Elmer Manual DSIP-MC

CSC, the Finnish IT Center for Science

February 8, 2006

Legal Notice

This document describes the DSMC-IP module of Elmer. The module was developed at the Laboratory of Physics at Helsinki University of Technology (COMP) by Mika Nuutinen. The rights of the module remain in the possession of COMP.

CSC has been granted the right to use and distribute the code together with the Elmer software. The module is provided free of charge for public and non-commercial use only. All commerical use requires a separate written agreement with COMP.

No part of the DSMC-IP module may be reproduced or further transmitted in any form or by any means without the written permission of COMP.

CSC assumes no responsibility or liability on any errors or inaccuracies in Elmer program or documentation. Any kind of material damages or other indirect consequences resulting from any Elmer part, documentation discrepancies and errors or non-anticipated program behavior are limited to the value of appropriate products purchased from CSC.

This document is for informational use only. All information and specifications given have been carefully prepared by the best efforts of CSC, and are believed to be true and accurate as of time publishing. CSC reserves the right to modify Elmer and its documents without notice.

Contents

Chapter 1

DSMC-IP Model

Subroutine name: DSMCSolver
Module authors: Mika Nuutinen
Document authors: Mika Nuutinen
Document created: 13.8.2003
Document edited: 8.2.2006

1.1 Introduction

Direct Simulation Monte Carlo (DSMC) [1] and Information Preservation (IP) [2, 3, 4] methods are used to solve field variables such as flow velocity, pressure and temperature in mass or/and thermal flows of gases when the equations (Navier-Stokes Equations and Heat Equation) following from continuum assumption are no longer valid.

The continuum assumption may be violated when the flow dimensions become comparable with the mean free path of gas. The DSMC method is time consuming and the results are often very noisy, while the IP method gives smooth results with far less iterations. This is why only the IP part of the solution is used for now. At this stage the method is implemented for steady state, 2D rectangular channels with constant rectangular meshes.

In Elmer the DSMCSolver is used together with Navier-Stokes and Heat equations in the sense that they provide an initial guess of the field variables for the DSMC-Solver [5].

1.2 Theory

In DSMC method the flow field is divided into elements with dimensions smaller than the mean free path. Simulated molecules are created in the cells according to the local equilibrium Maxwell-Boltzmann distribution. Molecules are then moved according to their thermal speeds during a time step smaller than the mean collision time. Collisions and molecular moves are assumed to be uncoupled during a time step and after the molecules are moved the intermolecular and boundary collisions are calculated using statistical methods in the cells and diffuse reflection model on the boundary. When a molecule exits the flow-field it is removed from the simulation and new molecules are injected into the domain according to the boundary conditions. This DSMC step is then repeated for desired number of times to develop and sample the flow properties.

The IP method works parallel to the DSMC method and uses the simulated molecules as carriers for preserved or macroscopic information. At the beginning of simulation macroscopic velocity and temperature are assigned to each molecule as well as number density to the cells according to the initial conditions. This is justified because each simulated molecule usually represents an enormous number of real ones. Macroscopic information is then updated every time step according to the Euler equations in Lagrangian description for the molecules and in Eulerian description for the cells. The diffusive terms (viscous stress and heat flux) dropped from the full preservation equations are included implicitly in molecular moves and collisions. Flow properties are then averaged in each cell from the preserved quantities.

1.2.1 Governing Equations

The continuity equation used to update number density in cells reads as

$$\frac{\partial n}{\partial t} = -\vec{u} \cdot \nabla n - n' \nabla \cdot \vec{u}, \tag{1.1}$$

where n is the number density to be updated, $\vec{u}$ is the flow velocity and n' is the actual number density in the cell used to improve the stability. The partial derivatives are replaced by finite differences obtained from the averaged cell values and time step.

The momentum equation in Lagrangian description used to update macroscopic velocity of molecules reads as

$$\frac{D\vec{u}}{Dt} = -\frac{1}{n'm} \nabla p + \vec{f}, \tag{1.2}$$

where $\frac{D}{Dt}$ stands for full or material derivative, p is the scalar pressure and $\vec{f}$ is the acceleration due to an external force. Note that viscous stress does not appear here explicitly but is accounted for in the intermolecular collisions.

The energy equation in Lagrangian description used to update macroscopic temperature of molecules reads as

$$\frac{\xi n k_B}{2} \frac{DT}{Dt} = -n' k_B T \nabla \cdot \vec{u} + \Phi, \tag{1.3}$$

where T is the temperature, ξ is the degrees of freedom, k_B is the Boltzmann constant and Φ is the heat source in units W/m^3. Note that viscous heating and heat conduction do not appear here but they are accounted for in intermolecular collisions and molecular moves.

