Journal of Computational Physics 228 (2009) 7109–7124

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A particle–particle hybrid method for kinetic and continuum equations [☆]

Sudarshan Tiwari [a,*], Axel Klar [a], Steffen Hardt [b]

[a] Fachbereich Mathematik, TU Kaiserslautern, Gottlieb-Daimler-Strasse, 67663 Kaiserslautern, Germany
[b] Center of Smart Interfaces, TU Darmstadt, Petersenstr. 32, 64287 Darmstadt, Germany

ARTICLE INFO

Article history:
Received 4 February 2009
Received in revised form 13 May 2009
Accepted 16 June 2009
Available online 24 June 2009

Keywords:
Hybrid particle methods
Boltzmann equation
Navier–Stokes equations
Domain decomposition
Projection method
Shock tube problem

ABSTRACT

We present a coupling procedure for two different types of particle methods for the Boltzmann and the Navier–Stokes equations. A variant of the DSMC method is applied to simulate the Boltzmann equation, whereas a meshfree Lagrangian particle method, similar to the SPH method, is used for simulations of the Navier–Stokes equations. An automatic domain decomposition approach is used with the help of a continuum breakdown criterion. We apply adaptive spatial and time meshes. The classical Sod's 1D shock tube problem is solved for a large range of Knudsen numbers. Results from Boltzmann, Navier–Stokes and hybrid solvers are compared. The CPU time for the hybrid solver is 3–4 times faster than for the Boltzmann solver.

1. Introduction

The coupling of kinetic and fluid dynamic equations originated from simulations of hypersonic flows around a space vehicle during the re-entry phase, where it experiences several flow regimes that are characterised by the Knudsen number (ratio of the mean free path and the characteristic length). In the past decade the coupling of kinetic and fluid dynamic equations has become an important research area also for flow problems on micro- and nanoscales [1,34].

For a larger Knudsen number, which corresponds to a rarefied regime, the flow is computed with the help of a kinetic equation, i.e., the Boltzmann equation. Usually particle methods, like DSMC [6] and its variants (see, for example, [2,29]) are used for the simulation of the Boltzmann equation. However, for smaller Knudsen numbers DSMC type particle methods are becoming increasingly expensive since the cell size must be the order of the mean free path. On the other hand, in the continuum regime one can solve Euler or Navier–Stokes equations. However, the continuum equations are not valid everywhere in the computational domain, for example, in shock and boundary layers, etc. This leads to domain decomposition approaches including continuum and kinetic domains. For these approaches it is first necessary to define the domains of validity and then to solve the equations in their respective domains. Several criteria have been suggested for the breakdown of the continuum equations [7,8,22,28,35]. Many efforts have been reported in the development of hybrid solvers, see for example [5,9,10,14,15,17,18,25–28,30,31,35,37,40,41].

Most of these papers deal with the coupling of a particle method for the Boltzmann equation and with a Finite Volume or Finite Element code for the fluid dynamic equations. The more natural choice (and more straightforward to implement also for complicated applications) is to choose particle methods for both equations. This simplifies the treatment of interface

[☆] This work was partially supported by the German Research Foundation (DFG), KL 1105/17-1, HA 2696/16-1.

[*] Corresponding author. Tel.: +49 631 205 4496; fax: +49 631 205 4986.
E-mail addresses: tiwari@itwm.fhg.de (S. Tiwari), klar@mathematik.uni-kl.de (A. Klar), hardt@csi.tu-darmstadt.de (S. Hardt).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.019

conditions between the two domains considerably. In particular, this is important if the decomposition process results in complicated (time dependent) domains for Boltzmann and Navier–Stokes equations. In [37] the continuum equations (Euler) have been solved by a kinetic particle method [19,24]. However, this method is not the optimal one in terms of CPU time, since it is – for the same spatial and temporal grids – as expensive as the particle method for the Boltzmann equation.

In the present work we use a meshfree Lagrangian, SPH-type, particle method for the continuum (Navier–Stokes) equations [36,38]. This method is several times faster than the kinetic particle method. Here, particles are numerical interpolation points, which move with fluid velocities and carry all necessary fluid information, like density, velocity, pressure, etc. with them. Differential operators at an arbitrary particle position are approximated from its neighbouring cloud of particles. Meshfree methods are in particular suitable for the coupling of the Boltzmann and fluid dynamic equations since they allow for an arbitrary treatment of the interface between the two regimes [37]. To the authors's knowledge SPH type particle methods for the Navier–Stokes equations have up to now not been coupled to Monte-Carlo methods for the Boltzmann equation for time dependent problems.

The particle methods for both the Boltzmann and the Navier–Stokes equations utilize a grid on which the particles move. We use different grid spacings and time steps in both cases. In general, the Boltzmann grid size must be of the order of the mean free path and the Navier–Stokes grid size (i.e., the distance between Navier–Stokes particles) is independent of the mean free path and can be several times larger in the continuum regime. The adaptive grid refinement technique used here is similar to that of above mentioned earlier work [37]. To determine the domains of validity for the Boltzmann and Navier–Stokes equations we use the breakdown criterion suggested in [35]. It can be computed as a function of the stress tensor and heat flux vector, which in turn can be computed from the Navier–Stokes solver. The numerical example we consider in this paper is the 1D unsteady shock tube problem (Sod's problem [32]), using full Boltzmann, Navier–Stokes and hybrid solvers in a large range of Knudsen numbers. As expected, it is shown that for large Knudsen numbers, the solutions of the Navier–Stokes solver have large discrepancy compared to the solutions of the Boltzmann solver. However, the solutions from the hybrid solver are close to the ones from the Boltzmann solver. This indicates that one can avoid the unnecessary use of the Boltzmann solver in the entire domain even for larger Knudsen numbers.

The paper is organized as follows. In Section 2, we present the mathematical models. In Section 3 the numerical methods for the Boltzmann and the Navier–Stokes equations are described. The description of the hybrid method is explained in Section 4. Finally, some numerical tests are presented in Section 5.

The above methodology can be easily extended to more complicated problems. Extensions to 2D, 3D problems and multiphase flows in nano-devices are in preparation.

2. Mathematical model

The Boltzmann equation describes the time evolution of a distribution function $f(t,x,v)$ for particles of velocity $v \in \mathbb{R}^3$ at point $x \in D \subset \mathbb{R}^d$ and time $t \in \mathbb{R}_+$. It is given by

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = J(f,f), \tag{1}$$

where

$$J(f,f) = \int_{\mathbb{R}^3} \int_{S^2} \beta(|v-w|,\eta)[f(v')f(w') - f(v)f(w)]d\omega(\eta)dw$$

with

$$v' = T_{v,w}(\eta) = v - \eta\langle\eta, v-w\rangle, \quad w' = T_{w,v}(\eta).$$

Here, β denotes the collision cross-section and $\langle , \rangle$ is the scalar product. Writing the equations in dimensionless form one observes that J is of the order $\mathcal{O}(\frac{1}{\epsilon})$, where ϵ is a dimensionless quantity, the Knudsen number, which is related to the mean free path of the molecules. The local mean free path $\lambda = \lambda(x,t)$ is given by

$$\lambda = \frac{kT}{\sqrt{2}\pi p d^2}, \tag{2}$$

where k is the Boltzmann constant, $T = T(x,t)$ the temperature, $p = p(x,t)$ the pressure and d is the diameter of molecules. For more details we refer to [12,33].

For ϵ tending to zero, i.e., for small mean free paths, one can show that the Boltzmann distribution function f tends to the local Maxwellian [11]

$$f_M(v;x,t) = \frac{\rho}{(2\pi RT)^{3/2}} e^{-\frac{|v-u|^2}{2RT}}, \tag{3}$$

where $\rho = \rho(x,t)$ is the density, $u = u(x,t)$ the mean velocity and R is the gas constant. The parameters of the Maxwellian $\rho,\ u,\ T$ solve the compressible Euler equations. This can be verified from the asymptotic expansion of f in ϵ, where the zeroth order approximation gives the local Maxwellian distribution and the first order approximation [4] gives the Chapman–Enskog distribution

$$f_{CE}(v;x,t) = f_M(v;x,t)[1+\phi(v;x,t)] \tag{4}$$

with

$$\phi(v;x,t) = \frac{2}{5}\frac{q\cdot c}{\rho(RT)^2}\left(\frac{|c|^2}{2RT}-\frac{5}{2}\right) - \frac{1}{2}\frac{\tau : c\otimes c}{\rho(RT)^2}, \tag{5}$$

where $c = v - u$. Here, $\phi = \mathcal{O}(\epsilon)$ and the parameters $\rho,\ u,\ T,\ q,\ \tau$ satisfy the compressible Navier–Stokes equations

$$\begin{aligned}
&\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho u) = 0,\\
&\frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u\otimes u + pI - \tau) = 0,\\
&\frac{\partial(\rho E)}{\partial t} + \nabla\cdot[(\rho E + p)u - \tau\cdot u - q] = 0,
\end{aligned} \tag{6}$$

where $E = |u|^2/2 + e$ is the total energy and e is the internal energy, the stress tensor τ and heat flux vector q are of order ϵ and given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\nabla\cdot u\delta_{ij}\right), \quad q = -\kappa\nabla T. \tag{7}$$

The dynamic viscosity $\mu = \mu(x,t)$ and the heat conductivity $\kappa = \kappa(x,t)$ for a monatomic gas of hard sphere molecules are of order ϵ. They are given, see [6], by

$$\mu = \frac{5}{16d^2}\sqrt{\frac{mkT}{\pi}}, \quad \kappa = \frac{15k}{4m}\mu, \tag{8}$$

where m is the molecular mass.

