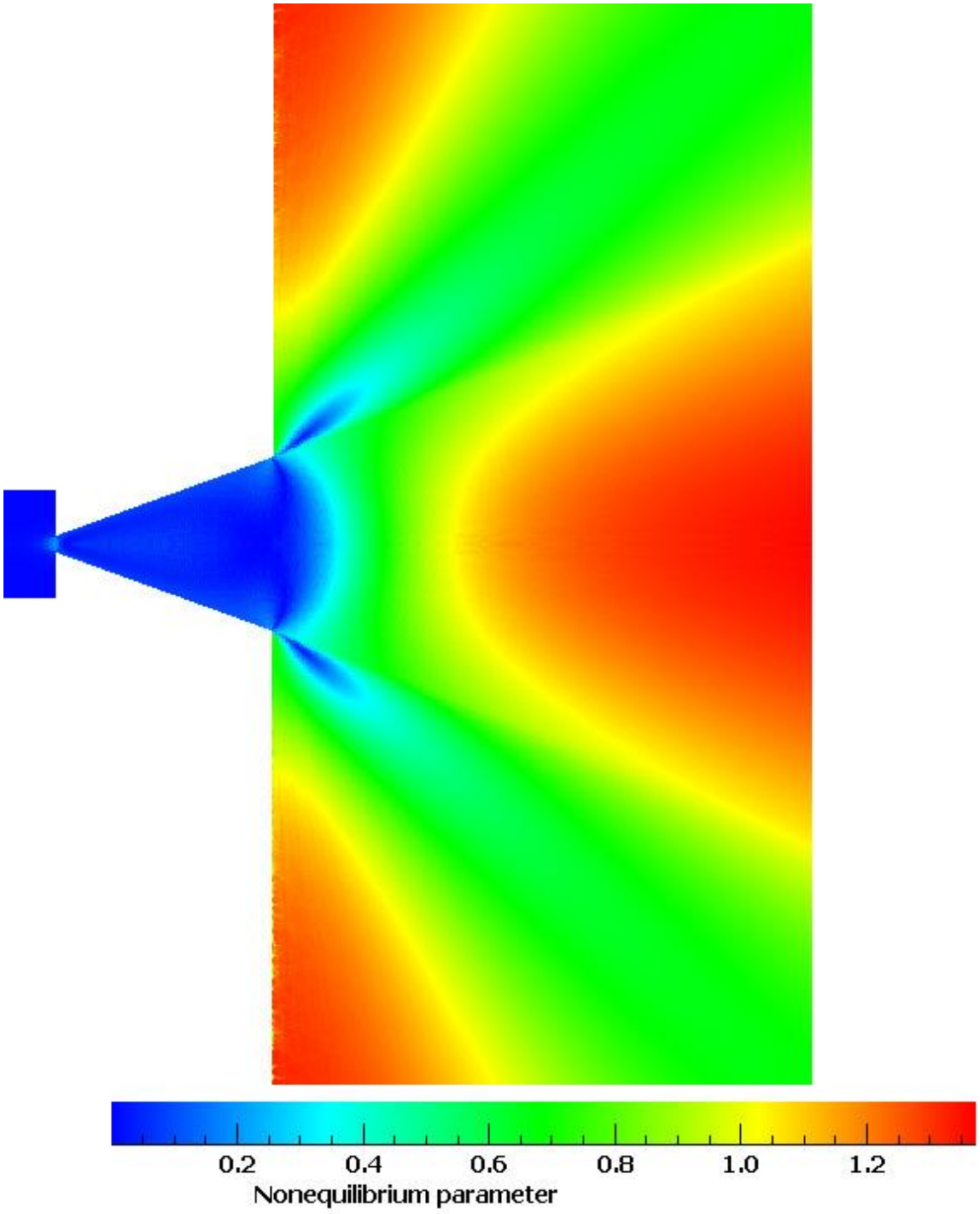


Figure 2.9 Pressure tensor asymmetry in an axially symmetric expansion.

3. Nearest-neighbor procedures

The finite difference methods of continuum CFD stores data only at grid points and finite volume methods assume uniform values within an element. On the other hand, the locations and velocities of the molecules within a DSMC cell define physically meaningful structure (e.g. vorticity) within the cell. Full advantage is not taken of this inherent DSMC advantage unless the program includes procedures that promote nearest-neighbor collisions.

As the geometry models become more complex, it becomes difficult to define what is meant by "cell size". However, early on in the development of DSMC, it became clear that the reason why smaller cells were beneficial was that they reduced the separation distance between the molecules that were selected as collision partners. This led to the introduction of sub-cells in order to reduce the separation distance. The immediate trigger for sub-cells was a paper critical of DSMC that demonstrated that a collision between molecules from opposite corners of a cell could reverse the angular momentum associated with that molecular pair. While that proved to be an unlikely extreme, excessive collisional separation does reduce the angular momentum of the molecules within a cell. The paper neglected to mention that the existence of any angular momentum within a cell is an advantage over most continuum CFD methods that carry angular only through the macroscopic velocity gradients from cell to cell or from grid-point to grid-point.

Sub-cells were first introduced in the G2 program that employed ray-tracing and the collisions of the molecules had to be calculated for the sub-cell boundaries as well as the cell boundaries. The resulting computational time penalty could be tolerated only for small numbers of sub-cells. This led to the introduction of "transient-adaptive" sub-cells in Version 4 of the DS2G program. This involved placing a rectangular grid over a cell only when collisions were to be calculated within that cell. The molecules within that cell were then indexed to the grid. Because it involves additional storage for only one cell at a time and the additional computation time depends only slightly on the grid size, the grid can be sufficiently fine to ensure nearest-neighbor collisions for that cell.

Early attempts had been made to implement the obvious nearest neighbor procedure that, following the random selection of the first molecule, the distance from this molecule of all other molecules is calculated and the nearest molecule is chosen as the second molecule. This was abandoned because the computation time is proportional to the square of the number of molecules and linear procedures had always been considered to be mandatory for DSMC.

Direct selection was implemented in the DAC program as "virtual sub-cells" and was found to be feasible. It was then implemented in the DS2V/3V programs alongside virtual sub-cells and it was found that direct selection was faster than transient-adaptive sub cells when the number of molecules per cell is less than a number between 30 and 40. For flows with large density variations an alternative

to direct selection is needed until the cells are adapted to a small and near equal number of molecules per cell.

A problem with direct selection is that, if either of the molecules in the preceding collision is chosen as the first molecule, the other molecule in the preceding collision will still be the nearest molecule and will therefore be again chosen. It is therefore necessary to keep a record of the previous collision partner of each molecule and to choose the next closest molecule if the previous partner is again chosen. This has a significant beneficial effect on the calculation, but test calculations indicate that it is not necessary to carry the exclusion back beyond the preceding collision.

The current procedures do not provide a full nearest-neighbor capability because, for molecules near the boundary of the cell, they do not look across to neighboring cells to find whether there are closer molecules in those cells. Doing so would add computing time as well as complication and it is not clear whether this would be worth the effort.

4. Separate collision and sampling cells

Why does it take so long to see obvious (in retrospect) solutions?

When the number of molecules per collision cell is as small as is now seen to be desirable, it is not practical to sample the flow properties to the same resolution and it is necessary to provide for separate collision and sampling cells. These were first implemented in the SMILE code. It is also desirable for the results to be based on a uniform sample and, as with the collision cells, the sampling cells should be adapted to a uniform number of molecules per cell. Because DSMC is a simulation at the molecular level, none of the macroscopic properties from the sampling cells should appear in the DSMC procedures for collisions or surface interactions.

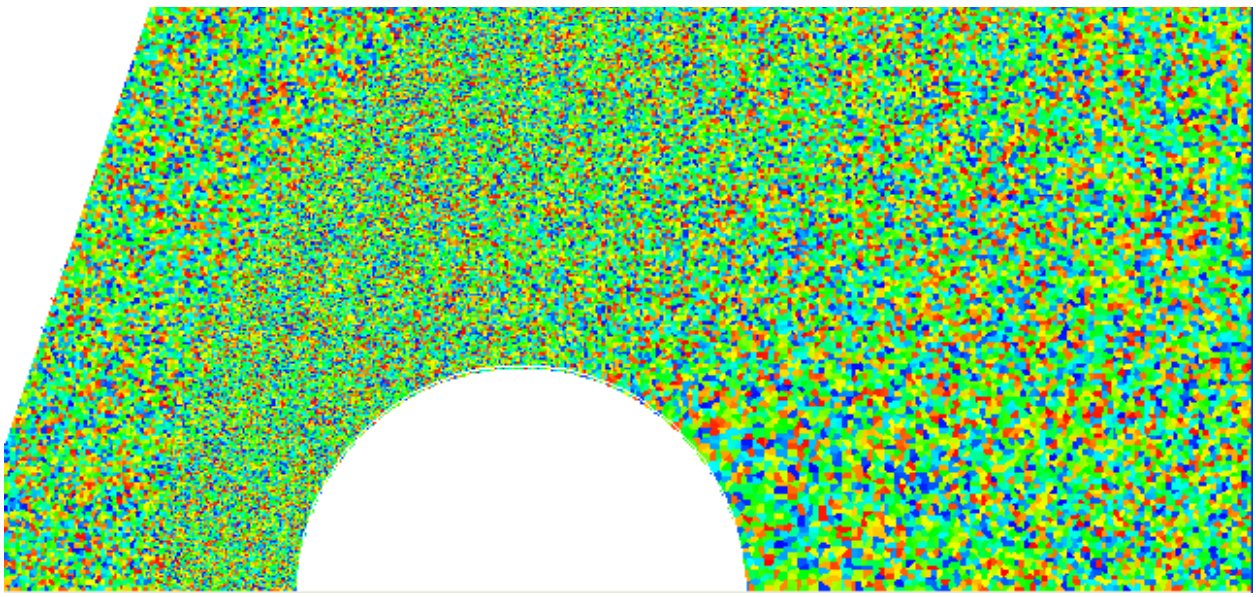


Figure 5.1 The adapted collision cells in the benchmark calculation

The adapted collision cells for the benchmark case are shown in Figure 5.1. This is again a cut and paste from the DS2V GUI and cells near the surface may be smaller than a single pixel. Random colors have been applied to the cells and some adjacent cells will have been assigned similar colors. The DS2V geometry model allows a very rapid adaption. The computation time that was required for the collision cell adaption was 2.268 seconds and the adaption of the sampling cells required 0.742 seconds. These timings were measured in the code compiled by the slower CVF compiler.

5. Very small adaptive collision cells

The lucky number is…….seven!

Nearest neighbor collisions minimize the mean collisional separation, but the collision rate depends on the number density that is resolved by the collision cell. The smaller the cell, the greater the resolution of the flow and the more accurate the solution. The early DSMC guidelines were generally to the effect that the number of molecules per cell should not fall below 10 and that the larger the number, the better the calculation. This is no longer the case and the DS2V program currently recommends 8 molecules per collision cell. Several recent papers indicate that the SMILE program appears to have a preferred value of 5 or 6 for this number. Furthermore, it is now recommended that the number should not go above this value and this means that the program must include procedures that adapt the cells to the flow density so that the average number of molecules per cell is uniform at the specified value.