1.2.2 Material properties

1.3 Keywords

`Simulation`

The simulation section gives the case control data:

`Simulation Type` `String`

Current DSMCSolver is implemented only for `Steady State`.

Table 1.1: Molecular parameters under standard atmospheric conditions (101325Pa and 273K).

gas	molecular mass m 10^{27}kg	diameter[a] d 10^{10}m	viscosity μ 10^{5}Nsm^{-2}	visc. index ω	α	C^{b}_{visc}	C^{c}_{cond}
H_2	3.34	2.88	0.845	0.67	1.35	-	-
He	6.65	2.30	1.865	0.66	1.26	-	
Ne	33.5	2.72	2.975	0.66	1.31	-0.62	1.10
N_2	46.5	4.11	1.656	0.74	1.36	-	-
O_2	53.12	4.01	1.919	0.77	1.40	-	-
Ar	66.3	4.11	2.117	0.81	1.40	-0.36	1.122
CO_2	73.1	5.54	1.380	0.93	1.61	-	-
Kr	139.1	4.70	2.328	0.80	1.32	-0.41	1.24
Xe	218.	5.65	2.107	0.85	1.44	-0.29	1.29

[a] Only mono atomic molecules are spherical, but effective values may be defined also for polyatomic molecules to match the measured coefficient of viscosity.

[b] Simulation results. This parameter can be defined also for (nearly) isothermal polyatomic gases.

[c] Simulation results. Applicable values only for mono atomic gases.

`Coordinate System` String

Defines the coordinate system to be used, only `Cartesian 2D` is currently implemented in DSMCSolver.

`Symmetric` Logical

`True` if flow is symmetric about the x-axis, e.g. Poiseuille flow, `False` if not e.g. Couette Flow.

`DSMC Time Step` Real

The time step used in DSMC-IP simulation.

`Molecules In Cell` Integer

The number of simulated molecules per cell (corresponds to maximum initial density).

`Steps To Develop` Integer

The number of time steps to develop the flow from initial guess.

`Steps To Sample` Integer

The number of time steps to sample and average the flow after `Steps to Develop`.

`Solver` solver id

The solver section defines equation solver control variables. Most of the possible keywords – related to linear algebra, for example – are common for all the solvers and are explained elsewhere.

`Equation` String

The name of the equation is `“DSMCSolver”`.

`Procedure` File

The file where the procedure named `DSMCSolver` is. Type e.g. `“./DSMCSolver”` `“DSMCSolver”` if procedure `“DSMCSolver”` is in file `“./DSMCSolver”` in working directory.

In the `Solver` section there is also a long list of additional variables that are used in the simulation.

`Variable` String DSMCSolver

`Variable DOFs` Integer 1

`Exported Variable 1` String MCPressure
: The DSMC pressure.

`Exported Variable 1 DOFs` Integer 1

`Exported Variable 2` String MCTemperature
: The DSMC temperature.

`Exported Variable 2 DOFs` Integer 1

`Exported Variable 3` String MCVelocity
: The DSMC velocity.

`Exported Variable 3 DOFs` Integer 3

`Exported Variable 4` String Kn_nodal_u
: The local Knudsen number for velocity (optional, used in user function for effective viscosity).

`Exported Variable 4 DOFs` Integer 1

`Exported Variable 4` String Kn_nodal_t
: The local Knudsen number for temperature (optional, used in user function for effective conductivity).

`Exported Variable 4 DOFs` Integer 1

`Equation` eq id
: The equation section is used to define a set of equations for a body or set of bodies.

`DSMCSolver` String
: If set to `True`, compute the DCMC-IP solution.

`Material` mat id
: The material section is used to give the material parameter values. The following material parameters may be usefully when the flow/heat solution is to be used as an initial condition for DSMCSolver.

`Viscosity Model` String
: If viscosity is assumed variable of temperature or if gas rarefaction is taken into account by an effective viscosity model, type `”User function”`.

`Viscosity Function` File
: Referring to Viscosity Model, if the viscosity function is in working directory, type e.g. `“./funcV” “EffectiveViscosity”`.

`Heat Conductivity Model` String
: If heat conductivity is assumed variable of temperature or if gas rarefaction is taken into account by an effective conductivity model, type `”User function”`.