3. Numerical methods

For the both types of equations particle methods are used. The Boltzmann equation is solved by a DSMC type Monte-Carlo method, whereas the Navier–Stokes equations are treated with a meshfree method which is similar to Smoothed Particle Hydrodynamics (SPH) [20]. In this paper we consider problems in one spatial dimension.

3.1. Particle method for the Boltzmann equation

For solving the Boltzmann equation we have used a variant of the DSMC method [6], developed in [29,2,3]. The method is based on the time splitting of the Boltzmann equation. Introducing fractional steps one solves first the free transport equation (the collisionless Boltzmann equation) for one time step. During the free flow, boundary and interface conditions are taken into account. In a second step (the collision step) the spatially homogenous Boltzmann equation without the transport term is solved. To simulate this equation by a particle method an explicit Euler step is performed. The result is then used in the next time step as the new initial condition for the free flow. To solve the homogeneous Boltzmann equation the key point is to find an efficient particle approximation of the product distribution functions in the Boltzmann collision operator given only an approximation of the distribution function itself. To guarantee positivity of the distribution function during the collision step a restriction of the time step proportional to the Knudsen number is needed. That means that the method becomes exceedingly expensive for small Knudsen numbers.

3.2. Meshfree particle method for the Navier–Stokes equations

The 1D Navier–Stokes equations are solved with a meshfree Lagrangian method, where we approximate the spatial derivatives with the help of the weighted least squares method. We consider the 1D domain Ω with boundary Γ. In 1D the Navier–Stokes equations (6) can be reexpressed in the Lagrangian form in the primitive variables as

$$\begin{aligned}
&\frac{dx}{dt} = u,\\
&\frac{d\rho}{dt} = -\rho\frac{\partial u}{\partial x},\\
&\frac{du}{dt} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\rho}\frac{\partial}{\partial x}\left(\frac{4}{3}\mu\frac{\partial u}{\partial x}\right),\\
&\frac{dT}{dt} = \frac{\gamma-1}{\rho R}\left[-p\frac{\partial u}{\partial x} + \frac{4}{3}\mu\left(\frac{\partial u}{\partial x}\right)^2 + \frac{\partial}{\partial x}\left(\kappa\frac{\partial T}{\partial x}\right)\right].
\end{aligned} \tag{9}$$

Here d/dt is the material derivative, γ is the ratio of specific heats and the other parameters have already been introduced in the previous section. Moreover, we consider the equation of state

$$p = \rho RT. \tag{10}$$

The system of Eqs. (9) and (10) are solved with some initial and boundary conditions. Note that from the continuity equation in (9) and (10) we have

$$\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{d\rho}{dt} = -\frac{1}{p}\frac{dp}{dt} + \frac{1}{T}\frac{dT}{dt}. \tag{11}$$

To solve the Eqs. (9) and (10) by a SPH type of method, one fills first the computational domain by particles, and then approximates the spatial derivatives occurring on the right hand side of (9) at each particle position from its neighbouring particles. This reduces the system of partial differential equation (9) to a system of ordinary differential equations with respect to time. The number of equations equals four times the total number of particles.

Using an explicit Euler method the time step would be restricted by the CFL condition and by the value of the transport coefficient, for example, $\nu = \max[\mu/\rho, \kappa/(\rho c_v)]$, where c_v is the heat capacity. The larger ν the smaller the time step has to be chosen. In the situations considered here, the time step for an explicit scheme for the Navier–Stokes equations would be much smaller than the time step for the Boltzmann solver. Therefore, we propose an explicit movement of the particles combined with an implicit solver for the momentum and energy equations. We treat system (9) using Chorin's pressure projection method, which was originally introduced for incompressible flows [13]. It consists of two fractional steps. Since the method is fully Lagrangian, we first compute the new particle positions at time level $t^{n+1} = (n+1)\Delta t$, where Δt is the time step, by

$$x^{n+1} = x^n + \Delta t u^n, \tag{12}$$

where x is the position of a particle. Then, for each particle position, we compute the intermediate velocity u^* by

$$u^* - \Delta t\frac{4}{3}\frac{1}{\rho^n}\frac{\partial \mu^n}{\partial x}\frac{\partial u^*}{\partial x} - \Delta t\frac{4}{3}\frac{1}{\rho^n}\frac{\partial^2 u^*}{\partial x^2} = u^n. \tag{13}$$

We note that the viscosity is not constant since it is the function of the temperature. The second step consists in establishing the new velocity u^{n+1} by correcting the intermediate velocity u^*. For this, we need to solve the equation

$$u^{n+1} = u^* - \frac{\Delta t}{\rho^n}\frac{\partial p^{n+1}}{\partial x} \tag{14}$$

with the constraints that u^{n+1} satisfy the continuity equation in (9). Hence, differentiating equation (14) with respect to x and using the relation (11), we obtain the equation for the pressure

$$\frac{-1}{\Delta t RT^n}p^{n+1} - \frac{\Delta t}{\rho^n}\frac{\partial \rho^n}{\partial x}\frac{\partial p^{n+1}}{\partial x} + \Delta t\frac{\partial^2 p^{n+1}}{\partial x^2} = -\frac{\rho^n}{\Delta t} + \rho^n\frac{\partial u^*}{\partial x} + \frac{\rho^n}{T^n}\frac{dT^n}{dt}. \tag{15}$$

On the right hand side of (15) dT/dt is replaced using the energy equation in (9).

The boundary condition for p^{n+1} is obtained by projecting Eq. (14) on the unit normal vector $n = +1$ and -1 to the boundary Γ. Thus, we obtain the Neumann boundary condition

$$\frac{\partial p^{n+1}}{\partial n} = -\frac{1}{\Delta t}\left(u_\Gamma^{n+1} - u_\Gamma^*\right)\cdot n, \tag{16}$$

where u_Γ is the value of u on Γ. Assuming that $u \cdot n = 0$ on Γ, we obtain

$$\frac{\partial p^{n+1}}{\partial n} = 0 \tag{17}$$

on Γ. Furthermore, we solve the energy equation implicitly as

$$T^{n+1} - \Delta t\frac{\gamma-1}{R\rho^n}\frac{\partial \kappa^n}{\partial x}\frac{\partial T^{n+1}}{\partial x} - \Delta t\frac{\gamma-1}{R\rho^n}\kappa^n\frac{\partial^2 T^{n+1}}{\partial x^2} = T^n - \Delta t\frac{\gamma-1}{R\rho^n}\left[-p^n\frac{\partial u^n}{\partial x} + \frac{4}{3}\frac{\mu^n}{\rho^n}\left(\frac{\partial u^n}{\partial x}\right)^2\right]. \tag{18}$$

Note that Dirichlet boundary conditions for $u = u_\Gamma$ and $T = T_\Gamma$ are prescribed on the boundary Γ. Finally, we update the density by

$$\rho^{n+1} = \frac{p^{n+1}}{RT^{n+1}} \tag{19}$$

and the viscosity and the heat conductivity according to (8).

The remaining task is the approximation of Eqs. (13), (15), (18) and (14). In the following we present a short description of the least squares method, developed in [36,38].

