Steffen Jebauer

Justyna Czerwińska

IMPLEMENTATION OF VELOCITY SLIP AND TEMPERATURE JUMP BOUNDARY CONDITIONS FOR MICROFLUIDIC DEVICES

Redaktor Naczelny:
　　　doc. dr hab. Zbigniew Kotulski

Recenzent:
　　　prof. dr hab. inż. Stanisław Drobniak –
　　　Politechnika Częstochowska

Praca wpłynęła do redakcji 13 czerwca 2007 r.

Contents

Abstract

The motivation of this work is to obtain a simulation tool which will be capable of modelling simple micro-devices in the gas slip flow regime. To achieve this goal *Maxwell Smoluchowski* boundary conditions were implemented by means of a udf routine into the commercial fluid finite volume solver Fluent. The applied velocity slip and temperature jump boundary condition are varied for small *Knudsen* numbers $Kn \leqslant 0.1$. In this regime there are several industrial applications such as micro-electro-mechanical systems (MEMS), heat exchange on chips and boundary layer problems for aerospace and turbomachinery applications. This report shows how the boundary condition implementation was done and is applied to a number of test cases: flow through a rectangular channel, Couette flow between two cylindrical surfaces, thermal creep flow between two heated tanks, flow through a bended diverging channel, a box heated with different wall temperatures and a vortex pattern flow between cylindrical surfaces heated with different temperatures.

    Among the test case setups are a few validation cases where good agreement with numerical or analytical results from references was obtained.

    Additional verification test cases were created in order to demonstrate certain abilities of the implementation.

List of Symbols

Δ_m	Residual of momentum equation
γ	Specific heat ratio
λ	Mean free path
μ	Dynamic viscosity
ω	Rotational velocity
ρ	Density
σ	Molecular diameter
σ_T	Thermal accommodation coefficient

σ_v	Tangential momentum accommodation coefficient
ζ	Slip length
A, B	Constants in analytical equation
b	Constant in boundary condition formulation
$\boldsymbol{c}$	Transformation coefficients
f	Under-relaxation factor
h	Channel height
k	*Boltzmann* constant
Kn	*Knudsen* number
L	Channel length
l	Characteristic length scale
$\vec{n}$	Unit wall-normal vector
p	Pressure
Pr	*Prandtl* number
R	Specific gas constant, radius
T	Temperature
u_τ	Circumferential velocity
$\vec{U}$	Velocity vector in fixed coordinate system
$\vec{u}$	Velocity vector in local wall coordinate system

Introduction

A classification of flows into no-slip, slip, transition and free molecular flows for gases can be made according to their respective *Knudsen* number Kn defined as:

$$Kn = \frac{\lambda}{l},\tag{1.1}$$

$$\lambda = \frac{k\,T}{\sqrt{2}\,\pi\,\sigma^2\,p}\ .\tag{1.2}$$

Here λ is the mean free path, l a characteristic length scale of the considered problem, T the temperature, σ the molecular diameter, p the pressure and k the *Boltzmann* constant.

Flows with $Kn \leqslant 0.001$ are regarded as no-slip flows, $0.001 \leqslant Kn < 0.1$ are slip flows; $0.1 \leqslant Kn < 10$ are flows in transitional regime and with *Knudsen* number $Kn > 10$ are regarded as free molecular flows [9]. Above the slip flow regime the *Navier Stokes* equations can no longer be applied. However, flows in the slip flow regime can be handled with the *Navier Stokes* equations, if appropriate boundary conditions are provided.

Several formulations of such boundary conditions were proposed [5, 6, 9, 12, 15]:

$$u_{fluid} - u_{wall} = \frac{2 - \sigma_v}{\sigma_v}\lambda\frac{\partial u}{\partial y} + \frac{3}{4}\frac{\mu}{\rho T}\frac{\partial T}{\partial x},\tag{1.3}$$

$$T_{fluid} - T_{wall} = \frac{2 - \sigma_T}{\sigma_T}\frac{2\gamma}{\gamma + 1}\frac{\lambda}{Pr}\frac{\partial T}{\partial y}.\tag{1.4}$$

Equations (1.3) and (1.4) are the *Maxwell* velocity-slip boundary condition in its velocity formulation and the *Smoluchowski* temperature jump condition, stating the difference between the actual value of the wall and of the fluid at the wall. Here Pr is the *Prandtl* number, σ_v the tangential momentum accommodation coefficient, σ_T the thermal accommodation coefficient, ρ the gas density,

u the stream-wise velocity, γ the specific heat ratio, x and y the tangential and wall-normal coordinate direction respectively

These formulations apply to flat surfaces in non-rotating domains. As it was noted by other investigators neglecting the influence of surface curvature or rotating motion on the slip behaviour leads to false predictions. Introducing necessary modifications for those cases extended formulations have been proposed [11]:

$$u_{fluid} - u_{wall} = \frac{2 - \sigma_v}{\sigma_v}\lambda\left(\frac{\partial u}{\partial y} + \frac{\partial u}{\partial x}\right) + \frac{3}{4}\frac{\mu}{\rho T}\frac{\partial T}{\partial x}. \tag{1.5}$$

A further extension taking higher order terms into account has been proposed by the same authors [11]. Their formulation has been implemented only partly here (neglecting higher-order terms),

$$u_{fluid} - u_{wall} = \frac{2 - \sigma_v}{\sigma_v}\lambda\left[\frac{\partial u}{\partial y} + \frac{\partial u}{\partial x} + \frac{\mu}{\rho}\left(\frac{1}{\rho}\frac{\partial^2 \rho}{\partial x \partial y} - \frac{1}{T}\frac{\partial^2 T}{\partial x \partial y}\right)\right] + \frac{3}{4}\frac{\mu}{\rho T}\frac{\partial T}{\partial x}. \tag{1.6}$$

2

Numerical Model

The numerical investigation has been carried out with the commercial *Navier Stokes* fluid solver Fluent$^{\textcircled{R}}$ version 6.2.16. In this solver slip boundary conditions can be implemented in two different ways:

- by the velocity slip as a moving wall; the corresponding value can be provided in $\left[\dfrac{m}{s}\right]$

- providing the shear stress term at the wall in $[Pa]$

Similarly, the temperature or temperature jump at the walls can be stated in terms of the temperature directly $[K]$ or the resulting heat flux $\left[\dfrac{W}{m^2}\right]$ [8].

By means of additional function such as an udf library (see appendix A) the mentioned boundary conditions have been implemented. Herein an under-relaxation term was introduced that accounts for the comparably high changes of slip and temperature especially with higher *Knudsen* number values when started from a rather different initial guess. It will be shown later that with under-relaxation term the speed of convergence can be increased.

All calculations were performed as laminar flows with the three-dimensional segregated solver. In here the gas was regarded as being an ideal compressible gas thus having a non-constant density ρ which is a crucial fact for the estimation of the mean free path throughout the computational domain.

Specific information about the used solver are given below:

- Fluent solver 6.2.16;

- three-dimensional, double precision variant;

- laminar approach;

- segregated, implicit solving algorithm.

For surface curvature the local wall coordinate system is in general not aligned with the solver's coordinate system. This makes a transformation of coordinate system necessary. Figure 2.1 illustrates this problem. The velocity

vector $\vec{U}$ is described in the solver coordinate system $\vec{X}_N{=}(\vec{X},\vec{Y},\vec{Z})$. A transformation into the local wall coordinate system $\vec{x}_n{=}(\vec{x},\vec{y},\vec{z})$ leading to the velocity $\vec{u}$ will be applied.
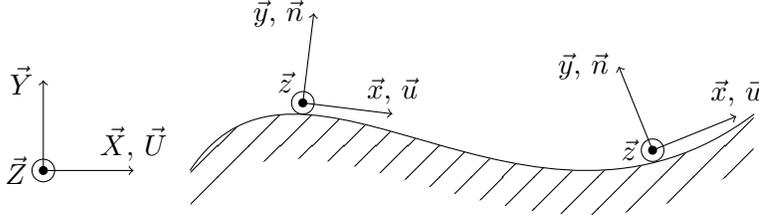

Fig. 2.1. Curved surface shown with local surface-fitted coordinate systems at two arbitrary points; the vectors $\vec{x}$, $\vec{y}$, $\vec{z}$ (small letters) indicate the local coordinate system and $\vec{X}$, $\vec{Y}$, $\vec{Z}$ (capital letters) indicate the fixed solver coordinate system; symbols $\vec{u}$ and $\vec{U}$ are the velocities in the local and fixed coordinate system; $\vec{n}$ is the unit wall-normal vector.

The transformation can be as in equation (2.1). Similarly the velocity gradient $\nabla^x \vec{u}$ is transformed as in equation (2.2).

$$\vec{u} = \boldsymbol{c} \cdot \vec{U},$$
$$\nabla^x T = \boldsymbol{c} \cdot \nabla^X T, \tag{2.1}$$
$$\nabla^x \vec{u} = \boldsymbol{c} \cdot \boldsymbol{c} \cdot \nabla^X \vec{U}. \tag{2.2}$$

Nabla operators ∇^x and ∇^X refer to the gradient operator evaluated in the local and fixed solver coordinate system respectively. The symbols $\boldsymbol{c}$ are the scalar products of both coordinate system base vectors $(\vec{x},\vec{y},\vec{z})$ and $(\vec{X},\vec{Y},\vec{Z})$. They represent 9 equations:

$$\boldsymbol{c} = \vec{x}_n \cdot \vec{X}_N. \tag{2.3}$$

In general the following properties are used in order to define the wall coordinate system: in the wall coordinate system the unit normal vector $\vec{n}$ equals $\vec{y}$ and the velocity vector u_m is aligned with $\vec{x}$ as the flow is parallel to the wall in its vicinity. If the flow is unsteady or the domain is in motion the transformation coefficients have to be computed in every single iteration step.

All coordinates of the velocity vector have to be stated in terms of the fixed solver coordinate system and thus are transformed back from the wall coordinate system.

Test Cases

3.1. Rectangular Channel

3.1.1. Setup

The calculations were performed with the model of a rectangular channel whith aspect ratio 20:1:1 in x, y, z direction. The gauge pressure [8] is set to 0 at the outlet and 5 Pa at the inlet, which applies for all configurations of this test case. The gauge pressure is added to the ambient pressure given in table 3.1 resulting in the respective static pressure p at the outlet boundary. The upper and lower z-boundaries are symmetry planes. At the lower and upper y-boundary planes the wall boundary conditions are applied. Details of the chosen grids for the finite volume method are given in section 5.1.
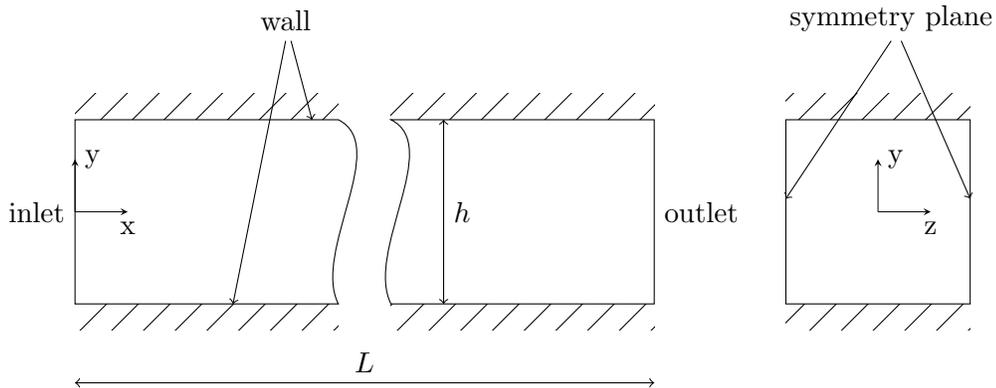

Fig. 3.1. Rectangular channel with length L and height h.

As it is clear from (1.2) the *Knudsen* number can be changed by either the characteristic length scale or the mean free path (e.g. changing the density). This is done with three sets of *Knudsen* numbers in the range of $0.0068 \leqslant Kn \leqslant 0.52$.

group	Kn_{out}	set	h [m]	p [Pa]	T [K]	ρ [kg/m^3]
1	0.0068	I	$\mathbf{10^{-5}}$	101.325	300	1.225
	0.0069	II	10^{-4}	**20.000**	**600**	$1.165 \cdot 10^{-1}$
	0.0069	III	10^{-3}	**2.000**	**600**	$1.165 \cdot 10^{-2}$
2	0.068	I	$\mathbf{10^{-6}}$	101.325	300	1.225
	0.069	II	10^{-4}	**2.000**	**600**	$1.165 \cdot 10^{-2}$
	0.069	III	10^{-3}	**400**	**1.200**	$1.162 \cdot 10^{-3}$
3	0.1	I	$\mathbf{6.8 \cdot 10^{-7}}$	101.325	300	1.225
	0.1	III I	10^{-3}	**400**	**1.740**	$8 \cdot 10^{-4}$
4	0.2	I	$\mathbf{3.4 \cdot 10^{-7}}$	101.325	300	1.225
5	0.34	I	$\mathbf{10^{-6}}$	101.325	300	1.225
	0.3445	II	10^{-4}	**400**	**600**	$2.33 \cdot 10^{-3}$
	0.3445	III	10^{-3}	**100**	**1.500**	$2.35 \cdot 10^{-4}$
6	0.4	I	$\mathbf{1.7 \cdot 10^{-7}}$	101.325	300	1.225
7	0.52	I	$\mathbf{2 \cdot 10^{-7}}$	101.325	300	1.225
	0.52	II	10^{-4}	**400**	**900**	$1.55 \cdot 10^{-3}$
	0.52	III	10^{-3}	**100**	**2.250**	$1.58 \cdot 10^{-4}$

Table 3.1. Simulation parameters for all cases at the outlet plane. The *Knudsen* number increases from group number 1 to 7. For set number I *Knudsen* number varies with geometry, for set number II and III with temperature and pressure. Sets number II and III differ in the channel height h.

