Journal of Computational Physics 256 (2014) 17–33

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Construction and comparison of parallel implicit kinetic solvers in three spatial dimensions

Vladimir Titarev [a,b,*], Michael Dumbser [c], Sergey Utyuzhnikov [d,b]

[a] Dorodnicyn Computing Centre of Russian Academy of Sciences, Moscow, Russia
[b] Moscow Institute of Physics and Technology, Moscow, Russia
[c] University of Trento, Trento, Italy
[d] University of Manchester, Manchester, UK

ARTICLE INFO

Article history:
Received 23 April 2013
Received in revised form 17 August 2013
Accepted 26 August 2013
Available online 5 September 2013

Keywords:
Kinetic
S-model
Unstructured
Micro
Rarefied
Implicit
Re-entry vehicle
Parallel

ABSTRACT

The paper is devoted to the further development and systematic performance evaluation of a recent deterministic framework Nesvetay-3D for modelling three-dimensional rarefied gas flows. Firstly, a review of the existing discretization and parallelization strategies for solving numerically the Boltzmann kinetic equation with various model collision integrals is carried out. Secondly, a new parallelization strategy for the implicit time evolution method is implemented which improves scaling on large CPU clusters. Accuracy and scalability of the methods are demonstrated on a pressure-driven rarefied gas flow through a finite-length circular pipe as well as an external supersonic flow over a three-dimensional re-entry geometry of complicated aerodynamic shape.

1. Introduction

In the last ten years or so there has been rapid development of explicit numerical methods and associated computer codes for solving the kinetic equations of the rarefied gas dynamics in three space dimensions [20,17,2,13]. A more recent approach is the high-order unstructured implicit Nesvetay-3D framework [38,39]. The main advantages of Nesvetay-3D over other existing three-dimensional numerical methods for kinetic equations are two-fold. Firstly, its ability to use arbitrary unstructured meshes, comprising not only tetrahedrons, but also elements of other shapes, makes Nesvetay-3D suitable for industrial problems with complex geometries and various flow regimes. Secondly, the use of the efficient one-step implicit time evolution method significantly accelerates convergence for steady-state problems. A recent application of the method is the calculation of the flow in long finite-length pipes across a wide range of Knudsen numbers [42,41]. Results were provided for length to radius ratios up to fifty, which so far has not been achieved by using other methods and codes.

The current implementation of the framework on parallel computers uses the Message Passing Interface (MPI) and is based on the decomposition of the molecular velocity mesh while keeping the spatial mesh as a single block. Such approach is simple and strictly equivalent to the sequential implementation. However, it has restrictions on the size of the spatial mesh on the most of the existing high-performance computing systems. This is because the data structure of the complete spatial mesh is stored at each processor and the amount of the required memory for its storage does not decrease with

* Corresponding author at: Dorodnicyn Computing Centre of Russian Academy of Sciences, Moscow, Russia.
E-mail addresses: titarev@ccas.ru (V. Titarev), michael.dumbser@ing.unitn.it (M. Dumbser), s.utyuzhnikov@manchester.ac.uk (S. Utyuzhnikov).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.08.051

the increasing number of processors used. Another problem is that the use of the data reduction operators can affect the scalability of the parallel code, see e.g. [40] for the studies of the analogous methods in two space dimensions.

The present work is a follow-on to [39] and has two objectives. Firstly, the developments of the existing implicit parallel method over the last four years are reviewed and further improvements are introduced. Secondly, the multi-block version of the numerical framework is presented, which does not use the data exchange between blocks during the solution of the linear system of equations for the time increments in the implicit scheme. Although the resulting parallel method is not strictly equivalent to the single-block method, the resulting method is free of the problems of the single-block parallel implementation and is simple enough for practical implementation.