3.3. Meshfree approximation of a function and its derivatives

Since we deal with a one-dimensional problem, we describe the least squares approximation in 1D. Extension to higher dimensional cases is straight forward. Let $f(x)$ be a scalar function and f_i its values at $x_i \in [0,1]$ for $i = 1,2,\ldots,N$. Consider the problem to approximate the function and the spatial derivatives of the function $f(x)$ at x in terms of the values of a set of neighboring points. In order to limit the number of neighbor points we associate a weight function $w = w(x_i - x; h)$ with small compact support, where h determines the size of the support. The weight function can be quite arbitrary but in our computations, we consider a Gaussian weight function in the following form:

$$w(x_i - x; h) = \begin{cases} exp\left(-\alpha \frac{(x_i-x)^2}{h^2}\right), & \text{if } \frac{|x_i-x|}{h} \leqslant 1 \\ 0, & \text{else} \end{cases} \tag{20}$$

with α a positive constant. Appropriate numerical values lie between 2 and 6. The radius of interaction h defines a set of neighboring particles around x. We define h such that there are enough neighbouring points for the least squares approximation. In general, we define $h = 3dx$, where dx is the initial spacing of particles. Let $P(x,h) = \{x_i : |x_i - x| \leqslant h,\ i = 1,2,\ldots,m\}$ be the set of m neighboring points of x inside the radius h. The distribution of neighboring points needs not to be uniform and it can be quite arbitrary. For consistency reasons some obvious restrictions are required, namely for example all neighbor particles should not be on the same point. Furthermore, in certain situations new particles have to be introduced in order to stabilize the code. The fluid quantities are approximated from the neighbor particles with the help of the least squares method. We approximate the function $f(x)$ by

$$f_h(x) = \Pi f_h(x) = \sum_{i=1}^{N} f_i \phi_h(x_i, x), \tag{21}$$

where the shape function $\phi_h(x_i, x)$ is computed at each point x by the least squares method over its own compact support h. It is important to stress that this expression is consistent only if the function ϕ_h is 1 at x_i, namely $\phi_h(x_i, x_j) = \delta_{ij}$ for all $i,\ j = 1,2,\ldots,N$.

The approximations of the first and second order derivatives can be computed directly from $f_h(x)$ or directly by using the least squares method. The first method is known in literature as moving least squares method. In this paper we approximate the derivatives $\partial f(x)/\partial x$ by

$$(f_h(x))_x = \Pi_x f_h(x) = \sum_{i=1}^{N} f_i \eta_h(x_i, x), \tag{22}$$

where $\eta_h(x_i, x)$ is directly computed by the least squares interpolation. In a similar manner we define the approximation for the second order derivatives $\partial^2 f(x)/\partial x^2$ by

$$(f_h(x))_{xx} = \Pi_{xx} f_h(x) = \sum_{i=1}^{N} f_i \psi_h(x_i, x). \tag{23}$$

The operators $\Pi,\ \Pi_x$ and Π_{xx} are well defined and satisfy the following properties:

(1) the operator Π is linear and the approximation depends linearly from the particle point values;
(2) the approximation obtained by applying the least square method is consistent and the evaluation at the particle points gives the interpolating value. Therefore, $\Pi f(x_i) = f_i = f(x_i)$ for all $i = 1,2,\ldots,N$;
(3) From the above formalism we have $\Pi f_x(x) = \Pi_x f(x)$ and $\Pi f_{xx}(x) = \Pi_{xx} f(x)$.

The functions $f_h(x),\ (f_h(x))_x$, and $(f_h(x))_{xx}$ can be computed easily and accurately by using Taylor series expansion and least squares approximations. We use Taylor's expansion around the point x and then compute the coefficients by minimizing a weighted error over the neighboring points. The optimization is constrained to satisfy $\phi_h(x_1, x_1) = 1$ where x_1 is the closest point, i.e., the approximation must interpolate the closest point.

In order to approximate the function and its derivatives at x by using a quadratic approximation through the m neighboring points sorted with respect to its distance from x we let

$$f(x_i) = f_h(x) + (f_h(x))_x (x_i - x) + \frac{1}{2}(f_h(x))_{xx}(x_i - x)^2 + e_i, \tag{24}$$

where e_i is the error in the Taylor's expansion at the point x_i. The unknowns $f_h,\ (f_h)_x$ and $(f_h)_{xx}$ are computed by minimizing the error e_i for $i = 2,3,\ldots,m$ and setting the constraint $e_1 = 0$. Our method to solve this constrained least squares problem is straightforward. By subtracting the first equation with $e_1 = 0$ to all the other equations the system can be written as $e = Ma - b$, where

$$M = \begin{pmatrix} x_2 - x_1 & \frac{1}{2}(x_2 - x_1)^2 \\ x_3 - x_1 & \frac{1}{2}(x_3 - x_1)^2 \\ \vdots & \vdots \\ x_m - x_1 & \frac{1}{2}(x_m - x_1)^2 \end{pmatrix}, \tag{25}$$

$a = [(f_h)_x, (f_h)_{xx}]^T$, $b = [f_2 - f_1, f_3 - f_1, \ldots, f_m - f_1]^T$ and $e = [e_2, e_3, \ldots, e_m]^T$.

For $m > 2$, this system is over-determined for the two unknowns $(f_h)_x$ and $(f_h)_{xx}$.

The unknowns a are obtained from a weighted least squares method by minimizing the quadratic form

$$J = \sum_{i=1}^{m} w_i e_i^2 = (Ma - b)^T W (Ma - b), \tag{26}$$

where $W = \delta_{ij} w_i$. The minimization of J with respect to a formally yields

$$a = (M^T W M)^{-1} (M^T W) b. \tag{27}$$

Since $(M^T W M)^{-1}$ is a (2×2) matrix its inverse can be computed analytically. When we equate the components of the vector $a = [(f_h)_x, (f_h)_{xx}]^T$ to the components of the right hand side vector, we obtain the derivatives as linear combinations of the discrete neighbor values f_i in the form (22) and (23).

Now, from Eq. (24) for the closest point x_1 we can compute the value of $f_h(x)$ at x as

$$f_h(x) = f(x_1) - (x_1 - x)(f_h)_x - \frac{1}{2}(x_1 - x)^2 (f_h)_{xx}, \tag{28}$$

since $(f_h)_x$ and $(f_h)_{xx}$ are now known and we can express Eq. (28) in the form (21).

For an explicit solver, the above approximations of first and second order derivatives are sufficient. However, we use an implicit scheme and have to deal with the second order linear partial differential equations given in (13), (15) and (18). All these Eqs. (13), (15) and (18) have the following form of second order linear partial differential equation

$$Af + Bf_x + Cf_{xx} = g, \tag{29}$$

where the coefficients A, B, C are given real numbers and $g = g(x)$ is a given real valued function. We solve these equations with Dirichlet $f = f_\Gamma$ or Neumann boundary conditions

$$\frac{\partial f}{\partial n} = G \quad \text{on } \Gamma. \tag{30}$$

Eqs. (29) and (30) are solved by adding them as constraints in the least squares approximation (21). This yields big linear sparse systems of equations, where the matrix represents the discrete approximation of the differential operators involved, and the right hand side reflects the source terms. The linear system can be solved using any iterative methods. The method gives second order convergence [23].

4. Hybrid method

For the coupling procedure we use the particle methods for the Boltzmann and the Navier–Stokes equations described in the last section. Both methods are Lagrangian methods. Whereas the particle method for the Navier–Stokes equations is a meshfree method, the method for the Boltzmann equation needs a mesh, where molecules have to be sorted into cells in order to compute the intermolecular collisions. Our main goal is to develop a hybrid code, which switches automatically from one domain into another domain, where the respective methods are used, and vice versa with the help of some breakdown criteria for the Navier–Stokes equations.

We start by defining regular cells as usual for DSMC simulations. In the cell centers we store the macroscopic quantities obtained by averaging over the particles contained in the cell. As far as the Navier–Stokes equations are concerned we use the cell centers as starting positions for the corresponding particles. Then, we prescribe the initial conditions on each Navier–Stokes particle and solve the Navier–Stokes equations in the whole domain. Next, depending on a breakdown criterion, the cell may either remain a Navier–Stokes cell or it may be redefined as a Boltzmann cell which means that it may need to be subdivided and new particles have to be defined according to the requirements of the DSMC method. To predict the breakdown of the continuum equations many criteria are reported [7,8,22,28]. In this paper we use a breakdown criterion suggested in [35], where the distribution function $f(t, x, v)$ is written as a deviation from the local Maxwellian $f_M(t, x, v)$ according to (4). The size of the deviation ϕ is then estimated with the norm $\|\phi\|^2 = \int_{\mathbb{R}^3} \frac{f_M}{\rho} \phi^2 \, dv$. A local equilibrium can be assumed if $\|\phi\| \ll 1$. Using a first order expansion in ϵ one obtains the explicit expression (5) for ϕ and

$$\|\phi\| = \frac{1}{\rho R T} \left[\frac{2}{5} \frac{|q|^2}{RT} + \frac{1}{2} |\tau|^2 \right]^{1/2}. \tag{31}$$

From the point of view of solutions of the Navier–Stokes equations the quantities q and τ are given by (7). From the point of view of the Boltzmann equations we have

$$q = \frac{1}{2}\int_{\mathbb{R}3} c|c|^2 f\,dv, \quad \tau = \int_{\mathbb{R}3} c \cdot c^T f\,dv - \rho RTI.$$

The main difficulty is to find the cutoff value for the breakdown criteria. q and τ are of the order of the Knudsen number ϵ. Hence, the breakdown parameter is of the order of ϵ. Therefore, we use ϵ as cutoff value and define a cell as Navier–Stokes cell if $\|\phi\| \leqslant \epsilon$, otherwise it is a Boltzmann one.