Pressure and temperature values were chosen arbitrarily. The given variable values are computed at the position of the outlet in each case and presented in table 3.1. The values subject to variation within one set are displayed in bold-face.

The dynamic viscosity μ was set according to equation from [10] where R is the specific gas constant:

$$\mu = 5/16 \sqrt{\frac{\pi}{R\,T}} \frac{k\,T}{\pi\,\sigma^2}. \tag{3.1}$$

The options common for all configurations of this test case setup read as follows:

- steady state calculation;
- first order upwind spatial discretisation scheme;
- laminar simulation with ideal gas: air;
- isothermal calculations;
- wall slip velocity boundary conditions applied at the walls.

3.1.2. Stream-wise Dependence

The outlet properties of the flow are presented in table 3.1. As a consequence of equation (1.2) the mean free path length depends on the pressure, thus the value changes along the flow direction. In the framework of this investigation a constant pressure difference $\Delta p = 5\ Pa$ for all cases was applied. Even in the case of high *Knudsen* number $Kn = 0.3445$ (set III) this value corresponds to a pressure ratio not higher than $p_{in}/p_{out} = 1.05$. Thus one could reason that the variation of *Knudsen* number along the wall in the channel is rather low. For all other cases the influence should be on an equal level or even lower.

Figure 3.2 compares two profiles for two different Knudsen numbers. In that case only slight influence of the position in the channel with length L on the resulting channel velocity profiles can be seen. In fact the increase between the $x = 0.25\,L$ and the $x = 0.75\,L$ position of wall velocity in the case of $Kn = 0.3445$ is 3.3%, and at the centerline is 2.2%. For *Knudsen* number $Kn = 0.1$ a wall velocity difference of 1.0% was found.
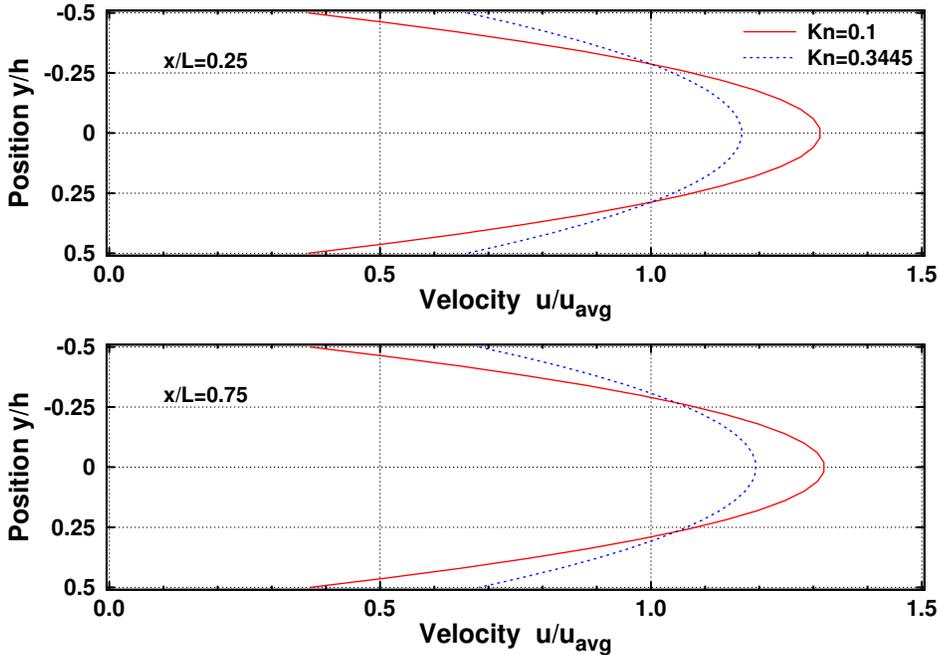

Fig. 3.2. Channel flow; velocity profiles at two positions along the channel ($x/L = 0.25$ and $x/L = 0.75$) for group 3 and 5 of set I; all velocities are normalised by the average velocity u_{avg} of the $x/L = 0.25$ profiles; the pressure ratio 1.05 was set.
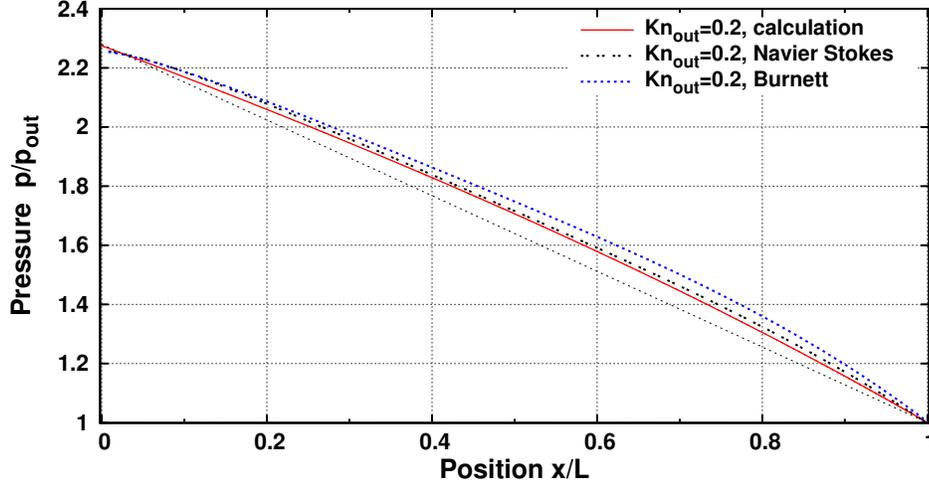
Fig. 3.3. Channel flow; pressure distribution along the channel using the setup of set III; values are normalised with the corresponding outlet pressure p_{out}; curves are compared with results for *Navier Stokes* and *Burnett* equations [9]; additionally a straight line for p/p_{out} is plotted from inlet to outlet.

As the difference in spanwise velocity is comparably small an additional case has been set up keeping Kn_{out} constant but increasing the inlet pressure ratio $p_{in}/p_{out} = 2.28$. The rectangular channel setup of reference [9] serves to validate the implementation and demonstrate the influence of comparably high applied pressure gradients.

An analysis [2] of the *Navier Stokes* equations showed that then the distribution of the pressure along the channel must be non-linear. Furthermore it was stated that the non-linearity in the pressure distribution at constant pressure ratio must decrease with increasing *Knudsen* number [1]. As can be seen in figure 3.3 the pressure is non-linear depending on the channel position x. The increased pressure ratio leads to a non-linear dependence of wall slip velocity on the channel position as well. The results used here for comparison are taken from reference [9] where the boundary condition is shown here for completeness in equation (3.2). That boundary condition has been applied in *Burnett* and *Navier Stokes* equations.

$$u_{fluid} - u_{wall} = \frac{2 - \sigma_v}{\sigma_v} \frac{Kn}{1 - b\,Kn} \frac{\partial u}{\partial y} \ , \tag{3.2}$$

where b is a constant, that can be determined analytically for applications in

the slip flow regime. Comparing equation (1.3) and (3.2) and neglecting the thermal creep term the value $\dfrac{Kn}{1 - b\,Kn}$ is of the same order as the mean free path λ.
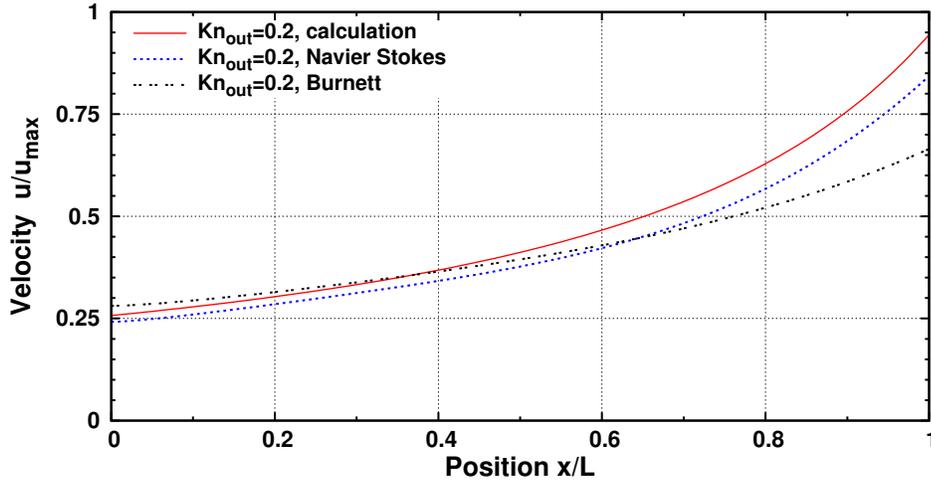

Fig. 3.4. Channel flow; wall velocity along the channel using the setup of set III; normalised with the maximum velocity at $x = 0$ position; curves are compared with results for *Navier Stokes* and *Burnett* equations [9].

The detectable deviation of pressure from linear characteristics is caused by the increase of velocity with decreasing Kn number (growing stream position along x axis), as is seen in figure 3.4. Near the inlet region all curves are considerably close to each other, but show differences when approaching the outlet region. In general the results found here agree better with the *Navier Stokes* approach [9]. As the *Knudsen* number grows towards the outlet the flow is outside of the validity of *Navier Stokes* equations. Therefore the *Burnett* equations approach predicts lower wall slip velocities near the outlet.

For the computed cases according to table 3.1 and noticing the influence on the pressure and velocity fields we can conclude that it is sufficient to examine only one velocity profile at an arbitrarily chosen position along the channel.
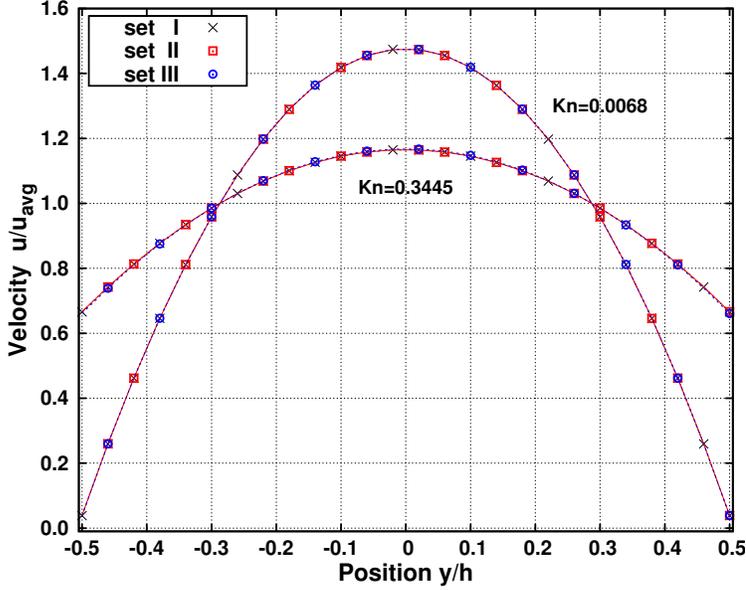
Fig. 3.5. Channel flow; normalised velocity at channel position $x = 0.25\,L$, for two different *Knudsen* numbers $Kn = 0.0068$ (group 1) and $Kn = 0.3445$ (group 5) for nominally two-dimensional grid 25x125x1.

3.1.3. Knudsen Number Dependence

The dependence of velocity profiles and slip velocity on the *Knudsen* number for the slip boundary conditions for all three sets at two different Kn values is shown in figure 3.5. Basically, the way the *Knudsen* number is varied shows no influence on the velocity profile. Hence only one set of velocity profiles over a range of Kn numbers is plotted in figure 3.6. For the ease of readability the curves are plotted solely on one half of the channel. In [4] similar investigations were performed using the *Direct Simulation Monte Carlo* method (DSMC). They found the function (3.3) to fit their velocity profile results. Herein $b = -1$ (mentioned in equation (3.2)) was chosen with h being the channel height in y direction

$$\frac{u}{u_{avg}} = \frac{-\left(\frac{y}{h}\right)^2 + \frac{y}{h} + \frac{Kn}{1-b\,Kn}}{\frac{1}{6} + \frac{Kn}{1-b\,Kn}} \ . \tag{3.3}$$

The values presented in [4] were in agreement with the *linearised Boltzmann* equation results of [14].