The adaption in the benchmark calculation was to 6 molecules per cell and the adaption in the DS2V case in Table 1 was to 8 molecules per collision cell. This was repeated with adaption to other values. The results (± 0.1) for the stagnation point heat transfer (converged 38.0) were:-

Unadapted	39.4 KW/sq m
Adapted to 32 mols./cell	39.0 KW/sq m
Adapted to 16 mols./cell	38.7 KW/sq m
Adapted to 8 mols./cell	38.3 KW/sq m
Adapted to 6 mols./cell	38.2 KW/sq m
Adapted to 4 mols./cell	37.9 KW/sq m
Adapted to 2 mols./cell	37.8 KW/sq m

This was a disturbing result because it indicated convergence to a lower value of the stagnation point heat transfer than was obtained with convergence based on the total molecule number. However, it was not expected that the collision procedures would be employed with such small number of molecules per collision cell and a check was made of the total number of collision events in the whole flowfield per unit time. The results were:

Adapted to 32 mols./cell	1.005e+11 /sec
Adapted to 16 mols./cell	1.009e+11 /sec
Adapted to 8 mols./cell	1.018e+11 /sec
Adapted to 6 mols./cell	1.032e+11 /sec
Adapted to 4 mols./cell	1.085e+11 /sec
Adapted to 2 mols./cell	1.362e+11 /sec

The fact that the magnitude of these numbers tends to unity as the number of molecules per cell increases is coincidental and the correct number is not unity. At the same time, there is consistent upward trend that accelerated rapidly as the number falls below six. At an average two simulated molecules per collision cell, there are about one third too many collisions. This means that the stagnation point heat transfer values for a small number of molecules do not indicate a converged value less than 38.0 KW/sq m. They merely reflect the fact that the 2 mols./cell case was effectively made at a lower Knudsen number.

The most useful test case is always a uniform equilibrium gas for which an analytical result is available. The collision rate was measured in a DSMC simulation of a hard sphere gas and the collision rate was compared with the theoretical value. The results for the ratio of the computational rate to the theoretical rate are:

Adapted to 32 mols./cell	0.993
Adapted to 16 mols./cell	0.992
Adapted to 8 mols./cell	0.995
Adapted to 6 mols./cell	1.001
Adapted to 4 mols./cell	1.041
Adapted to 2 mols./cell	1.516

The 32 mols./cell case had a sample size such that the collision rate ratio would have been 1.000 in a simple DSMC calculation. It appears that the sophisticated DSMC procedures have a systematic error that causes the collision rate to be about 0.7% too small. The excess number of collisions when there are only two molecules per collision cell is even greater than with the benchmark calculation. Because the collision rate goes from slightly low to much too great, there is a value of molecules per collision cell that fortuitously leads to the correct collision rate. This value is less than eight, but greater than six.

A problem with increasing the resolution of the flow is that unphysical effects may also be enhanced. Unphysical effects are most likely near the axis of an axially symmetric flow in a calculation with radial weighting factors. The purpose of weighting factors is to make the uniform flow molecule sample independent of radius even though the number of molecules in a real axially symmetric flow increases with radius. It is therefore necessary to remove some molecules moving away from the axis and to duplicate some molecules moving towards the axis.

There are obvious problems when identical molecules are present in a cell and these are avoided by imposing a time delay on the molecule duplication process. Even so, the molecule number in these flows exhibits enhanced fluctuation on a timescale much larger than that of the normal fluctuations. It appears that the enhanced fluctuation can produce weak acoustic waves that can grow in strength if they move towards the axis. This sometimes results in a spurious spike in heat transfer at the center of a blunt body. A probable fix is to run identical calculations side-by-side and to swap molecules between them in a checkerboard fashion. Because the flows are identical, physically real disturbances would not be affected but, because they occur at random, spurious disturbances would be broken up.

6. Automatically adaptive variable time steps

The use of a single fixed time step becomes inefficient when the ratio of maximum to minimum density becomes large. The time step should be a set fraction of the local mean collision time and this requires a time step that varies over the flowfield and adapts automatically to the local mean collision time and/or, for high Knudsen number flows, to the transit time of the molecules through the cells.

Variable time steps have been implemented in the DS2V program by continuously updating a desired time step (DTS) in every collision cell. It is set to a specified fraction of the local mean collision time in that cell or, if it is smaller, a specified fraction of the local collision cell transit time. A time parameter is assigned to all molecules and all collision cells. The flow time is advanced in steps equal to the smallest value of DTS. Then, if the time parameter of a molecule falls a time DTS/2 behind the flow time, it is moved the distance appropriate to DTS. Similarly, if the time parameter of a collision cell falls DTS/2 behind the flow time, collisions appropriate to the time interval DTS are calculated for that collision cell.

There is a computation time penalty in that the program cycles through all molecules and collision cells even though only a fraction will be selected for moves and collisions. The penalty is proportional to the ratio of the maximum to the minimum time step. A more severe problem is that the arrays that index the molecules to the collision cells have to be generated at each time step after the computation of the molecule moves and before the computation of the collisions. The indexing array is essentially a list of the molecule numbers in order of the cells. The numbers in this array act as pointers to the molecules that are stored in random order. There is a second array that is dimensioned to the number of cells. The numbers in this array point to the locations of the cell boundaries in the molecule array.

The alternative is an index array that is continuously updated as the molecules move. It requires a prohibitive amount of number shuffling for the length of this array to be kept equal to the total number of molecules. A two-dimensional indexing array with the number of collision cells as one dimension and the maximum number of molecules in any cell as the other must have the second dimension sufficiently large to accommodate any likely value of this maximum number. This often requires an excessive amount of memory, particularly if the cells are not adapted.

The average time step in the benchmark case is seven times the size of the minimum time step and the ratio of the largest to the smallest step would be more than 100. It is therefore a demanding case as far as "normal indexing" is concerned. Even so, the computation time associated with the very frequent indexing was only five percent of the total computation time. A five percent time penalty is small when compared with the complexity (which would itself impose a time penalty within the move and collision routines) and the memory penalty. The "continuous indexing' option within the DS2V/3V programs has now been removed.

7. Modified NTC procedure for collisions

While there are alternatives such as the "majorant frequency" procedure used in the SMILE code, most DSMC codes employ the NTC method for the selection of representative collision pairs. The NTC method is that

$$\tfrac{1}{2} N \overline{N} F_N (\sigma_T c_r)_{\max} \Delta t / V_c \qquad (7.1)$$

potential collision pairs are selected from a collision cell containing N molecules at each time step Δt, and the collisions occur between each pair with probability

$$\frac{\sigma_T c_r}{(\sigma_T c_r)_{\max}}. \qquad (7.2)$$

Here, F_N is the number of real molecules represented by each simulated molecule, V_c is the volume of the collision cell, σ_T is the total collision cross-section, and c_r is the relative speed in the collision. The maximum value of the product of the collision cross-section and the relative speed is the maximum value that has occurred in the collision cell.

There are three problems. First, $\overline{N} F_N / V_c$ is the number density n and it is physically undesirable to have a microscopic collision procedure depend on a macroscopic flow property. The second problem is that the instantaneous number N of molecules in the collision cell is subject to statistical scatter and, for the small number of molecules per collision cell that is now recommended, one molecule in a cell is a frequent occurrence. A collision requires two molecules and it is not clear what action should be taken when N=1. The final problem is that it takes time to establish the value of averaged quantities such as the number density.

The fluctuations in the number N follow the Poisson distribution such that, if the mean value of N is M the probability of a particular value is

$$P(N) = M^N \exp(-M) / N! \qquad (7.3)$$

Eq. (7.1) shows that the overall collision rate is set by the mean value of the product of N and $\overline{N}$ and, because $\overline{N} \equiv M$ and M is a constant in Eq. (7.3), this can be written

$$\sum_{N=0}^{\infty} NM^{N+1} \exp(-M)/N! \qquad (7.4)$$

This may be compared with the mean value of $N(N-1)$ in a Poisson distribution

$$\sum_{N=0}^{\infty} N(N-1)M^{N} \exp(-M)/N! \qquad (7.5)$$

The first term of both series is zero and the second term of Eq. (7.5) is also zero. It is easily shown that, for n>0, term n+1 of Eq. (7.5) is equal to term n of Eq. (7.4), so the two series are identical.

This means that $N\overline{N}$ in Eq. (7.1) can be replaced by $N(N-1)$. This solves all three problems.

8. Discontinuous and event-driven physical processes

Continuum methods are unsuited to the modeling of the physical processes in high temperature gas flows and, when two molecules collide, they are completely unaware of the overall gas temperature.

Early DSMC procedures necessarily employed the molecular models that had been developed in the context of classical kinetic theory and were strongly influenced by continuum models and data. The DSMC method has led to the development of a number of phenomenological gas models that are, in many respects, superior to the models of classical kinetic theory. A phenomenological model reproduces the physical behavior of a real gas in the simplest possible manner. While some aspects of these models may appear to be unphysical, there is physical information associated with the determination of the physical features that can be neglected.

The quantum vibration model that was developed by Frank Bergemann as a PhD student at Göttingen has been the most exciting of these models. This appeared when the manuscript for the 1994 monograph [3] was almost complete and its presentation was combined with that of the clumsy and computationally inefficient classical model that had previously been employed. The classical model had employed an unsatisfactory definition of vibrational temperature and the quantum definition appeared only in the discussion of the demonstration program. This is

$$T_{vib} = \Theta_v / \ln(1+1/\bar{i}), \qquad (8.1)$$

where Θ_v is the characteristic temperature of the mode. Note that the effective number of degrees of freedom of vibration ζ_{vib} is

$$\varsigma_{vib} = 2\bar{i} \ln(1+1/\bar{i}). \qquad (8.2)$$

An advantage of the quantum model is that it avoids the use of this quantity, but it is required for the computation of the overall temperature of the gas.

The most serious problem was that equipartition between the energy modes in an equilibrium gas was not achieved at that time unless the overall flow temperature was employed in the procedure. It is undesirable for any DSMC procedure to be based on a macroscopic property such as the gas temperature. The two molecules in a collision have no knowledge of the overall temperature and, if highly nonequilibrium flows are to be treated adequately, the procedures for energy redistribution in a collision must be based entirely on the energies and impact parameters associated with that collision.

This issue may be illustrated through the current implementation of the quantum vibration model. The author is indebted to Mr. Derek Liechty of NASA Langley and Dr Richard Wilmoth (formerly with NASA Langley) for recent discussions about the collision energy based procedure.

The code in the current Version 4.4.02 of the DS2V program is:-

```
1  IF (ISPV(KS) > 0) THEN      !-considering molecule K (one of the collision pair) which is species KS
2    DO KV=1,ISPV(KS)          !-ISPV(KS) is the number of vibrational modes of species KS
3      EVIB=IPV(KV,K)*BOLTZ*SPV(KV,KS)  !-IPV(KV,K) is pre-collision level for this mode of molecule K, it
is multiplied by the Bolltzmann constant and the characteristic vibrational temperature of this mode SPV(KV,KS) to
give the pre-collision vibrational energy EVIB of molecule K
4      ECC=ECT+EVIB              !-the redistribution of energy to this mode is from the sum of EVIB and the relative
translational energy in the collision ECT (n.b. Bergemann's discovery that the redistribution does not have to be from
the total energy in the collision, as had previously been assumed.)
5      IF (SPVM(2,KV,KS) > 0.) THEN    !- SPVM(2,KV,KS) is the constant C1 in eqn (6.53) of  for Zv
6        MAXLEV=ECC/(BOLTZ*SPV(KV,KS))  !-the highest level  with energy under ECC
7        COLT=MAXLEV*SPV(KV,KS)/(3.5-SPM(3,KS,JS))         !- COLT is the "quantized collision temperature".
The energy associated with MAXLEV is divided by the factor Ξ in eqns.(5.42) and (11.34) of (Bird, 1994).
8          A=SPVM(2,KV,KS)*(COLT**(-0.3333333))
9          IF (A < 50.) THEN
10           ZV=(SPVM(1,KV,KS)/(COLT**SPM(3,KS,JS)))*EXP(A)   !-Zv from eqn (6.53) of (Bird, 1994)
11         ELSE
12           ZV=1.E7  !- avoids 32 bit floating point error with large exponential arguments
13         END IF
14       ELSE     !- Zv is a constant if  SPVM(2, is 0
15         ZV=SPVM(1,KV,KS)  !—for SPVM(2 =0, a  constant Zv is given by SPVM(1
16       END IF
17       CALL RANDOM_NUMBER(RANF)
18       IF (1./ZV > RANF) THEN     !-perform the redistribution for this collision
19         II=0       !- variable to indicate when a successful choice has been made
20         DO WHILE (II == 0)      !-acceptance-rejection procedure
21           CALL RANDOM_NUMBER(RANF)
22           IV=RANF*(MAXLEV+0.99999)   !-select a random level within the energy range
23           IF (IV > 127) IV=127  !- four bit integer for the state (dissociation generally occurs around state 50)
24           IPV(KV,K)=IV  !- provisionally set the post-collision level of mode KV of molecule K
25           EVIB=IV*BOLTZ*SPV(KV,KS)
36           IF (EVIB < ECC) THEN    !- this check is probably a redundant
27             PROB=(1.-EVIB/ECC)**(1.5-SPM(3,KS,JS))     !-PROB is Bergemann's probability ratio and the
quantum vibration model is essentially contained in this single short statement
28             CALL RANDOM_NUMBER(RANF)
29             IF (PROB > RANF) II=1   !-accepted
30           END IF
31         END DO
32         ECT=ECC-EVIB  !-adjusts the value of the relative translational energy
33       END IF
34     END DO
35   END IF
```

The physical law that has to be obeyed is the equipartition of energy between the energy modes when the gas is in equilibrium. The procedure must also lead to relaxation rates that match the accepted Millikan-White data for the vibrational relaxation rates as a function of the flow temperature. It appeared for some years that vibrational equipartition could be achieved only if the vibrational collision number was based on the macroscopic temperature and this was explicitly stated in a paper by Bird (Phys. Fluids, **14**, May 2002). Fortunately, the statement proved to be incorrect and the eventual solution to that problem appears in lines 6 and 7 of the above code. With the "collision temperature" quantized in the same fashion as the vibrational energy and not otherwise dependent on the collision energy, exact equipartition is achieved.