4.1. Adaptive grid refinement

The spatial grid size for the Boltzmann solver has to be chosen of the order the mean free path λ. However, the grid size for the Navier–Stokes solver is independent of λ. This means that we usually need to subdivide a Boltzmann cell into smaller units to achieve the desired numerical accuracy. In the following, we refer to these units as Boltzmann subcells. Let N be the total number of Navier–Stokes particles for the unit interval $[0,1]$. The particle distance for the Navier–Stokes solver is given by $dx_{NS} = 1/N$. Due to the explicit movement of the particles the time step for the Navier–Stokes equations is restricted by

$$\Delta t_{NS} \leqslant \frac{dx_{NS}}{|u_{max}|}, \tag{32}$$

where u_{max} is the maximum fluid velocity. The number of Boltzmann subcells per Boltzmann cell is defined by

$$L = \max\left[int\left(\frac{dx_{NS}}{\lambda}\right) + 1, 2\right]. \tag{33}$$

This means, we define at least two Boltzmann subcells per Boltzmann cell. The Boltzmann grid size dx_B and time step Δt_B are defined by

$$dx_B = \frac{dx_{NS}}{L}, \quad \Delta t_B = \frac{\Delta t_{NS}}{L}. \tag{34}$$

The number L is the total number of time steps for the inner Boltzmann solver.

4.2. Coupling condition

If a Navier–Stokes cell is predicted to be a Boltzmann cell in the next time step, we use the macroscopic quantities $\rho,\ u,\ T,\ q,\ \tau$ from the Navier–Stokes solver and generate the Boltzmann particles according to the Chapman–Enskog distribution using the acceptance–rejection method suggested by Gracia and Alder [21]. Then, the Boltzmann cell is refined according to the size of the mean free path. The macroscopic quantities are stored in the cell centers of the coarse cells, and the same quantities are used to generate the Chapman–Enskog distribution in all finer cells. If the cell was a Boltzmann cell in the previous time step and the breakdown criterion determines it to be a Navier–Stokes one with a corresponding particle in its center in the current time step, we remove all Boltzmann particles from the cell.

In order to apply the boundary conditions for the Boltzmann equation, we have to define the boundary cells (or interface cells) between the Boltzmann and Navier–Stokes domains. The interface cells are, in fact, the Navier–Stokes cells which are adjacent to the Boltzmann domain. Hence, the boundary conditions for the Boltzmann equation at the interface are obtained by generating again the Boltzmann particles according to a Chapman–Enskog distribution from the Navier–Stokes values at the interface. Then we perform the Boltzmann simulation for L time steps. At the Lth time step of the Boltzmann simulation we compute the macroscopic quantities in each subcell and then take the average values in the (coarse) cell. The boundary conditions for the Navier–Stokes equations are chosen as Dirichlet boundary conditions, where the macroscopic quantities from the Boltzmann cells at the interface are used as boundary conditions. As in all DSMC codes there are some statistical fluctuations in the Boltzmann data. These fluctuating data destabilize the Navier–Stokes solver. Therefore, we need a smoothing operator. Here we have used the Shephard interpolation. For example, for the density at cell center x, the Shephard interpolation is defined as

$$\tilde{\rho} = \frac{\sum_{i=1}^{m} w_i \rho_i}{\sum_{i=1}^{m} w_i}, \tag{35}$$

where m is the number of neighbouring cell centers x_i, which are Boltzmann cells and w is the weight function given by (20). Similarly, we smooth u and T and then update μ and κ.

Initially, we define coarse regular cells in the domain. The center of these cells are the positions of the Navier–Stokes particles. Since Navier–Stokes particles move with their fluid velocity according to (12), they will be redistributed arbitrarily. In this case it is meaningful to reproject all Navier–Stokes solutions to the old positions and then consider the old positions as the current positions of particles. This is called particle remeshing and is used by several authors, see [16] and other references therein. The fluid quantities at the old positions are approximated from the neighbour particles at the new positions with the help of the least squares method. To reconstruct the flow field accurately one needs at least a second order least squares approximation. The solutions obtained from the remeshing particle method are consistent with the moving particle method as well as with corresponding analytical solutions [36,39].

4.3. Coupling algorithm

Summarising the above, we present the following coupling algorithm:

1. Define coarser cells with cell size dx_{NS} and define cell centers as NS particles. Prescribe initial value on NS particles.
2. Solve the NS equations in all NS cells for a time step Δt_{NS}. Boundary conditions are taken from the values of the Boltzmann cells (if there exist any).
3. Compute $\|\phi\|$ from (31) in all coarser cells and define

   $$\text{coarser cell} = \begin{cases} \text{NS}, & \text{if } \|\phi\| \leqslant \epsilon, \\ \text{Boltzmann}, & \text{otherwise}. \end{cases}$$

   Here ϵ is the local Knudsen number computed in each cell.
4. If flag of a cell changes from NS into Boltzmann, generate particles according to the Chapman–Enskog distribution.If flag of cell changes from Boltzmann into NS, remove all Boltzmann particles from the cell.
5. Refine the Boltzmann cells by $dx_B = dx_{NS}/L$ and choose a time step of $\Delta t_B = \Delta t_{NS}/L$.
6. Do for i = 1, L
   - (a) Generate particles on boundaries between Boltzmann and NS domains according to the Chapman–Enskog distribution.
   - (b) Free flow of Boltzmann particles over a time step Δt_B.
   - (c) If Boltzmann particles leave Boltzmann domain, delete them.
   - (d) Sort particles into Boltzmann subcells and perform intermolecular collisions.

     end do
7. Compute macroscopic quantities (moments) in Boltzmann subcells and store the mean values in the centers of coarser cells.
8. Smooth the macroscopic quantities in the Boltzmann cells.
9. Reproject Navier–Stokes particles onto old positions (coarser cell centers).
10. Goto 2 and repeat until the final time.

5. Numerical tests

We consider the classical Sod's 1D shock tube problem [32]. It is an unsteady situation with three regions, namely rarefaction, shock and contact discontinuity, where the continuum equations fail. We present three solutions: full Navier–Stokes, full Boltzmann and hybrid using the above described solvers. The hard sphere model for the collision cross section is considered for the Boltzmann equation. We consider three cases, where the Knudsen number ranges from 0.8 to 0.001. We have used $N = 200$ Navier–Stokes particles or coarse cells in all test cases. In order to compare the CPU time for both hybrid and full Boltzmann simulations the number of Boltzmann particles per cell are of the same order for both simulations. Moreover, we have used the following parameters, which are the same for all cases. The gas is chosen as Argon with a molecular mass $m = 6.63 \times 10^{-26}$ kg. Furthermore, the Boltzmann constant $k = 1.38 \times 10^{-23}\ \mathrm{J\,K^{-1}}$, the molecular diameter $d = 3.68 \times 10^{-10}$ m, the ratio of specific heats $\gamma = 5/3$. These parameters give the gas constant $R = 208\ \mathrm{J\,kg\,K^{-1}}$.

We choose $\Omega = [0, 1]$ as computational domain. Initially, there is a discontinuity in the density and temperature at $x = 0.5$. The velocity is taken zero everywhere. The parameters with the index l denote the fluid quantities in the left half interval [0,0.5] and those with the index r denote them in the right half interval (0.5,1]. The initial temperatures are $T_{0l} = 273.008012$ K and $T_{0r} = 273.00641$ K. These temperature values are equivalent to the internal energy considered in the classical Sod's problem [32]. In the following test cases, we vary the densities ρ_{0l} and ρ_{0r} such that we have different ranges of Knudsen numbers. We have simulated the problem up to the final time $t_{final} = 6.8 \times 10^{-4}$ s. On the boundaries the velocities are taken to be zero and the temperatures are equal to the initial values.