During the transition from a low Kn number to a higher value the non-
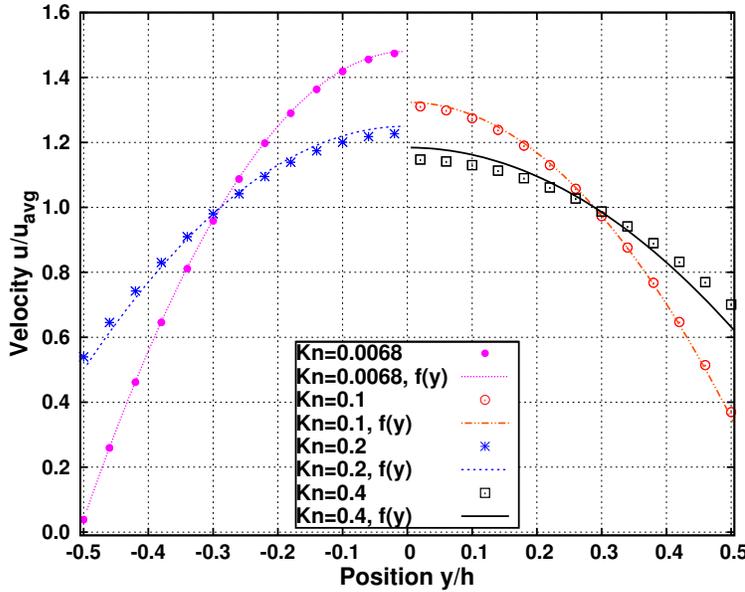
Fig. 3.6. Channel flow; normalised velocity profiles at channel position $x = 0.25\,L$ for four different *Knudsen* numbers (groups 1, 3, 4 and 6 of set I) for nominally two-dimensional grid 25x125x1. Only half of each profile is plotted to avoid overcrowding.

dimensional wall velocity increases and centreline velocity decreases. As a result both values approach the average value and thus flatten the whole curve.

For *Knudsen* number higher than $Kn = 0.2$ the wall slip and centreline velocity values deviate from the given curve $u = f(Kn)$ (equation 3.3), and they overestimate the value of wall slip velocity. Clearly, those values lie outside the range of the slip flow regime. Beyond this the usage of the *Navier Stokes* equations with first order slip boundary conditions, as used here, is no longer appropriate. However, the examples of flow within the slip-flow regime are in good agreement with the curves given by reference [4].

3.1.4. Influence of Tangential Momentum Accommodation Coefficient

In equations (1.3), (1.5) and (1.6) the parameter σ_v has been introduced which allows for the consideration of the amount of momentum that is contributed to the wall slip. A value of $\sigma_v = 1$ is related to diffusive reflections of the particles at the wall. As can be seen from equations (1.3) and (1.5) lowering σ_v increases the non-dimensional wall velocity slip value. As was noted before, in the same instant the non-dimensional centreline velocity decreases. However,
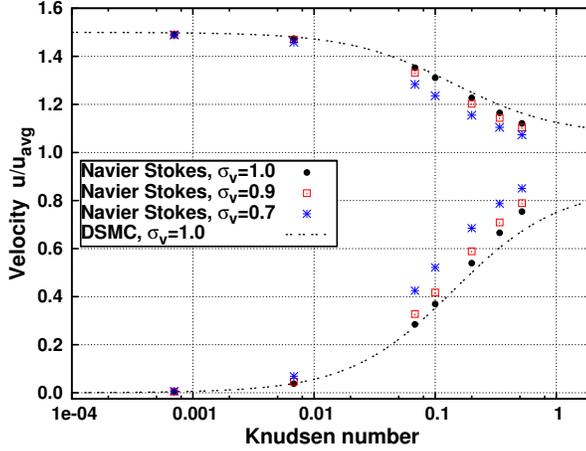
Fig. 3.7. Channel flow; normalised centreline (upper curve) and wall slip velocity (lower curve) in the function of *Knudsen* number for all groups in set I and nominally two-dimensional grid 25x125x1.

the value $\sigma_v = 0$ would lead to an infinite velocity slip value. Figures 3.7 and 3.8 show how the value of σ_v influences the wall slip and centreline velocity of set I and II respectively. The velocity curves shown in figure 3.7 and 3.8 (dashed line) follow equation (3.3). It has been pointed out before that above $Kn \approx 0.2$ or a density ρ below ≈ 0.01 the wall slip and centreline values deviate from the given curve $f(Kn)$, $f(\rho)$ due to the invalid descriptions of the flow regime by *Navier Stokes* equations.

3.2. Couette Flow

In this section a validation of the slip velocity implementation on curved surfaces has been carried out. The simulations of Couette flow between two cylinders with common axis of rotation where the inner one rotates and the outer one is at rest were performed. This test case serves two purposes: wall motion together with wall slip will be considered and the influence of wall curvature will be examined.

The outer and inner radii are expressed as multiples of the mean free path λ; $R_{out} = 25\lambda$ and $R_{in} = 15\lambda$. The gas is air with uniform temperature $T = 300\ K$. Taking the radial distance between both cylinders as a characteristic length the *Knudsen* number is $Kn = 0.1$. The computational domain consists of a 30° rotational-symmetric slice of the whole domain. The azimuthal end
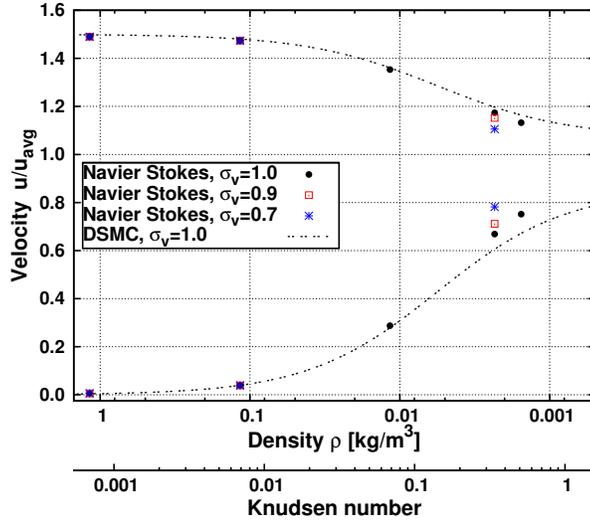
Fig. 3.8. Channel flow; normalised centreline (upper curve) and wall slip velocity (lower curve) in the function of density and *Knudsen* number (lower x-axis) for set II and nominally two-dimensional grid 25x125x1.

planes are set as a periodic pair. In direction of the rotational axis one grid element is placed with bounding symmetry planes representing a nominally two-dimensional domain. Inner and outer cylinder are set as walls with the velocity slip condition applied. Figure 3.9 illustrates this setup.
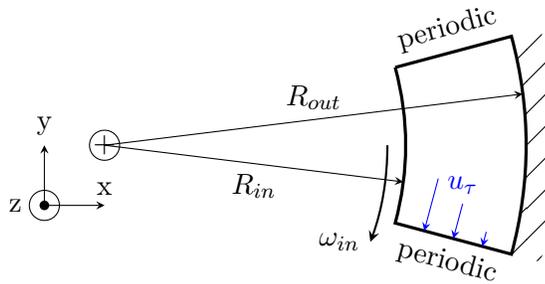

Fig. 3.9. Setup of Couette flow case; inner wall bounded by R_{in} is moving with rotational speed ω_{in}; the outer wall bounded by R_{out} is at rest.

Holding the inner surface in steady motion with a rotational speed of $\omega_{in} = 0.25$ *rpm* the wall slip effects have been studied for various values of the tangential momentum accommodation coefficient σ_v.

In a study of slip effects on macro- or microscopically curved surfaces formulations of the slip length ζ were considered [7, 17]. This expression is the term $\dfrac{2 - \sigma_v}{\sigma_v}\lambda$ from equation (1.3). It was stated that surface curvature contributes to the slip length and an expression for the velocity profile of the Couette shear flow was provided there. With ω_{in} being the angular velocity the equations read:

$$\zeta = \frac{2 - \sigma_v}{\sigma_v}\,\lambda, \tag{3.4}$$

$$A = (1 - 2\frac{\zeta}{R_{out}})/R_{out}^2, \;\; B = (1 + 2\frac{\zeta}{R_{in}})/R_{in}^2, \tag{3.5}$$

$$u_\tau(R) = \frac{\omega_{in}}{A - B}(A\,R - 1/R) \quad R = \{R_{in}\ldots R_{out}\} \tag{3.6}$$

The computational setup can be summarised as follows:

- steady state calculation;
- first order upwind spatial discretisation scheme;
- laminar simulation with ideal gas: air;
- isothermal calculations with $T = 300\ K$;
- wall slip velocity boundary conditions with wall motion applied at walls;
- grids: 30x26x1, 60x52x1.

Together with the computational results the function $u_\tau(R)$ for three different σ_v is plotted in figure 3.10. In general the velocity profile for slip *Navier Stokes* approach is in good agreement with the function given by reference [7].

3.3. Thermal Creep Flow

In equation (1.5) the thermal creep term of the slip velocity has been introduced. This term accounts for the slip induced by a tangential temperature gradient. Validation will be performed using the test case introduced in reference [3].

The setup configuration consists of two large gas tanks with two different constant temperatures ($T_1 = 300\ K$ and $T_2 = 400\ K$). The tanks are connected with a channel of length L and height h. Due to the tangential temperature gradient thermal creep flow in the wall's vicinity is induced. The resulting normal velocity gradient activates the slip flow mechanism. After pumping in the early stages of the process from cold to hot the pressure in the right tank is
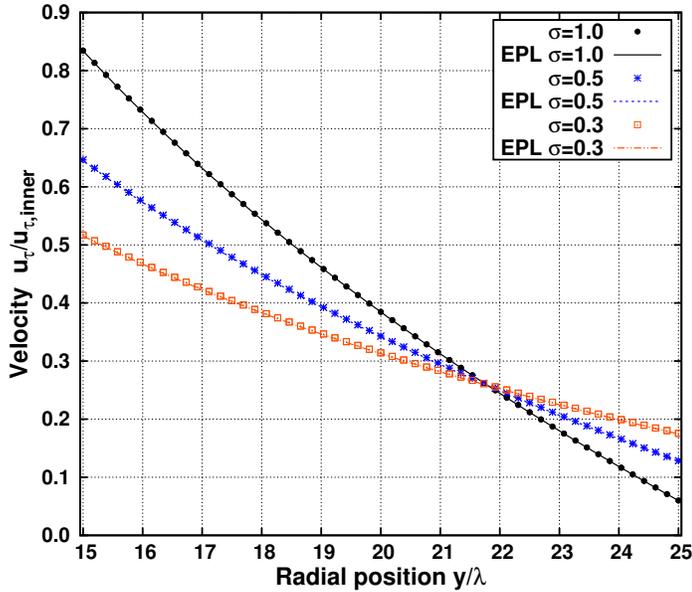
Fig. 3.10. Couette flow case; circumferential velocity u_τ compared with the theoretical prediction from [7], labelled as EPL; the radial position is normalised by the mean free path λ; result for nominally two-dimensional grid 60x52x1.

increased. This eventually leads to a back flow through the channel centre. As a result a steady state is reached where the thermal creep and pressure difference flow balance to a zero net mass flow in the channel.

The flow is simulated as an unsteady one with a constant time step. The *Knudsen* number was defined to be the average mean free path along the channel walls related to the channel height. In the following the main properties of the computational setup are summarised:

- second order implicit time discretisation scheme;

- second order upwind spatial discretisation scheme;

- laminar simulation with ideal gas air;

- wall slip velocity and temperature jump conditions applied at channel walls;

- zero wall shear stress constrained at tank walls;

- used grids:

  – middle channel section: 220x80x1;

  – tank section: 40x140x1.

