Implementation and Validation of Wall Slip Velocity and Temperature Jump Boundary Conditions for Gas Flow in Microdevices

Steffen Jebauer

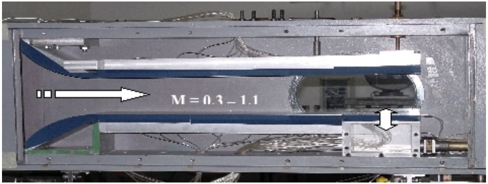
Department of Mechanics and Physics of Fluids
Institute of Fundamental Technological Research
Polish Academy of Sciences

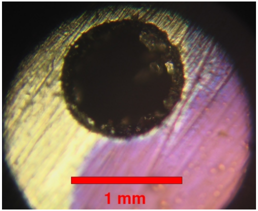
11.04.2007

Objective

Use *Navier Stokes* Flow Solver for Microdevices

Flow in drilled holes on airfoil



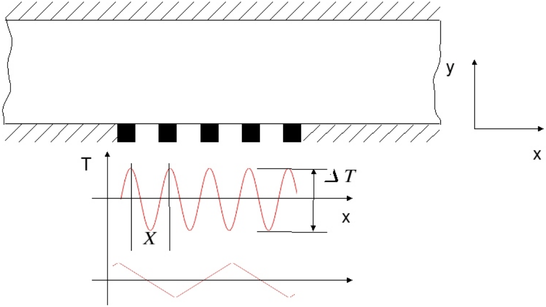


IMP PAN
Institute of Fluid Flow Machinery
Gdańsk
Prof. Doerffer

Objective

Use *Navier Stokes* Flow Solver for Microdevices

- Cooling of microfluidic devices



Outline

Knudsen Number

Mean free path λ

Definition

$$\lambda = \frac{k \cdot T}{\sqrt{2} \cdot \pi \cdot d^2 \cdot p}$$

$k \ldots$ Boltzmann constant
$d \ldots$ Molecular diameter

Knudsen number Kn

Classification of flows

$$Kn = \frac{\lambda}{L}$$

$L \ldots$ Length scale

Alternative length scale expression

$$L = \frac{\rho}{\left|\frac{\partial \rho}{\partial y}\right|}$$

Knudsen Number

Mean free path λ

Definition

$$\lambda = \frac{k \cdot T}{\sqrt{2} \cdot \pi \cdot d^2 \cdot p}$$

$k \ldots$ Boltzmann constant
$d \ldots$ Molecular diameter

Knudsen number *Kn*

Classification of flows

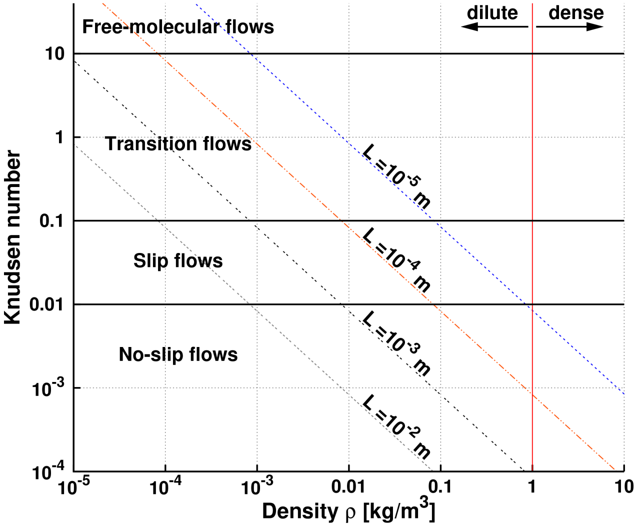
$$Kn = \frac{\lambda}{L}$$

$L \ldots$ Length scale

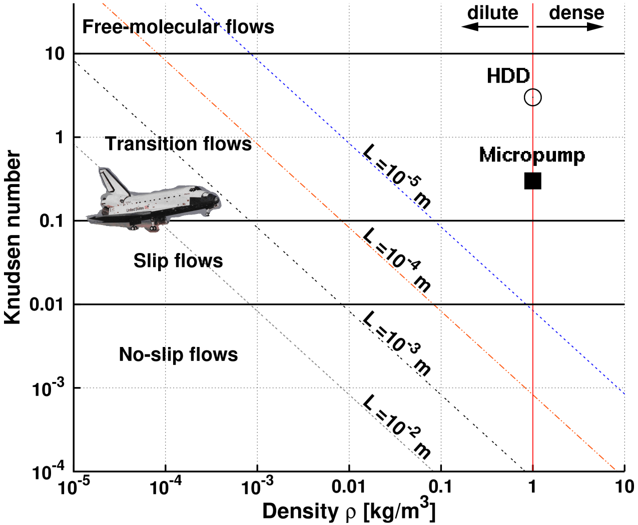
Alternative length scale expression

$$L = \frac{\rho}{\left|\frac{\partial \rho}{\partial y}\right|}$$

Air Flows - Classification



Air Flows - Classification



Maxwell Slip Velocity Condition

Equation

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

Symbols

σ_V... Tangential momentum accommodation coefficient (TMAC)

Maxwell, J.C. (1879)

Maxwell Slip Velocity Condition

Equation

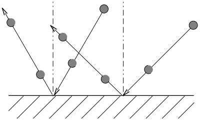
$$u_{fluid} - u_{wall} = \frac{2-\sigma_V}{\sigma_V} \lambda \frac{\partial u}{\partial y} + \frac{3}{4} \frac{\mu}{\rho T} \frac{\partial T}{\partial x}$$

Symbols