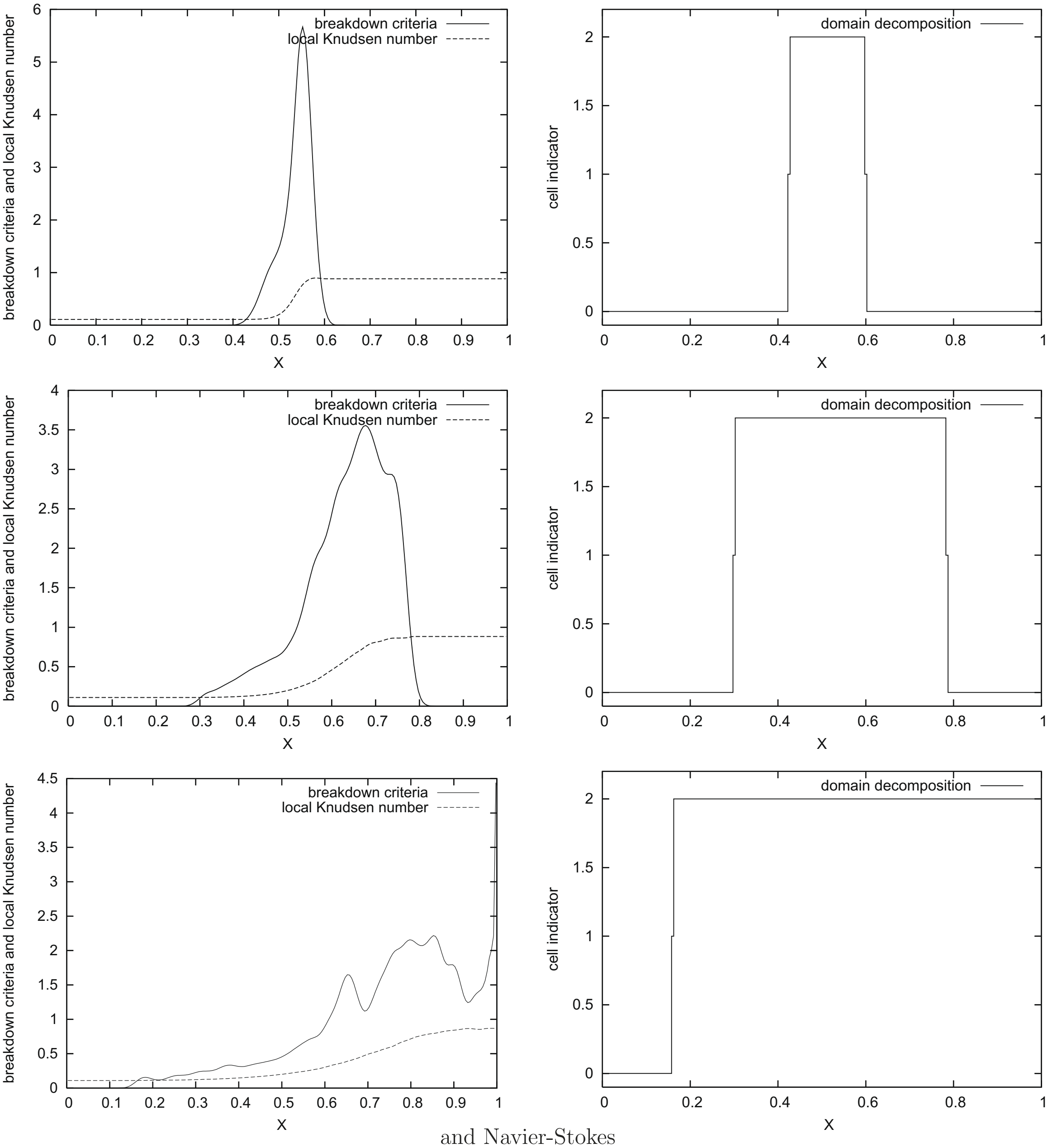
5.1. Test 1

We consider the first test case with the initial densities $\rho_{0l} = 1 \times 10^{-6}\ \mathrm{kg\,m^{-3}},\ \rho_{0r} = 0.125 \times 10^{-6}\ \mathrm{kg\,m^{-3}}$. This corresponds to the Knudsen number $\epsilon_l = 0.1$ and $\epsilon_r = 0.8$, where the characteristic length D is equal to 1. The Boltzmann cell size could be less or equal to the mean free path $\lambda_l = 0.1$. This corresponds to the order of 10 Boltzmann cells which gives a poor approximation. Since we have used 200 Navier–Stokes particles (or coarser cells) and the minimum number of Boltzmann subcells per coarser cell is $L = 2$, we consider 400 cells for the pure Boltzmann simulation. The number of Boltzmann particles is proportional to the density ρ. In the left half of the domain we used $n0 = 5000$ initial Boltzmann particles per coarser cell. The corresponding number of particles on the right half of the domain is given by

$$int\left(\frac{\rho_{0r}n0}{\rho_{0l}L}\right). \tag{36}$$

Similarly, in the hybrid code, if the cell becomes a Boltzmann cell, or in the interface boundary cells of the Boltzmann domain, the number of particles is given by

$$int\left(\frac{\rho(t)n0}{\rho_{0l}L}\right).$$



and Navier-Stokes

Fig. 1. Test 1: Breakdown criterion and the local Knudsen number for times $t = 8.9 \times 10^{-5}$ s (top left), $t = 3.709 \times 10^{-4}$ s (middle left) and $t = 6.8 \times 10^{-4}$ s (bottom left). Domains for Boltzmann and Navier–Stokes equations (the value 2 indicates the Boltzmann domain, the value 0 indicates the Navier–Stokes domain) for the same times.

The time step for the pure Boltzmann simulation is approximately 7.41×10^{-06}. The time step for the Navier–Stokes simulation is $\Delta t_{NS} = dx_{NS}/u_{max}$. In the first time step we solve the Navier–Stokes equations in all coarser cells. Then we compute the breakdown criterion $\|\phi\|$ and the local Knudsen number $\epsilon = \lambda$ from (2) in all cells. If $\|\phi\| > \epsilon$ the cell is defined as a Boltzmann cell, otherwise a Navier–Stokes one. In Fig. 1, we have plotted the time evolution of $\|\phi\|$ and ϵ. We also present the corresponding domain decomposition, where the cells with value 2 correspond to Boltzmann cells, those with value 0 the Navier Stokes cells. The cells of value 1 indicate the interface boundary cells, where we use the boundary conditions for the Boltzmann domain.

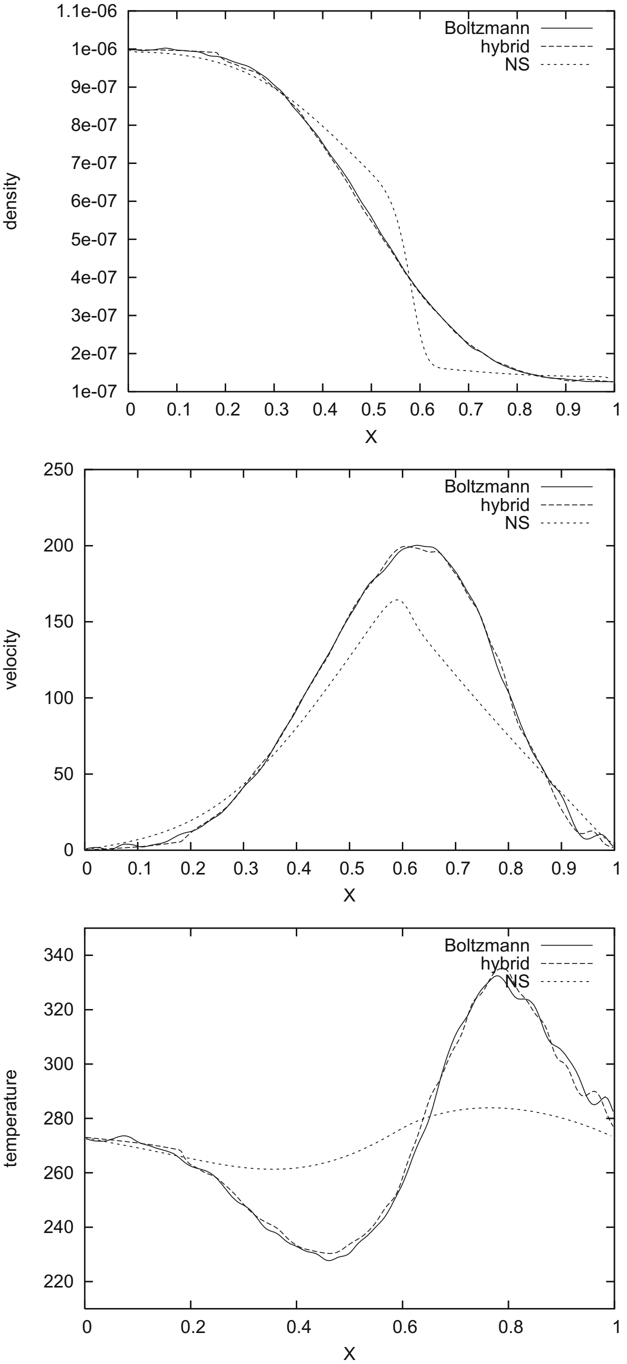


Fig. 2. Test 1: Density (top), velocity (middle) and temperature (bottom) at time $t = 6.8 \times 10^{-4}$ s for all three solvers and a maximum Knudsen number of 0.8.