`Heat Conductivity Function` File

Referring to Heat Conductivity Model, if the conductivity function is in working directory, type e.g. “`./funcC`” “`EffectiveConductivity`”.

`Compressibility Model` String

This setting may be used to set the compressibility model for the flow simulations. Choices are `Incompressible` and `Perfect Gas`. Always use the latter with DSMCSolver.

`Specific Heat Ratio` Real

The ratio of specific heats (in constant pressure versus in constant volume) may be given with this keyword. The default value of this setting is $5/3$, which is the appropriate value for mono atomic ideal gas. DSMCSolver is implemented only for mono atomic gases, but if the temperature variation is insignificant, the model can still be used to compute the flow field accurately even for di- or polyatomic gases.

In the `Material` section there are also some DSMC-IP specific parameters:

`Molecular Mass` Real

The molecular mass [1]. (see table 1.)

`Molecular Diameter` Real

See table 1. for reference diameters at 273K [1]. (see table 1.)

`Viscosity Power Law` Real

The viscosity (heat conductivity) power law index $\mu \propto T^{\omega}$, ($\kappa \propto T^{\omega}$) [1]. (see table 1.)

`VSS Scattering Parameter` Real

Variable Soft Sphere scattering parameter [1]. (see table 1. for α)

`Collisional Viscosity Parameter` Real

Viscosity parameter made to match DSMC- and IP-results. (see table 1. for $Cvisc$)

`Collisional Conductivity Parameter` Real

Conductivity parameter made to match DSMC- and IP-results- (see table 1. for $Ccond$)

`Body Force` bf id

The body forces like gravity and frictional heating are defined in this section for Navier-Stokes and Heat Solvers. DSMCSolver has somewhat different definitions for external sources:

`Force Density` Real

The volumetric constant external force in x-direction in units N/m^3.

`Power Density` Real

The volumetric constant external power in units W/m^3.

`Boundary Condition` Integer

The boundary conditions need some definitions for the DSMCSolver.

`Boundary Name` String

The type of the boundary needs to be specified “`Lower Boundary`”, “`Outlet`”, “`Upper Boundary`” or “`Inlet`” respectively. ‘“`Lower`

Boundary" and "Upper Boundary" are solid or "Lower Boundary" may be a symmetry line. "Inlet" and "Outlet" are open boundaries at the ends of the channel.

Plate Velocity 1 Real

The constant velocity of the lower/upper boundary in x-direction, the value should be equal to Velocity 1 on the corresponding boundary and to Plate Velocity 1 on the opposing boundary if Symmetric is set to True. To get better results a user function to model slip velocity should be used.

Chapter 2

Example: Rarefied gas flow in a rectangular 2D channel

Solvers: HeatSolve, FlowSolve, DSMCSolver
Tools: ElmerGrid
Dimensions: 2D, Steady-state

2.1 Problem description

In the example we study the flow of rarefied gas in a straight rectangular channel under strong pressure- and temperature- gradients. The results are compared to the solutions of the Navier-Stokes- and Heat-equation solvers.

A gas (Ar) flow in the channel is induced by a pressure difference between the two open ends of the channel (15μm$\times$0.5μm). The pressure at the right and left ends are kept at 100kPa and 150kPa respectively. A very strong heat flux is induced by a temperature difference between the solid upper and lower edges of the channel. The temperatures of the upper and lower walls are kept at 283K and 263K respectively. Additionally we may guess (by inspecting the Navier-Stokes and Heat Equations) that the pressure drop in flow direction and the frictional heating of gas also affect the resulting temperature distribution. The problem is now to solve the field variables: velocity, pressure and temperature, first by the conventional Navier-Stokes and Heat Equations with first order slip boundary conditions (for comparison these solutions are also computed using effective viscosity-, heat conductivity- and slip-models corrected with local Knudsen number [5]), and then by using these solutions as an initial condition for the DSMCSolver, obtain a more realistic solution for the field variables at this standard Knudsen number region (approximately 0.64<Kn<1.04) which is well in the transition region where the standard continuum equations are expected to give inaccurate results.