σ_V . . . Tangential momentum accommodation coefficient (TMAC)

Maxwell, J.C. (1879)

Tangential Momentum Accommodation Coefficient σ_V



- $\sigma_V = \dfrac{\text{particles contributing streamwise momentum}}{\text{total number of impinging particles}}$
- $\sigma_V = 1$ - diffuse reflection of particles
- $\sigma_V < 1$ - diffuse/specular reflection

Experimental observations:

$\sigma_V \approx 0.2$	$\sigma_V \approx 0.65 \ldots 0.96$	$\sigma_V \approx 0.75 \ldots 1$
Molecular beams on metal-coated surfaces	He, Ar, Kr on titanium-coated surfaces	Ar, N_2, CO_2 in silicon micro-channels
Steinheil et al. (1977)	Nakarjakov et al. (1994)	Arkilic (1997)

Smoluchowski Temperature Jump

Equation

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

$$Pr = \frac{\mu \cdot c_p}{k_L}$$

Symbols

$\sigma_T \ldots$ Thermal accommodation coefficient
$\gamma \ldots$ Specific heat ratio
$c_p \ldots$ Specific heat
$k_L \ldots$ Thermal conductivity

Smoluchowski, M. (1898)

Higher Order Expansion

Expansion

- Slip velocity, temperature gradient neglected:

$$u_{fluid} - u_{wall} = \frac{2-\sigma_V}{\sigma_V}\left[\lambda\frac{\partial u}{\partial y} + \frac{\lambda^2}{2}\frac{\partial^2 u}{\partial y^2} + \frac{\lambda^3}{6}\frac{\partial^3 u}{\partial y^3}\dots\right]$$

- Temperature jump:

$$T_{fluid} - T_{wall} = \frac{2-\sigma_T}{\sigma_T}\frac{2\gamma}{\gamma+1}\frac{1}{Pr}\left[\lambda\frac{\partial T}{\partial y} + \frac{\lambda^2}{2}\frac{\partial^2 T}{\partial y^2} + \frac{\lambda^3}{6}\frac{\partial^3 T}{\partial y^3}\dots\right]$$

- Generally higher order expressions not applied with *Navier Stokes* equations approach

Extended Slip Velocity Formulation

Curved surface influence

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

Thermal Stress Slip Flow Term

$$\frac{2-\sigma_v}{\sigma_v}\lambda\left[\frac{\mu}{\rho}\left(-\frac{1}{T}\frac{\partial^2 T}{\partial x \partial y}\right)\right]$$

Lockerby, D.A., Reese, J.M., Emerson, D.R., Barber, R.W. (2004)

Sone, Y. (2000)