In Fig. 2, we have plotted the solutions of the pure Navier–Stokes solver, the pure Boltzmann solver and of the hybrid solver. For the pure Boltzmann solver, we have smoothed the data according to the Shephard interpolation (35). As expected, Fig. 2 shows that the pure Navier–Stokes solution is far away from the pure Boltzmann and hybrid solutions. In this Knudsen regime the gas is too rarefied such that the pure Navier–Stokes equations give unphysical results.

5.2. Test 2

In this case we increase the density by a factor 10. We consider $\rho_{0l} = 1 \times 10^{-5}$ kg m^{-3}, $\rho_{0r} = 0.125 \times 10^{-5}$ kg m^{-3}. The corresponding Knudsen numbers are $\epsilon_l = 0.01$ and $\epsilon_r = 0.08$. The number of Boltzmann cells and Navier–Stokes particles are same as in the Test 1. Furthermore, the other parameters are also the same as in the Test 1. The time dependence of

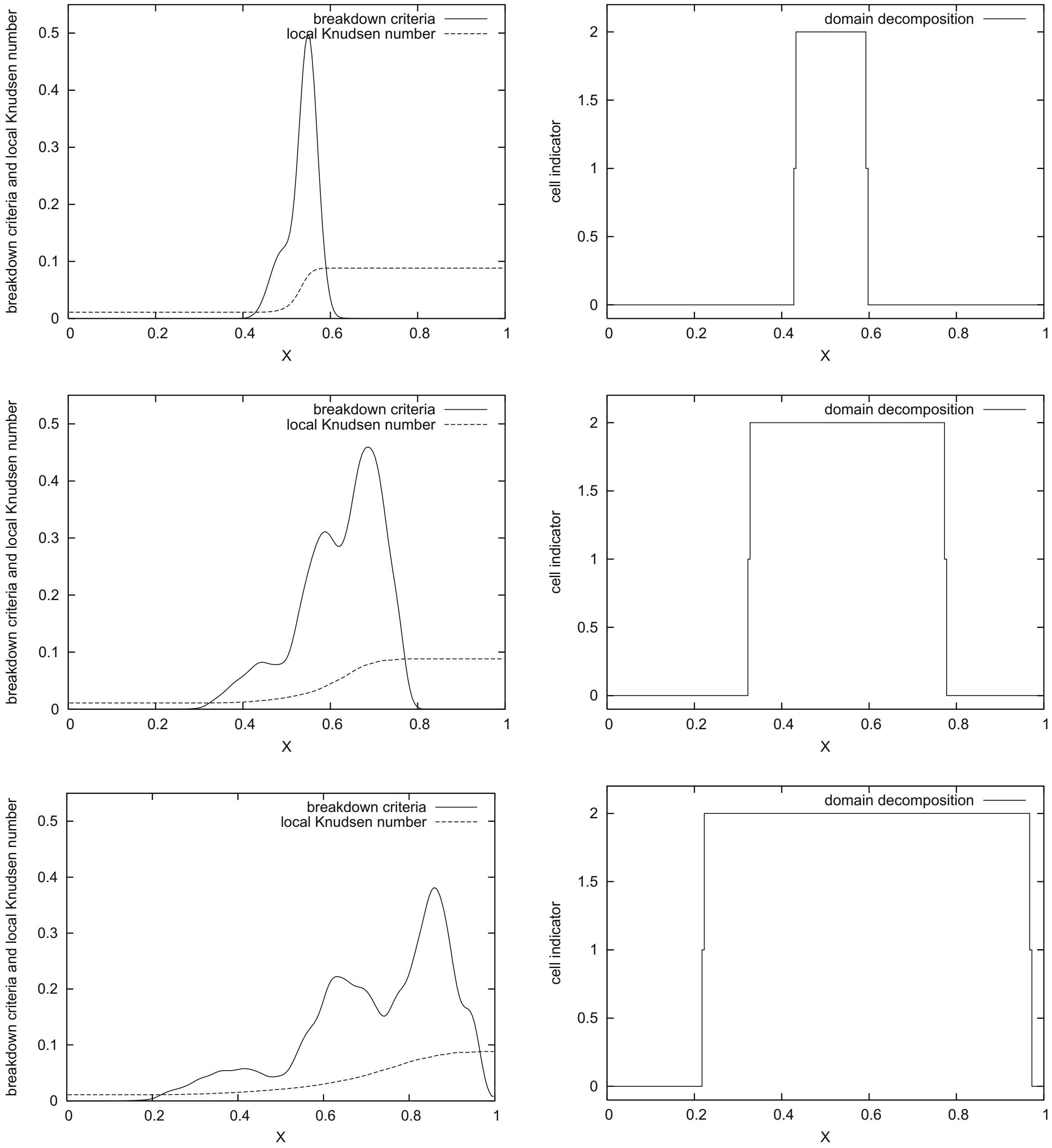


Fig. 3. Test 2: Breakdown criterion and the local Knudsen number for time $t = 8.9 \times 10^{-5}$ s (top left), $t = 3.709 \times 10^{-4}$ s (middle left) and $t = 6.8 \times 10^{-4}$ s (bottom left). Domains for Boltzmann and Navier–Stokes equations (the value 2 indicates the Boltzmann domain, the value 0 indicates the Navier–Stokes domain) for the same times.

the domain decomposition of the Boltzmann and Navier–Stokes is similar as in Test 1. In Fig. 3, we have plotted the breakdown criterion, the local Knudsen number and the corresponding domain decomposition at different times.

In Fig. 4, we have plotted the density, velocity and temperature for all three solvers. Again the agreement between the hybrid and the Boltzmann solutions is very good. However, the velocity and the temperature from the Navier–Stokes solver on the right half of the domain deviate slightly from both solutions. The reason is that we are still in the rarefied regime, where the local Knudsen number is of the order of 0.08.

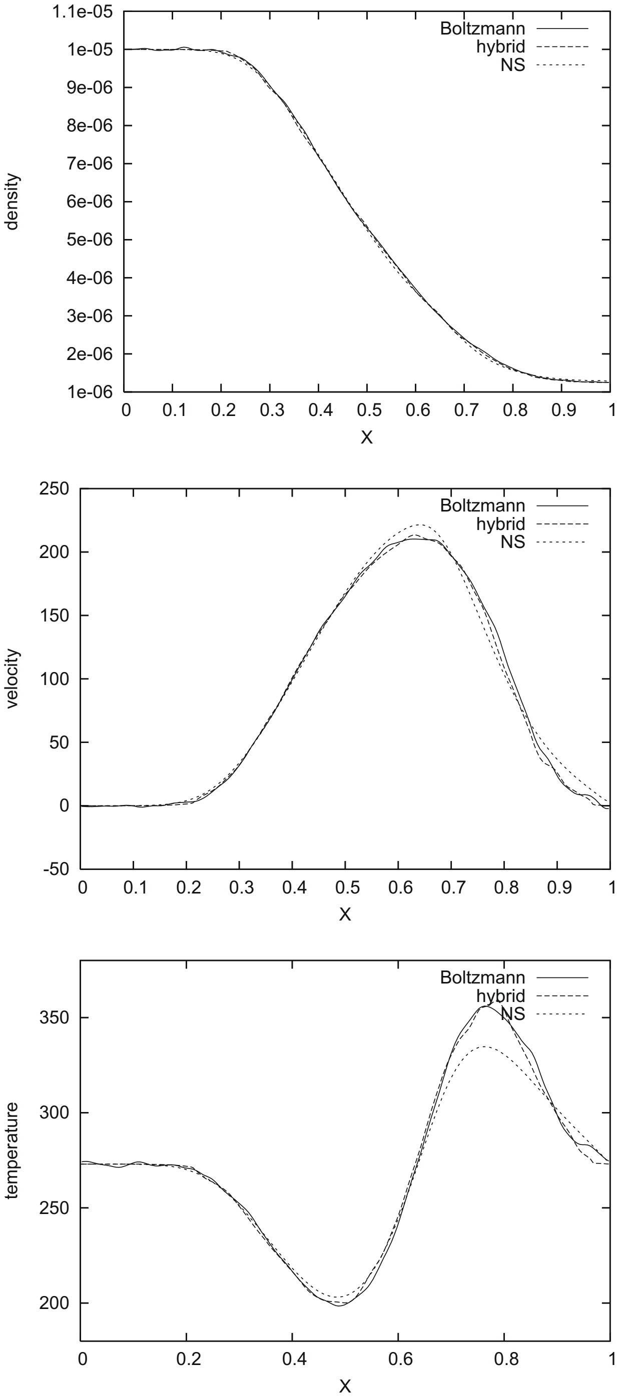


Fig. 4. Test 2: Density (top), velocity (middle) and temperature (bottom) at time $t = 6.8 \times 10^{-4}$ s for a maximum Knudsen number of 0.08.