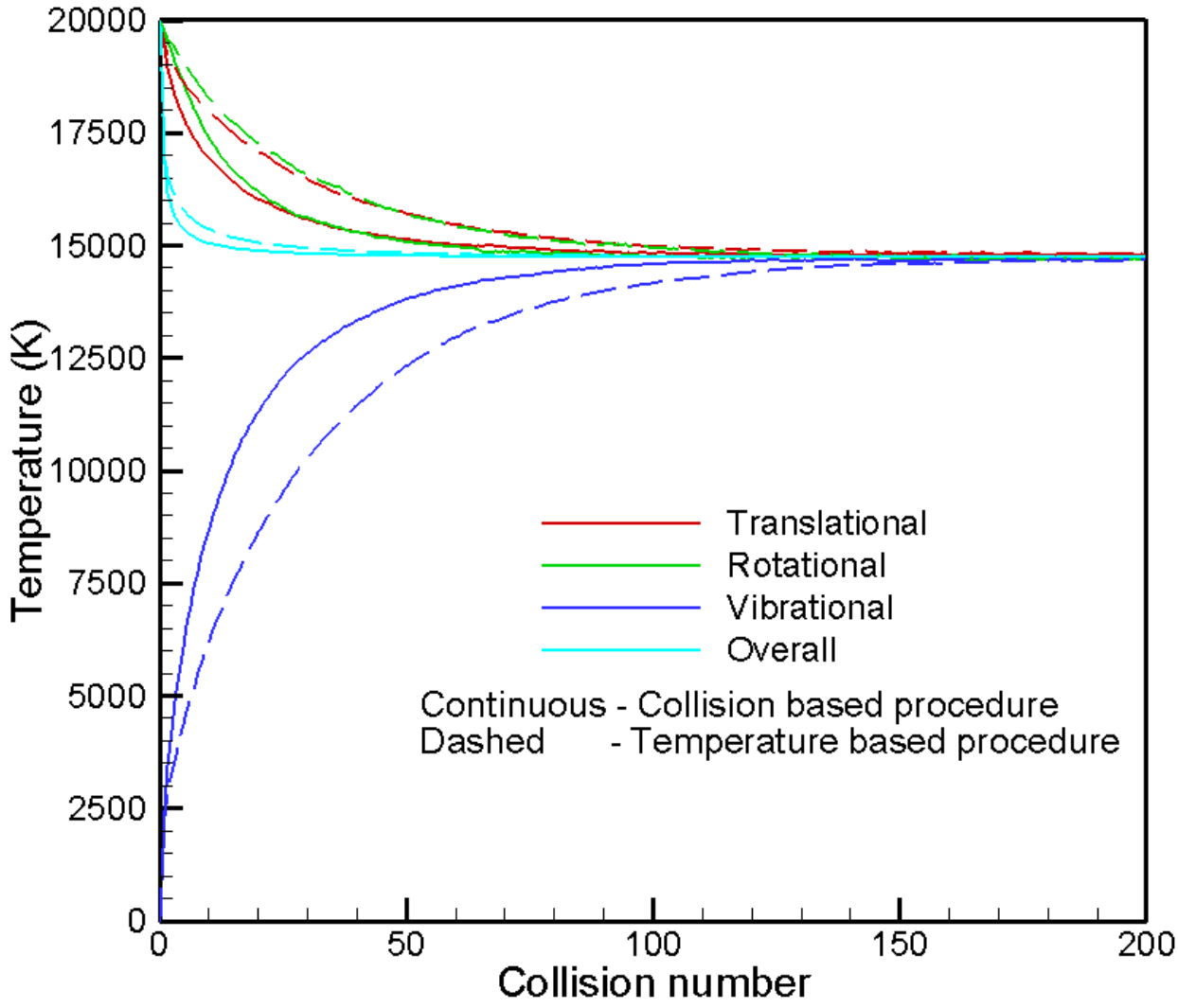


Figure 8.1 Vibrational relaxation in nitrogen

Figure 8.1 shows the results from 1,200,000 molecule DS2V calculations for nitrogen relaxation. The gas is initially at a temperature of 20,000 K as far as the translational and rotational modes are concerned, but with no energy in vibration.

It is a homogeneous gas calculation, but the initial state is similar to that behind the bow shock of a hypersonic blunt body. The figure also shows the relaxation curves that are produced when the relaxation probability is based on the overall flow temperature rather than on the energies in the collisions. The initial vibrational temperature gradients are similar and are consistent with the Millikan-White collision number of $Z_v = 19.8$ for nitrogen. This increases to $Z_v = 58.9$ at the equilibrium temperature of 14,720 K. and, because the overall temperature reflects the total energy in the gas and falls almost instantly to the equilibrium value, the temperature based vibrational temperature lags the collision based value. This is unphysical and the rate from the collision based procedure correctly reflects the fact that the translational and rotational temperatures are well above the overall temperature. The temperature based procedure would have led to better values if the translational temperature had been used in place of the overall temperature.

The correct vibrational excitation rate was obtained only after the form of the vibrational rate data was changed from that in the above code listing. The rate from the above code is too high and this was the subject of the recent discussions.

The rate in the above code is based on equation (6.53) of Reference [3] that employs the VHS collision rate to convert the Millikan-White data fit for τ_v to the following equation

$$Z_v = (C_1/T^{\omega}) \exp(C_2\, T^{-1/3}) \quad , \qquad (8.1)$$

where ω is the temperature-viscosity power law and C_1 and C_2 are constants. With the temperature based procedure, T is a constant in any collision cell and the relaxation rate is consistent with its value. However, the "collision temperature" in the collision based procedure varies over a wide range and the rate is correct only if the constants in equation (8.1) are consistent over a wide range of temperatures. The problem was that the suggested values of the constants C_1 and C_2 for nitrogen led to $Z_v = 0.015$ at the dissociation temperature. This is unphysical and the collision number for many collisions was far too small.

The constants C_1 and C_2 have no physical significance and it is preferable to replace them by Z_d which is the value of Z_v at the characteristic temperature of dissociation Θ_d and Z_c which is the value of Z_v at the characteristic temperature of vibration Θ_v. Equation (8.1) then becomes

$$Z_v = Z_d \left(\frac{\Theta_d}{T}\right)^{\omega} \left[\frac{Z_c}{Z_d}\left(\frac{\Theta_v}{\Theta_d}\right)^{\omega}\right]^{\left[\left(\frac{\Theta_d}{T}\right)^{1/3}-1\right]\Big/\left[\left(\frac{\Theta_d}{\Theta_v}\right)^{1/3}-1\right]}. \qquad (8.2)$$

The collision numbers are kept in a physically realistic range if Z_d is set to unity. This choice is supported by a classical analysis in Reference [3] that showed that setting Z_d to unity leads to the correct dissociation rate. Also, it is preferable to base the data on the value of Z_v at a reference temperature rather than at θ_v. i.e.

$$Z_v = \left(\frac{\Theta_d}{T}\right)^{\omega} \left[Z_{ref}\left(\frac{\Theta_d}{T_{ref}}\right)^{-\omega}\right]^{\left[\left(\frac{\Theta_d}{T}\right)^{1/3}-1\right]\Big/\left[\left(\frac{\Theta_d}{T_{ref}}\right)^{1/3}-1\right]}. \qquad (8.3)$$

For a procedure based on equation (8,3), the required data for each vibrational mode is just the vibrational collision number at the reference temperature and this should be representative of the range of temperatures in the application. The one-third power law is preserved, but there is a decrease in the rate at higher temperatures. It is essentially a more precise version of the "Park correction factor".

A practical problem is that the current database for the temperature dependent values of Z_v for real gases is unsatisfactory. Z_v is not measured directly and the conversion from the raw data to the presented data is often based on questionable and undocumented theory. Physicists and chemists almost invariably employ hard sphere kinetic theory and, at high temperatures, the percentage errors can be in the hundreds. DSMC simulations of the actual experiments should be made whenever possible. Also, it should be noted that the above DSMC procedures are applied separately to each vibrational mode. One often sees gas rather than mode based data for polyatomic molecules such as carbon dioxide and this has to be nonsense.

This example exemplifies the challenge associated with the development of the phenomenological models in DSMC. Equipartition is built into the relaxation terms in the Navier-Stokes equations. On the other hand, DSMC models and procedures deal with individual intermolecular collisions and must mimic the elements of the basic physics that promote equipartition. Note the use of the word "mimic" rather than "contain" because the solution of a problem in a phenomenological model may be through mathematics rather than physics. The requirement to have an entirely microscopic model is in order to at least open up the possibility of having a valid model for highly nonequilibrium situations. Validation through comparison with experiment is still required, but note that we are dealing with physics that is beyond the reach of any continuum model.

The time from the emergence of the vibrational equipartition problem to its solution through the simple expressions in lines 6 and 7 was more than ten years. The inability to solve such a basic problem has until recently inhibited further work on DSMC chemistry. While DSMC was able to deal with vibrational excitation at a superficial level during that time through the procedure that employs the overall temperature, any need to use a macroscopic property in a DSMC procedure constitutes a defeat.

The ability of DSMC to directly employ quantum models is a major advantage over continuum models that require discontinuous physical data to be approximated by smoothed functions. The fact that the quantum models are computationally incredibly simple (see Line 27 in the annotated listing) is an enormous bonus

It is noted in the annotation of the listing that dissociation generally occurs at levels of the order of 50. Dissociation is currently implemented in the DS2V/3V programs through the continuum rate equations, although kinetic theory is used to convert the temperature dependent rates to collision energy dependent rates. It should not be necessary to use a rate equation for dissociation because, in the quantum vibration procedure, it occurs naturally when the selected level is above the level that corresponds to dissociation. This is an excellent example of an event driven process that can be modeled directly in DSMC and the calculation that led to Figure 8.1 was repeated for dissociating nitrogen.