With the initial condition of zero velocities and temperature distribution as in figure 3.11 the transient simulation was started. Results at an early time step and a solution where steady-state is reached are shown in figure 3.12 and 3.13. The transition from pumping towards the hot tank and the later zero net mass flow can be observed. Clearly during the initial pumping all x velocity values are positive thus pumping the fluid from cold to hot. From the curves of the later stages the velocity balance can be observed. In figure 3.14 and 3.15 the pressure distributions for the reached steady state solution along a horizontal plane ($y = 1/2h$, channel centre) are shown. The comparison of steady state pressure distributions for *Knudsen* number $Kn = 0.052$ shows only small differences. The static pressure difference is $\approx 0.05\%$ in the left side tank. In the $Kn = 0.122$ case a relative pressure difference of $\approx 0.2\%$ in the right side tank remains. However, due to the high changes in temperature and pressure e.g. along the channel a non-constant *Knudsen* number is obtained. Thus the way of averaging has to be regarded as the procedure here is not necessarily according to that of the reference case. The reference does not clearly state the definition of *Knudsen* number used in the simulation. Therefore detailed comparison is difficult to obtain.
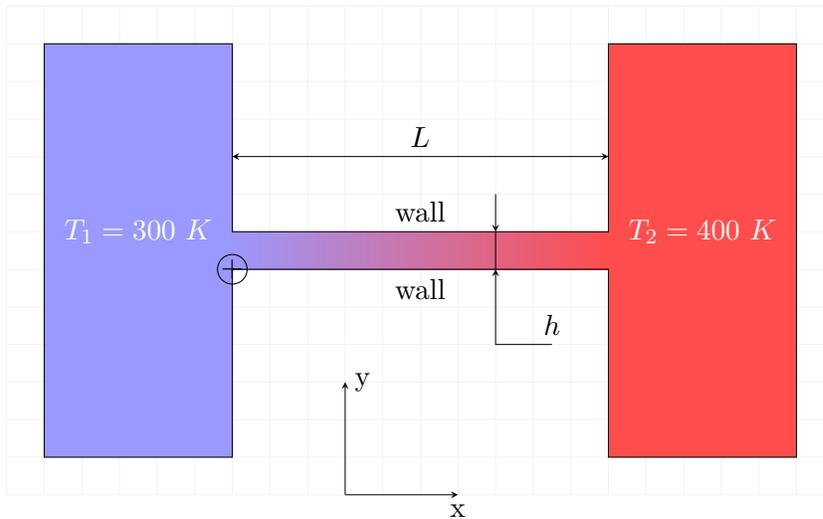

Fig. 3.11. Thermal creep setup. Colours indicate the temperature distribution; coordinate system origin marked with (+). Figure presents two large tanks filled with ideal gas connected with a small channel. Temperature between two tanks rises linearly from to 300 K to 400 K; L is the length of the connecting channel and h is the channel height.
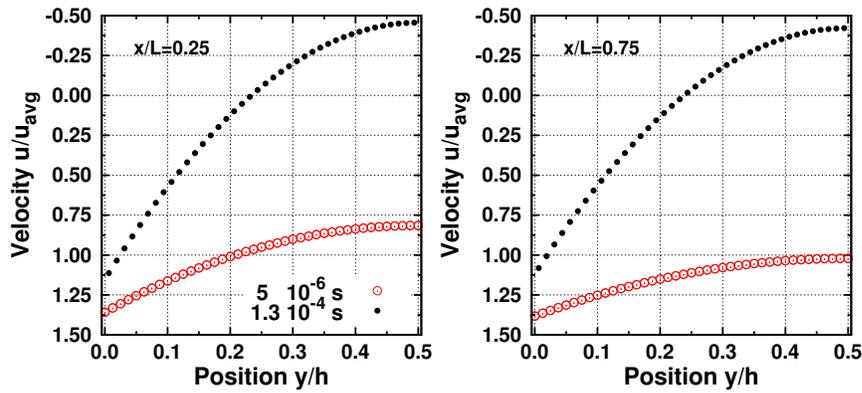
Fig. 3.12. Thermal creep setup; normalised x velocity distribution in the function of normalised channel height; left hand side labels indicate the stream-wise x position ($x/L = 0.25$ and $x/L = 0.75$); curve symbols correspond to the simulation time for two different times $t = 5\,10^{-6}s$ and $t = 1.3\,10^{-4}s$; velocities are normalised with the average velocity of the $x/L = 0.25$ profile of the first time step; calculations were performed for average *Knudsen* number $Kn = 0.052$.
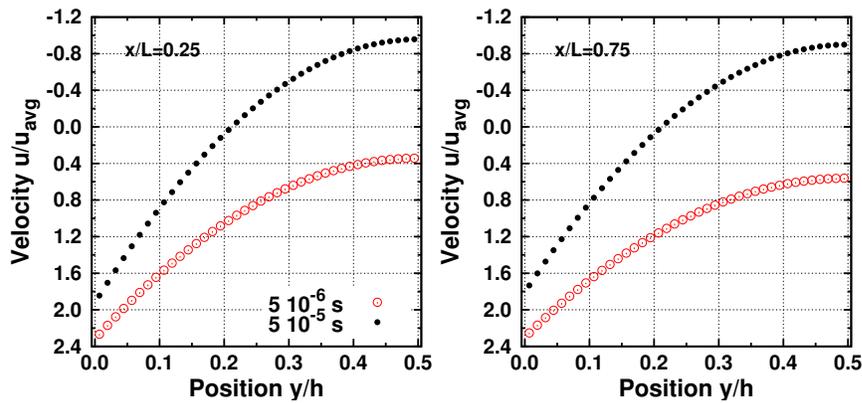

Fig. 3.13. Thermal creep setup; normalised x velocity distribution in the function of normalised channel height; left hand side labels indicate the streamwise x position ($x/L = 0.25$ and $x/L = 0.75$); curve symbols correspond to the simulation time for two different times $t = 5\,10^{-6}s$ and $t = 5\,10^{-5}s$; velocities are normalised with the average velocity of the $x/L = 0.25$ profile of the first time step; calculations were performed for average *Knudsen* number Kn=0.122
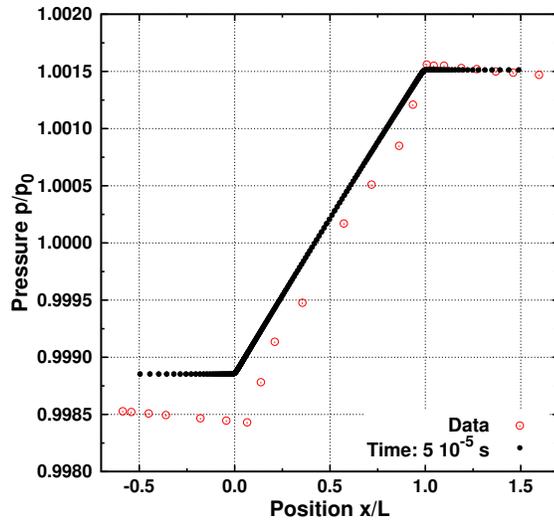
Fig. 3.14. Thermal creep setup; pressure distribution on horizontal plane located in the channel centre in the function of normalised x position; pressure is normalised with the ambient pressure p_0; Kn=0.052; the comparison data are obtained from Direct Simulation Monte Carlo performed by [3].
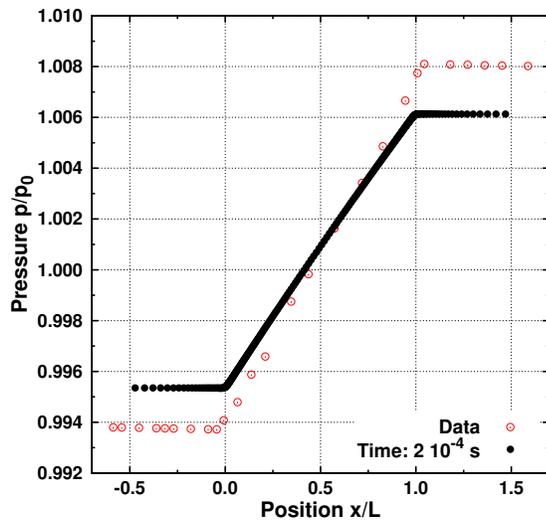

Fig. 3.15. Thermal creep setup; pressure distribution on horizontal plane located in the channel centre in the function of normalised x position; pressure is normalised with the ambient pressure p_0; Kn=0.122; the comparison data are obtained from Direct Simulation Monte Carlo performed by [3].

4

Verification Cases

4.1. Temperature Jump Boundary Condition

The condition for temperature jump was applied to the rectangular channel geometry setup shown in figure 4.1. Here a temperature gradient between the lower and the upper wall was imposed by setting the wall target temperature values to $T_L = 300K$ and $T_U = 350K$ ($T_U/T_L = 1.167$). The left and right walls were set as wall boundaries with $T = 300K$ as well.
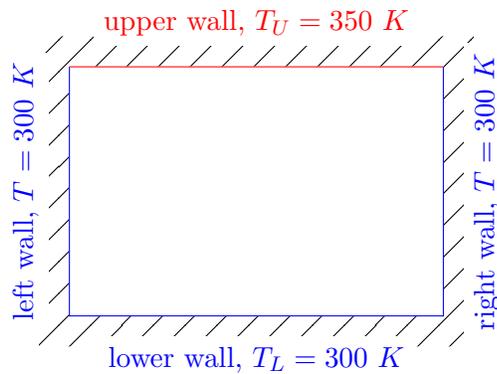
Fig. 4.1. Setup of the box for investigation of temperature jump boundary condition. All walls apart from the upper wall are heated with $T = 300\ K$, the upper wall has a temperature $T_U = 350\ K$.

The temperature profiles in comparison to a profile without the *Smoluchowski* boundary condition applied are shown in figure 4.2. It can be seen that the fluid near the walls does not reach the wall temperature value. In the temperature profiles the difference between the wall temperature values and the gas wall temperature increases with increasing *Knudsen* number.
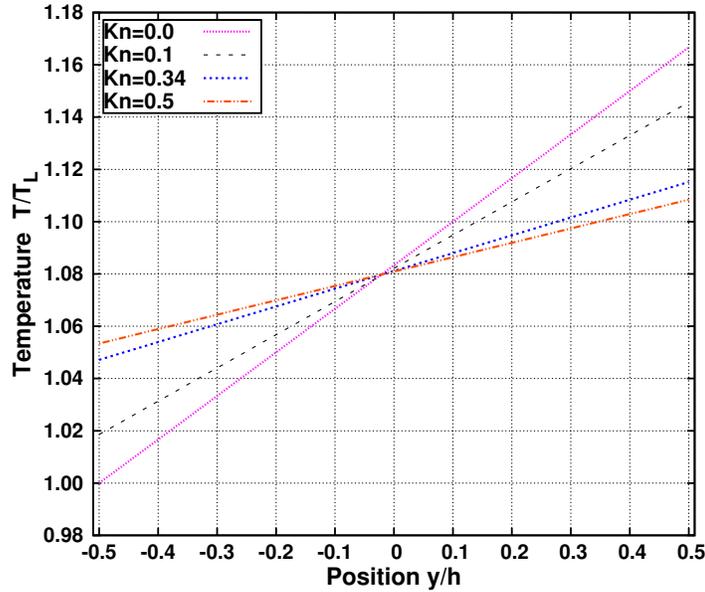
Fig. 4.2. Temperature jump boundary condition case; temperature profiles in the function of the normalised position for nominally two-dimensional grid 25x125x1 for four different *Knudsen* numbers.

4.2. Bended Diverging Channel

A simple verification test case that shows the influence of surface curvature has been created: the flow through a bended diverging channel.

The magnitude of the resulting velocity field is presented in figure 4.4 on a symmetry plane for a computation where a no-slip condition at the walls was applied. The *Knudsen* number $Kn = 6.89 \cdot 10^{-2}$ is defined here with the channel height at the inlet.

Figure 4.5 shows the wall velocity on the outer and inner wall surface. This velocity is in the function of the arc angle ϕ defined as being $90°$ in the upper left corner (inlet zone) and $0°$ in the lower right corner (outlet zone). The total wall velocities with its tangential and normal components are visualised. In figure 4.6 the distribution of velocity component pointing normal to the corresponding inlet and outlet is shown along the radius r. Both profiles are asymmetric with respect to their mid-plane; $r/h = 0$ in radial direction. The overall velocity is higher in the inlet plane due to its smaller cross section area compared to the outlet, thus the difference in wall velocity is higher here. The relative difference in the wall slip velocity at the inlet position is $\Delta u = 7.5\%$ whereas at the
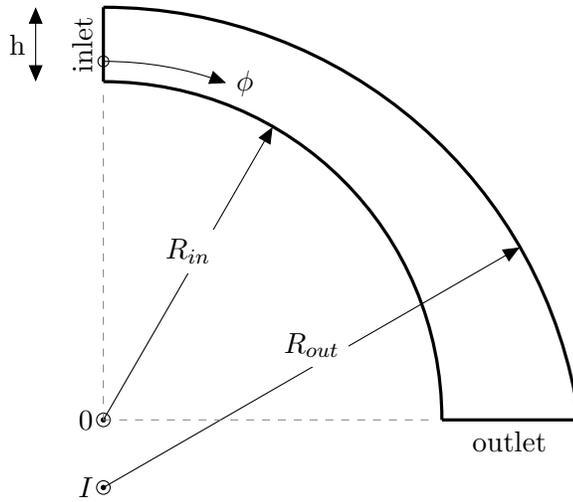
Fig. 4.3. Bended diverging channel setup; the radii are $R_{in} = 5.5\,10^{-4}m$ and $R_{out} = 7.5\,10^{-4}m$ with different midpoints; h is the inlet channel height defined as the characteristic length scale; the midpoint of angle ϕ is point 0; the flow is from the inlet at position $\phi = 0$ to the outlet.
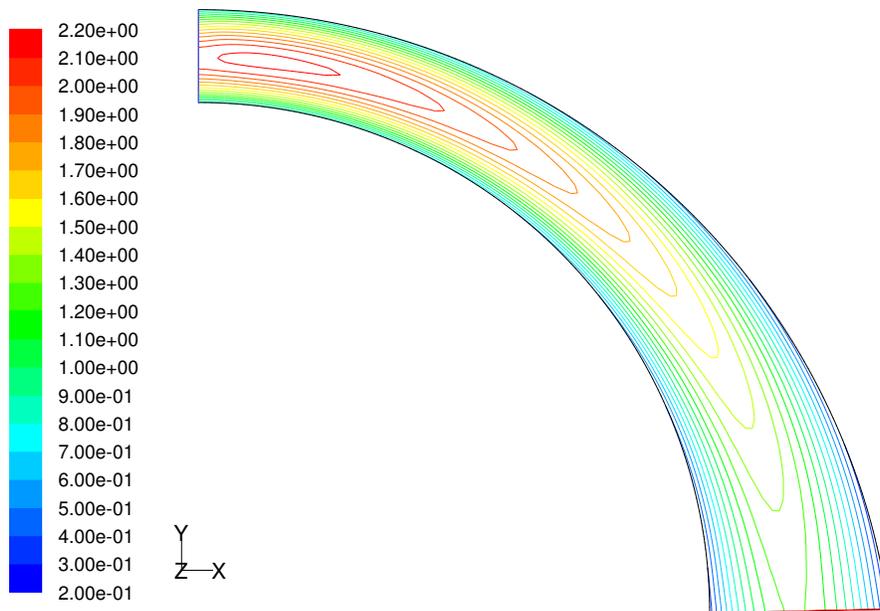
Fig. 4.4. Bended diverging channel flow; the colour scale indicates the normalised velocity magnitude; the *Knudsen* number is $6.89 \cdot 10^{-2}$.

