

## Two Dimensional Isotropic Mesh Adaptation for viscous flow of a kinetic theory gas using TDEFM

M. R. Smith and M. N. Macrossan

Centre for Hypersonics  
 The University of Queensland, St. Lucia Queensland, 4076 AUSTRALIA

### Abstract

Adaptive Mesh Refinement, or AMR, has been used as a tool in CFD to better resolve and capture compressible flows, both for continuum solvers [5, 4, 8] and DSMC (Direct Simulation Monte Carlo) solvers [6, 7, 21]. Presented is the True Direction Equilibrium Flux Method, or TDEFM [1, 2, 3] applied with an adaptive meshing technique and diffusely reflecting boundary conditions. Continuum methods generally transfer fluxes between directly adjacent cells only; TDEFM is capable of transferring fluxes of mass, momentum and energy from any source cell to any given destination cell. TDEFM has been previously shown to provide superior results when compared to existing continuum solvers [1, 2, 3] for unaligned flows on cartesian grids. In the present method, cells are isotropically divided in order to more accurately resolve the flow field. Various mesh adaption parameters are employed, taken from both continuum and direct solver applications on adaptive meshes. Results have shown that use of an adaptive mesh with an adaptation parameter based on the local mean free path  $\lambda$  has reduced computational requirements considerably while maintaining resolution of important features of the flow.

### Introduction

Recent advancements of computing technology and increased demands of industry and the scientific community have led to the increased use of simulations using very large numbers of cells to ensure high fidelity. Examples of such simulations are simulations performed on the Boeing 787 aircraft by Boeing and recent simulations by NASA describing particle impact models [20] as shown in Figure 1. The application of continuum solvers on adaptive, cartesian meshes has been shown to be a useful tool for computing flows on complicated geometries [5, 6].

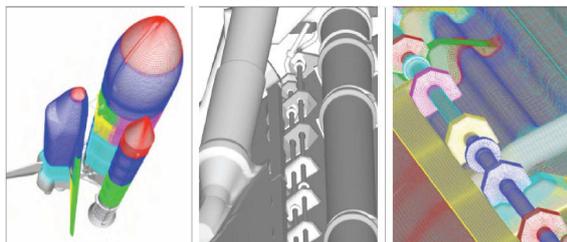


Figure 1: Examples of a high fidelity simulation using a large number of cells [20].

Conventional adaptive mesh methods work either by increasing mesh resolution around specific features of the flow, such as shock waves or solid surfaces, or by increasing mesh resolution in order to reduce the error in the result [4, 8, 5]. By doing so, less cells can be used to obtain the same accurate results obtained by using a large number of cells. However, existing continuum methods applied to adaptive cartesian meshes often encounter problems with ‘hanging nodes’, when a single cell

interface is shared by multiple adjacent cells [6]. Also, the evaluation of fluxes at such split interfaces in finite volume solvers often requires a complicated, higher order scheme to maintain stability [4].

Adaptive meshing strategies have been successfully employed in many direct solvers, such as DSMC [6, 7, 21]. In DSMC, the flow is separated into a ‘free flight phase’, where particles move without any interactions with other gas particles, and a ‘collision phase’, where particles undergo a pre-calculated number of collisions. For this assumption to be valid, cell sizes are typically 2 or 3 times smaller than the local mean free path. Therefore, in many such adaptive mesh simulations, the cells are refined according to local mean free paths or Knudsen numbers [6, 7].

Presented is the True Direction Equilibrium Flux Method, or TDEFM, applied to an adaptive meshing scheme to simulate a high speed lid driven cavity problem. TDEFM is based on the Kinetic theory of gases and represents the analytical solution to the free flight phase of a direct simulation assuming thermal equilibrium and uniform distribution of mass throughout each cell. A new diffuse reflection model is presented based upon the integral of the velocity probability distribution function for diffusely reflected particles. Three adaptive meshing parameters are employed - variance of density in the source cell and its surrounding neighbours, local density gradients and a parameter based upon the local mean free path. Results show that accurate determination of the location of the main circulation present in the driven cavity problem is found when cell sizes are based upon the local mean free path length. Addition of cells near flow features or walls causes an artificially high viscosity to exist and shifts the location of the circulation region significantly. TDEFM is considerably faster than a direct solver and creates no statistical scatter.

### TDEFM

The True Direction Equilibrium Flux Method, or TDEFM, is based upon the integration of the Maxwell-Boltzmann equilibrium velocity distribution over velocity space and physical space. The fluxes represent the analytical result to the ‘free flight’ phase of a direct solver (such as DSMC). If the simulation particles were uniformly distributed throughout the source cell and the gas was in thermal equilibrium, the fluxes represent what would be achieved if an infinite number of simulation particles were used. In the same way as a direct solver, TDEFM allows fluxes to be transferred from the source region to any specified destination region. The flux expressions are

$$\begin{aligned}
 f_M &= \mathbf{f}_M(m, s, \Delta t, x_R, x_L, x_I, x_T) & (1) \\
 &= M_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_I)^2}{2s^2\Delta t^2} \right) + M_1 \text{Erf} \left( \frac{m\Delta t + x_R - x_I}{\sqrt{2}st} \right) \\
 &- M_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_T)^2}{2s^2\Delta t^2} \right) - M_2 \text{Erf} \left( \frac{m\Delta t + x_R - x_T}{\sqrt{2}st} \right)
 \end{aligned}$$