5.3. Test 3

In our final test case we further increase the density by a factor of 10 compared to Test 2. Here the densities are taken as $\rho_{0l} = 1 \times 10^{-4}$ kg m^{-3}, $\rho_{0r} = 0.125 \times 10^{-4}$ kg m^{-3}. The corresponding Knudsen numbers are $\epsilon_l = 0.001$ and $\epsilon_r = 0.008$. The total number of Boltzmann cells equals 1200, which guarantees that the cell size is smaller than the mean free path. The total number of particles for the Boltzmann simulation is the same as in the earlier test cases. The other parameters are the same as in the earlier test cases. The refinement factor (36) yields in this case $L = 6$. This guarantees that the Boltzmann subcell size and time steps in the hybrid solver are the same as in the pure Boltzmann solver. In Fig. 5, we have plotted the breakdown criterion and local Knudsen number and the corresponding domain decomposition at different times. In this

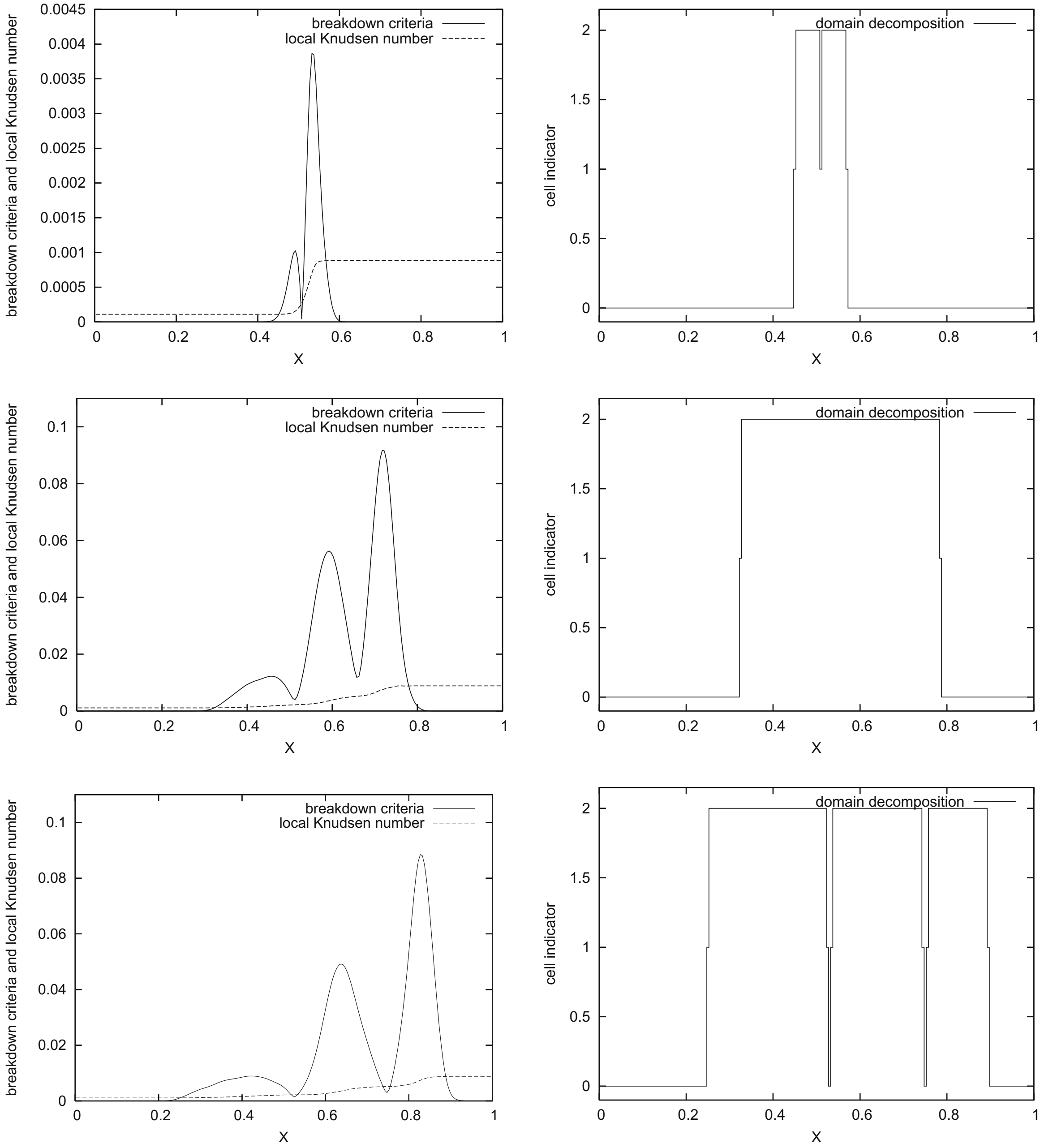


Fig. 5. Test 3: Breakdown criterion and local Knudsen number for time $t = 8.9 \times 10^{-5}$ s (top left), $t = 3.709 \times 10^{-4}$ s (middle left) and $t = 6.8 \times 10^{-4}$ s (bottom left). Right side: Domains for Boltzmann and Navier–Stokes equations (the value 2 indicates the Boltzmann domain, the value 0 the Navier–Stokes domain) for the same times.

range of Knudsen numbers we observe the separation of the Boltzmann and Navier–Stokes domain near the rarefaction and shock. For even smaller Knudsen numbers the Boltzmann domain decreases significantly. In Fig. 5, we observe some isles in the regions of contact discontinuity and shocks. These isles do not create any problems for explicit Navier–Stokes solvers. However, for the implicit one, presented above, some care must be taken and domains have to be smoothened. In the 1D case considered there are one or two cells in the isles which can be considered as Boltzmann cells. For higher dimensional cases this point has to be investigated in more detail.

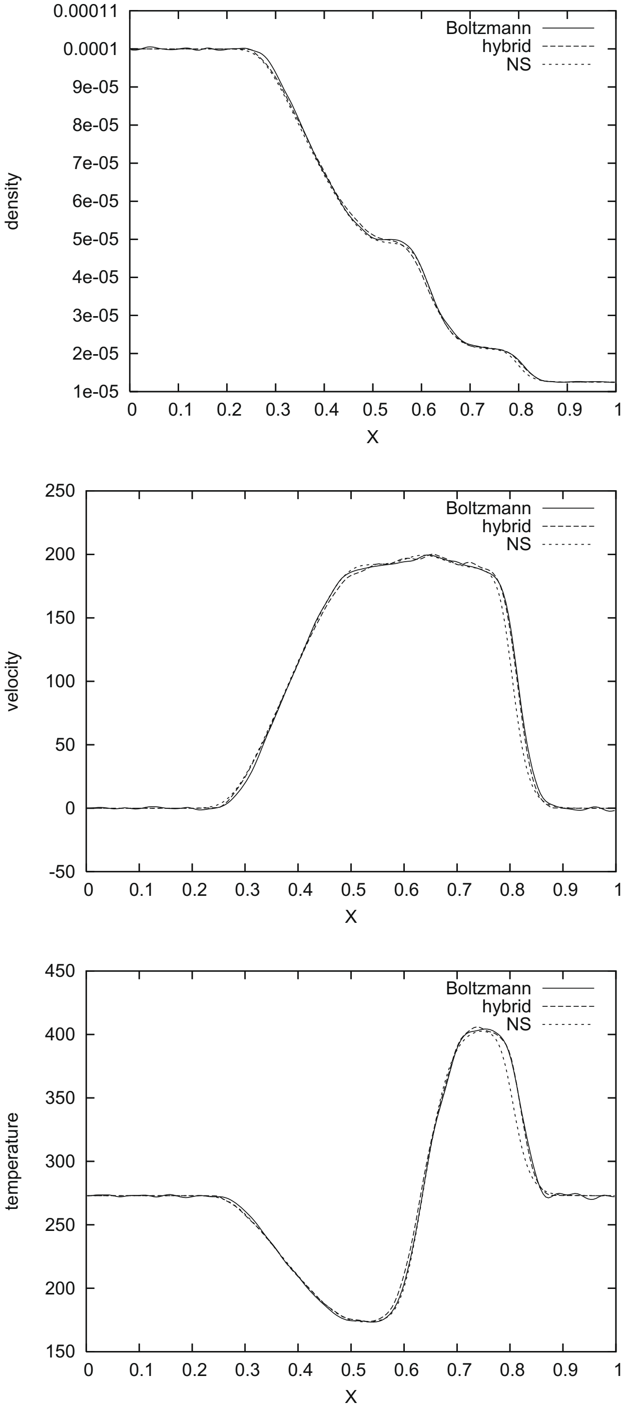


Fig. 6. Test 3: Density (top), velocity (middle) and temperature (bottom) at time $t = 6.8 \times 10^{-4}$ s for all three solvers and a maximum Knudsen number of 0.008.

Table 1
Comparison of CPU time.