2.2 Solution procedure

The *.sif file with comments used in this example is given below.

```
Check Keywords "Ignore"
```

```
Header
   Mesh DB "MESHDIR" "m1"
End

Simulation
  Min Output Level = Integer 0
  Max Output Level = Integer 31
  Output Caller = Logical True
  Coordinate System = String "Cartesian 2D"
  Coordinate Mapping(3) = 1 2 3
  Simulation Type = String "Steady State"
  Steady State Max Iterations = Integer 20
  Output Intervals = Integer 1
  Solver Input File = File "loota.sif"
  Output File = File "loota100.dat"
  Post File = File "loota103.ep"

  !!!----------SIMULATION PARAMETERS FOR DSMCSOLVER----------!!!
  Symmetric = Logical False
  DSMC Time Step = Real 2E-11
  Molecules In Cell = Integer 100
  Steps To Develop = Integer 2000
  Steps To Sample = Integer 20000
  !!!--------------------------------------------------------!!!
End

Constants
  Gravity(4) = Real 0 -1 0 9.82
  Stefan Boltzmann = Real 5.67e-08
End

Body 1
  Name = String "Body1"
  Initial Condition = 1
  Body Force = 1
  Equation = 1
  Material = 1
End

Equation 1
  Name = "Equation1"
  Active Solvers(3) = 1 2 3
  Convection = "Computed"
  NS Convect = Logical True
  DSMCSolver = Logical True
End

Solver 2
  Exec Solver = "Always"
  Equation = "Heat Equation"
  Variable = "Temperature"
  Variable Dofs = 1
  SolverId = Integer 2
  Linear System Solver = "Iterative"
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 350
  Linear System Convergence Tolerance = 1e-08
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU3"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1e-05
  Stabilize = False
  Bubbles = True
  Nonlinear System Convergence Tolerance = 1e-05
  Nonlinear System Max Iterations = 10
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 1.0e-2
  Nonlinear System Relaxation Factor = 1.0
End

Solver 1
  Exec Solver =  "Always"
  Equation = "Navier-Stokes"
  Variable = "Flow Solution"
  Variable Dofs = 3
  SolverId = Integer 1
  Linear System Solver = "Iterative"
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1e-08
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU3"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1e-05
  Stabilize = True
  Nonlinear System Convergence Tolerance = 1e-05
  Nonlinear System Max Iterations = 20
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 0.01
  Nonlinear System Relaxation Factor = 1.
End

Solver 3
  Exec Solver = "After All"
  Equation = "DSMCSolver"
  Procedure = File "./DSMCSolver" "DSMCSolver"
  Variable = "DSMCSolver"
  Variable DOFs = 1
  Exported Variable 1 = "MCPressure"
  Exported Variable 2 = "MCTemperature"
  Exported Variable 3 = "MCVelocity"
  Exported Variable 3 DOFs = 3
  Exported Variable 4 = "Kn_nodal_u"
  Exported Variable 5 = "Kn_nodal_t"
End

Material 1
  Name = String "Material1"
  Viscosity = Real 0.
  Viscosity Model = String "User function"
  Viscosity Function = File "./funcV" "EffectiveViscosity"
  Reference Temperature = Real 273.
  Compressibility Model = String "Perfect Gas Equation 1"
  Reference Pressure = Real 100000.
  Specific Heat Ratio = Real 1.666666666
  Heat Capacity = Real 520
  Heat Conductivity = Real 0.
  Heat Conductivity Model = String "User function"
  Heat Conductivity Function = File "./funcC" "EffectiveConductivity"

  !!!!----------MATERIAL PARAMETERS FOR DSMCSOLVER----------!!!
  Molecular Mass = Real 66.3E-27
  Molecular Diameter = Real 4.11E-10
  Viscosity Power Law = Real 0.81
  VSS Scattering Parameter = Real 1.40
  Collisional Viscosity Parameter = Real -0.360
  Collisional Conductivity Parameter = Real 1.122
  !!!-------------------------------------------------------!!!
End

Body Force 1
  Name = String "BodyForce1"
  Friction Heat = Logical True

  !!!----------EXTERNAL BODY FORCES FOR DSMCSOLVER----------!!!
  Force Density 1 = Real 0.
  Power Density = Real 0.
  !!!-------------------------------------------------------!!!
End

Initial Condition 1
  Name = String "InitialCondition1"
   Temperature = Real 273.
   Velocity 1 = Real 0.
   Velocity 2 = Real 0.
   Pressure = Real 0.
End

Boundary Condition 1
```

```
  Name = String "Down"
  Target Boundaries = 1
  Boundary Name = String "Lower Boundary"
  Velocity 2 = Real 0.
  External Temperature = Real 273.
  Flow Force Bc = Logical True
  Heat Flux Bc = Logical True

  Heat Transfer Coefficient = Variable Pressure
    Real Procedure "TSlip" "TSlip"

  Slip Coefficient 1 = Variable Pressure
    Real Procedure "Slippery" "Slippery"

  Plate Velocity 1 = Real 0.
End

Boundary Condition 2
  Name = String "Out"
  Target Boundaries = 2
  Boundary Name = String "Outlet"
  Pressure 1 = Real 0.
  Flow Force Bc = Logical True
End

Boundary Condition 3
  Name = String "Up"
  Target Boundaries = 3
  Boundary Name = String "Upper Boundary"
  Velocity 2 = Real 0.
  External Temperature = Real 273.
  Flow Force Bc = Logical True
  Heat Flux Bc = Logical True

  Heat Transfer Coefficient = Variable Pressure
        Real Procedure "TSlip" "TSlip"

  Slip Coefficient 1 = Variable Pressure
        Real Procedure "Slippery" "Slippery"

  Plate Velocity 1 = Real 0.
End

Boundary Condition 4
  Name = String "In"
  Target Boundaries = 4
  Boundary Name = String "Inlet"
  Pressure 1 = Real 50000.
  Flow Force Bc = Logical True
End
```