$$\begin{aligned}
& - M_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_l)^2}{2s^2\Delta t^2} \right) - M_3 \text{Erf} \left( \frac{m\Delta t + x_L - x_l}{\sqrt{2}st} \right) \\
& + M_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_r)^2}{2s^2\Delta t^2} \right) + M_4 \text{Erf} \left( \frac{m\Delta t + x_L - x_r}{\sqrt{2}st} \right) \\
f_P &= \mathbf{f}_P(m, s, \Delta t, x_R, x_L, x_l, x_r) \\
&= P_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_l)^2}{2s^2\Delta t^2} \right) + P_1 \text{Erf} \left( \frac{m\Delta t + x_R - x_l}{\sqrt{2}st} \right) \\
& - P_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_r)^2}{2s^2\Delta t^2} \right) - P_2 \text{Erf} \left( \frac{m\Delta t + x_R - x_r}{\sqrt{2}st} \right) \\
& - P_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_l)^2}{2s^2\Delta t^2} \right) - P_3 \text{Erf} \left( \frac{m\Delta t + x_L - x_l}{\sqrt{2}st} \right) \\
& + P_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_r)^2}{2s^2\Delta t^2} \right) + P_4 \text{Erf} \left( \frac{m\Delta t + x_L - x_r}{\sqrt{2}st} \right) \\
f_E &= \mathbf{f}_E(m, s, \Delta t, x_R, x_L, x_l, x_r) \\
&= E_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_l)^2}{2s^2\Delta t^2} \right) + E_1 \text{Erf} \left( \frac{m\Delta t + x_R - x_l}{\sqrt{2}st} \right) \\
& - E_c \text{Exp} \left( \frac{-(m\Delta t + x_R - x_r)^2}{2s^2\Delta t^2} \right) - E_2 \text{Erf} \left( \frac{m\Delta t + x_R - x_r}{\sqrt{2}st} \right) \\
& - E_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_l)^2}{2s^2\Delta t^2} \right) - E_3 \text{Erf} \left( \frac{m\Delta t + x_L - x_l}{\sqrt{2}st} \right) \\
& + E_c \text{Exp} \left( \frac{-(m\Delta t + x_L - x_r)^2}{2s^2\Delta t^2} \right) + E_4 \text{Erf} \left( \frac{m\Delta t + x_L - x_r}{\sqrt{2}st} \right)
\end{aligned} \tag{2}$$

where  $x_L - x_R$  is the region occupied by the source cell,  $x_l - x_r$  the region occupied by the destination cell,  $m$  is the bulk flow velocity in the  $x$  direction,  $s = \sqrt{RT}$  and  $M_c, P_c, E_c, M_{1-4}, P_{1-4}$  and  $E_{1-4}$  are constants found in [1, 3]. In a two dimensional flow simulation, the net flux of mass, momentum and energy to any 2D rectangular region is given by

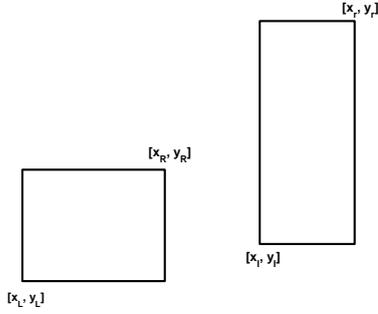


Figure 2: Computational domain for flows in two dimensions. The source cell (left) uses capitalised subscripts. The destination cell (right) uses lower case subscripts.

$$M = M_0 \mathbf{f}_M(U, \sqrt{RT}, \Delta t, x_R, x_L, x_l, x_r) \times \mathbf{f}_M(V, \sqrt{RT}, \Delta t, y_R, y_L, y_l, y_r) \tag{4}$$

$$P_x = M_0 \mathbf{f}_P(U, \sqrt{RT}, \Delta t, x_R, x_L, x_l, x_r) \times \mathbf{f}_M(V, \sqrt{RT}, \Delta t, y_R, y_L, y_l, y_r) \tag{5}$$

$$P_y = M_0 \mathbf{f}_M(U, \sqrt{RT}, \Delta t, x_R, x_L, x_l, x_r) \times \mathbf{f}_P(V, \sqrt{RT}, \Delta t, y_R, y_L, y_l, y_r) \tag{6}$$

$$E_x = M_0 \mathbf{f}_E(U, \sqrt{RT}, \Delta t, x_R, x_L, x_l, x_r) \times \mathbf{f}_M(V, \sqrt{RT}, \Delta t, y_R, y_L, y_l, y_r)$$

$$E_y = M_0 \mathbf{f}_M(U, \sqrt{RT}, \Delta t, x_R, x_L, x_l, x_r) \times$$

$$\begin{aligned}
& \mathbf{f}_E(V, \sqrt{RT}, \Delta t, y_R, y_L, y_l, y_r) \\
E &= E_x + E_y \tag{7}
\end{aligned}$$

where  $([x_L, y_L], [x_R, y_R])$  give the size and location of the rectangular source region,  $([x_l, y_l], [x_r, y_r])$  describe the size and location of the destination region,  $U$  is the  $X$  velocity,  $V$  is the  $Y$  velocity,  $M$  is the net mass flux,  $P_x$  and  $P_y$  are the  $X$  and  $Y$  momentum fluxes and  $E$  is the energy flux.

### Diffuse Reflection Model

Previous implementations of TDEFM have focused on specular reflections of particles of boundaries [1, 2, 3]. Energy and momentum are conserved and reflected back into the source (or correct destination) cell. This boundary condition is of limited usefulness. In practise, most particle reflections of engineering surfaces are considered diffuse. The velocity probability distribution function for the component of reflected velocity normal to the wall surface is:

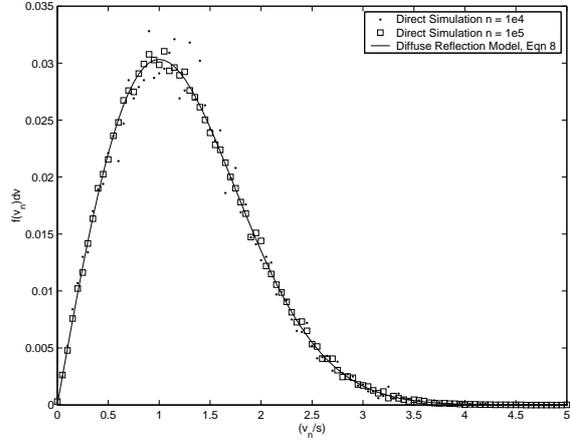


Figure 3: Numerical verification using direct simulation of Equation 8.