channel outlet $\Delta u = 6.7\%$ remain.
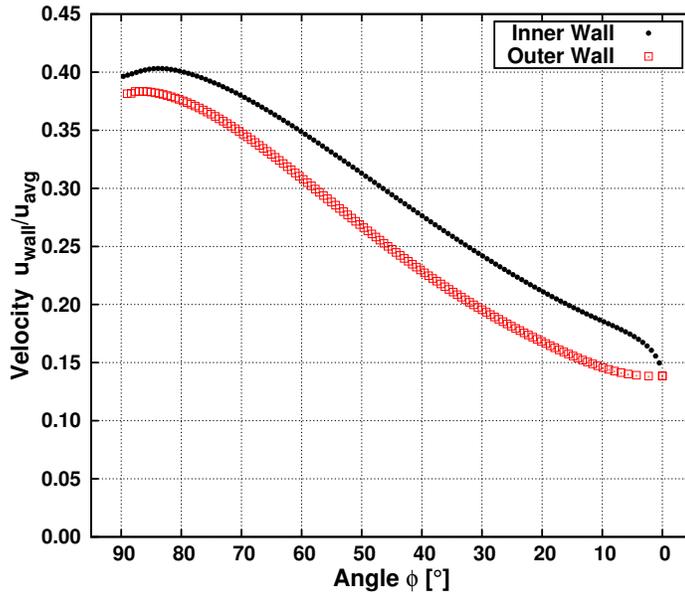
Fig. 4.5. Bended diverging channel flow; wall velocity normalised by the average velocity magnitude of the whole domain in the function of the arc angle ϕ; the *Knudsen* number is $6.89 \cdot 10^{-2}$.

4.3. Thermal Stress Slip Flow

An example test case for the verification of thermal stress slip flow is the two-dimensional setup proposed by Sone [16] and shown in figure 4.7. The wall temperatures in figure 4.7 are $T_{in} = 300\ K$, $T_{out} = 320\ K$ where the radii are $R_{in} = 2.5\,10^{-4}m$, $R_{out} = 5\,10^{-4}m$ and the centre-point distance is $d = 1.3\,10^{-4}m$. Defining the difference of radii $R_{out} - R_{in}$ as the characteristic length scale with the temperature $T = 300\ K$ and the ambient pressure $p = 20000\ Pa$ for the ideal gas air the *Knudsen* number $Kn = 0.0014$ is obtained here.

In the figure the dotted line marks a plane of symmetry here, with the two co-axial cylinders. Due to the different temperatures the second temperature derivative $\partial^2 T / \partial x \partial y$ does not vanish everywhere, which holds similarly for other terms in equation (1.6) with assumption of ideal gas behaviour. It has been pointed out that this temperature gradient contribution drives the flow in a
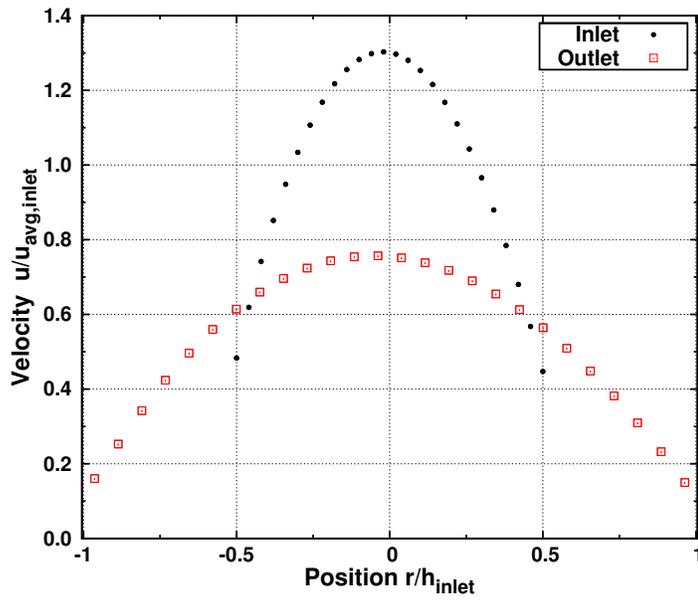
Fig. 4.6. Bended diverging channel flow; velocity profiles at inlet and outlet plane; both curves are normalised with inlet width h and average velocity; the *Knudsen* number is $6.89 \cdot 10^{-2}$.
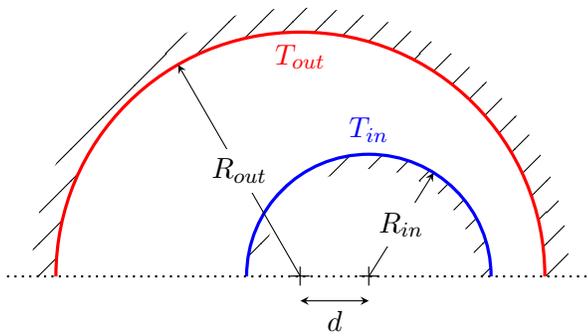
Fig. 4.7. Thermal stress slip flow setup; temperature at inner surface T_{in} is lower than outer surface temperature T_{out}; d is the distance between the rotation axes; the dotted line indicates the symmetry plane.

reverse direction against the thermal creep induced flow. However, thermal creep is not present in this case due to constant wall temperatures.

Because of missing quantitative information in the reference only a qualitative comparison is performed here. In fact, the result visualised here is in qualitative agreement with the papers of [11] and [16].
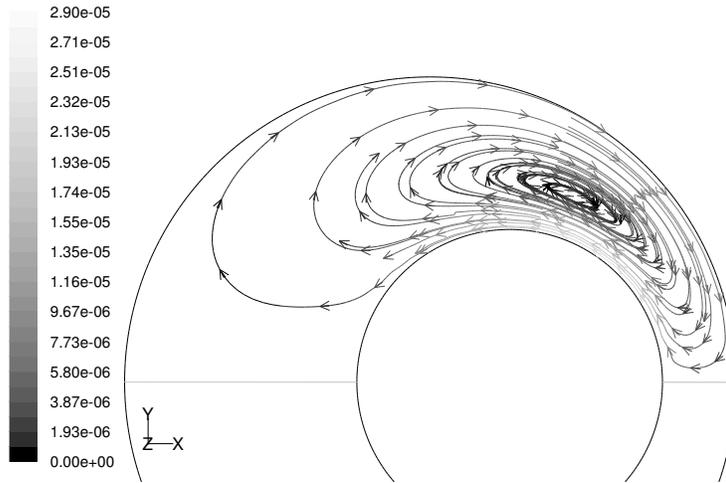
Fig. 4.8. Thermal stress slip flow stream lines; brightness corresponds to the velocity magnitude in $[m/s]$; $T_{in} = 300\ K$ and $T_{out} = 320\ K$; axis of symmetry shown in lower part of the image; the *Knudsen* number is $Kn = 0.0014$.

Numerical Errors

5.1. Spatial Discretisation Error

Using the rectangular channel setup (see section 3.1.1.) numerical errors are computed. Moreover the influence of the under-relaxation parameter is examined. Based on a preliminary investigation of the spatial discretisation influence on the velocity profile for a no-slip *Poiseuille* flow a final set of three grids was created. The influence of the discretisation error will be shown for this set. In respect to its coarser predecessor each subsequent grid has a scaling factor of 2 in the spatial directions x and y. In z direction a layer with a thickness of one cell only was created. This implies a two-dimensional flow field.

The wall and centreline velocities are compared in figure 5.1 and 5.2 respectively. They show the influence of the discretisation size in y direction normalised by the channel height h. Centreline and wall velocity are used then to estimate error values. Chosen grids sizes consist of 25x125x1 (coarse), 50x250x1 (medium) and 100x500x1 (fine) grid cells. The values of the discretisation error for a transition from the coarser grid to the finer grid are presented in table 5.1. The relative wall velocity difference for a coarse to fine grid transition is $\Delta u = 1.63\%$. Considering these values the approach to an asymptotic velocity profile during subsequent grid refinement can be assumed. With feasibility being a crucial part in the analysis of a rather large parameter set in this test case the choice of the coarsest grid 25x125x1 seems justified.

Grid transition	Errors	
	u_{centre}	u_{wall}
25x125x1→ 50x250x1	0.31%	1.07%
50x250x1→100x500x1	0.14%	0.55%

Table 5.1. Rectangular channel flow; spatial discretisation errors; the error values are calculated for the transition from coarser to the subsequent finer grid; the *Knudsen* number is $Kn = 0.1$.
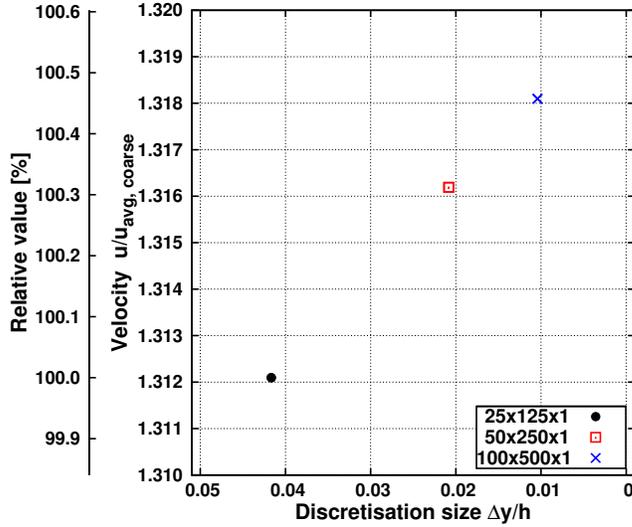
Fig. 5.1. Rectangular channel flow; normalised centreline velocities in the function of the discretisation size for three different grid levels at the stream-wise position $x = 0.25\,L$; all values are normalised with the average velocity of the coarsest grid result $\boldsymbol{u_{avg,coarse}}$; the outer y axis shows the relative velocity difference referring to the coarsest grid result; the *Knudsen* number Kn=0.1 was chosen.

5.2. Iteration Error

An investigation of the dependence of the iteration error is carried out by means of the domain mass imbalance between inlet and outlet and the obtained wall velocities. It is again applied for the flat rectangular channel test case. Normalisation of the values is achieved by the inlet mass flow and the average velocity respectively. Calculations here were performed with an under-relaxation factor of 0.5 for the slip velocity expression. Initial values for the calculation were taken from a converged solution with a no-slip wall velocity condition.

As can be seen from figure 5.3 and 5.4 a scaled residual for x-velocity momentum [8] value of $\Delta_m = 10^{-7}$ is sufficient to obtain a converged solution. Below a residual criterion of $\Delta_m \approx 5 \cdot 10^{-5}$ the relative mass flow difference between inlet and outlet exceeds the lower limit of the solver's capabilities for number display.

Figure 5.5 demonstrates how the velocity profile at the chosen channel position evolves from the no-slip profile to the converged solution within different residual levels. No substantial changes occur between the velocity momentum
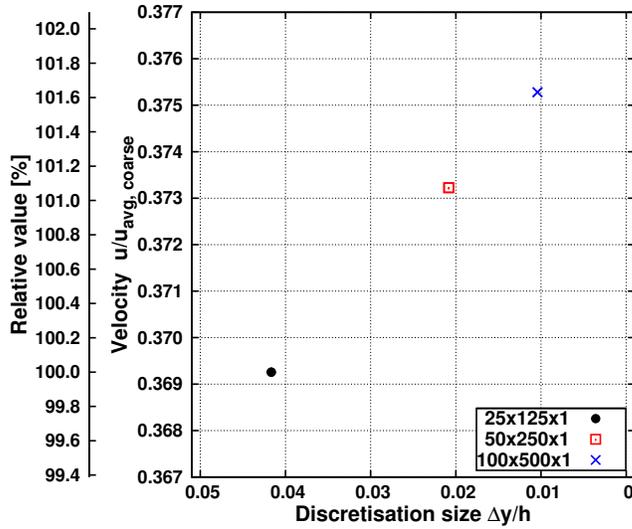
Fig. 5.2. Rectangular channel flow; normalised wall slip velocities in the function of the discretisation size for three different grid levels at the stream-wise position $x = 0.25\,L$; all values are normalised with the average velocity of the coarsest grid result $\boldsymbol{u}_{avg,coarse}$; the outer y axis shows the relative velocity difference referring to the coarsest grid result; the *Knudsen* number Kn=0.1 was chosen.

residual level $\Delta_m = 10^{-6}$ and $\Delta_m = 10^{-7}$. Both curves overlap each other completely.