2.3 Results

The nature of the resulting flow fields can be deduced beforehand. The velocity field in x-direction and the temperature field are presented in figures 2.1 and 2.2 for clarity. We are mainly interested in the differences between the continuum and DSMC-IP results. In figure 2.3 the difference in temperature solutions (DSMC-IP - Heat Eq.) is presented. Since the temperature profiles are approximately linear the difference is mainly due to the temperature slip on the solid walls. In figures 2.5 and 2.7 the effect of the Knudsen layers is evident: the difference in velocity solutions changes rapidly near the walls, indicating reduced effective viscosity, and remains nearly constant in the region well away from the Knudsen layers. The local Knudsen number corrected continuum results are superior to those obtained without corrections. In figure 2.4 the temperature difference between corrected heat equation and DSMC-IP solutions is presented. Although the maximum temperature difference is not much smaller than in figure 2.3, the error is more confined to the proximity of the walls (Knudsen layers). In figures 2.6 and 2.8 the corresponding velocity differences are presented. The error is much smaller in magnitude and the effects of the Knudsen layers near the walls are again more sharply confined.

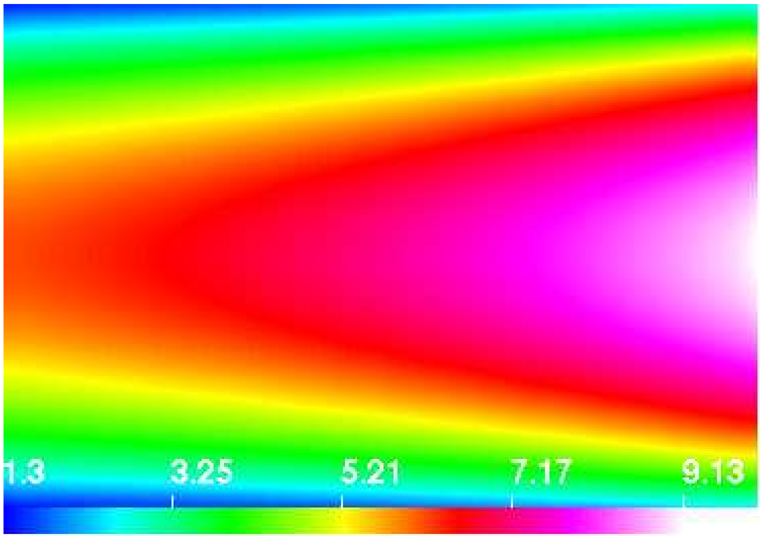

Figure 2.1: Velocity component in x-direction [m/s], DSMC-IP result.

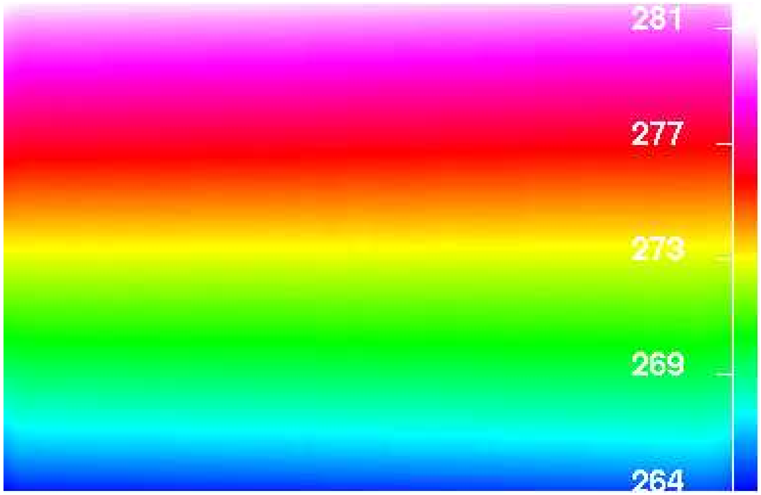

Figure 2.2: Temperature profile, DSMC-IP result.