Test cases	Boltzmann (s)	Hybrid (s)
Test 1	68	28
Test 2	70	25
Test 3	216	66

Finally, in Fig. 6 we have again plotted the density, velocity and temperature for all three solvers. In this case we observe close agreement for all solutions.

5.4. CPU time

In Table 1 the CPU times are shown for the hybrid and the Boltzmann solvers. The CPU time for the Navier–Stokes solver is negligible compared to the other two. The computation was carried out on a single processor Intel Xeon E5420 (2.5 GHz). For these tests the hybrid code is significantly faster than the Boltzmann solver, for example by more than a factor of 3 in Test 3.

6. Conclusion and outlook

We have presented a particle–particle hybrid method for time dependent problems. The Boltzmann equation is simulated by a variant of the DSMC method and the Navier–Stokes equations are solved by a meshfree Lagrangian particle method. To decompose the domains we use a breakdown criterion in each time step. The coupling between the two domains is done by sampling macroscopic field quantities from the particle ensembles in one direction and by creating particle ensembles from a Maxwell distribution with suitable parameters in the other direction. Sod's 1D shock tube problem was considered as a test case. A very satisfactory agreement between the solutions of the hybrid and the Boltzmann code was found. However, the hybrid code is more than 3 times faster than the pure Boltzmann code.

Future work will be concentrating on the extension of the code to higher dimensional problems as well as to special micro- and nanoscale geometries.

References

[1] O. Aktas, N.R. Aluru, A combined continuum/DSMC technique for multiscale analysis of microfluidic filters, J. Comput. Phys. 178 (2002) 342–372.
[2] H. Babovsky, A convergence proof for Nanbu's Boltzmann simulation scheme, Eur. J. Mech. 8 (1989) 41.
[3] H. Babovsky, R. Illner, A convergence proof for Nanbu's simulation method for the full Boltzmann equation, SIAM J. Numer. Anal. 26 (1989) 45.
[4] C. Bardos, F. Golse, D. Levermore, Fluid dynamic limits of kinetic equations, JSP 63 (1991) 323–344.
[5] M. Bennoune, M. Lemou, L. Mieussens, Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible Navier–Stokes asymptotics, J. Comput. Phys. 277 (2008) 3781–3803.
[6] G.A. Bird, Molecular Gas Dynamics and Direct Simulation of Gas Flows, Oxford University Press, New York, 1994.
[7] G.A. Bird, Breakdown of translational and rotational equilibrium in gaseous expansions, AIAA J. 8 (1970).
[8] I.D. Boyd, G. Chen, C.V. Candler, Predicting failure of the continuum fluid equations in transitional hypersonic flows, Phys. Fluid 7 (1995) 210.
[9] J.F. Bourgat, P. LeTallec, B. Perthame, Y. Qiu, Coupling Boltzmann and Euler equations without overlapping, Domain Decomposition Methods in Science and Engineering, Contemporary Mathematics, vol. 157, AMS, Providence, RI, 1994, pp. 377–398.
[10] J.F. Bourgat, P. LeTallec, M.D. Tidriri, Coupling Boltzmann and Navier–Stokes equations by friction, J. Comput. Phys. 127 (1996) 227.
[11] R. Caflish, The fluid dynamical limit of the nonlinear Boltzmann equation, CPAM 33 (1980) 651.
[12] C. Cercignani, R. Illner, M. Pulvirenti, The Mathematical Theory of Dilute Gases, Springer, 1994.
[13] A. Chorin, Numerical solution of the Navier–Stokes equations, J. Math. Comput. 22 (1968) 745–762.
[14] N. Crouseilles, P. Degond, M. Lemou, A hybrid kinetic/fluid models for solving the gas dynamics Boltzmann–BGK equation, J. Comput. Phys. 199 (2004) 776.
[15] S. Chen, Weinan E, Y. Liu, C.-W. Shu, A discontinuous Galerkin implementation of a domain decomposition method for kinetic hydrodynamic coupling multiscale problems in gas dynamics and device simulations, J. Comput. Phys. 225 (2007) 1314–1330.
[16] A.K. Chaniotis, D. Poulikakos, P. Koumoutsakos, Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows, J. Comput. Phys. 182 (2002) 67–90.
[17] H.A. Carlson, R. Roveda, I.D. Boyd, G.V. Candler, A hybrid CFD–DSMC method of modeling continuum-rarefied flows, AIAA Paper 2004-1180.
[18] P. Degond, G. Dimarco, L. Mieussens, A moving interface method for dynamic kinetic–fluid coupling, J. Comput. Phys. 227 (2007) 1176–1208.
[19] S.M. Deshpande, A second order accurate kinetic theory based method for inviscid compressible flows, Tech. Paper 2613, NASA-Langley Research Center, Hampton, VA, 1986.
[20] R.A. Gingold, J.J. Monaghan, Smoothed Particle Hydrodynamics: theory and application to non-spherical stars, Mon. Not. Roy. Astron. Soc. 181 (1977) 375–389.
[21] A.L. Gracia, B.J. Alder, Generation of the Chapmann–Enskog distribution, J. Comput. Phys. 140 (1998) 66–70.
[22] A.L. Gracia, J.B. Bell, W.Y. Crutchfield, B.J. Adler, Adaptive mesh and algorithm refinement using direct simulation Monte Carlo, J. Comput. Phys. 154 (1999) 134.
[23] O. Iliev, S. Tiwari, A generalized (meshfree) finite difference discretization for elliptic interface problems, in: I. Dimov, I. Lirkov, S. Margenov, Z. Zlatev (Eds.), Numerical Methods and Applications, Lecture Notes in Computer Science, 2002.
[24] S. Kaniel, A kinetic model for the compressible flow equations, Indiana Univer. Math. J. 37 (3) (1998) 537–563.
[25] A. Klar, Domain decomposition for kinetic problems with noneqilibrium states, Eur. J. Mech. B: Fluid 15 (2) (1996) 203–216.
[26] V.I. Kolobov, R.R. Arslanbekov, V.V. Aristov, A.A. Frolova, S.A. Zabelok, Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement, J. Comput. Phys. 223 (2) (2007) 589–608.
[27] P. Le Tallec, F. Mallinger, Coupling Boltzmann and Navier–Stokes equations by half fluxes, J. Comput. Phys. 136 (1997) 51–67.

[28] D. Levermore, W.J. Morokoff, B.T. Nadiga, Moment realizability and the validity of the Navier–Stokes equations for rarefied gas dynamics, Phys. Fluid 10 (12) (1998).
[29] H. Neunzert, J. Struckmeier, Particle methods for the Boltzmann equation, Acta Numer. (1995) 417.
[30] T. Ohsawa, T. Ohwada, Deterministic hybrid computation of rarefied gas flows, RGD23, in: Andrew D. Ketsdever, E.P. Munz (Eds.), Rarefied Gas Dynamics: 23rd International Symposium, AIP Conference Proceedings, vol. 663, 2003, pp. 931–938.
[31] T.E. Schwartzentruber, I.D. Boyd, A hybrid particle–continuum method applied to shock waves, J. Comput. Phys. 215 (2006) 402–416.
[32] G.A. Sod, A survey of several finite difference methods for systems on nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31.
[33] Y. Sone, Molecular Gas Dynamics, Theory, Techniques and Applications, Birkhaueser, 2007.
[34] Q. Sun, I.D. Boyd, G.V. Candler, A hybrid continuum/particle approach for modeling subsonic, rarefied gas flows, J. Comput. Phys. 194 (2004) 256–277.
[35] S. Tiwari, Coupling of the Boltzmann and Euler equations with automatic domain decomposition, J. Comput. Phys. 144 (1998) 710–726.
[36] S. Tiwari, A LSQ-SPH approach for solving compressible viscous flows, in: Freistüler, Warnecke (Eds.), International Series of Numerical Mathematics, vol. 141, Birkhäuser, 2001.
[37] S. Tiwari, A. Klar, An adaptive domain decomposition procedure for Boltzmann and Euler equations, J. Comput. Appl. Math. 90 (1998) 233.
[38] S. Tiwari, J. Kuhnert, Modeling of two phase flows with surface tension by Finite Pointset Method (FPM), J. Comput. Appl. Math. 203 (2007) 376–386.
[39] S. Tiwari, S. Manservisi, Modeling incompressible Navier–Stokes flows by least squares approximation, Nepali Math. Sci. Rep. 20 (1) (2002).
[40] D.C. Wadsworth, D.A. Erwin, Two-dimensional hybrid continuum/particle approach for rarefied flows, AIAA Paper 92-2975, 1992.
[41] H.S. Wijesinghe, N.G. Hadjiconstantinou, A discussion of hybrid atomistic-continuum methods for multiscale hydrodynamics, Int. J. Multiscale Comput. Eng. 2 (2004) 189.