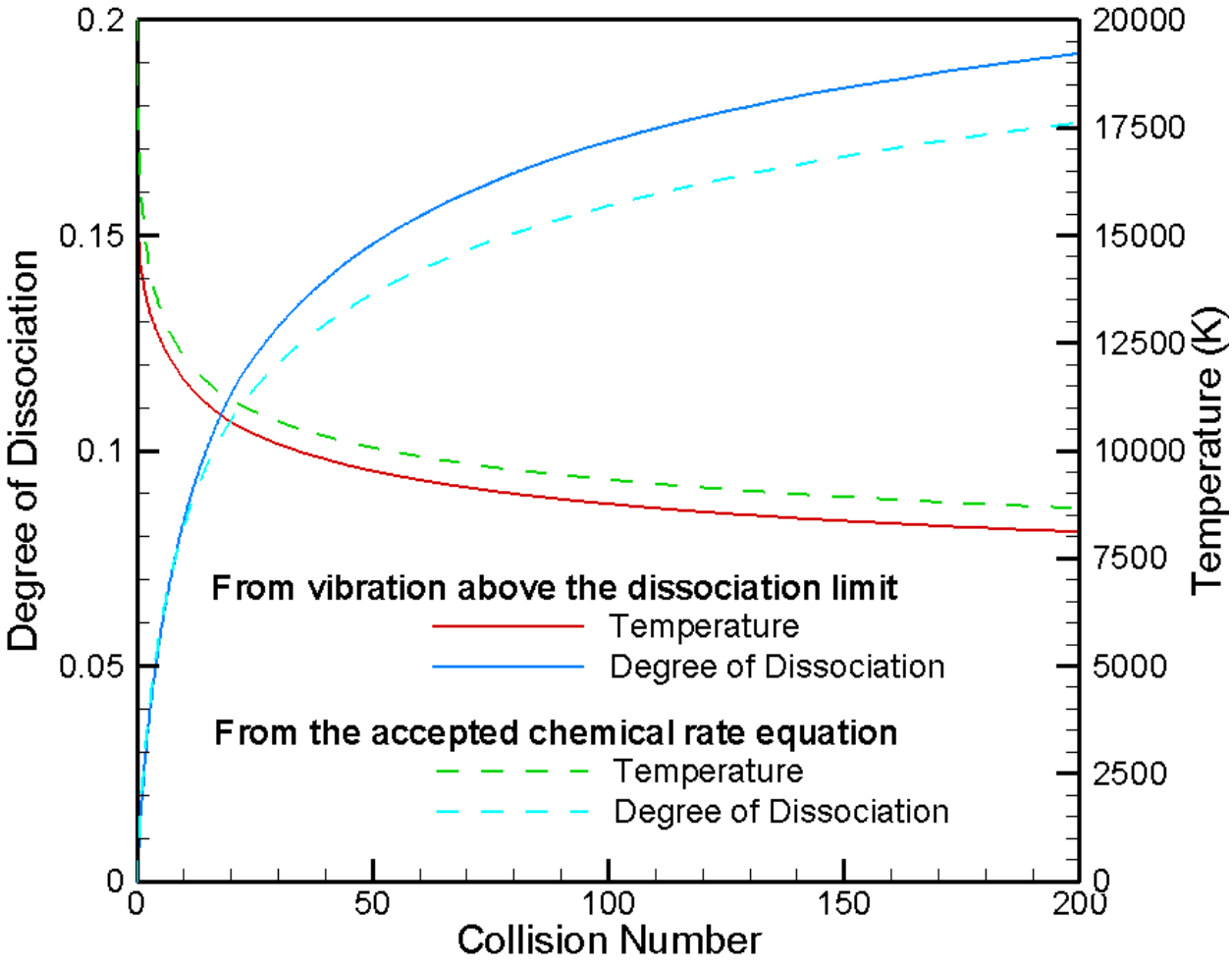


Figure 8.2 Combined vibrational excitation and dissociation in nitrogen at a number density of 10^{20} m^{-3} from an initial (T_{tr} and T_{rot}) temperature of 20,000K.

The dashed curves in Figure 8.2 were obtained from Version 4.4.02 of the DS2V program. The only modification was the addition of a special output file with the information that is plotted. The chemical reaction data for the dissociation was entered manually using the values from the built-in data for real air calculations. At the number density that was employed, the probability of any recombination reactions occurring over this time interval is negligible. Equilibrium would eventually be attained as a result of these recombination reactions that, because they require three-body collisions, are so rare at this density that the collision number to equilibrium would be many millions.

The calculation that led to the solid curves employed a modified program. Lines 8 and 10 in the above code that are based on Equation (8.1) were replaced by similar lines based on Equation (8.3). **The data did not include chemical reaction rates and the dissociation rate as well as the vibrational relaxation rate is based entirely on the reference value Z_v=19.8 at 20,000 K.** The COLLISION subroutine was followed by a new DISSOCIATION subroutine. This cycles through the molecules and, if the

vibrational level of an N_2 molecule is above the dissociation limit, it is changed to an N atom and another N atom is added. The vibrational energy above that corresponding to the dissociation energy is combined with the rotational energy and becomes the relative translational energy of the two atoms.

However, if the selection of the vibrational level was left in the form that appears in the code listing, the dissociation rate would be more than an order of magnitude too slow. This was appreciated when the DSMC0V.FOR demonstration program was prepared for Reference [3]. The solution at that time was to replace the uniform simple harmonic levels by the more realistic anharmonic levels of the Morse potential. These were continued beyond the dissociation limit so that the probability of levels above the dissociation limit was increased. This was complex and was not precise because the number of fictitious levels above the dissociation limit was an arbitrary choice. (Note that line 22 in the code segment represents an even choice of the available levels that precedes the Larsen-Borgnakke selection). Dr. Gordon Lord of Oxford University pointed out that there is a "continuum", i.e. an infinite number, of states above the dissociation limit and, if dissociation is energetically possible, it occurs. Therefore, if the variable MAXLEV in the code segment is above the dissociation limit, it is assigned to the molecule without further tests. This molecule is then picked up by the DISSOCIATION subroutine to be dissociated. Note that this is consistent with the setting of the vibrational collision number to unity at the characteristic temperature of dissociation, so that aspect of Equation (8.3) is more than just an unsupported hypothesis. The new procedure is incomparably simpler and contains no adjustable parameters.

The quantum vibrational procedure is incomplete to the point of being incorrect at high temperatures unless dissociation is integrated with the vibrational excitation.

Assuming that the coding is correct and is now based on sound physics, the comparison of the sets of curves in Figure 8.2 provides a check on the consistency of the experimentally based data on vibrational rates and the chemical rate data for dissociation reactions. Given the uncertainties associated with these data sets and the large variation of temperature in this calculation, the agreement is remarkable.

Version 4.5.01 of the DS2V program offers two options with regard to the chemistry model:

(i) The ***particle-based chemistry model*** in which the vibrational excitation rate depends on the "quantized collision temperature" based on the energies in each individual collision. Dissociation occurs naturally when the excitation is to the quantum level that corresponds to dissociation.

(ii) The ***continuum or temperature-based chemistry model*** in which the vibrational excitation rate depends on the local flow temperature. Dissociation is based on the conventional temperature dependent chemical rate equations. This model has been employed in almost all DSMC calculations and leads to agreement with the corresponding Navier-Stokes CFD results at low Knudsen number, but cannot be expected to provide realistic results for highly nonequilibrium flows.

The options have been compared in calculations for the Mach 25 flow of nitrogen past a cylinder at a Knudsen no. of 0.01. This test case in has been taken from Reference [4], although dissociation was not employed in those calculations.

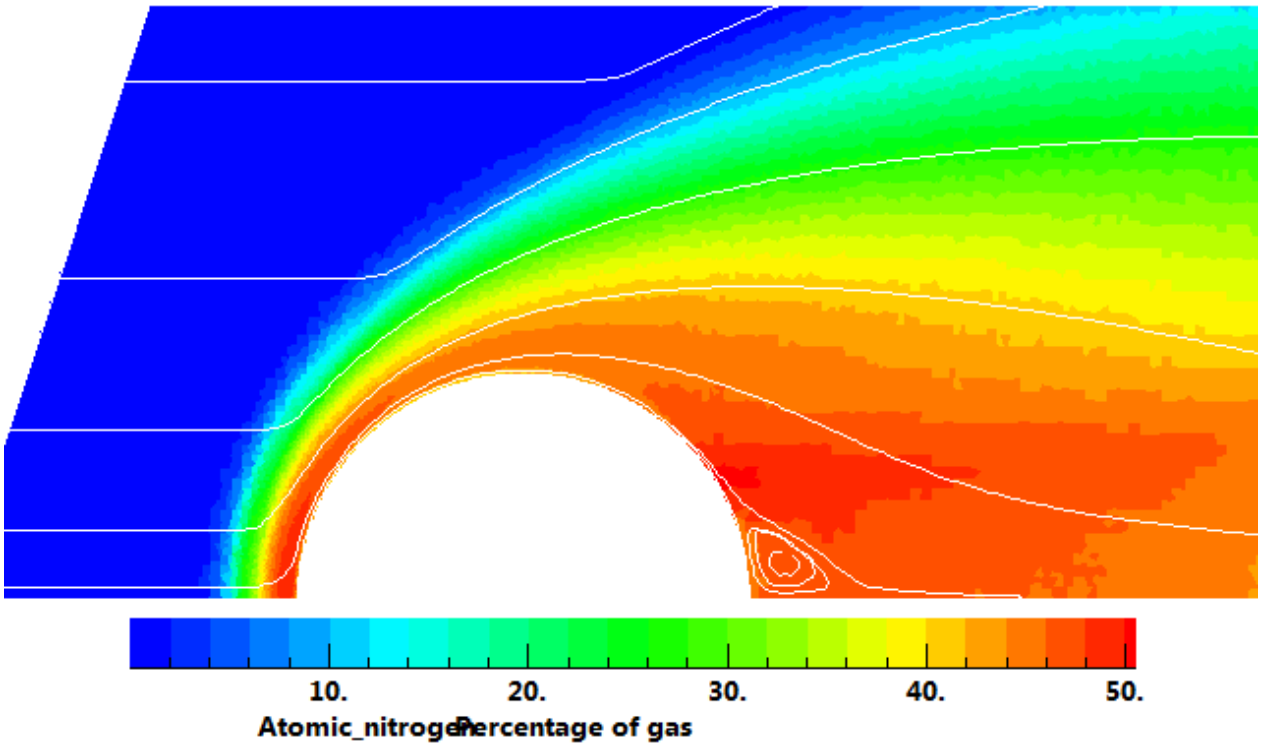


Figure 8.3 Atomic nitrogen contours from a calculation with "natural dissociation" based on a Z_v reference value of 10,000 at a reference temperature of 5000 K [4].

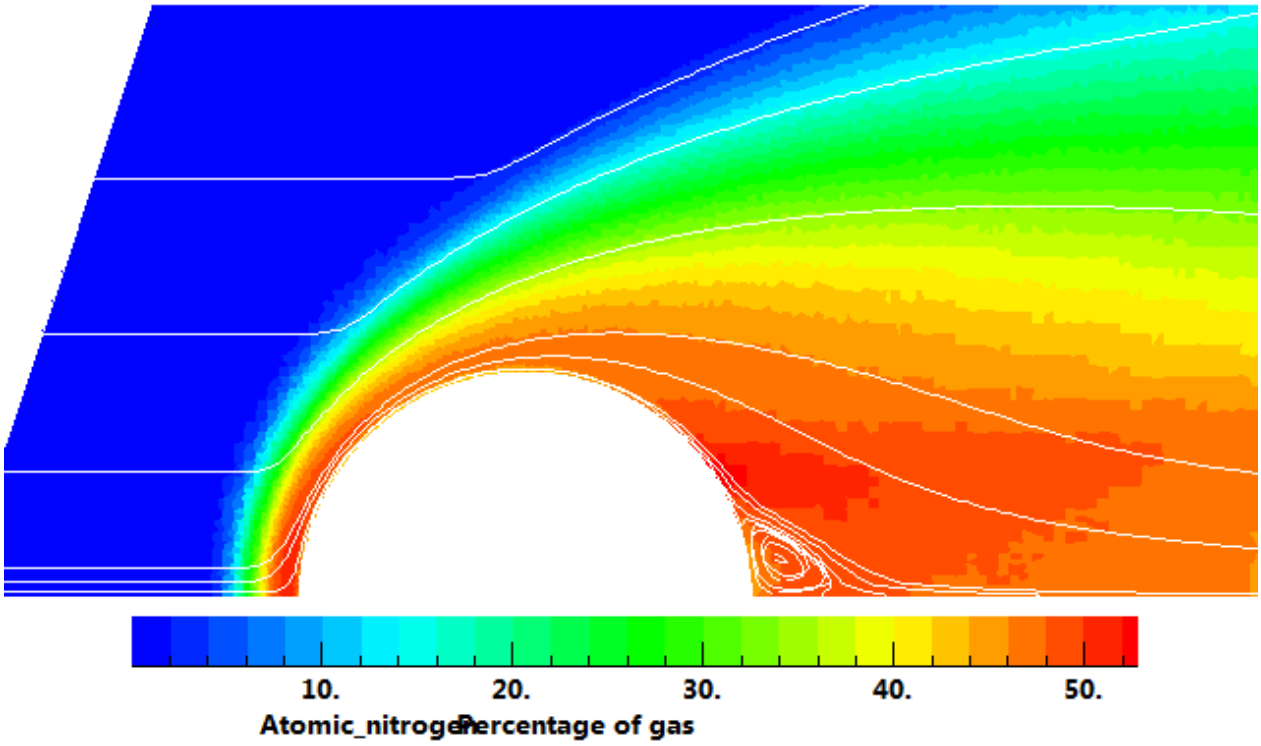


Figure 8.3 Atomic nitrogen contours based on a conventional DSMC calculation that employed the standard chemical rate equations.