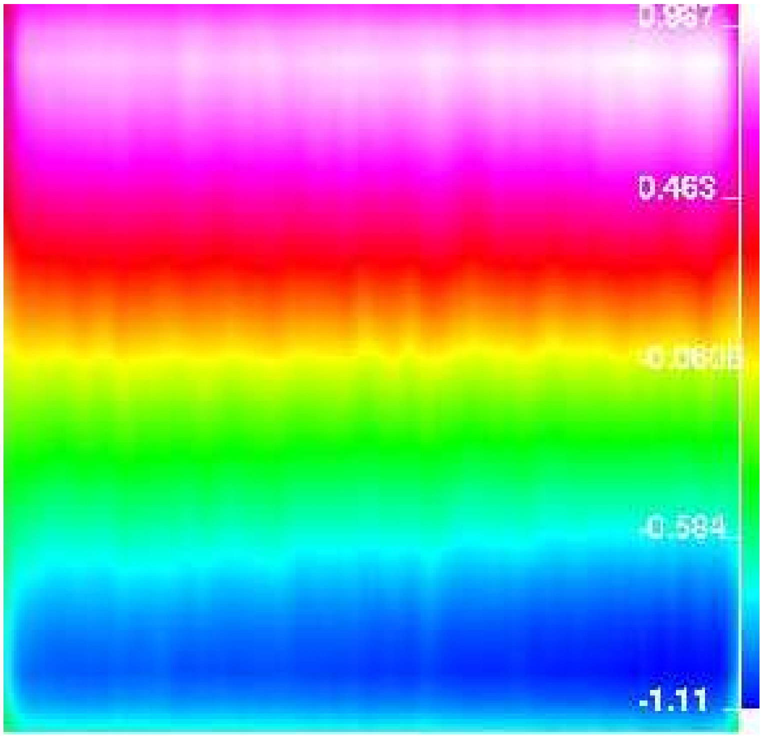

Figure 2.3: Temperature difference $[K]$ between DSMC-IP and FEM methods. The figure is flipped about the x-axes so the lower edge is now hotter.

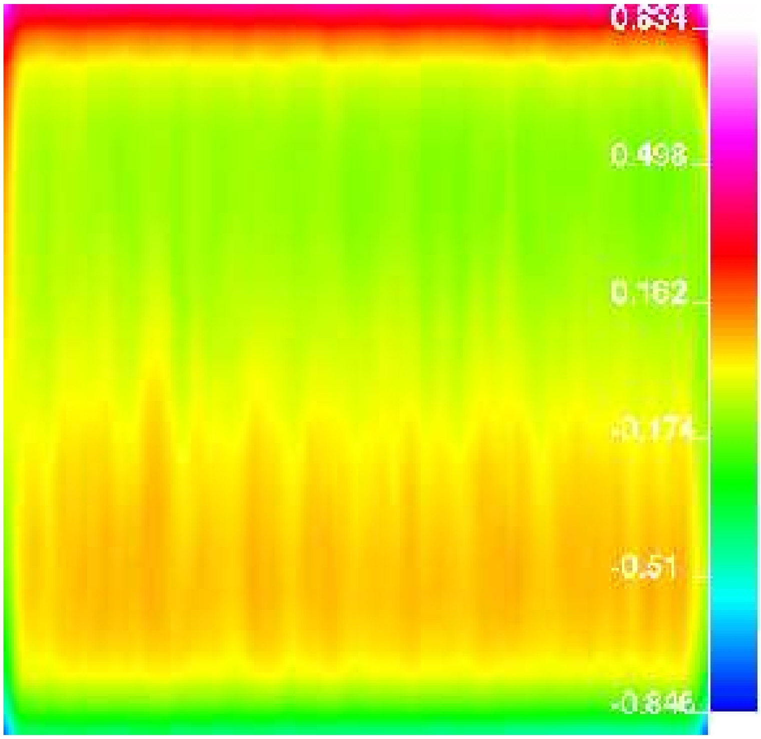

Figure 2.4: Temperature difference $[K]$ between DSMC-IP and Kn corrected FEM methods.