Extended Slip Velocity Formulation

Curved surface influence

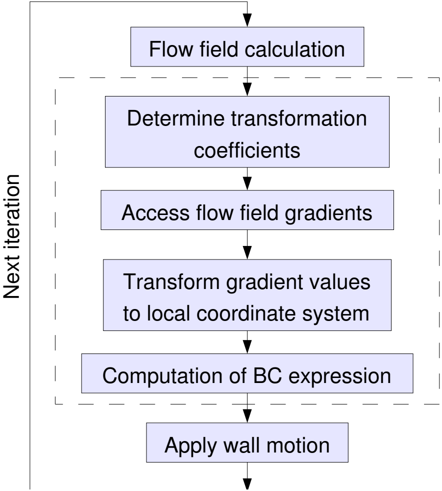
$$
\begin{aligned}
u_{fluid} - u_{wall} \quad = \quad & \frac{2-\sigma_V}{\sigma_V}\lambda\left[\frac{\partial u}{\partial y} + \frac{\partial u}{\partial x} + \right. \\
& \left. \frac{\mu}{\rho}\left(\frac{1}{\rho}\frac{\partial^2 \rho}{\partial x \partial y} - \frac{1}{T}\frac{\partial^2 T}{\partial x \partial y}\right)\right] + \frac{3}{4}\frac{\mu}{\rho T}\frac{\partial T}{\partial x}
\end{aligned}
$$

Thermal Stress Slip Flow Term

$$
\frac{2-\sigma_V}{\sigma_V}\lambda\left[\frac{\mu}{\rho}\left(-\frac{1}{T}\frac{\partial^2 T}{\partial x \partial y}\right)\right]
$$

Lockerby, D.A., Reese, J.M., Emerson, D.R., Barber, R.W. (2004)
Sone, Y. (2000)

Schematic Algorithm



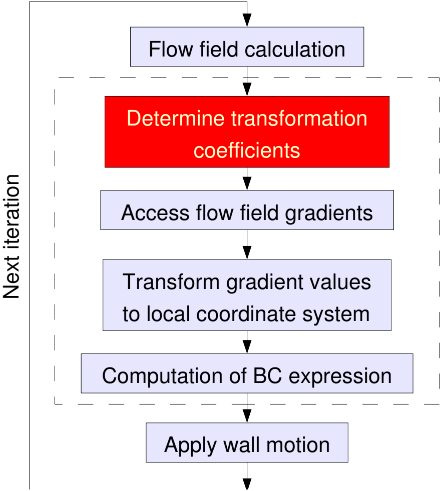
```
        .
ND_SET(F_UDMI(face,f_thread,0),
  F_UDMI(face,f_thread,1),
  F_UDMI(face,f_thread,2),
  NVD_DOT(u,1,0,0)/a,...
        .

        .
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))) ;
normal_slip=   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))) ;
        .

        .
slip = ((2-TMAC)/TMAC) * MeanFreePath *
  (tangential_slip+normal_slip) ;
        .

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

Schematic Algorithm



Next iteration

Flow field calculation

Determine transformation coefficients

Access flow field gradients

Transform gradient values to local coordinate system

Computation of BC expression

Apply wall motion

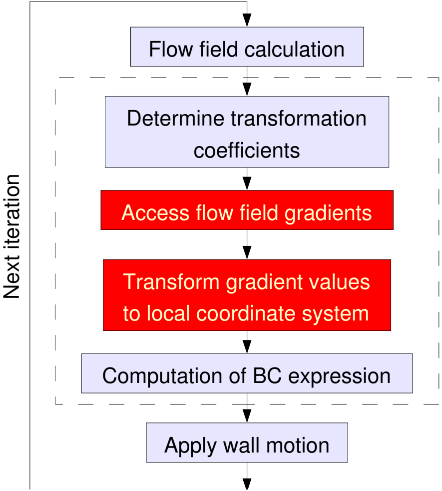
```
      .
ND_SET(F_UDMI(face,f_thread,0),
  F_UDMI(face,f_thread,1),
  F_UDMI(face,f_thread,2),
  NVD_DOT(u,1,0,0)/a,...
      .

      .
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))) ;
normal_slip=   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))) ;
      .

      .
slip = ((2-TMAC)/TMAC) * MeanFreePath *
  (tangential_slip+normal_slip) ;
      .