The reverse reactions (recombinations in this case) can be incorporated in DSMC through the application of the law of mass action at each collision, preferably based on a "collision temperature", to determine the probability of the reverse reaction. The calculation of partition functions is computationally intensive but, as in DSMC0V.FOR, the computation may be made for just a small fraction of the collisions and the probability increased accordingly. Exchange reactions pose a more serious problem, but it was shown in Reference [3] that satisfactory rates for the forward (endothermic) reactions in air can be obtained through a Larsen-Bognakke process based on the activation energy. The law of mass action is again available for the reverse reactions. As long as fifteen years ago, Dr. Ann Carlson of NASA LaRC employed the older DSMC procedures to show that satisfactory results could be obtained for real air re-entry flows without there being any need for experimentally based data for the chemical reaction rates.

The rotational energy is currently assumed to be continuously distributed, but it is planned to move to a quantum rotation model. This will permit the precise modeling of the line radiation from bound-bound transitions, although it will be necessary to move beyond the simple harmonic model for vibration. The Morse potential model will probably prove to be adequate but, in particle based methods such as DSMC, there is no difficulty associated with moving to tabulated data.

Electronic excitation and ionization have to be treated as chemical reactions in continuum formulations, but it is planned to implement in DSMC an electronic state procedure similar to the above quantum vibrational model. This will be complicated by the irregular spacing of the electronic states and tabular data will almost certainly be required. A precise treatment of thermal radiation will again be a major objective. A further point is that, because DSMC has a physical time parameter, it can easily handle the absorption of radiation, induced emission and spontaneous emission. Continuum treatments of high temperature flows generally assume steady flow and the absorption of radiation poses particular difficulties because it can convert a set of parabolic equations to an elliptic system.

9. Molecule I/O files for flow dimension and time scale changes

Whatever the capabilities of DSMC, there is always a demand for calculations that are beyond these capabilities. This has led to special procedures that extend DSMC capabilities for special cases. A procedure that was useful for hypersonic blunt-body flows was to progressively reduce both the number of real molecules represented by each simulated molecule (FNUM) and the time step (DTM) from cell-to-cell as the surface was approached. As long as both are reduced by the same ratio, the molecular flux remains unchanged, and there is no need for molecule duplication or removal. While it permitted blunt-body calculations for "continuum-overlap" Knudsen numbers on slow computers, this "DTM/FNUM scaling" destroyed the inherent time-accuracy of DSMC and is harmful rather than beneficial in

regions of expanding flow. It is not used in the DS2V/3V programs, although these programs do employ the "molecule input/output" files that were first introduced in calculations of jet plume expansions.

A very large file is generated from the molecules leaving a flow across a boundary, either at the edge of or within the flow. As long as the normal component of the outflow velocity is well supersonic, the flow beyond the boundary has no influence on the flow that generates the molecule file. As long as the flow is steady, the molecules from this file can be used as inflow molecules for a calculation of the flow in an adjoining region. **Also, the placing of molecule file generation boundaries within the flow allows molecule I/O files to be used even when the normal velocity component at the boundary is subsonic.** The molecules may be input on a different time scale from that associated with their generation but, unlike DTM/FNUM scaling, there is no timescale variation within either calculation.

The other and probably more important use of molecule input/output files is to supply two-dimensional boundary conditions for three-dimensional calculations. A three-dimensional calculation is two or three orders of magnitude more demanding than a two dimensional calculation and it is essential that the portions of the overall flow that are two-dimensional or axially symmetric are not calculated as three-dimensional. This is illustrated by the example shown in Figure 9.1.

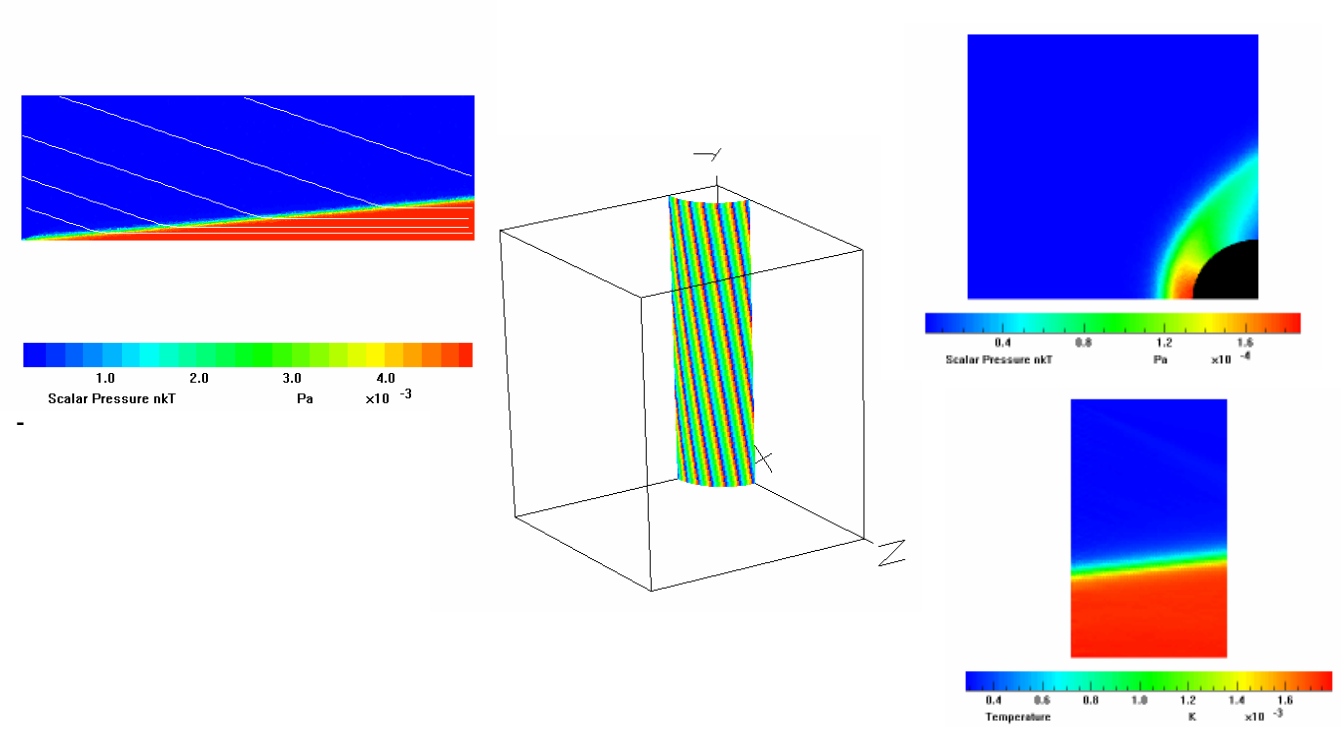


Figure 9.1. Linking DS2V and DS3V calculations by molecule I/O files. Four of the six faces of the three-dimensional region are interfaces with two-dimensional flows.

The three-dimensional flow is the interaction of an oblique shock wave with a circular cylinder. The flow is symmetrical about the plane z=0 and the calculation includes only the forward half of the cylinder. The shock wave is generated in a 2-D calculation with height (in the y direction) equal to the height of the 3-D flowfield. A file of molecules is generated at the downstream end of the 2-D calculation and molecules from this file are spread across the forward face (minimum x) of the 3-D flowfield. These molecules also serve as the entry molecules to another 2-D calculation with dimensions that match the maximum z face of the 3-D flowfield. A file of molecules is generated from that notionally exit from this flow in the negative z direction. These serve as entry molecules to the maximum z plane of the 3-D flowfield. Another 2-D calculation is made of the flow over the quadrant of the cylinder with dimensions that match a cut through the 3-D flow by a plane normal to the y direction. Separate molecule output files are generated from the molecules that notionally leave the flow in the positive and negative z directions. With the y direction transformed to the z direction, these files serve as molecule input files at the minimum and maximum y planes of the 3-D flowfield.

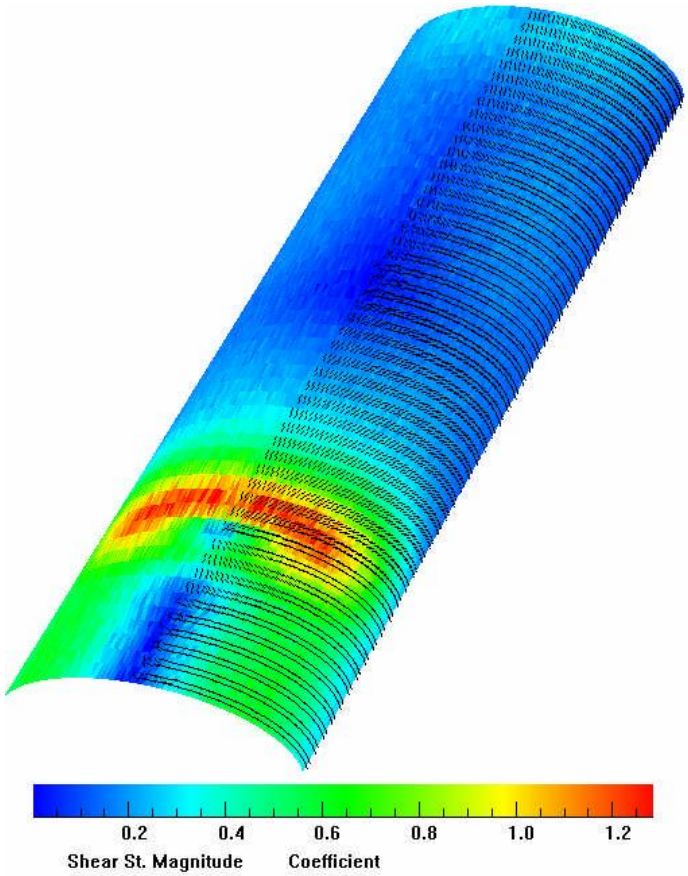


Figure 9.2. The surface shear stress distribution in the intersection of an oblique shock wave with a cylinder.

The 3-D flow was found to be reminiscent of the Type IV Edney flowfield in two dimensions. A typical result for the surface properties is shown in Figure 9.2.

10. Hybrid codes and parallelization strategies

DSMC calculations have often been combined with continuum Navier-Stokes calculations. The earliest examples involved applications to jet plumes and the DSMC calculations employed "specified flow input boundaries" with data based on continuum nozzle flow calculations. These specified flow boundaries are an essential feature of general DSMC codes and are widely used.