$$f_n(v_n) = v_n s^{-2} \text{Exp} \left( \frac{-v_n^2}{2s^2} \right) \tag{8}$$

such that the probability of a reflected particle having a normal velocity between  $v_n$  and  $v_n + dv_n$  is  $f_n(v_n)dv_n$ . This has been confirmed using direct simulations using increasing numbers of simulation particles as shown in Figure 3. The probability distribution functions for parallel components of velocity is assumed to be the Maxwell-Boltzmann equilibrium distribution function. Figure 4 shows the computational domain surrounding the diffusely reflecting surface between  $x_L - x_R$ . The fluxes of mass, momentum and energy (per unit mass) of diffusely reflected particles from region  $x_L - x_R$  to fall in region  $x_l, y_l - x_r, y_r$  will be:

$$\begin{aligned}
f_M &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(y_l-x)}{\Delta}}^{\frac{(y_r-x)}{\Delta}} \int_{\frac{(y_l-y_{Wall})}{\Delta}}^{\frac{(y_r-y_{Wall})}{\Delta}} f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \\
f_{P_x} &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(y_l-x)}{\Delta}}^{\frac{(y_r-x)}{\Delta}} \int_{\frac{(y_l-y_{Wall})}{\Delta}}^{\frac{(y_r-y_{Wall})}{\Delta}} v_n f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \\
f_{E_n} &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(y_l-x)}{\Delta}}^{\frac{(y_r-x)}{\Delta}} \int_{\frac{(y_l-y_{Wall})}{\Delta}}^{\frac{(y_r-y_{Wall})}{\Delta}} E_n f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \tag{9}
\end{aligned}$$

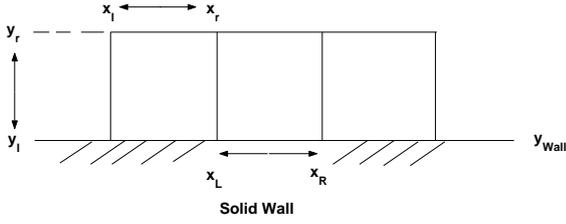


Figure 4: Computational domain for diffuse reflection from a surface.

where  $\Delta x$  is the width of the source region  $x_R - x_L$  and  $E_n$  is the energy of a particle in the direction normal to the wall  $E_n = 0.5v_n^2 + C$  where  $C$  is the internal energy per simulated degrees of freedom  $C = (1/2SD)((2C_v/R) - SD)s^2$  where  $SD = 2$  in a 2D simulation. The Maxwell-Boltzmann equilibrium distribution function  $f_{eq}$  is used in the directions parallel to the wall surface. In Equation 9, no assumptions are made regarding destination cell location. The destination region does not have to be adjacent to the wall surface, and any fraction of reflected particles can fall into the destination region. If the CFL number in the region near the wall is small, it is reasonable to assume that all reflected particles will be captured in the region between  $y_l - y_r$ . In this instance, the equations simplify to:

$$\begin{aligned}
 f_M &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(x_l-x)}{\Delta}}^{\frac{(x_r-x)}{\Delta}} \int_0^\infty f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \\
 &= \mathbf{f}_M(V_{wall}, s, \Delta t, x_R, x_L, x_l, x_r) \\
 f_{P_n} &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(x_l-x)}{\Delta}}^{\frac{(x_r-x)}{\Delta}} \int_0^\infty v_n f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \\
 &= \left( \sqrt{\frac{\pi}{2}} s \right) \times \mathbf{f}_P(V_{wall}, s, \Delta t, x_R, x_L, x_l, x_r) \\
 f_{E_n} &= \frac{1}{\Delta x} \int_{x_L}^{x_R} \int_{\frac{(x_l-x)}{\Delta}}^{\frac{(x_r-x)}{\Delta}} \int_0^\infty E_n f_n(v_n) f_{eq}(v_p) dv_n dv_p dx \\
 &= (s^2 + C) \times \mathbf{f}_E(V_{wall}, s, \Delta t, x_R, x_L, x_l, x_r)
 \end{aligned} \tag{10}$$

where  $s \equiv \sqrt{RT_{Wall}}$ ,  $\sqrt{\pi/2}s$  is the mean normal velocity of all reflected particles,  $s^2 + C$  is the mean normal energy of the reflected particles and  $V_{Wall}$  is the velocity of the wall. The proposed model provides an accommodation coefficient of 1.0, which is valid for many engineering applications with a few exceptions [9]. The momentum and energy components parallel to the surface of the wall are found using Equations 1-3 exclusively since all particles are assumed to fall into the cell adjacent to the wall. If the wall has a non-zero velocity in the normal direction to its surface the above equations can be modified to accommodate, however, this study is limited to a wall moving parallel to its surface.

#### Cell Division/Combination Criterion

General mesh adaptation methods can be separated into three branches - mesh generation (remeshing), mesh stretching or movement, and mesh enrichment (*h refinement*) [6]. Previous work has shown that mesh enrichment has numerous advantages over other techniques [10, 11]. One of the large disadvantages to using this form of refinement is the creation of hanging nodes which conventionally causes error and computational difficulty [6].

Conventional studies into adaptive meshing have relied upon various solution gradients, especially density, to locate regions for remeshing [4, 6]. Cell size restrictions derived through error calculations have been used extensively in recent adaptive meshing techniques, though typically are formed from lower order terms of the Taylor series expansion for truncation error [15, 14, 4]. Solvers based on the kinetic theory of gases or the direct simulation of a gas (such as DSMC) typically aim to ensure cell sizes are less than the mean free path  $\lambda$ . This can result in large amounts of wasted computational effort in the free stream [6, 7]. Therefore, previous efforts to use DSMC on unstructured adaptive meshes aim to divide cells according to a local Knudsen number in addition to a density parameter [6, 7] which ensures cells in the free stream are not resized or enriched.

In this study, three separate mesh adaptation parameters and criteria will be investigated, namely:

- Addition of cells near flow features - The local mathematical variance of a flow quantity sampled from the source cell and its neighbours is calculated and compared to a cutoff value  $\phi_s$ .
- Addition of cells near large flow gradients - The local value of a flow gradient (typically density) is calculated and compared to a cutoff value.
- Addition (or subtraction) of cells according to the local mean free path - Cells can be divided where the cell size is a certain fraction larger than the local mean free path.