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

Schematic Algorithm



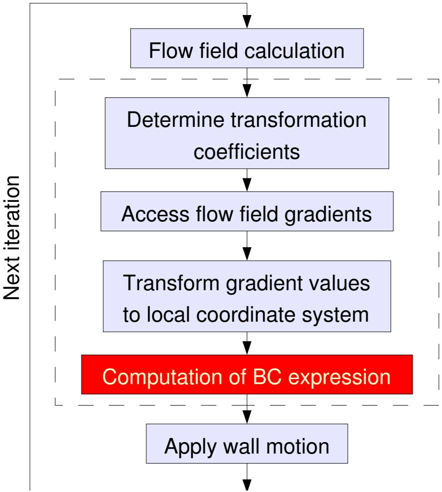
```
    .
ND_SET(F_UDMI(face,f_thread,0),
  F_UDMI(face,f_thread,1),
  F_UDMI(face,f_thread,2),
  NVD_DOT(u,1,0,0)/a,...
    .

    .
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))) ;
normal_slip=   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))) ;
    .

    .
slip = ((2-TMAC)/TMAC) * MeanFreePath *
  (tangential_slip+normal_slip) ;
    .

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

Schematic Algorithm



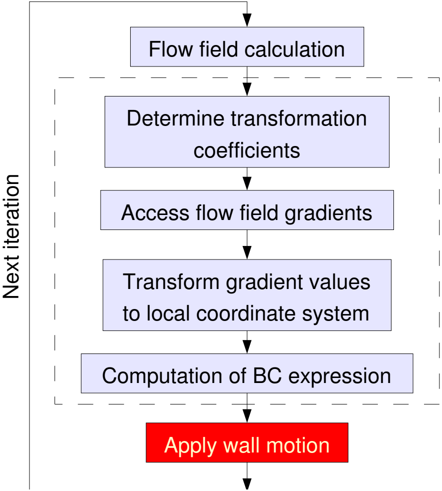
```
        .
ND_SET(F_UDMI(face,f_thread,0),
  F_UDMI(face,f_thread,1),
  F_UDMI(face,f_thread,2),
  NVD_DOT(u,1,0,0)/a,...
        .

        .
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))) ;
normal_slip=    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))) ;
        .

        .
slip = ((2-TMAC)/TMAC) * MeanFreePath *
  (tangential_slip+normal_slip) ;
        .

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

Schematic Algorithm



Next iteration

Flow field calculation

Determine transformation coefficients

Access flow field gradients

Transform gradient values to local coordinate system

Computation of BC expression

Apply wall motion

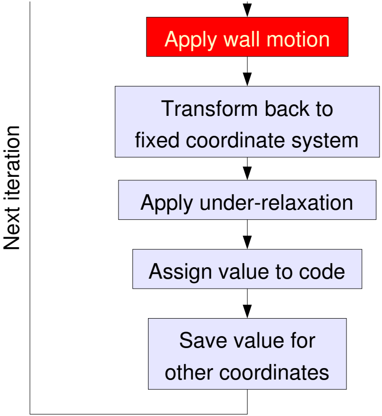
```
      .
ND_SET(F_UDMI(face,f_thread,0),
  F_UDMI(face,f_thread,1),
  F_UDMI(face,f_thread,2),
  NVD_DOT(u,1,0,0)/a,...
      .

      .
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))) ;
normal_slip=    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))) ;
      .

      .
slip = ((2-TMAC)/TMAC) * MeanFreePath *
  (tangential_slip+normal_slip) ;
      .

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

Schematic Algorithm (contd.)



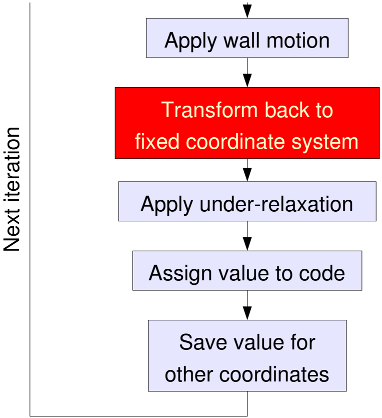
```
        .
dveloc *= Coeff1[0];
        .

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

        .
F_PROFILE(face,f_thread,index) = dveloc;
        .

        .
C_UDMI(cell,c_thread,6) = dveloc;
C_UDMI(cell,c_thread,7) = normal_slip;
        .
```

Schematic Algorithm (contd.)



Apply wall motion

Transform back to fixed coordinate system

Apply under-relaxation

Assign value to code

Save value for other coordinates

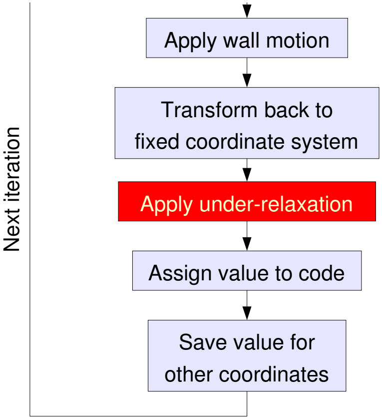
```
        .
dveloc *= Coeff1[0];
        .