5.3. Influence of Under-Relaxation

For the case of slip flow regime a no-slip initial guess solution could be used. Though the change of wall velocity for the subsequent iterations is comparably large. In a first iteration the slip velocity is set using the no-slip velocity gradient known beforehand. In general this leads to an increased mass flow through the considered cross section. With the changed wall velocity in the same moment the wall shear stress is decreased and the whole inside channel flow has to be updated to reestablish the equilibrium between the flow driving pressure gradient and the stress. With the flattened channel profile the velocity gradient in the wall's vicinity is decreased and results in a lower wall velocity in the next iteration.

This procedure of continious wall velocity and gradient updating might slow down the overall convergence of the simulation or even cause divergence.

Therefore, it will be shown that the under-relaxation factor $f = 0 \ldots 1$,
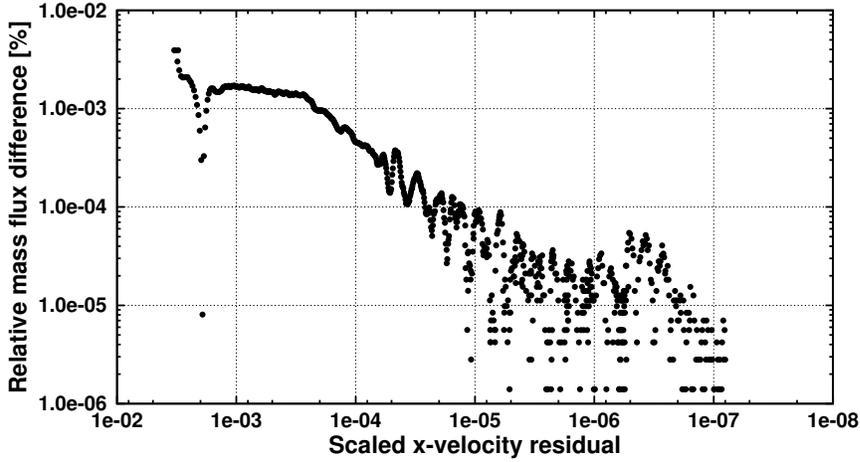
Fig. 5.3. Relative mass flux difference between inlet and outlet in the function of the scaled residual for x-velocity momentum.

which was introduced before, serves to increase the speed of convergence. The updated velocity value u_k^n depends on the former value u_k^{n-1} and the actual value $u_k^{'n}$ as,

$$u_k^n = (1 - f) \cdot u_k^{n-1} + f \cdot u_k^{'n}. \tag{5.1}$$

A similar expression is used for temperature under-relaxation. Setting the under-relaxation coefficient value to a high level again might lead to divergence of the solution process for some cases. An investigation of *Knudsen* numbers $Kn = 0.1$ and $Kn = 0.4$ was performed varying f according to a set of values. All calculations were started with the initial distribution from a converged no-slip solution of the same Kn value from set I. As a convergence criterion all scaled equation residuals had to fall below the value $\Delta_m = 10^{-7}$.

Figures 5.6 and 5.7 show that the lower the Kn value is set the higher the value of f can be chosen here. In general f can be increased with decreasing Kn value, which follows from the smaller quantitative difference between the no-slip initial guess and the resulting wall slip. Both plots show that for high values of f the calculation is diverging. It needs to be underlined that for all chosen values of f for a converged solution the same final wall velocity value is obtained.
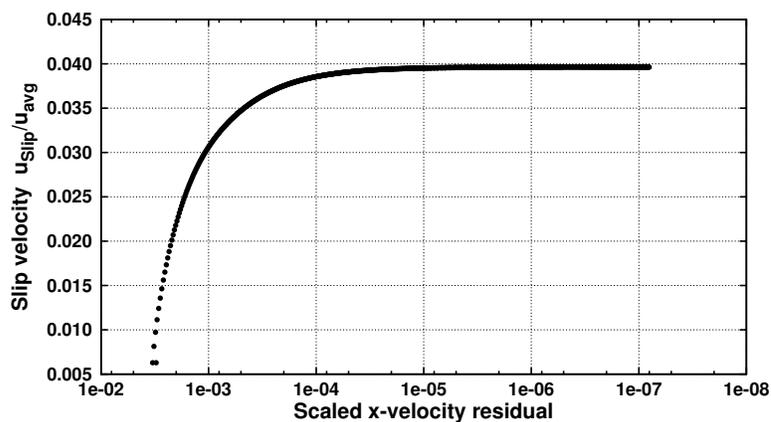
Fig. 5.4. Rectangular channel flow; normalised wall slip velocity in the function of the scaled residual for x-velocity momentum.
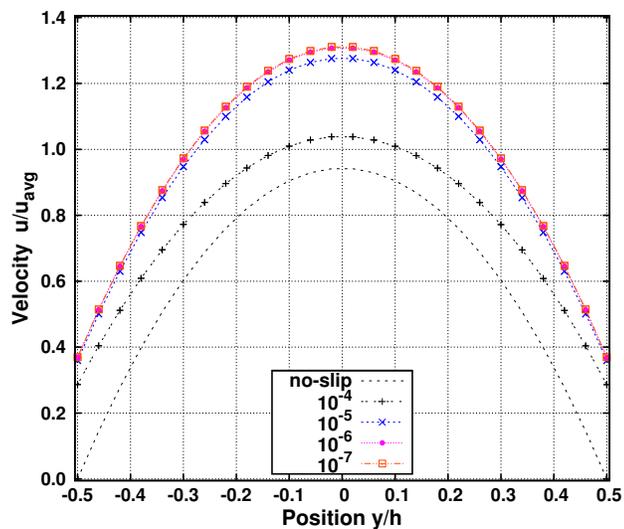
Fig. 5.5. Rectangular channel flow; velocity profiles at different convergence levels; all curves are normalised with the same average velocity from the $\Delta_m = 10^{-7}$ level; the initial guess (no-slip) is plotted with a line only; the subsequent iteration levels are marked with point-lines; curves of convergence level $\Delta_m = 10^{-6}$ and $\Delta_m = 10^{-7}$ lie one over another.
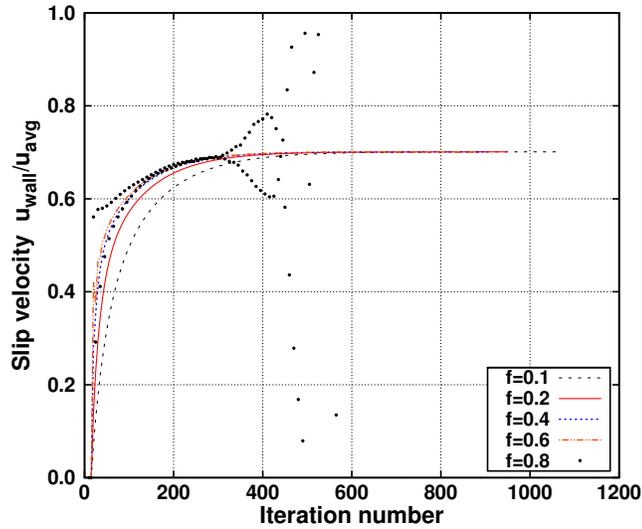
Fig. 5.6. Rectangular channel flow; normalised wall velocity in the function of iteration number for *Knudsen* number $Kn = 0.1$; all values are normalised with the average velocity $\boldsymbol{u}_{\mathrm{avg}}$ of the $f = 0.1$ case result; the value of f represents different under-relaxation coefficients; it can be seen that all under-relaxation coefficients except $f = 0.8$ lead to convergent solutions.
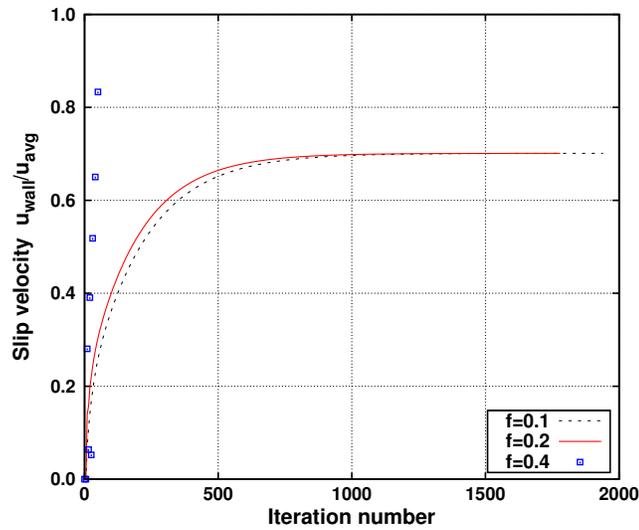
Fig. 5.7. Rectangular channel flow; normalised wall velocity in the function of iteration number for *Knudsen* number $Kn = 0.4$; all values are normalised with the average velocity $\boldsymbol{u}_{\mathrm{avg}}$ of the $f = 0.1$ case result; the value of f represents different under-relaxation coefficients; for under-relaxation coefficient $f = 0.4$ or higher solutions do not converge.

6

Conclusions

The boundary conditions for gas flows operating in the slip flow regime (see figure 6.1) have been implemented for usage with the finite volume *Navier Stokes* equations solver Fluent. The velocity slip and temperature jump boundary conditions include the formulations of *Maxwell*, *Smoluchowski* as well as an extended one that accounts for surface curvature.

Simulations on a number of test cases were performed in order to validate the implemenation by comparison with results of reference cases. Among the validation cases is the flow through a channel with constant rectangular cross section. For the no-slip flow regime the channel flow driven by pressure differences between inlet and outlet is a *Poiseuille* flow with parabolic velocity profile. In the slip flow regime the velocity profiles are flattened and have the non-zero wall velocity. Profiles of the velocity were compared at distinct distances away from the inlet plane. Furthermore the dependence of certain flow variable values in the function of stream-wise position was observed. All obtained results were in good agreement with the chosen reference test cases. Additionally this case setup served to estimate the errors caused by the chosen discretisation scheme and the convergence criteria. As a result it could be shown that the chosen target variables approach asymptotic values as the computational grid is refined. Also the convergence level used for the entire range of mentioned test cases proved to be sufficient. The necessity for the implemented under-relaxation factor was demonstrated for two different *Knudsen* number configurations.

A second validation case was chosen to check slip flow effects in curved geometries. Two concentric cylinder surfaces with a *Couette* shear flow between them provided such a test case. Furthermore the influence of the tangential momentum accommodation coefficient was studied. The resulting radial profiles of the circumferential velocity agreed with the analytically derived functions values of the chosen case reference.

As a third case *Knudsen* pump was chosen to validate the temperature jump boundary condition together with thermal creep flow. Thermal creep is the wall slip velocity component induced by wall-tangential temperature gradients. It is

the driving component in a *Knudsen* pump. Such a setup, following a reference case, was imitated with a configuration of a channel that connects two tanks heated at different temperatures. The whole transient process beginning with pumping from cold to hot tank and the final steady state with backflow in the channel centre was simulated. Result comparisons of the steady state pressure distribution showed significant differences especially for the higher *Knudsen* number. This might be caused by missing information about the *Knudsen* number definition in the reference case.
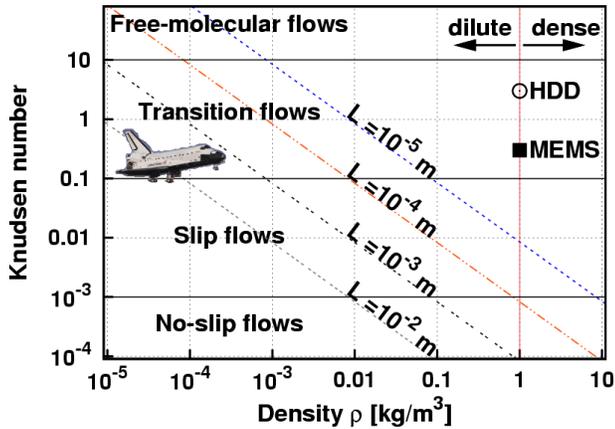
Fig. 6.1. Overview of flow regimes; the *Knudsen* number is shown in the function of density ρ and characteristic length scale L; flow situations for space shuttle on re-entry, flow between disk and head in a hard disk drive (HDD) and other micro-electro-mechanical systems (MEMS) are indicated.

Additional verification cases demonstrated several other aspects. A wall-bounded box was used to show the temperature distribution with temperature jump applied. In a bended channel surface curvature was further taken into account. A vortex flow pattern between cylinder surfaces that were placed co-axially was observed as well. It is found in different references, but quantitative information is missing. Thus only the qualitative behaviour was found to agree with them.