More recently, there have been many attempts to produce hybrid codes that include both particle and continuum components and which attempt to automatically switch between the modes. This has proved to be a difficult undertaking. This is partly because the requirements of the two types of calculation are very different with regard to meshes and data. Problems also arise as a consequence of the generally implicit but questionable assumption that the continuum model is to be preferred when both approaches are feasible. The Navier-Stokes model does not contain the physical information that would enable it to know when its assumptions become invalid. DSMC contains that information but, as long as there is an overlap of the Knudsen number ranges for practical calculations of the complete flow by the two approaches, it is hard to see any real need for hybrid codes. As noted in the Introduction, there is already a large overlap for two-dimensional flows and an overlap will soon exist also for three-dimensional flows.

DSMC can be applied to almost all one-dimensional flows but, to date, there have been very few applications that exploit its potential in this area. "One-dimensional" includes flows with cylindrical and spherical symmetry and also special extensions such as that to stagnation streamline flows. The latter extension would permit calculation of the stagnation point conditions on a blunt re-entry body beyond the maximum heating altitude. In addition, it is not generally appreciated that a two-dimensional flow can have velocity components the third direction that vary over the flowfield. Similarly, an axially symmetric flow can have circumferential velocity components that vary over the plane of axial and radial components. The only requirement is that there must be no gradients in the third direction.

There is an increasing requirement for DSMC codes to be "parallel". Many of the more demanding calculations in recent years have been made on clusters. These have distributed memory with limited bandwidth between the nodes and the existing parallel codes employ domain decomposition. However, all high performance CPU's are now dual core, quad cores are becoming common and the number of cores is set to increase. Multi-core CPU's have a shared-memory architecture with a comparatively high bandwidth between the cores. This favors parallelism through multithreaded code that splits loops between the cores. At the same time, clusters have become clusters of multi-core CPU's and domain decomposition will almost certainly continue to have a place. However, because the continuation of Moore's law now depends on the increasing number of cores in the CPU, priority has to be given to the development of multi-threaded code. Domain decomposition introduces a great deal of unproductive complication within programs and does not scale well as the number of processors increases. However, it is a

proven approach and is available as fall-back or a supplementary strategy if multithreading proves to be impractical.

The DS3V program has an option for "dual core" domain decomposition at planes of constant z but, as far as the author knows, Prof. Zuppardi of Naples University has been the sole user of this feature. This was written in Fortran 95 and employs non-standard file waiting loops to synchronize the calculations. The recently released Version 10 Intel Fortran contains the "WAIT" instructions from the Fortran 2002 specification that would enable domain decomposition to be implemented with standard code.

Problems caused by round-off errors in 32 bit arithmetic have been responsible for about half the effort associated with the development of the DS2V/3V codes. In order for the programs to cope with nano and microscale flows, internal scale factors are applied within the codes so that most calculations involve numbers of order unity. The newer CPU's are almost all 64 bit and there are now 64 bit versions of Windows as well as Linux, so that memory is now virtually unlimited as well as cheap. There now appears to be a strong case for all 64 bit code, although some procedures might employ shorter number formats for better vectorization.

The latest Intel Version 10 includes OpenMP commands that permit loops to be split between processor cores. These have been tested on small programs and computation times have been halved on a computer with a dual core processor. There are other new compilers, e.g. Absoft, with features that will hopefully lead to practical multithreaded DSMC programs.

11. Dynamic load libraries for custom applications

No matter how far the data for a general-purpose DSMC code is extended, there will always be demands for further extensions. Similarly, there is often a requirement for additional output quantities in non-standard formats. For example, the data that is plotted in Figures 8.1 and 8.2 would have been difficult to extract from the standard data files. Because the author had access to the source code, the required data file was easily added. Open source code would solve a simple problem like this, but additions to the standard data menus require careful changes to the linked GUI program as well as to the Fortran source code. This would involve excessive work for most users and, in any case, it is undesirable to continually modify a standard user interface to accommodate additional data that might be unique to a particular user's application.

Mr. Martin Rose of the Fraunhofer IKTS in Dresden has demonstrated a solution for this problem. The approach is described in the following quote from the Abstract of a draft paper being written by Mr. Rose:

"This paper presents an interface to the 2D DSMC program (DS2 (Ref)) written by Dr. G. A. Bird. It enables the user to read and to modify certain variables and data structures inside the simulation. With this tool the user can apply the DSMC method to his special problems and rely on a mature, working DSMC code instead of starting from scratch with an implementation of DSMC.
Some of the most interesting data structures are the surface properties like adsorption coefficients, temperature, reaction probabilities and the properties of the inlet streams. All those quantities can be altered at runtime with the proposed interface. For example the surface properties can be changed to

implement more complex surface reactions. It is possible to change the density, velocity, temperature, and composition of the entering streams to model the gas flow in complex systems.
Another feature is the direct access to sampled flow properties inside the flow field and at the surface. These properties include quantities like density, temperature, pressure, composition, number fluxes and velocity. Also the coordinates of the simulated particles can be read. This allows tracking of particles, a feature unique to discrete particle methods.
In general the author of a program needs to change his code whenever a user requests a new feature. This is time consuming and leads to a very specialized code. The proposed interface leaves this work to the user without revealing the source code of the actual DSMC program. The user does not need to worry about the details behind DSMC. In a windows environment the interface is based on a dynamic load library (DLL). An analogue mechanism is possible in a UNIX environment. The user needs to know a programming language that can create DLL's, e.g. C/C++, Visual Basic or FORTRAN. A Template for the creation of this library is provided in C++."

The description of the procedures was supplemented by several examples. Figure 11.1 shows the density in a typical vacuum system with cyclical input conditions.

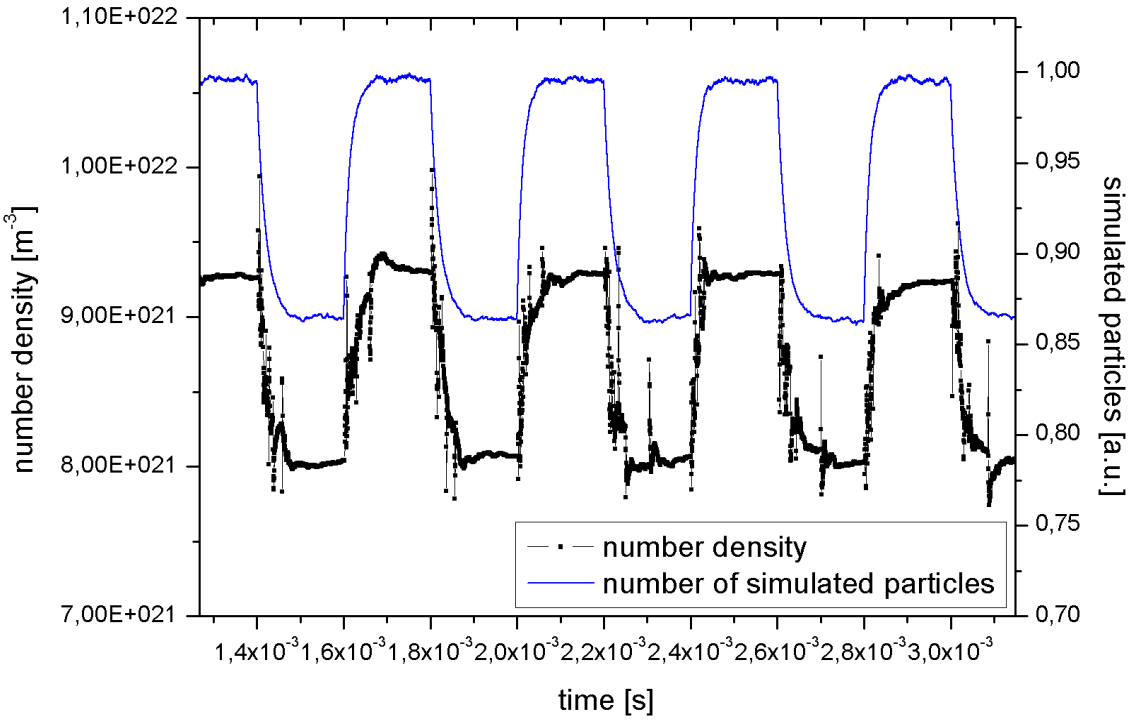


Figure 11.1 Results from a test calculation by Mr. Rose

Mr Rose was provided with a copy of the source in order to incorporate the entry points for the external .DLL's. He found the appropriate variables in the code and was able to add the additional statements without having to ask a single question about the details of the coding. He then implemented the external libraries in a way that would make them accessible to other users of the programs.

This is an example of what could be achieved if the DS2V/3V programs became part of a formal OpenDSMC initiative.

12. Self-validating and user-friendly codes

A good calculation requires a good user as well as a good program!

- Chairman's closing remark at a continuum CFD conference (Melbourne, 2001).

If the above announcement had been made at the beginning rather than the end of the conference, one could have asked why it was that, after so many years of effort by so many people, the Navier-Stokes CFD community had been unable to produce programs that allow non-specialist users to readily produce reliable results for their applications. On the other hand, it had become clear at that time that the level of expertise that was required of users of DSMC codes could readily be incorporated into the internal logic of the program. Moreover, the tests that were applied to determine whether a good calculation had been made could also be internal checks so that the programs could be made to be self-validating.

The DS2V and DS3V codes that have been referred to throughout these notes represent an attempt to produce "intelligent" programs that make DSMC a useful and reliable tool, even in the hands of casual users. The user does not have to be concerned with the spatial grid, but is responsible for setting the overall size of the flowfield. It is hard to prevent bad choices in this area for "aerodynamic" applications, although the "vacuum system" class of application is less prone to error because the boundary choices are largely constrained by the apparatus.

Assuming that a proper choice is made for the overall flow geometry and boundary conditions, a good DSMC calculation is made if the effective cell size is everywhere no larger than about one third of local mean free path, and if the time step is similarly small in comparison with the local mean collision time. As described in Section 6, the time step can be left entirely to the code. It is not easy to define the "cell size" in a modern code, but the mean separation distance of the pairs of molecules that are selected as collision partners can be used in its place. This local value of this distance can be calculated and compared with the local mean free path. The program can then use this "mcs/mfp ratio" to inform the user whether or not a valid calculation is being made. The separation distance is proportional to the total number of simulated molecules to the power $1/n$, where n is the number of spatial dimensions of the flowfield. The user controls the total number of molecules through the specification of the initial number of megabytes to be used in the calculation. This is the only computational variable that must be set by the user. Because the user as well as the program can monitor the value of the mcs/mfp ratio, it is easy to assess whether the available computer resources are sufficient for a good calculation to be made.

The program could be made to terminate the calculation if the mcs/mfp ratio is well over unity. However, the only consequence of excessively large ratios is that incorrect values are obtained for the transport properties. Just as the Euler equations are sometimes useful even though they completely neglect the transport properties, DSMC calculations at Knudsen numbers far higher than the desired number can be very useful.

The DS2V/3V programs are written in FORTRAN and early versions employed the WINTERACTER add-on that provides access to the Windows API (application programming interface) through FORTRAN subroutines. Version 4 of DS2V employs an all FORTRAN core DSMC program DS2 (.f90). This is linked by binary stream files to the REALbasic program DS2V (.rbp) for the interactive GUI and data input. This is more flexible because the programs do not need to be run on the same computer and both can be compiled for Windows, Linux or Apple computers. The DS2V program can also be run in a post-processing mode. Version 3 of DS3V will be similar, but this has to be preceded by a Version 5 of DS2V that will have an improved geometry model, 64 bit arithmetic for the geometry and multithreading for efficient operation on multi-core CPU's. The introductory screen of the DS2V program is shown in Figure 12.1 and should be self-explanatory.