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

        .
F_PROFILE(face,f_thread,index) = dveloc;
        .

        .
C_UDMI(cell,c_thread,6) = dveloc;
C_UDMI(cell,c_thread,7) = normal_slip;
        .
```

Schematic Algorithm (contd.)



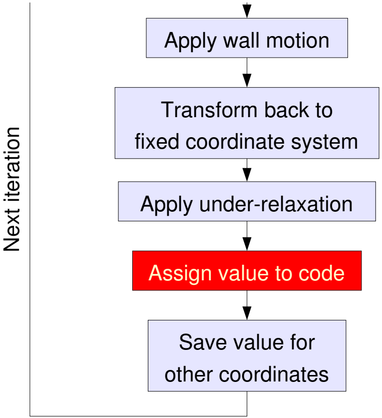
```
        .
dveloc *= Coeff1[0];
        .

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

        .
F_PROFILE(face,f_thread,index) = dveloc;
        .

        .
C_UDMI(cell,c_thread,6) = dveloc;
C_UDMI(cell,c_thread,7) = normal_slip;
        .
```

Schematic Algorithm (contd.)



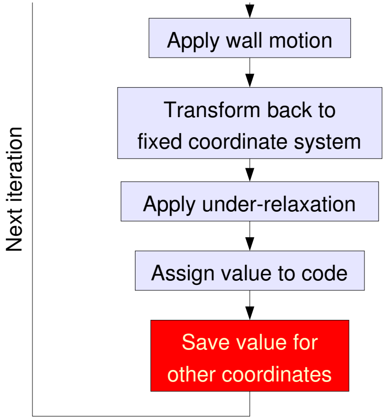
```
       .
dveloc *= Coeff1[0];
       .

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

       .
F_PROFILE(face,f_thread,index) = dveloc;
       .

       .
C_UDMI(cell,c_thread,6) = dveloc;
C_UDMI(cell,c_thread,7) = normal_slip;
       .
```

Schematic Algorithm (contd.)



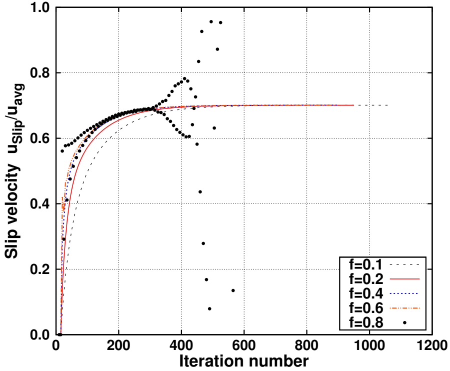
```
        .
dveloc *= Coeff1[0];
        .

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

        .
F_PROFILE(face,f_thread,index) = dveloc;
        .