#### Variance Criteria

In this option, the variance of the density in a source cell and its surrounding neighbours  $\sigma$  is multiplied by the effective width of the source cell  $\Delta x_e = (\Delta x \Delta y)^{1/2}$ , normalised by the source cell density and compared to cutoff values  $\phi_s$  and  $\phi_c$ :

$$\begin{aligned}
 \frac{\sigma \Delta x_e}{\phi_s \rho^2} &< 1 \rightarrow \text{Split Cell} \\
 \frac{\sigma \Delta x_e}{\phi_c \rho^2} &> 1 \rightarrow \text{Combine Cells}
 \end{aligned} \tag{11}$$

This method can be loosely related to Sun's [14] work on mesh adaptation, though Sun used a ratio of second and first order gradients as opposed to the variance. The advantage of the variance scheme is that cells cannot divide infinitely - each division makes a subsequent division more difficult. Another advantage is that it makes use of all neighbouring cell information, rather than those restricted by sharing a common interface. The main disadvantage is that the cutoff values  $\phi_s$  and  $\phi_c$  need to be selected manually and optimal values are not problem independent.

#### Flow Gradient Criteria

This scheme has been used extensively [13, 4] in adaptive mesh research. This scheme compares local density gradients to cutoff values such that:

$$\begin{aligned}
 \frac{|\rho_x| + |\rho_y|}{\phi_s} &< 1 \rightarrow \text{Split Cell} \\
 \frac{|\rho_x| + |\rho_y|}{\phi_c} &> 1 \rightarrow \text{Combine Cells}
 \end{aligned} \tag{12}$$

where  $\rho_x$  and  $\rho_y$  is the gradient of density in direction  $x$  and  $y$ , typically determined using a finite difference scheme. This scheme shares the same disadvantage as the proposed variance scheme in that the cutoff values are typically selected manually and not problem independent.

### Mean Free Path Criteria

TDEFM is built upon the assumption that the flow can be divided into a ‘free flight phase’ and a ‘collision phase’. For this approach to be valid, the cell sizes must be restricted to the local mean free path length. Previous work by Alexander *et.al.* [16, 17] has determined that the effective viscosity and thermal conductivity present in a direct solver are cell size dependent. To minimise error, the cell size must be less than the local mean free path. The criteria for cell division/recombination is therefore:

$$\begin{aligned} \frac{2\mu}{\rho\bar{c}\Delta x\phi_s} &< 1 \rightarrow \text{Split Cell} \\ \frac{2\mu}{\rho\bar{c}\Delta x\phi_c} &> 1 \rightarrow \text{Combine Cells} \end{aligned} \quad (13)$$

where  $\lambda \approx 2\mu/\rho\bar{c}$  is the mean free path,  $\bar{c} = (8RT/\pi)^{1/2}$  is the mean molecular speed and  $\mu$  is the viscosity. The desired ratio of cell size to mean free path size is  $\chi_s$  and  $\chi_c$ . In standard DSMC simulations, the cell size is typically 1/3 the length of the mean free path [19]. However, in regions where local flow gradients are low, the cell can be arbitrarily sized without consequence to the solution [6, 7, 18]. Here, the local gradient of the mean free path is used. It may be worth noting that the gradient of the local mean free path has been shown to be equivalent to the continuum breakdown parameter [12] though will not be used in that context here.

### Implementation

TDEFM does not require any special treatment of fluxes at cell interfaces since it analytically calculates the fluxes of free flight particles from any source region to an arbitrary destination region. No interpolation of interface states is required. Flux limiters can be used in the calculation of density gradients which might be applied to improve accuracy [2] though is not applied here. Hanging nodes do not present an issue with the accuracy or the ease of implementation of TDEFM. Since TDEFM is based upon the same foundations as a direct simulation, any CFL number can be used without fear of instability.

All simulations use an adaptive time step to ensure a ‘kinetic’ CFL number is kept below 1.0. The guide for the value of the time step with relation to the conditions in the cell is:

$$CFL = \frac{(|V| + \sigma\sqrt{RT})\Delta t}{\Delta x} \quad (14)$$

where  $\sigma$  is a selected number of variances of the equilibrium distribution and  $|V|$  is the magnitude of the velocity in the cell. Higher values of  $\sigma$  ensure that the surrounding neighbours of the source cell capture a larger fraction of the mass. Small values of  $\sigma$  allow the time step to be large enough for particles to travel in free-flight beyond the surrounding neighbours. If these distant cells are not registered as neighbours to the source cell, then this flux will be neglected and the results will be inaccurate. The results presented here use  $\sigma = 5$ .

It is important to realise that this is not a stability limitation. Any size time step can be selected, however, for large time steps the flow is free to move outside of the neighbouring cell locations and is not captured. As the time step approached infinity

| Simulation        | $K_n$ | $M$  | $\phi_s$ |
|-------------------|-------|------|----------|
| Density variance  | 0.04  | 8.73 | 0.005    |
| Density variance  | 0.05  | 8.0  | 0.005    |
| Density gradients | 0.04  | 8.73 | 2.0      |
| Density gradients | 0.05  | 8.0  | 1.0      |
| Mean free paths   | 0.04  | 8.73 | 2.0      |
| Mean free paths   | 0.05  | 8.0  | 3.0      |

Table 1: Cutoff values  $\phi_s$  for each condition and criteria.

all fluxes reduce to zero and the simulation solution will not advance. If the region of neighbouring cells was increased to include cells far away from the source, the simulation would progress and a steady solution would be reached, however, the results would not be valid since particles have (presumably) been allowed to travel far greater distances than the permitted mean free path length.