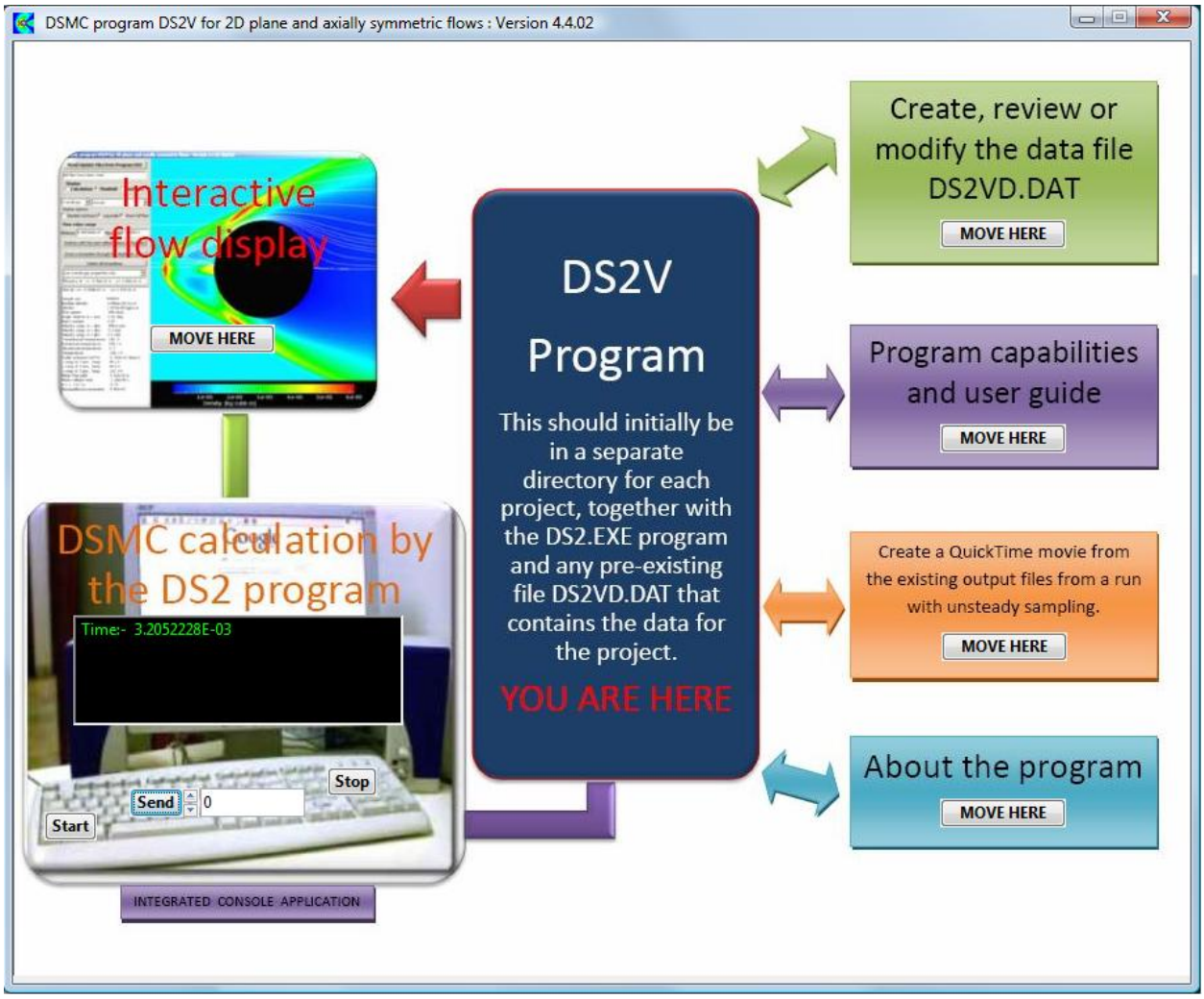


Figure 12.1 The introductory screen of Version 4 of the DS2V program.

The data items that are required for the calculation of the benchmark flow are listed in the Appendix. The user is not normally concerned with this list which appears in just one small field of the data input menus that are shown after the

listing. These menus can go beyond a user-friendly means for the generation, modification or review of the data. This is best illustrated through the menus for the chemical reaction rates that were employed in Section 8.

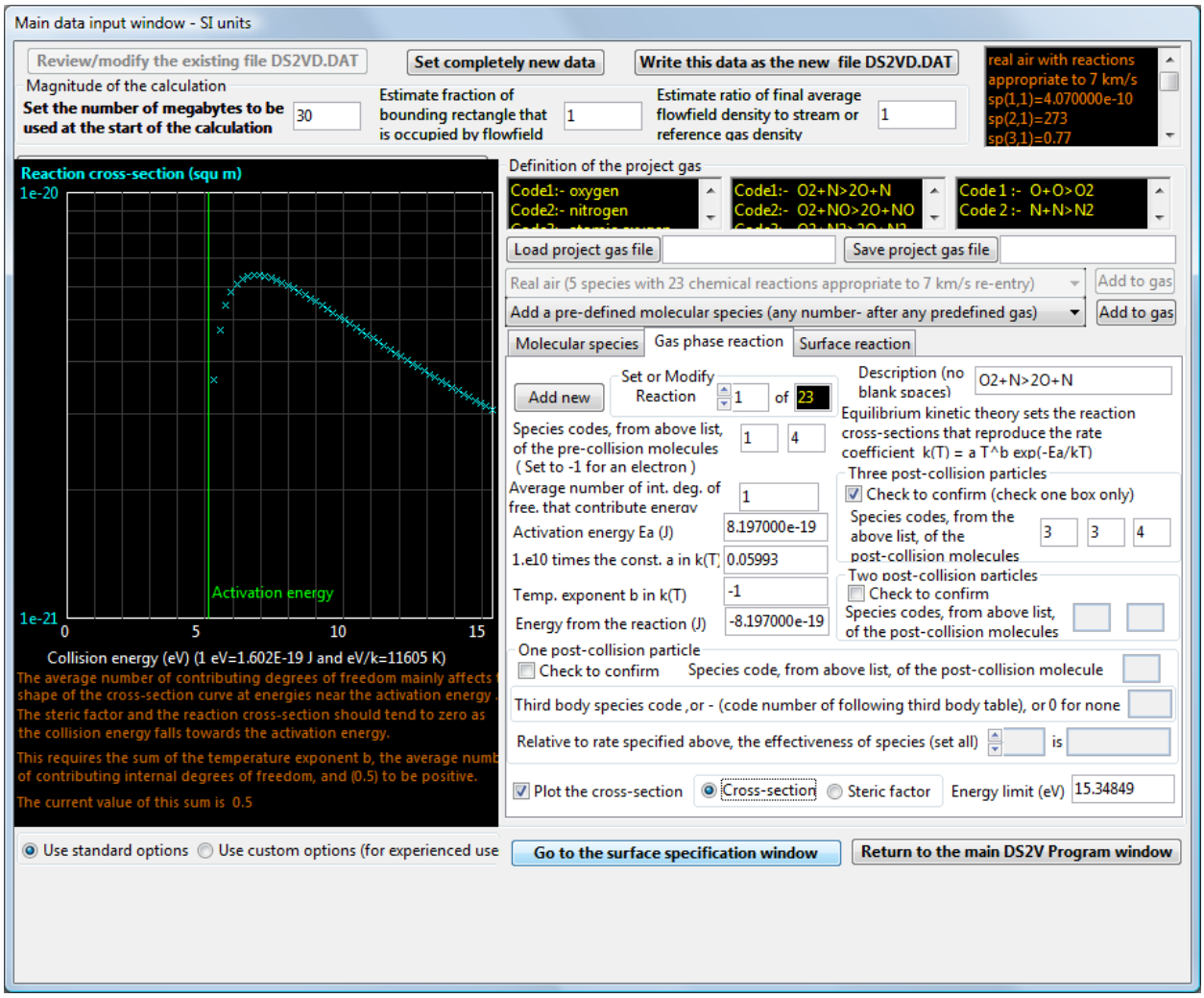


Figure 2.2 Interactive adjustment of chemical rate data within the menus.

The tab for "Gas phase reaction" has been chosen in the first data generation window and the check box at the lower left of this field has been ticked. This brings up a small overlaid window that plots the either the steric factor or the collision cross-section as a function of the collision energy for the selected reaction. The average number of internal degrees of freedom that contribute to the collision energy is an adjustable parameter and the rate equation is independent of its value. However, its value (which must be chosen to be physically valid for the molecules that are involved) affects the shape of the cross-section versus collision energy curve. Data may be available for spot values of the reactive cross-section and the value of the adjustable parameter should be chosen to match this data. Alternatively, extensive data may be available for the cross-sections of a particular reaction and the links between the data field and the overlaid window then allows a rate

equation to be fitted interactively to the cross-section data. The program then makes full use of the experimental data and the form of the rate equation just serves as a convenient curve-fitting mechanism.

There are many other windows within the DS2V program and an attempt has been made to produce a user interface of similar quality to those in familiar consumer programs. For example, there is a separate data window that permits the generation of QuickTime movies of unsteady flows. The movies are made in a post-processing mode and there are options for the choice of the flow variable, to zoom into part of the flowfield, set the frame rate etc. Most figures in these notes have been cut and pasted from the program GUI. At the same time, the objective is always to keep the basic procedures at the leading edge of DSMC development.

The production of the programs has involved almost ten years of work and has built on the experience associated with earlier generations of programs. The DS2V/3V programs have involved the writing of more than 100,000 lines of code. The testing of the programs has been dependent on the feedback from the hundreds of users of the programs. Special thanks are due to Dr. Jim Moss of NASA Langley who, when he raises an issue related to the results, almost invariably proves to be correct. Acknowledgement must also be made of the sustained support from NASA Langley. Without this, the programs would not have been developed. The programs were marketed for a time and the necessity to provide the expected level of support to purchasers rather than just users contributed to the development of what are intended to be "industrial strength" products. However, given the difference in the size of the markets, selling DSMC programs at a similar price to that of Microsoft Office was a commercial proposition only because the money was not needed!

The programs are anything but complete and the "to do" list grows every day. There is probably two years of work required just for the development of the planned Version 5 of DS2V and a Version 3 of DS3V to the same specifications. In addition, ideas for new and improved procedures come up continually and have to be investigated – several as a result of the work on these notes.

It is becoming harder for the author to drop everything whenever an e-mail arrives with a title starting "Problem with……". Posting the source code on the web site so that one could simply reply "OK, so fix it!" is not a viable option. There has to be some mechanism to protect the integrity of the codes, to make sure that problems are fixed for all users and, above all, to ensure that development continues.