        .
C_UDMI(cell,c_thread,6) = dveloc;
C_UDMI(cell,c_thread,7) = normal_slip;
        .
```

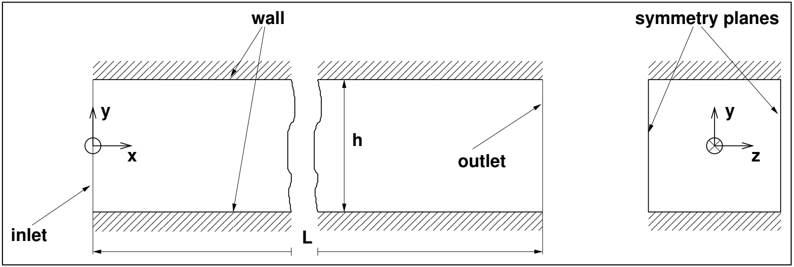
Under-relaxation factor *f*



- Application of under-relaxation:
  $u_k^n = (1-f)\cdot u_k^{n-1} + f\cdot u_k'^n$
- Case here - starting from initial guess
- Leading to final solution for most *f* values

Test Cases - Outline

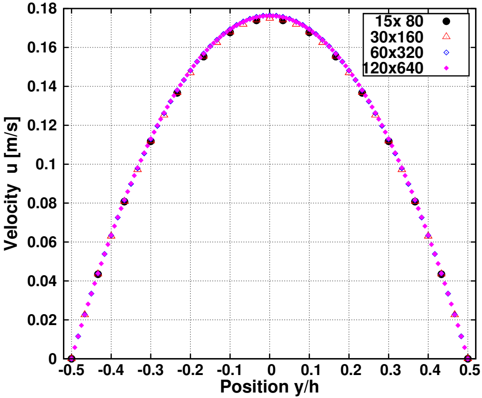
Test Case 1 - Setup and Boundary Conditions



- Laminar two-dimensional setup
- Ideal gas: air
- $h = 10^{-7} \dots 10^{-3} m$ (no-slip and slip flow regimes)
- Static pressure difference between inlet and outlet planes $\Delta p = 5\ Pa$
- *Slip velocity condition* applied at walls

Test Case 1 - Setup

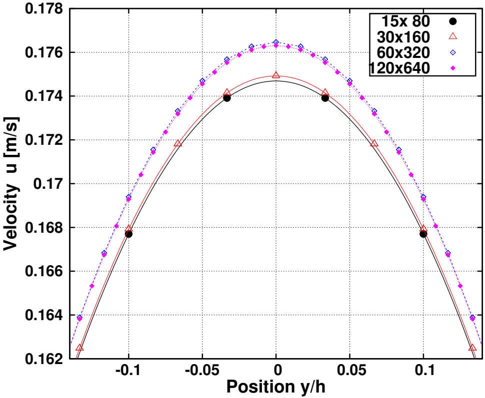
Grid Information



- Initial set of grids for no-slip flow tested
- Chosen grids are:
  - 25x125x1
  - 50x250x1
  - 100x500x1

Test Case 1 - Setup

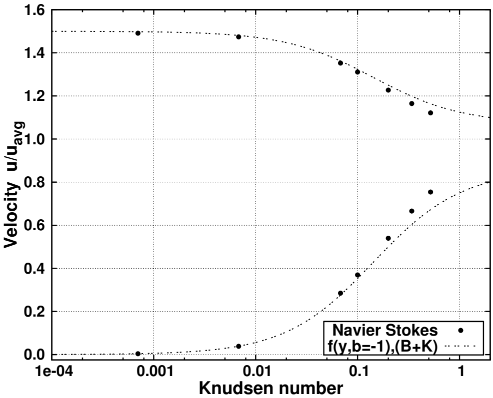
Grid Information



- Initial set of grids for no-slip flow tested
- Chosen grids are:
    - 25x125x1
    - 50x250x1
    - 100x500x1

Test Case 1 - Result

Knudsen Number versus Velocities



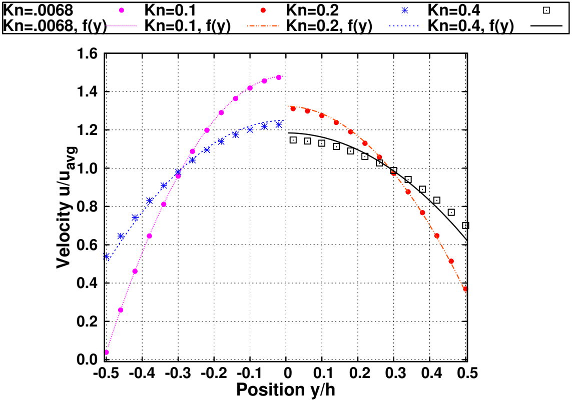
- Good agreement with reference(curve) up to $Kn \approx 0.1$

$$\frac{u(Kn,y)}{u_{avg}} = \frac{-(y/h)^2+y/h+\frac{Kn}{1-bKn}}{1/6+\frac{Kn}{1-bKn}}$$

Beskok, A., Karniadakis, G. E. (1999)

Test Case 1 - Result

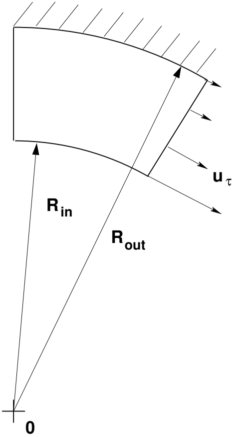
Channel Velocity Profiles