During simulation initialisation, neighbours of source cells are found through the following routine:

1. Select a source cell
2. Search though destination cells (i.e. all cells except the source cell) and calculate distance between cell centers  $R = [(c_{xd} - c_{xs})^2 + (c_{yd} - c_{ys})^2]^{0.5}$ .
3. Compare to a desired neighbour radius  $R_d$ . If cells which are immediately adjacent (including diagonal cells) are desired then  $R_d = [0.25((\Delta x_s - \Delta x_d)^2 + (\Delta y_s - \Delta y_d)^2)]^{0.5}$ .  $R_d$  can also be a function of flow speed and radial location (if desired), though in this study only immediate neighbours are considered.

This is a lengthy procedure, and is performed during the initialisation of the simulation only. For the addition or removal of cells mid-simulation a similar routine is used, though source and destinations are limited to neighbours of the cell to be split/combined. This means that the computational cost of splitting or combining cells is very low.

Boundary conditions are managed through the use of ghost cells, which store information regarding the mass, momentum and energy passed from real (internal cells) through the surfaces of the simulation region. These fluxes can either be calculated using a kinetic theory based vector split flux method (such as EFM) or approximated using a TDEFM flux into the ghost cell region. For small values of  $\Delta t$  these fluxes has been shown to be equivalent [1]. The momentum and energy in each ghost cell can be stored for calculation of heat transfer and drag coefficients, but is not used in calculating fluxes from diffusely reflective surfaces. The general procedure for calculation of fluxes is:

1. For all real (internal) cells - calculate TDEFM fluxes to all neighbours, including ghost cells, using Equations (1-7).
2. Subtract calculated quantities of mass, momentum and energy from the source cell and add it to the destination cell.
3. For all ghost cells - calculate reflected diffuse TDEFM fluxes into all real neighbours (not into neighbouring ghost cells) using Equations 10.
4. Subtract calculated quantities of mass, momentum and energy from ghost cells and place into real cells.
5. Check to ensure remaining mass in ghost cells is 0.

| Quantity            | Wu [6]     | Smith (current)                        |
|---------------------|------------|--|
| $M_{wall}$          | 8.73       | 8.0 and 8.73                           |
| $K_n$               | 0.04       | 0.05 and 0.04                          |
| $T_{wall}/T_{init}$ | 1.0        | 1.0                                    |
| Gas                 | Argon, VHS | Monatomic, $\mu = \mu_o(T/T_o)^{0.75}$ |
| Elements            | Tri, Rect. | Rectangular                            |

Table 2: Comparison between conditions used by Wu [6] and Smith (current) in the Lid Driven cavity problem.

Since the results presented are at steady state, only cell division is examined. At each remesh attempt, the tests shown in Equations (11-13) are carried out. Any cells found failing the tests are divided into four equal portions, each with equal amounts of mass, momentum and energy. This is for ease of implementation only - the method can be modified such that the number of divisions in each direction is different. The standard method is for the flow to be progressed with the minimum number of cells, and after a certain time remeshing attempts begin. In all simulations presented here remeshing was attempted every 50 time steps, and remeshing commenced after the simulation had progressed to past 75 percent of the steady flow time.

## Results

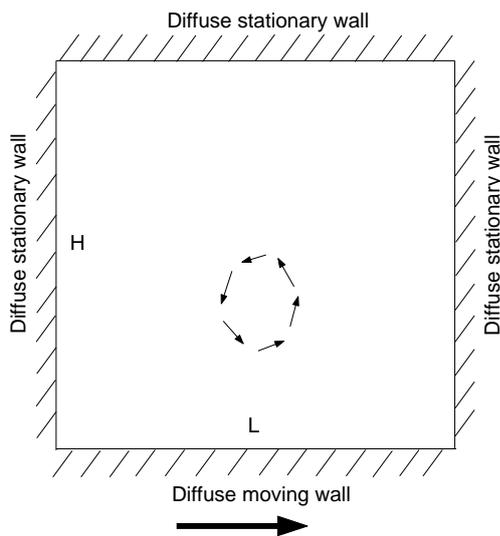


Figure 5: Diagram of a high speed viscous flow inside a lid driven cavity ( $M_{wall} = 8$  or  $8.73, T_{wall}/T_{init} = 1, K_n = 0.05$  or  $0.04, \mu = \mu_o(T/T_o)^{0.75}$ ).

Results are for viscid flow inside a lid-driven cavity as displayed in Figure 5. The lid-driven cavity problem has been extensively used to test viscous flow solvers and results here are compared to results from DSMC simulations on an adaptive mesh obtained by Wu *et.al.* [6]. The conditions are similar to Wu's initial conditions, and are compared in Table 3. The values of  $\phi_s$  for each adaptive simulation is shown in Table 1.

Figure 6 shows velocity vectors obtained from DSMC [6] and TDEFM on a uniform rectangular mesh. There is reasonable agreement between the flow features present. A large circulating region of relatively low density and high temperature (compared to regions far from the moving wall) is present near the moving wall. Inside this region, the mean free path is larger than in regions far from the moving wall. The coordinates of the center of circulation is provided in Table 3. It can be seen that the effect of increasing the Mach number from 8 to 8.73