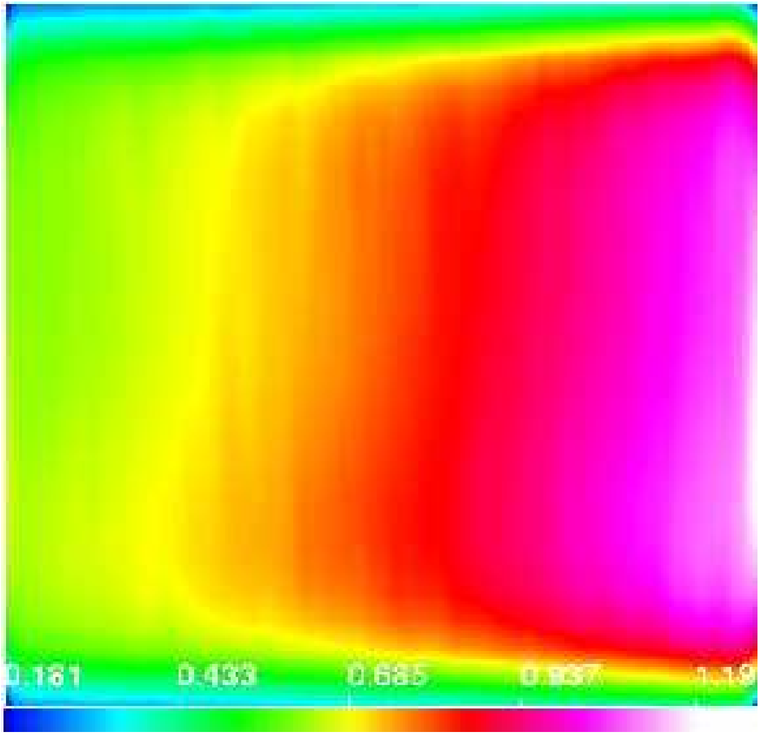

Figure 2.5: Velocity difference $[m/s]$ between DSMC-IP and FEM methods

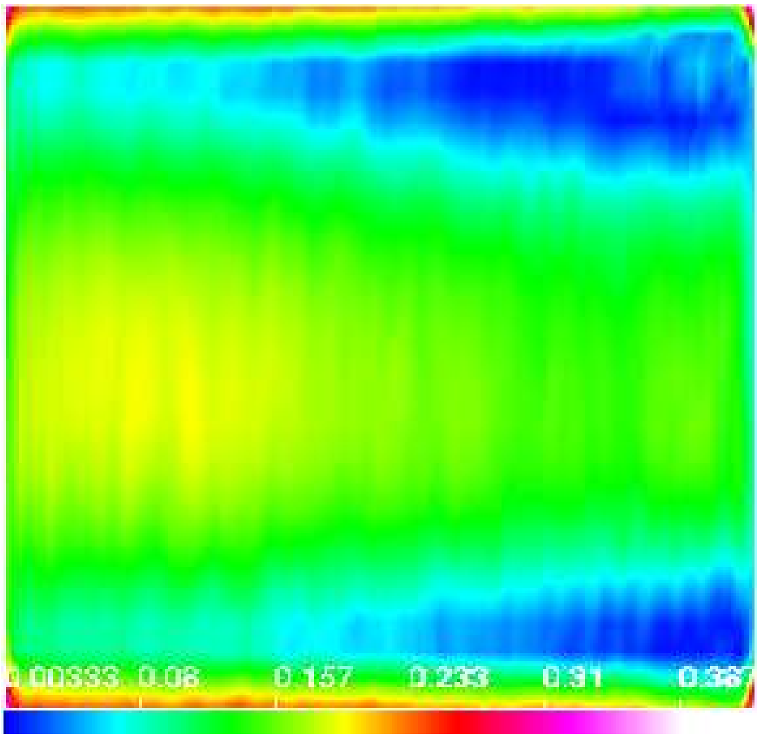

Figure 2.6: Velocity difference $[m/s]$ between DSMC-IP and corrected FEM methods