Experimental verification for the chosen reference cases remain to be performed, because all of them are either numerical simulations or an analytical approach. The implementation of the complete boundary condition allows complex flow configurations for the gas slip flow regime such as micro-electronic-mechanical systems (MEMS). In the future the developed code will be used for simulations of chip cooling processes and slip flow in micro bearings.

<div align="right">

A

</div>

Computational Routine

A.1. Slip Velocity Routine

The slip velocity is computed using equation (1.5). The influence of sur-
face curvature and arbitrarily rotated domains is included. For clarity only one
out of three velocity vector coordinate routines is shown here. Sections differ-
ing between the three coordinate routines are indicated as appropriate within
comment lines. Note that the lines preceding the slip velocity routine itself are
needed for all the routines mentioned here.

```
/****************************************************************
 Slip_Thermal_BC.c
 UDF - User Defined Function
 for slip velocity(Maxwell) + temperature jump(Smoluchowski)
 ****************************************************************/

#include "udf.h"    /* must be at the beginning of every UDF */
#include "sg.h"     /* must be at the beginning of this UDF */
#include "math.h"   /* must be at the beginning of this UDF */

/*Set preprocessor directives here in order to change
behaviour of the routine:*/

#define not SCHEMEMFP /* shall MeanFreePath be scheme var.? */
#define SCHEMETMAC  /* shall TMAC ... */
#define SCHEMEUDRLXCOEFF   /* shall the under-relaxation
                             coefficient... */
#define not SCHEMEThAC  /* shall ThAC ... */
#define not SCHEMESpHR  /* shall SpHR ... */
#define SCHEMESIGMASQUARE  /* shall sigma-square value... */
#define SCHEMEAMBPRESS  /* shall ambient pressure value... */
```

```
#define UNDERRLX    /* shall slip veloc be under-relaxated in
                       severe cases? */
#define WALLMOTION   /* shall wall-motion be regarded? */
/*
If the scheme variables are not used fall back to some default
values as below, also some constants are defined
*/
#ifndef SCHEMESIGMASQUARE
#define sigma_square 1.35305239e-19  /* squared value of
                                       sigma(molecule diameter) */
#endif


#ifndef SCHEMEAMBPRESS
#define ambpress     101325  /* ambient pressure */
#endif


#ifndef SCHEMETMAC
#define TMAC         1.0   /* tangential momentum accomodation
                             coefficient */
#endif


#ifndef SCHEMEThAC
#define ThAC         1.0   /* thermal accomodation coefficient */
#endif


#ifndef SCHEMESpHR
#define SpHR         1.4   /* specific heat ratio; Air, Oxygen,
                             Nitrogen */
#endif


#ifndef SCHEMEUDRLXCOEFF
#define UDRLXCOEFF   0.02    /* under-relaxation coefficient */
#endif


#define Boltzmann    1.3806505e-23    /* Boltzmann constant */
#define PI           3.14159265358979323846 /* number pi */
#define SQRT_2       1.41421356237309504880 /* sqrt(2) */
```

```
/*
   =======================================
      Velocity slip at wall boundaries,
      a separate routine for every velocity coordinate
      this is x-coordinate routine
      with thermal creep term
   =======================================
*/
DEFINE_PROFILE(maxwell_slip_velocity_x_full,f_thread,index)
{

#ifdef  SCHEMETMAC
        real TMAC=1;
        TMAC=RP_Get_Real("tmac");
#endif


        real MeanFreePath=6.8e-8;
#ifdef  SCHEMEMFP
        MeanFreePath=RP_Get_Real("meanfreepath");
#endif


#ifdef  SCHEMEUDRLXCOEFF
        real UDRLXCOEFF=0.2;
        UDRLXCOEFF=RP_Get_Real("udrlxcoeff");
#endif


#ifdef  SCHEMESIGMASQUARE
        real sigma_square=1.0e-19;
        sigma_square=RP_Get_Real("sigmasquare");
#endif


#ifdef  SCHEMEAMBPRESS
        real ambpress;
        ambpress=RP_Get_Real("ambpress");
#endif


        face_t face;

        cell_t cell;
```

```
        Thread *c_thread;

        real slip, thcreep, dveloc;
        real normal_slip, tangential_slip, tangential_thcreep;
        real Coeff1[ND_ND], Coeff2[ND_ND];

        real u[ND_ND];

/* call the routine that computes transformation coefficients,
   once per iteration is sufficient
   coord_coeff(f_thread) is called only in the velocity
   $x$-coordinate routine and activated by the coordeval
   scheme variable(integer) */
        if (RP_Get_Integer("coordeval")==1) coord_coeff(f_thread)

        if ((RP_Get_Integer("tempgradeval")>1)&&(Data_Valid_P()))
{
begin_f_loop(face,f_thread)
        {

/* get cell and cell thread pointer */
                cell=F_C0(face,f_thread);
                c_thread=THREAD_T0(f_thread);

/* compute mean free path at every position */
#ifndef  SCHEMEMFP
                MeanFreePath = Boltzmann * F_T(face,f_thread) / (
                    sigma_square * (F_P(face,f_thread)+ambpress)*
                    PI * SQRT_2 ) ;
#endif

/* save the velocity coordinates into u[ND_ND] */
                ND_SET(u[0],u[1],u[2],
                    F_U(face,f_thread),
                    F_V(face,f_thread),
                    F_W(face,f_thread)) ;
/* save the transformation coefficients c_Mm:
   c_11, c_21, c_31 into Coeff1[] */
                ND_SET(Coeff1[0],Coeff1[1],Coeff1[2],
```

```
                         F_UDMI(face,f_thread,0),
                         F_UDMI(face,f_thread,1),
                         F_UDMI(face,f_thread,2)) ;
/* save the transformation coefficients c_Mm:
   c_12, c_22, c_32 into Coeff2[] */
                  ND_SET(Coeff2[0],Coeff2[1],Coeff2[2],
                         F_UDMI(face,f_thread,3),
                         F_UDMI(face,f_thread,4),
                         F_UDMI(face,f_thread,5)) ;


/* evaluate the du/dl (tangential to surface) term in the
   local coord. system */
                  tangential_slip=NVD_DOT(Coeff1,
                      NV_DOT(Coeff1,C_U_G(cell,c_thread)),
                      NV_DOT(Coeff1,C_V_G(cell,c_thread)),
                      NV_DOT(Coeff1,C_W_G(cell,c_thread))) ;


/* evaluate the du/dy (normal to surface) term in the
   local coord. system */
                  normal_slip=-1 * NVD_DOT(Coeff1,
                      NV_DOT(Coeff2,C_U_G(cell,c_thread)),
                      NV_DOT(Coeff2,C_V_G(cell,c_thread)),
                      NV_DOT(Coeff2,C_W_G(cell,c_thread))) ;


/* add theses values and multiply with MFP and TMAC */
                  slip = ((2-TMAC)/TMAC) * MeanFreePath *
                      (tangential_slip+normal_slip) ;


/* evaluate the dT/dl (tangential to surface) term in the
   local coord. system */
                  tangential_thcreep=NV_DOT(Coeff1,
                      C_T_G(cell,c_thread)) ;
                  thcreep = 0.75 * C_MU_L(cell,c_thread)/
                      (C_R(cell,c_thread) *
                      C_T(cell,c_thread)) * tangential_thcreep ;


#ifdef   WALLMOTION
                  dveloc = Coeff1[0]* (F_UDSI(face,f_thread,1) +
                      (slip + thcreep)) ;
```

```
/* in order to set the wall motion velocity a user defined scalar
   variable F_UDSI(...,1) has to be created (its transport is not
   important and won't effect any result) */
#else
                dveloc = Coeff1[0]* (slip + thcreep);
/*                      ^--- transforms back to Fluent's coord.
                        system thus delivers only the velocity
                        component for x-direction */
#endif
/* the index '0' in Coeff1[..] is replaced by 0,1,2 for x, y, z;
   Underrelaxation is strictly recommended for Knudsen# at the
   upper bounds of the slip velocity regime
   thus the preprocessor directive is activated here */

#ifdef   UNDERRLX
                dveloc =  (1-UDRLXCOEFF) *u[0] + UDRLXCOEFF *
                    dveloc;
#endif

/* boundary condition value is returned */
                F_PROFILE(face,f_thread,index) = dveloc;
        }
        end_f_loop(face,f_thread)
        else {
        begin_f_loop(face,f_thread)
        {
/* save the velocity coordinates into u[ND_ND] */
                ND_SET(u[0],u[1],u[2],
                    F_U(face,f_thread),
                    F_V(face,f_thread),
                    F_W(face,f_thread)) ;
/* save the transf. coefficients into Coeff1[ND_ND] */
                ND_SET(Coeff1[0],Coeff1[1],Coeff1[2],
                    F_UDMI(face,f_thread,0),
                    F_UDMI(face,f_thread,1),
                    F_UDMI(face,f_thread,2)) ;
#ifdef   WALLMOTION
                F_PROFILE(face,f_thread,index) = Coeff1[0] *
                    F_UDSI(face,f_thread,1);
```

```
#else
                F_PROFILE(face,f_thread,index) = Coeff1[0] *
                    NV_DOT(u,Coeff1);
#endif
        }
        end_f_loop(face,f_thread)
        }
}
```

A.2. Second Order Slip Velocity Routine

In a similar way to the routine before the slip velocity condition including second order terms has been implemented.

The fluid solver provides no access to second order gradient terms. Those terms are computed using a simple forward difference implementation for the temperature gradient. Density gradients are only available in the so-called density-based solver. That solver computes the momentum and mass conservation equations in a coupled way. It imposes additional practical difficulties during the usage. Instead the segregated solver is in use here which requires that the first and second spatial density derivatives have to be computed from scratch. In this sense the gradient values in surrounding cells are used inside a central difference scheme. Note that at boundaries a forward or backward difference scheme respectively has to be retained.