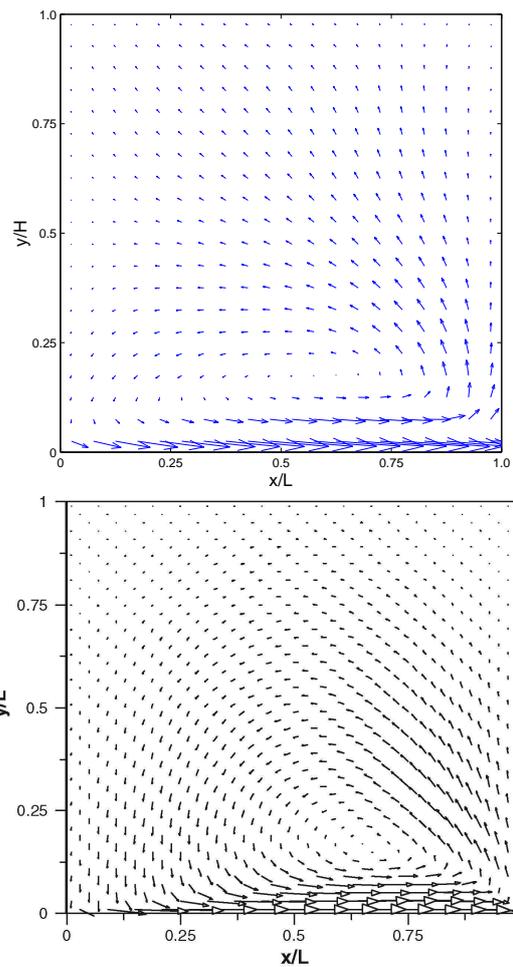


Figure 6: Comparison of velocity vectors of a high speed lid driven cavity. (Top) Uniform mesh using TDEFM with 400 quadrilateral cells,  $M_{wall} = 8.73$  and  $K_n = 0.04$ . (Bottom) Wu *et.al* [6] DSMC with 2500 quadrilateral cells,  $M_{wall} = 8.73$  and  $K_n = 0.04$ .

and decreasing the Knudsen number results in the center of circulation moving further to the right and closer to the wall. Placing too many cells in this circulating region effects the location of the center of rotation by pushing it further downstream and closer to the wall. This is due to the increased effective viscosity present in the solver - by using too many cells in a large mean free path region, we are forcing particles to collide (with an infinite collision rate [2]) when they should remain in free flight. The higher the mesh density, the higher the effective viscosity. This confirms the relationship between transport quantities and cell size as shown by Alexander *et.al.* [16, 17]. Therefore, it is critical that the cell size in all regions be the correct size.

Figure 7 shows the meshes used by the various adaptive mesh procedures. Using the local density variance criteria results in too many cells being placed along the wall and in the large mean free path region, resulting in an exaggerated viscosity. The same effect can be seen using the local density gradients criteria, though not to the same effect. All criteria capture the region of recirculation occupying the upper right hand corner as noted by Wu *et.al.* [6]. However, the mesh density in the region, as well as the mesh density in the large circulating region, has an effect on the magnitude of velocities and the center of

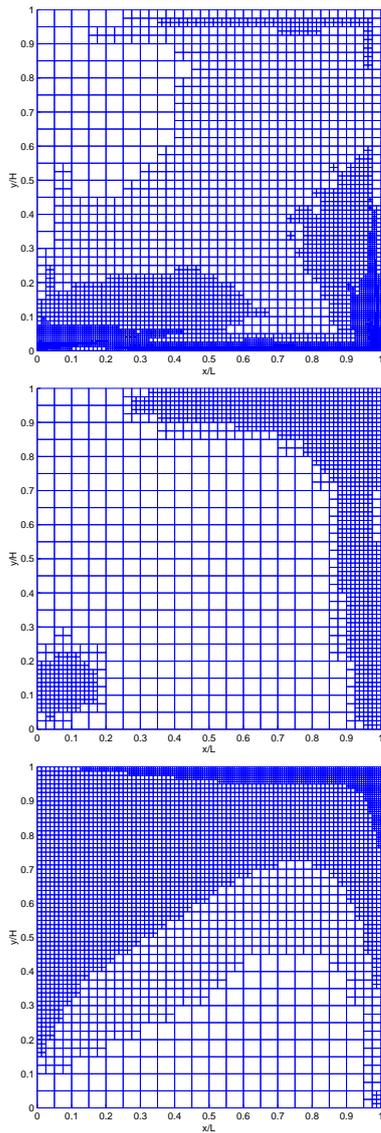


Figure 7: Meshes resulting from the use of (Top) density variance, (Middle) density gradients, and (Bottom) the local mean free path  $\lambda$  ( $K_n = 0.05, M = 8.0, \mu = \mu_o(T/T_o)^{0.75}$ ).

rotation of this recirculating region. For instance, the maximum Mach number in the circulating region was calculated to be 0.2 using the Mean free path criteria. If the Variance criteria is used, the maximum Mach number in the same region was calculated to be 0.07. The velocity vectors showing recirculation in this region is shown in Figure 8.

Figure 9 shows color graphs of the local mean free path  $\lambda$ . The mean free path is largest in the center of the circulation region and in the lower left hand corner region against the moving wall. Solutions using large mesh densities in the circulation region or the lower left hand corner show inconsistencies in the results which can be seen in these color graphs. The high mesh density solution has a non-physical ‘kink’, as does the adaptive mesh using density variance as a guideline. Using a density gradient performs better, since density gradients are generally quite low in the circulation region. The use of the local mean path as a guideline, although producing coarse results here, provides a better representation of the flow and results in the correct capture of the recirculation region.

| Simulation         | $K_n$ | $M$  | # Cells | Location     |
|--------------------|-------|------|---------|--------------|
| Wu [6]             | 0.04  | 8.73 | 2500    | (0.67, 0.16) |
| Coarse Mesh        | 0.05  | 8    | 400     | (0.57, 0.2)  |
| Coarse Mesh        | 0.04  | 8.73 | 400     | (0.6, 0.18)  |
| Fine Mesh          | 0.05  | 8    | 6400    | (0.7, 0.17)  |
| Fine Mesh          | 0.04  | 8.73 | 6400    | (0.77, 0.15) |
| Variance Criteria  | 0.05  | 8    | 5134    | (0.75, 0.11) |
| Variance Criteria  | 0.04  | 8.73 | 6241    | (0.76, 0.09) |
| Gradient Criteria  | 0.05  | 8    | 1750    | (0.57, 0.2)  |
| Gradient Criteria  | 0.04  | 8.73 | 1819    | (0.6, 0.17)  |
| $\lambda$ Criteria | 0.05  | 8    | 4252    | (0.6, 0.18)  |
| $\lambda$ Criteria | 0.04  | 8.73 | 6010    | (0.63, 0.16) |

Table 3: Comparison of the location of the main circulation obtained Wu [6] and Smith (current) in the Lid Driven cavity problem. Superscript \* indicates the Knudsen number and Mach number used match that used in [6].