- Good agreement with reference(curves) up to $Kn \approx 0.1$

Beskok, A., Karniadakis, G. E. (1999)

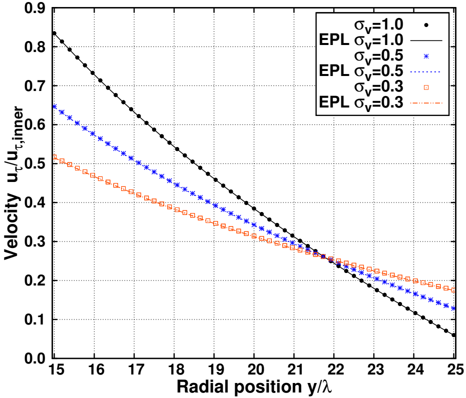
Test Case 2 - Setup and Boundary Conditions



- Laminar two-dimensional setup, symmetry planes in depth direction
- Periodicity in circumferential direction
- Ideal gas: air
- $R_{out} = 25\ \lambda$, $R_{in} = 15\ \lambda$, $\lambda(T = 300\ K) = 6.9 \cdot 10^{-7} m$
- *Slip velocity condition* with wall motion applied

Test Case 2 - Result

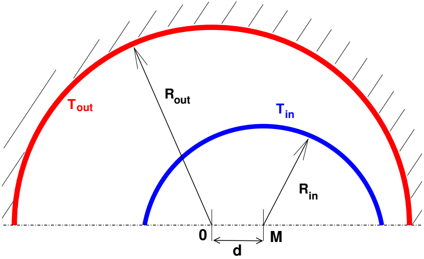
Radial Velocity distribution



- Radial position given as a multiple of mean free path λ
- Good agreement with analytical approach(curves)

Einzel, D., Panzer, P., Liu, M. (1990) [EPL]

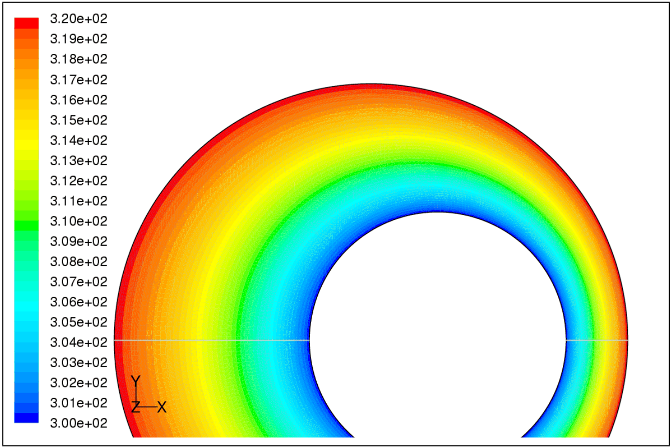
Test Case 3 - Setup and Boundary Conditions



- Two-dimensional setup, symmetry planes present
- Laminar case; ideal gas: air
- Temperature at walls: $T_{in} = 300\ K$, $T_{out} = 320\ K$
- $R_{out}/R_{in} = 2$, $R_{out} = 5 \cdot 10^{-4} m$, $d = 1.3 \cdot 10^{-4} m$
- *Extended slip velocity condition* and *temperature jump* applied at walls

Test Case 3 - Result

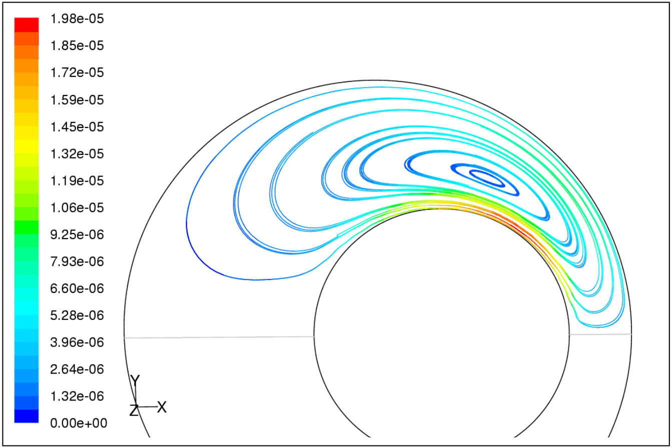
Temperature Field



- Values indicate temperature in [K]
- Upper part of domain shown

Test Case 3 - Result

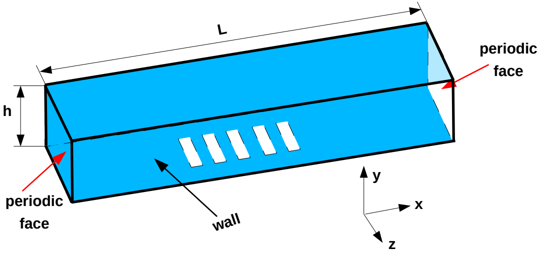
Path Lines



- Values indicate velocity magnitude in [m/s]
- Upper part of domain shown

Chip Cooling - Setup and Boundary Conditions