Additionally the evaluation is carried out only for the x-velocity coordinate and then copied into the user-defined memory location UDM(...,6). In the y and z routine this value is only read in and further processed.

```
/*
  ======================================
    Velocity slip at wall boundaries,
    a separate routine for every velocity coordinate
    this is x-coordinate
    2nd order expressions included
  ======================================
*/
DEFINE_PROFILE(maxwell_slip_velocity_x_2nd,f_thread,index)
{
#ifdef  SCHEMETMAC
```

```
        real TMAC=1;
        TMAC=RP_Get_Real("tmac");
#endif


        real MeanFreePath=6.8e-8;
#ifdef  SCHEMEMFP
        MeanFreePath=RP_Get_Real("meanfreepath");
#endif


#ifdef  SCHEMEUDRLXCOEFF
        real UDRLXCOEFF=0.2;
        UDRLXCOEFF=RP_Get_Real("udrlxcoeff");
#endif


#ifdef  SCHEMESIGMASQUARE
        real sigma_square;
        sigma_square=RP_Get_Real("sigmasquare");
#endif


#ifdef  SCHEMEAMBPRESS
        real ambpress;
        ambpress=RP_Get_Real("ambpress");
#endif


        face_t face, nface;

        cell_t cell, ncell;
        Thread *c_thread, *nc_thread, *nf_thread;

        real slip, thcreep, dveloc;
        real normal_slip, tangential_slip, tangential_thcreep;
        real tangential_tempgrad_b, tangential_tempgrad_n;
        real tempgrad_2ndorder, slip_tempgrad_2ndorder;
        real mu_rho_T;
        real Coeff1[ND_ND], Coeff2[ND_ND];
        real u[ND_ND], du[ND_ND];

        real A[ND_ND];
        real dr0[ND_ND], es[ND_ND], ds, A_by_es;
```

```
        real nA[ND_ND];
        real ndr0[ND_ND], ndr1[ND_ND], nes[ND_ND], nds, nA_by_es;
        real sclr1, mag_a, a;


        if (RP_Get_Integer("coordeval")==1) {
                coord_coeff(f_thread);
        }

        if (N_UDM < UDM_NUMBER) Error("Not enough user-defined
                    memory variable locations available!!");

        if ((RP_Get_Integer("tempgradeval")>1)&&(Data_Valid_P()))
        {
        begin_f_loop(face,f_thread)
        {
                cell=F_C0(face,f_thread);
                c_thread=THREAD_T0(f_thread);

#ifndef  SCHEMEMFP
                MeanFreePath = Boltzmann * F_T(face,f_thread) / (
                    sigma_square * (F_P(face,f_thread)+ambpress)*
                    PI * SQRT_2 ) ;
#endif

                ND_SET(du[0],du[1],du[2],
                    C_U_G(cell,c_thread)[0],
                    C_U_G(cell,c_thread)[1],
                    C_U_G(cell,c_thread)[2]);
                ND_SET(u[0],u[1],u[2],
                    F_U(face,f_thread),
                    F_V(face,f_thread),
                    F_W(face,f_thread)) ;
                a=NV_MAG(u);

                ND_SET(Coeff1[0],Coeff1[1],Coeff1[2],
                    F_UDMI(face,f_thread,0),
                    F_UDMI(face,f_thread,1),
                    F_UDMI(face,f_thread,2)) ;
```

```
            ND_SET(Coeff2[0],Coeff2[1],Coeff2[2],
                F_UDMI(face,f_thread,3),
                F_UDMI(face,f_thread,4),
                F_UDMI(face,f_thread,5)) ;


/* evaluate du/dl (surface tang.) term in local coord. system */
            tangential_slip=NVD_DOT(Coeff1,
                NV_DOT(Coeff1,du),
                NV_DOT(Coeff1,C_V_G(cell,c_thread)),
                NV_DOT(Coeff1,C_W_G(cell,c_thread))) ;


/* evaluate du/dy (surface normal) term in local coord. system */
            normal_slip=-1 * NVD_DOT(Coeff1,
                NV_DOT(Coeff2,du),
                NV_DOT(Coeff2,C_V_G(cell,c_thread)),
                NV_DOT(Coeff2,C_W_G(cell,c_thread))) ;


            BOUNDARY_FACE_GEOMETRY
                (face,f_thread,A,ds,es,A_by_es,dr0) ;


            n=0;
            mag_a=NV_MAG(A); /* normal vector's magnitude */


            c_face_loop(cell, c_thread, n)
            {
                    nface = C_FACE(cell,c_thread,n);
                    nf_thread = C_FACE_THREAD(cell,c_thread,n);

                    BOUNDARY_FACE_GEOMETRY(nface,nf_thread,
                        nA,nds,nes,nA_by_es,ndr0);
                    sclr1 = NV_DOT(A,nA)/(mag_a*NV_MAG(nA));
                    /* find out if the face is the opposite to
                        the boundary face */
                    if (((sclr1*sclr1)>0.85)&&((nface!=face)
                        || (nf_thread!=f_thread))) {
                    /* tangential term in very same cell */
                            ncell=F_C0(nface,nf_thread);
                            nc_thread=THREAD_T0(nf_thread);
                            if (sclr1<0)
```

```
                                        {
                        /* tangential term in cell's neighbour */
                                ncell=F_C1(nface,nf_thread);
                                nc_thread=THREAD_T1(nf_thread);
                                }
                                INTERIOR_FACE_GEOMETRY(nface,
                                        nf_thread,nA,nds,nes,nA_by_es,
                                        ndr0,ndr1);
                        }
                }

/* tangential term in the cell at the boundary */
                tangential_tempgrad_n=NV_DOT(Coeff1,
                        C_T_RG(ncell,nc_thread));
/* tangential term in the cell next to it */
                tangential_tempgrad_b=NV_DOT(Coeff1,
                        C_T_RG(cell ,c_thread ));
                tempgrad_2ndorder=tangential_tempgrad_n-
                        tangential_tempgrad_b;
/* 2nd derivative of temperature */
                tempgrad_2ndorder/=nds;

                mu_rho_T = C_MU_L(cell,c_thread)/
                        (C_R(cell,c_thread) * C_T(cell,c_thread)) ;
                slip_tempgrad_2ndorder = tempgrad_2ndorder *
                        mu_rho_T ;
/* add the values and multiply with MFP and TMAC */
                slip = tangential_slip + normal_slip -
                        slip_tempgrad_2ndorder ;
                slip *= ((2-TMAC)/TMAC) * MeanFreePath ;

/* evaluate the dT/dl [(tangential to surface),thermal creep]
   term in the local coord. system */
                tangential_thcreep=NV_DOT(Coeff1,
                        C_T_G(cell,c_thread)) ;
                thcreep = 0.75 * mu_rho_T * tangential_thcreep;

#ifdef   WALLMOTION
                dveloc = F_UDSI(face,f_thread,1) + slip +thcreep;
```

```
#else
               dveloc = slip + thcreep;
#endif
               C_UDMI(cell,c_thread,6) = dveloc;

               dveloc *= Coeff1[0];

#ifdef   UNDERRLX
               dveloc = (1-UDRLXCOEFF)*u[0] + UDRLXCOEFF*dveloc;
#endif

               F_PROFILE(face,f_thread,index) = dveloc;

       }
       end_f_loop(face,f_thread)
/* if the gradients are not known, set only the boundary value
   that is already known */
       }
       else {
       begin_f_loop(face,f_thread)
       {
               ND_SET(u[0],u[1],u[2],
                   F_U(face,f_thread),
                   F_V(face,f_thread),
                   F_W(face,f_thread)) ;
               ND_SET(Coeff1[0],Coeff1[1],Coeff1[2],
                   F_UDMI(face,f_thread,0),
                   F_UDMI(face,f_thread,1),
                   F_UDMI(face,f_thread,2)) ;
#ifdef   WALLMOTION
               F_PROFILE(face,f_thread,index) = Coeff1[0] *
                   F_UDSI(face,f_thread,1) ;
#else
               F_PROFILE(face,f_thread,index) = Coeff1[0] *
                   NV_DOT(u,Coeff1) ;
#endif
       }
       end_f_loop(face,f_thread)
       }
```

```
}
```

A.3. Temperature Jump Boundary Condition Routine

The implementation of the temperature jump boundary condition is made as follows:

```
/*
  =====================================
   Temperature change at wall boundaries
  =====================================
*/
DEFINE_PROFILE(temperature_jump,f_thread,index)
{

        real MeanFreePath;
#ifdef  SCHEMEMFP
        MeanFreePath=RP_Get_Real("meanfreepath");
#endif

#ifdef  SCHEMEThAC
        real ThAC=1;
        ThAC=RP_Get_Real("thac");
#endif

#ifdef  SCHEMEUDRLXCOEFF
        real UDRLXCOEFF=0.2;
        UDRLXCOEFF=RP_Get_Real("udrlxcoeff");
#endif

#ifdef  SCHEMESpHR
        real SpHR;
        SpHR=RP_Get_Real("sphr");
#endif

        face_t face;

        cell_t cell;
```

```
        Thread *c_thread;

        real Coeff2[ND_ND];

        real Prandtl, gamma, temp, normal;

/* call the routine that computes the transformation
   coefficients, once per iteration should be enough */
        coord_coeff(f_thread);

/* loops over all faces in the thread passed in the DEFINE
   macro argument */
        begin_f_loop(face,f_thread)
        {

                cell=F_C0(face,f_thread);
                c_thread=THREAD_T0(f_thread);

#ifndef  SCHEMEMFP
                MeanFreePath = Boltzmann * F_T(face,f_thread) / (
                    sigma_square * (F_P(face,f_thread)+ambpress)*
                    PI * SQRT_2 ) ;
#endif

                Prandtl = (C_MU_L(cell,c_thread)*
                    C_CP(cell,c_thread))/C_K_L(cell,c_thread) ;

                ND_SET(Coeff2[0],Coeff2[1],Coeff2[2],
                    F_UDMI(face,f_thread,3),
                    F_UDMI(face,f_thread,4),
                    F_UDMI(face,f_thread,5)) ;

/* evaluate the dT/dn (normal to surface) term in the
   local coord. system */
                normal=NV_DOT(Coeff2,C_T_G(cell,c_thread));

                gamma=(2*SpHR)/(SpHR+1);
                temp=((2-ThAC)/ThAC) * gamma *
                    MeanFreePath/Prandtl * normal ;
```

```
/* in order to set the wall temperature a user defined scalar
   variable F_UDSI(...,0) has to be created (its transport is not
   important and won't effect the result) */
#ifdef   UNDERRLX
                F_PROFILE(face,f_thread,index) =
                    F_T(face,f_thread)*(1-UDRLXCOEFF) +
                    (F_UDSI(face,f_thread,0)+temp)*UDRLXCOEFF ;
#else
                F_PROFILE(face,f_thread,index) =
                    (F_UDSI(face,f_thread,0)+temp) ;
#endif


        }
        end_f_loop(f,thread)
}
```

A.4. Computation of Transformation Coefficients

In the following paragraph the routine for computation of transformation coefficients is documented. It performs the transformation between the solver's coordinate frame and the local coordinate system. It seems reasonable to use Fluent's intrinsic vector handling routines in order to fulfil this requirement. The velocity vector coordinates are assigned to u[ND_ND] for better handling and the F_..() macros only once to be accessed.

```
void coord_coeff(Thread *f_thread)
{
        face_t face;
        cell_t cell;
        Thread *c_thread;
        real y[ND_ND];
        real u[ND_ND], a;

        real A[ND_ND];
        real dr0[ND_ND], es[ND_ND], ds, A_by_es;

        begin_f_loop(face,f_thread)
        {
                F_CENTROID(y,face,f_thread);
```

```
                cell=F_C0(face,f_thread);
                c_thread=THREAD_T0(f_thread);

                BOUNDARY_FACE_GEOMETRY(
                    face,f_thread,A,ds,es,A_by_es,dr0) ;

                ND_SET(u[0],u[1],u[2],
                    C_U(cell,c_thread),
                    C_V(cell,c_thread),
                    C_W(cell,c_thread)) ;
                a=NV_MAG(u);

                ND_SET(F_UDMI(face,f_thread,0),
                    F_UDMI(face,f_thread,1),
                    F_UDMI(face,f_thread,2),
                    NVD_DOT(u,1,0,0)/a,
                    NVD_DOT(u,0,1,0)/a,
                    NVD_DOT(u,0,0,1)/a) ;
/* c_mM -> M...fixed Fluent coord. system, m...local
        wall surface coord. system */
                /* c_11 -> 0 */
                /* c_12 -> 1 */
                /* c_13 -> 2 */

                ND_SET(F_UDMI(face,f_thread,3),
                    F_UDMI(face,f_thread,4),
                    F_UDMI(face,f_thread,5),
                    NVD_DOT(A, 1, 0, 0)/A_by_es,
                    NVD_DOT(A, 0, 1, 0)/A_by_es,
                    NVD_DOT(A, 0, 0, 1)/A_by_es) ;
                /* c_21 -> 3 */
                /* c_22 -> 4 */
                /* c_23 -> 5 */
        }
        end_f_loop(face,f_thread)
}
```

Bibliography

1. Agrawal, A., Agrawal, A. "Three-dimensional simulation of gaseous slip flow in different aspect ratio microducts." Phys.Fluids vol. 18, 103604, 2006.

2. Arkilic, E. B. "Measurement of the Mass Flow and Tangential Momentum Accomodation Coefficient in Silicon Micromachined Channels." Ph.D. thesis, Massachusatts Institute of Technology, Cambrigde, Massachusatts, 1997.

3. Beskok, A., "Thermal Creep Flows."
   http://www.cfm.brown.edu/people/beskok/creep.html, retrieved 05-22-2007

4. Beskok, A., Karniadakis, G. E. "A Model for Flows in Channels, Pipes, and Ducts at Micro and Nano scale." Microscale Thermophys. Eng. vol. 3, 1999: 43-77

5. Beskok, A., Karniadakis, G. E., Trimmer, W. "Rarefaction and Compressibility Effects in Gas Microflows." J. Fluids Eng. vol. 118, 1996: 448-456

6. Chapman, S., Cowling, T.G. "The Mathematical Theory of Non-Uniform Gases." Cambridge University Press, 1960.

7. Einzel, D., Panzer, P., Liu, M. "Boundary condition for fluid flow: Curved or rough surfaces." Phys. Rev. Lett. vol. 64-19, 1990: 2269–2272

8. Fluent Inc. "Fluent 6.2 Manual.", 2005.

9. Gad-el-Hak, M.(ed.). "The MEMS Handbook." CRC Press. 2002. 7.12-7.27.

10. Hirschfelder, J.O., Curtiss, C.F., Bird, R.B. "Molecular Theory of Gases and Liquids." John Wiley & Sons, Inc., New York, London, 1964.

11. Lockerby, D. A., Reese, J. M., Emerson, D. R., Barber, R. W. "Velocity boundary condition at solid walls in rarefied gas calculations." Phys. Rev. E vol. 70, 017303, 2004.

12. Maxwell, J. C., "On stresses in rarified gases arising from inequalities of temperature.", Philos. Trans. R. Soc. London vol. 170, 1879: 231-256

13. Ohwada, T., Sone, Y., Aoki, K. "Numerical analysis of the shear and thermal creep flows of rarified gas over a plane wall on the basis of the linearized Boltzmann equation for hard-sphere molecules." Phys. Fluids A. vol. 1-9, 1989: 1588-1599

14. Ohwada, T., Sone, Y., Aoki, K. "Numerical Analysis of the Poiseuille and Thermal Transpiration Flows between Two Parallel Plates on the Basis of the Boltzmann Equation for Hard Sphere Molecules." Phys. Fluids A. vol. 1-12, 1989: 2042-2049

15. Smoluchowski, M. "Über den Temperatursprung bei Wärmeleitung in Gasen." Akad. Wiss. Wien. CVII, 1898: 304-329

16. Sone, Y. "Flows Induced by Temperature Fields in a Rarefied Gas and their Ghost Effect on the Behavior of a Gas in the Continuum Limit." Annu. Rev. Fluid Mech vol. 32, 2000: 779-811

17. Tibbs, K.W., Baras, F., Garcia, A.L. "Anomalous flow profile due to the curvature effect on slip length." Phys. Rev. E vol. 56-2, 1997: 2282-2283