### Conclusions

Presented is the True Direction Equilibrium Flux Method (TDEFM) applied on an adaptive mesh using various adaptation criteria. Diffusely reflective boundary conditions have been implemented through the integration of the reflected particle velocity probability distribution function. TDEFM has been shown [1, 2, 3] to capture unaligned flows on regular cartesian grids with higher fidelity than existing direction split methods. The fluxes obtained using TDEFM represent the analytical solution to the free flight phase of a direct simulation under the condition of equilibrium. Thus, fluxes can be calculated from any source cell to any other destination cell. Unlike most existing adaptive mesh continuum methods, TDEFM requires no interpolation of states at interfaces or higher order methods such as flux limiting to maintain stability. Also, the issue of hanging nodes, previously the cause of many stability and computational complications, can be ignored completely. TDEFM, being a continuum flux method, is also significantly faster than a direct simulation and produces no statistical scatter.

Three different adaption criteria were tested and compared. The mesh was adapted using local density variance, local density gradients and the local mean free path as guides and results compared to each other and those obtained using DSMC. The correct location of the main circulation region was predicted best when cell sizes were based upon the local mean free path  $\lambda$ . Using density variance or density gradients led to increased numbers of cells in large mean free path regions, causing an artificially high viscosity and shifting the location of the circulation region significantly.

### Acknowledgements

This work is supported by an Australian Postgraduate Award (APA) provided by the Australian government and a Scholarship supplement provided by the University of Queensland. The support, wisdom and motivation provided by my advisor Dr. Michael Macrossan is also greatly appreciated.

### References

- [1] Smith, M.R., Macrossan, M.N. and Abdel-jawad, M.M., ‘Effects of Direction Decoupling in flux calculations in Euler Solvers’, submitted to *Journal of Computational Physics*, May 2007.
- [2] Smith, M.R., Macrossan, M.N., Abdel-jawad, M.M. and Ferguson, A., ‘DSMC in the Euler Limit and its approximate Kinetic Theory Fluxes’, In *Proceedings of the 14th*

*National Taiwan CFD Conference*, 16-18th August, 2007, Nantou, Taiwan.

- [3] Macrossan, M.N., Smith, M.R., Metchnik, M. and Pinto, P.A., 'True Direction Equilibrium Flux Method: Applications on Rectangular 2D Meshes', In *25th International Symposium on Rarefied Gas Dynamics*, 21-28th July, 2006, St. Petersburg, Russia, <http://eprint.uq.edu.au/archive/00004358>.
- [4] Keats, W.A. and Lien, F.S., 'Two dimensional anisotropic cartesian mesh adaptation for the Euler Equations', *International Journal for Numerical Methods in Fluids*, **46**:1099-1125, 2004.
- [5] Wu, Z.N. and Li, K. 'Anisotropic cartesian grid method for steady inviscid shocked flow computation', *International Journal for Numerical Methods in Fluids*, **41**:1053-1084, 2003.
- [6] Wu, J.S., Tseng, K.C. and Kuo, C.H., 'The direct simulation Monte Carlo method using unstructured adaptive mesh and its application', *International Journal for Numerical Methods in Fluids*, **38**:351-375, 2002.
- [7] Wu, J.S., Tseng, K.C. and Wu, F.Y., 'Parallel three-dimensional DSMC method using mesh refinement and variable time-step scheme', *Computer Physics Communications*, **162**:166-187, 2004.
- [8] Almeida, R.C. and Galeao, A.C., 'An adaptive Petrov-Galerkin formulation for the compressible Euler and Navier-Stokes Equations', *Computer methods in applied mechanics and engineering*, **129**:157-176, 1996.
- [9] Nocilla, S., 'Surface Interaction and Applications', In *Rarefied Gas Flows: Theory and Experiment*, Edited by Fiszdon, W., 1981, **3**, New York: Springer-Verlag, 1981.
- [10] Rausch, R.D., Batina, J.T. and Yang, H.T.Y., 'Spatial Adaptation procedures on Unstructured Meshes for accurate Unsteady Aerodynamics flow computation', *AIAA Paper 91-1106*.
- [11] Connell, S.D., Holms, D.G., 'Three Dimensional unstructured adaptive multigrid scheme for the Euler Equations', *AIAA Journal* **32**:1626-1632, 1994.
- [12] Macrossan, M.N., 'A particle only hybrid method for near continuum flows', In *AIP Conference Proceedings: 22nd International Symposium on Rarefied Gas Dynamics*, Edited by Bartel and Gallis, **585**:426-433, 2001.
- [13] Lien F.S., 'A pressure-based unstructured-grid method for all-speed flows', *International Journal for Numerical Methods in Fluids*, **33**:355-374, 2000.
- [14] Sun M., *Numerical and experimental studies of shock wave interaction with bodies*, Ph.D. Thesis, Tohoku University, 1998.
- [15] Ham F., Lien F.S., Strong A.B., 'A Cartesian grid method with transient anisotropic adaptation', *Journal of Computational Physics*, **179**:469-494, 2002.
- [16] Alexander, F.J., Garcia, A.L. and Alder, B.J., 'Cell size dependence of transport coefficients in stochastic particle algorithms', *Phys. Fluids*, **10(6)** : 1540-1542, 1998.
- [17] Alexander, F.J., Garcia, A.L. and Alder, B.J., 'Erratum: Cell size dependence of transport coefficients in stochastic particle algorithms [Phys. Fluids 10 (1998)]', *Phys. Fluids*, **12(3)** : 731, 2000.
- [18] Lilley, C. *A macroscopic chemistry method for the direct simulation of non-equilibrium gas flows*, PhD Thesis, The University of Queensland, Australia, 2005.
- [19] Bird, G.A., *Molecular Gas Dynamics and the direct simulation of gas flows*, Clarendon Press, Oxford, 1994.
- [20] Chang, C.L., 'Implementation Issues - A Parallel Code Framework Based on the CESE Method', In *Proceedings of the 1st Taiwan-USA workshop on the CESE Method*, **3**:1-56, 2007.
- [21] Wang L., and Harvey, J.K., 'The application of adaptive unstructured grid technique to the computation of rarefied hypersonic flows using the DSMC method', in *Rarefied Gas Dynamics*, edited by Harvey, J. and Lord, G., 19th International Symposium: 843-849, 1994.