References

[1] A.J. Lofthouse, I.D. Boyd, and M.J. Wright, AIAA Paper 2006-993 (2006)

[2] G.A. Bird, *Rarefied Gas Dynamics, 25th International Symposium*, ed. M.S. Ivanov and A.K. Rebrov, Publishing House of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia, pp 349-354 (2007)

[3] G.A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford University Press (1994)

[4] A.J. Lofthouse, L.C. Scalabrin, and I.D. Boyd, AIAA Paper 2007-3903 (2007)

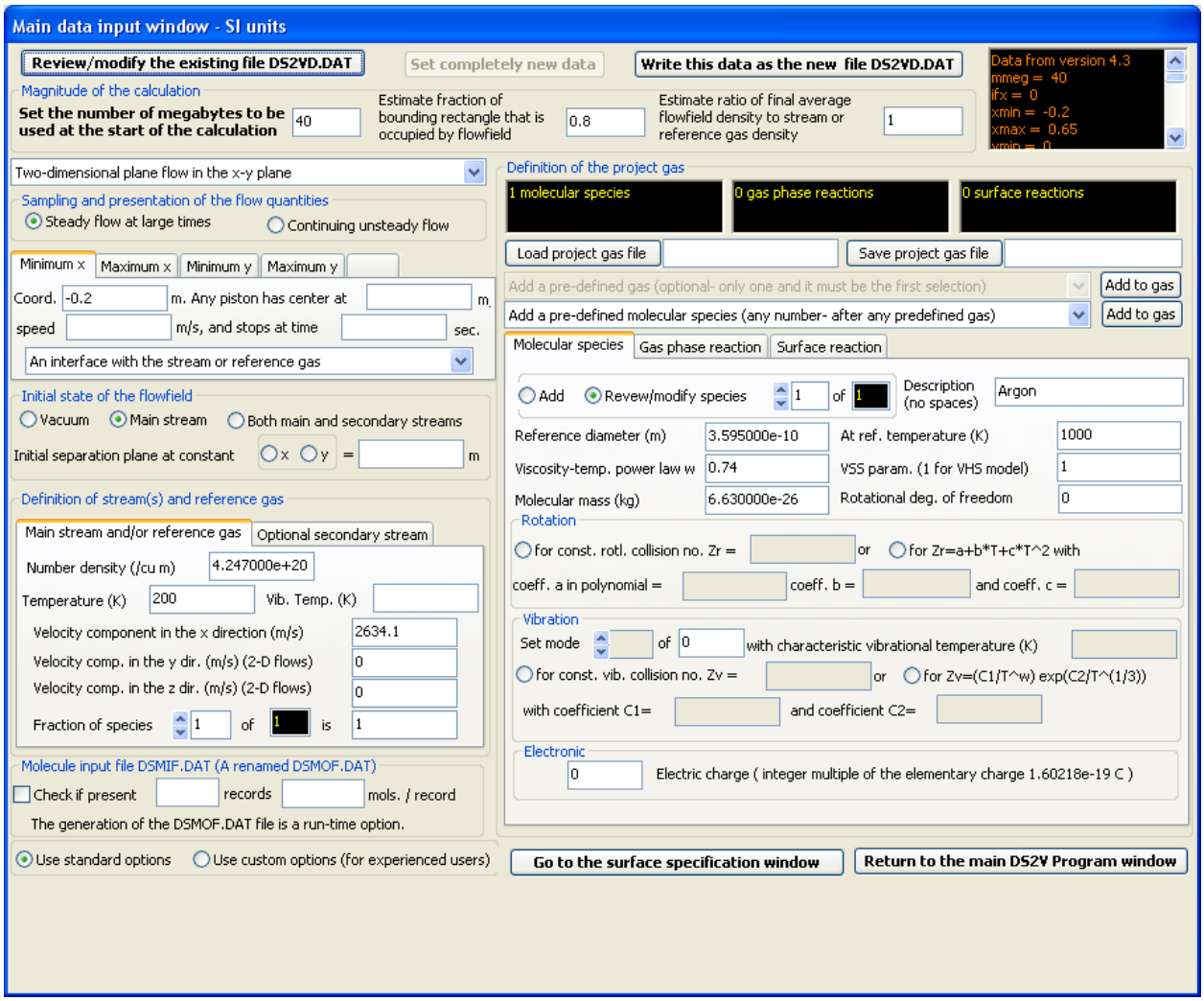
Appendix. Data for the benchmark case

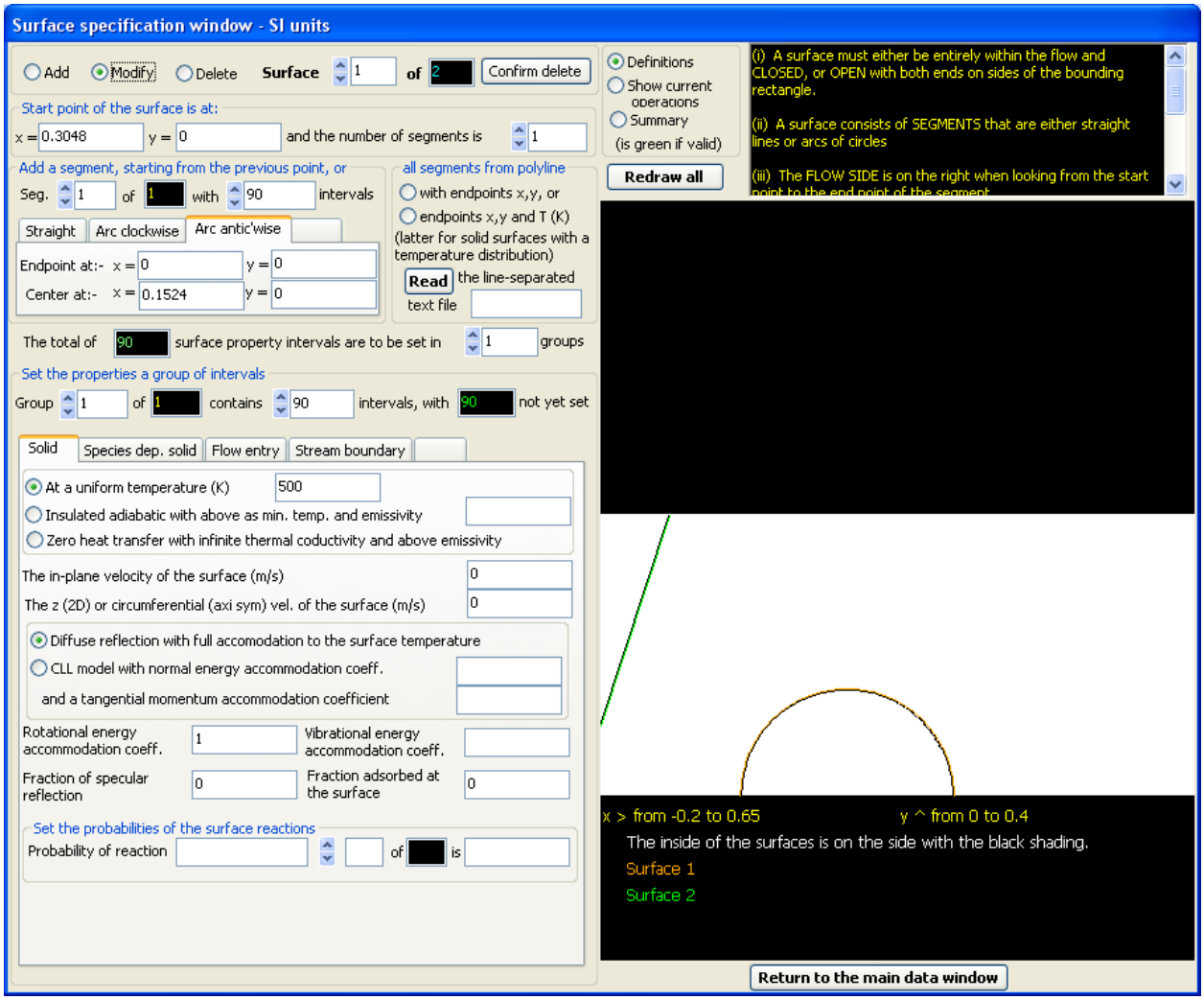
An annotated listing of the DS2VD.DAT that has been written from the DS2V data preparation windows is as follows:

4	This DS2VD.DAT data file was generated by version 4.
3	3 of the DS2V program.
40	Initial number of megabytes to be used in the calculation.
0	It is a two-dimensional flow.
-0.2	Minimum x coordinate of the bounding rectangle.
0.65	Maximum x coordinate of the bounding rectangle.
0	Minimum y coordinate of the bounding rectangle.
0.4	Maximum y coordinate of the bounding rectangle.
0.8	Approximate fraction of the bounding rectangle occupied by flow.
1	Estimated ratio of average final to initial density.
1	The number of molecular species.
0	Maximum number of vibrational modes of any species.
0	The number of gas phase chemical reactions.
0	The number of surface reactions.
3.595000e-10	The reference diameter of species 1.
1000	The reference temperature of species 1.
0.74	The viscosity-temperature index of species 1.
1	The VHS molecular model is employed for species 1.
6.630000e-26	The molecular mass of species 1.
Argon	The text description of species 1 in the output.
0	Species 1 is electrically neutral.
0	Species 1 has no rotational degrees of freedom.
4.247000e+20	Number density of the stream (or reference gas).
200	Stream temperature.
2634.1	Stream velocity component in the x direction.
0	Stream velocity component in the y direction.
0	Stream velocity component in the z direction.
1	The fraction of species 1 in the stream.
2	Number of unconnected surfaces.
91	Number of defining points along surface 1.
2	Number of defining points along surface 2.
91	Maximum number of points on any surface (sampling intervals+1)*.
1	Maximum number of surface type groups of intervals on any surface*.
90	Maximum number of sampling intervals on any surface*.
1	Maximum number of molecule entry intervals on any surface*.
1	The number of surface definition segments on surface 1 is 1.
-2	Segment 1 is a circular arc in the anticlockwise direction.
0.1524	The x coordinate of the center of the arc.
0	The y coordinate of the center of the arc.
0.3048	The x coordinate of the initial point on the arc.
0	The y coordinate of the initial point on the arc.
0	The x coordinate of the final point on the arc.
0	The y coordinate of the final point on the arc.
90	Number of sampling intervals along the arc.
1	The number of sampling interval groups along the arc.
90	The number of sampling intervals in group 1.
1	Group 1 is a solid surface with species independent properties.
500	The temperature of the group of intervals.
0	The speed of the group of solid surface elements in the x-y plane.
0	The speed of the group of solid surface elements in the z direction.
0	Diffuse reflection
1	The (not applicable) rotational accommodation factor is 1.
0	The fraction of specular reflection is 0.
0	The probability of adsorption is 0.
1	The number of surface definition segments on surface 2 is 1.

1	Segment 1 on surface 2 is a straight line.
-0.2	The x coordinate of the initial point of the segment.
0.1	The y coordinate of the initial point of the segment.
-0.1	The x coordinate of the final point of the segment.
0.4	The y coordinate of the final point of the segment.
1	There is one sampling interval along the line.
1	This interval must be in a single group.
1	This group must contain a single interval.
4	The group is an interface with the stream.
3	The boundary at the minimum x coordinate is an interface with the stream.
3	The boundary at the maximum x coordinate is an interface with the stream.
2	The boundary at the minimum y is a plane of symmetry.
3	The boundary at the maximum y coordinate is an interface with the stream.
1	The initial state of the flow is a uniform stream.
0	There are no file entry molecules.
0	There is no secondary stream
1	The flow sampling is for an eventual steady flow.
0	The calculation employs the default computational variables.

The DS2V user interface for the production of this data file is as follows:




Surface specification window - SI units
Add
Modify
Delete
Surface
1
of
2
Confirm delete
Start point of the surface is at:
x = 0.3048
y = 0
and the number of segments is
1
Add a segment, starting from the previous point, or
Seg.
1
of
1
with
90
intervals
Straight
Arc clockwise
Arc antic'wise
Endpoint at:- x = 0
y = 0
Center at:- x = 0.1524
y = 0
all segments from polyline
with endpoints x,y, or
endpoints x,y and T (K)
(latter for solid surfaces with a temperature distribution)
Read
the line-separated text file
The total of
90
surface property intervals are to be set in
1
groups
Set the properties a group of intervals
Group
1
of
1
contains
90
intervals, with
90
not yet set
Solid
Species dep. solid
Flow entry
Stream boundary
At a uniform temperature (K)
500
Insulated adiabatic with above as min. temp. and emissivity
Zero heat transfer with infinite thermal coductivity and above emissivity
The in-plane velocity of the surface (m/s)
0
The z (2D) or circumferential (axi sym) vel. of the surface (m/s)
0
Diffuse reflection with full accomodation to the surface temperature
CLL model with normal energy accommodation coeff.
and a tangential momentum accommodation coefficient
Rotational energy accommodation coeff.
1
Vibrational energy accommodation coeff.
Fraction of specular reflection
0
Fraction adsorbed at the surface
0
Set the probabilities of the surface reactions
Probability of reaction
of
is
Definitions
Show current operations
Summary
(is green if valid)
Redraw all
(i) A surface must either be entirely within the flow and CLOSED, or OPEN with both ends on sides of the bounding rectangle.
(ii) A surface consists of SEGMENTS that are either straight lines or arcs of circles
(iii) The FLOW SIDE is on the right when looking from the start point to the end point of the segment
x > from -0.2 to 0.65
y ^ from 0 to 0.4
The inside of the surfaces is on the side with the black shading.
Surface 1
Surface 2
Return to the main data window