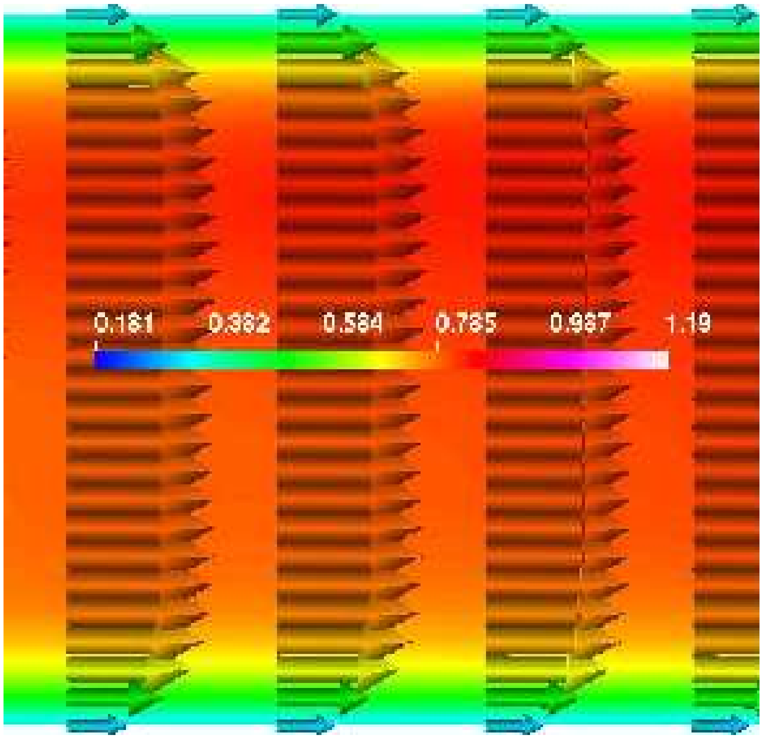

Figure 2.7: Velocity difference $[m/s]$ between DSMC-IP and FEM methods at the center of the channel.

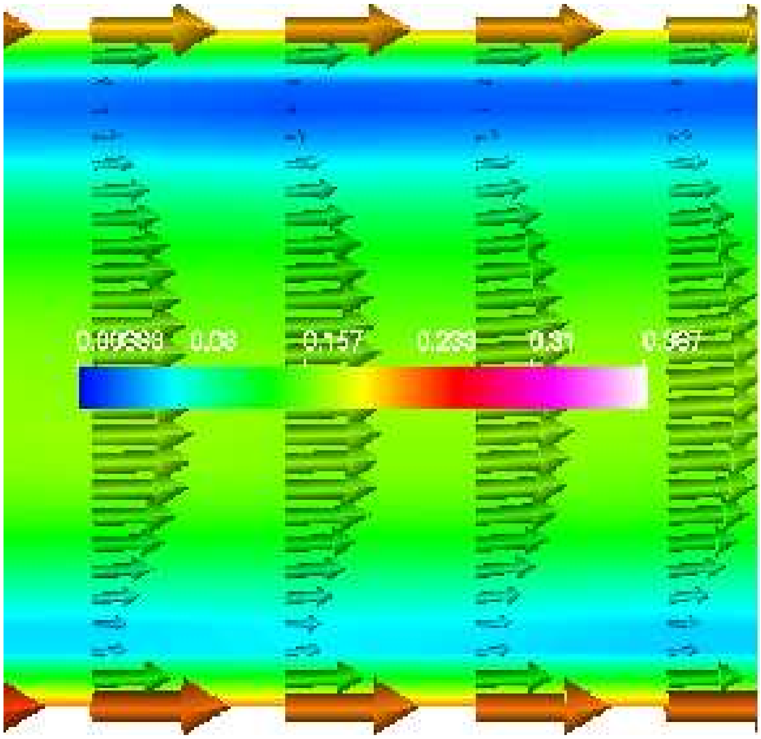

Figure 2.8: Velocity difference $[m/s]$ between DSMC-IP and corrected FEM methods at the center of the channel.

Bibliography

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

[2] J. Fan C. Cai, I.D. Boyd and G.V. Gandler. Direct simulation methods for low-speed microchannel flows. *Journal of thermophysics and heat transfer*, 14(3):368–378, 2000.

[3] Q. Sun and I.D. Boyd. A direct simulation method for subsonic microscale gas flows. *Journal of Computational Physics*, 179(2):400–425, 2002.

[4] I.D. Boyd Q. Sun and G.V. Gandler. A hybrid continuum / particle approach for micro-scale gas flows. 2002.

[5] Mika Nuutinen. Mikrovirtaukset. Master's thesis, Helsinki University of Technology, 2003.