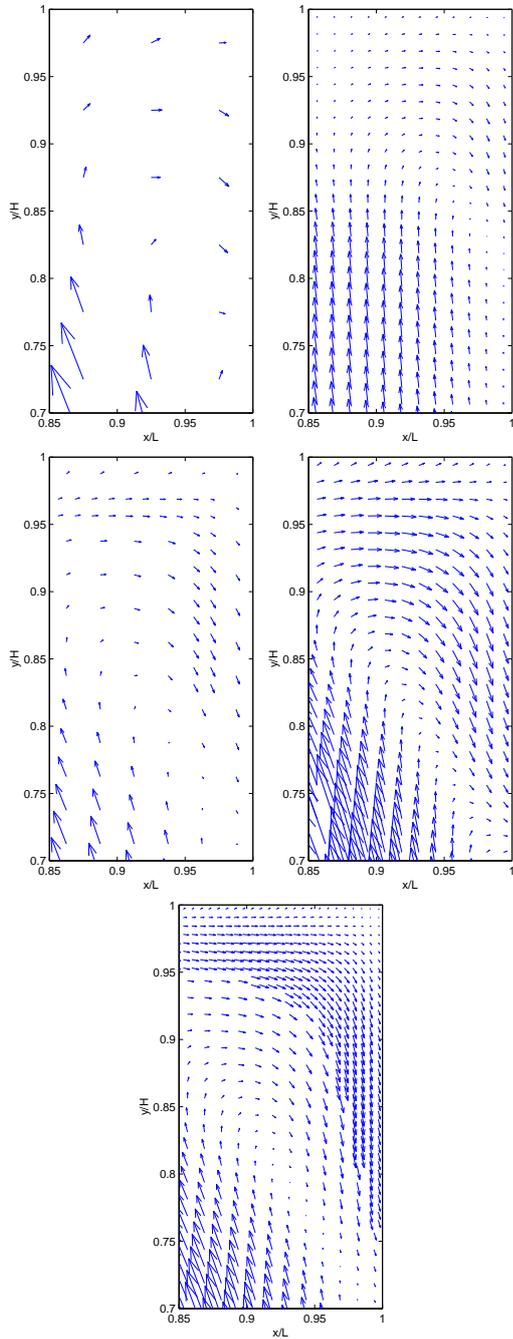


Figure 8: Quiver plots of secondary circulation described by Wu [6] in the region  $[0.85 < (x/L) < 1, 0.7 < (y/H) < 1]$ . (Top Left) Coarse regular mesh using 400 cells, (Top Right) Fine regular mesh using 6400 cells, (Middle Left) Adaptive mesh using variance criteria with 5134 cells, (Middle Right) Adaptive mesh using density gradient criteria with 1750 cells, (Bottom) Adaptive mesh using Mean free path criteria with 4252 cells ( $K_n = 0.05, M = 8.0, \mu = \mu_o(T/T_o)^{0.75}$ ).

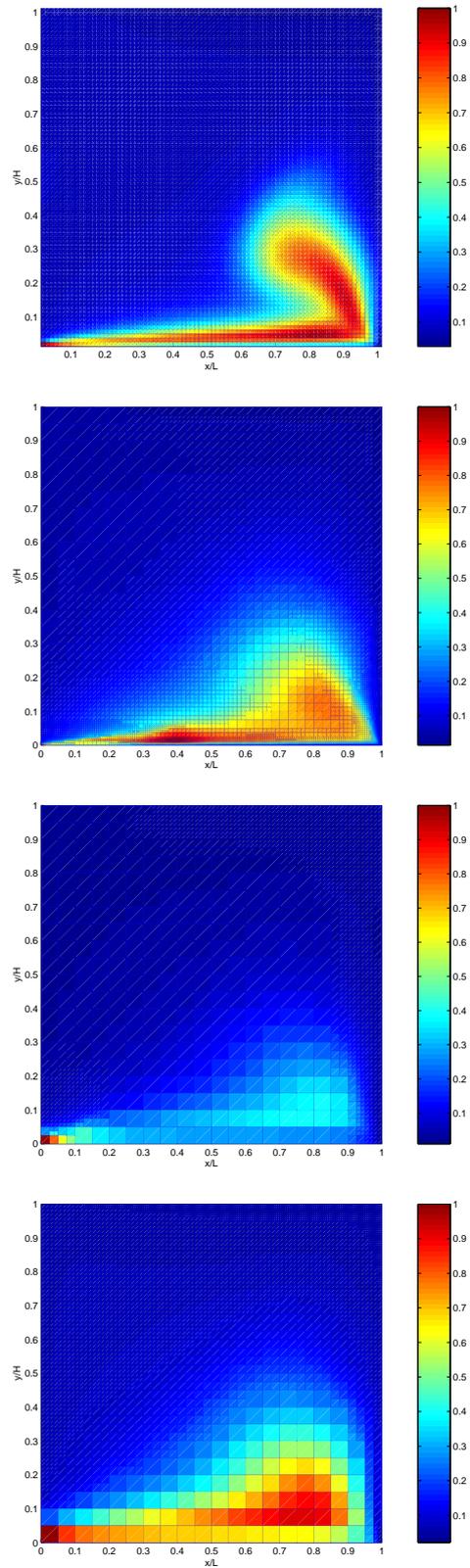


Figure 9: Mean free path length  $\lambda$  for various solvers (From top to bottom). (i) Fine regular mesh using 6400 cells, (ii) Adaptive mesh using density variance criteria with 5134 cells, (iii) Adaptive mesh using density gradient criteria with 1750 cells, (iv) Adaptive mesh using Mean free path criteria with 4252 cells ( $K_n = 0.05, M = 8.0, \mu = \mu_o(T/T_o)^{0.75}$ ).