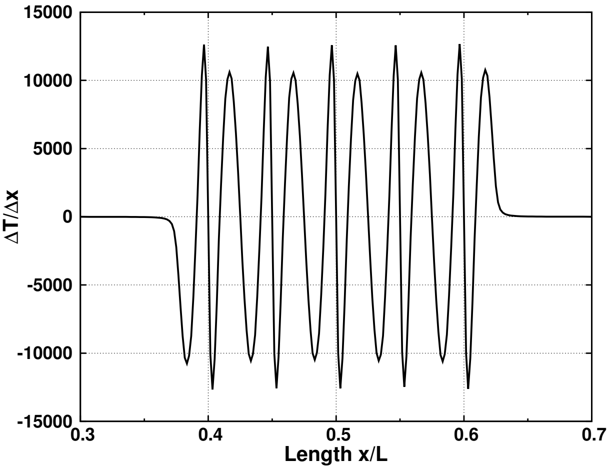
Domain



- Threedimensional laminar case, ideal gas: air
- Height h and depth 10^{-4} m, length $L = 2 \cdot 10^{-3}$ m
- Periodic pair inlet/outlet, pressure gradient 9 [Pa/m]
- *Slip velocity condition* and *temperature jump* conditions at lower y wall, T=300 K
- Temperature at plates 300 ± 1 K
- Other faces are symmetry planes

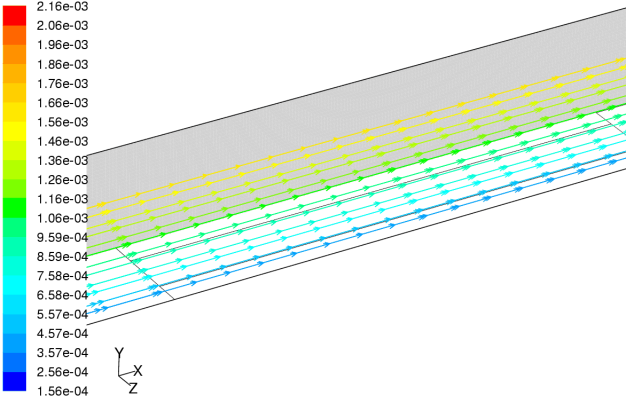
Chip Cooling - Result

Temperature Gradient vs Length



Chip Cooling - Result

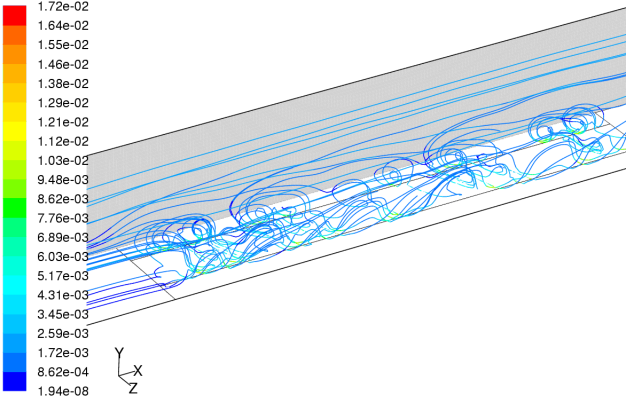
Path Lines for No-Slip Flow



- Colour indicates velocity magnitude in [m/s]

Chip Cooling - Result

Path Lines for Slip Flow



- Colour indicates velocity magnitude in [m/s]

Summary

- Boundary conditions for slip velocity and temperature jump implemented into finite volume fluid dynamics solver
- Validation of implemented routines performed
- Chip cooling application can be modelled

References

- 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
- Einzel, D., Panzer, P., Liu, M. “Boundary condition for fluid flow: Curved or rough surfaces.", Phys. Rev. Lett. vol. 64-19, 1990: 2269-2272
- Maxwell, J.C., "On stresses in rarified gases arising from inequalities of temperature.", Philos. Trans. R. Soc. London vol. 170, 1879: 231-256
- 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
- Smoluchowski, M., "Über den Temperatursprung bei Wärmeleitung in Gasen.", Akad. Wiss. Wien. CVII, 1898: 304-329