The convergence properties and parallel scalability of various versions of the method are compared on two test problems. The first problem concerns rarefied gas flow through a short circular pipe, connecting two reservoirs with gas under different pressures. The difficulty here is to compute the solution with satisfactory accuracy across all degrees of gas rarefaction from the free-molecular to nearly-continuum flow. In the present work for the first time a spatial mesh convergence study is presented, using three consequently refined meshes. The results of three advection schemes are compared with one another and with the well-resolved DSMC calculations [48]. It is known that the use of mesh partitioning may seriously degrade the convergence properties of the implicit time-marching schemes [34]. The results are presented that demonstrate the behaviour of the present implicit method for multi-block meshes.

The second problem concerns an external supersonic flow over a model winged re-entry space vehicle (RSV), proposed by Central Aerohydrodynamic Institute (TsAGI). The model has a rather complex shape, which includes a blunt fuselage, swept wings, keel and flap. Recently, the aerodynamics of this vehicle has been extensively studied on the basis of the compressible Euler equations for a wide range free-stream Mach numbers [43,44]. In the present work the rarefied regime of the flow is examined for a moderate free-stream velocity for the conditions, approximately corresponding to 100 km of altitude. To the best of our knowledge, it is the first time when the flow over such a complex vehicle has been computed using the kinetic equations.

The selected test problems require the ability to handle three-dimensional geometries and flow features with steep gradients at very different degrees of rarefaction. They are thus very suitable for testing the accuracy and robustness of the numerical methods for kinetic equations.

The rest of the paper is organized as follows. The governing equations are presented in Section 1. The sequential numerical algorithm is described in Section 2. The parallel strategies are discussed in Section 4. Numerical results are presented in Section 5 and conclusions are drawn in Section 6.

2. Governing equations

The present work concerns the monatomic rarefied gas flows. A three-dimensional state of the rarefied gas is determined by the velocity distribution function $f(\boldsymbol{x},\boldsymbol{\xi})$, where $\boldsymbol{\xi}=(\xi_1,\xi_2,\xi_3)$ are the components of the molecular velocity vector in the spatial directions $\boldsymbol{x}=(x_1,x_2,x_3)=(x,y,z)$. Let l_*, p_*, T_*, μ_* be characteristic scales of length, pressure, temperature and viscosity, respectively; $\beta_*=\sqrt{2kT_*/m}$ is used as the characteristic scale of velocity, $t_*=l_*/\beta_*$ is the scale of temporal variable. Here m is mass of a molecule, k is the Boltzmann constant. The non-dimensional macroscopic quantities, such as number density n, temperature T, mean velocity $\boldsymbol{u}=(u_1,u_2,u_3)$ and heat flux $\boldsymbol{q}=(q_1,q_2,q_3)$ vectors are defined as the integrals of the velocity distribution function with respect to the molecular velocity:

$$\begin{pmatrix} n \\ n\boldsymbol{u} \\ n(\frac{3}{2}T+u^2) \\ \boldsymbol{q} \end{pmatrix} = \int \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \xi^2 \\ \frac{1}{2}\boldsymbol{v}v^2 \end{pmatrix} f\,d\boldsymbol{\xi}, \tag{1}$$

where $u^2=u_\alpha u_\alpha$, $\xi^2=\xi_\alpha\xi_\alpha$, $v^2=v_\alpha v_\alpha$, $d\boldsymbol{\xi}=d\xi_1\,d\xi_2\,d\xi_3$.

In the non-dimensional variables the Boltzmann equation with the S-model collision integral [29,30] for the distribution function f has the following form:

$$\begin{aligned} &\xi_1\frac{\partial f}{\partial x}+\xi_2\frac{\partial f}{\partial y}+\xi_3\frac{\partial f}{\partial z}=\nu\left(f^{(S)}-f\right),\quad \nu=\frac{nT}{\mu}\delta,\quad \delta=\frac{l_*p_*}{\mu_*\beta_*},\\ &f^{(S)}=f_M\left[1+\frac{4}{5}(1-\mathrm{Pr})S_\alpha c_\alpha\left(c^2-\frac{5}{2}\right)\right],\quad f_M=\frac{n}{(\pi T)^{3/2}}\exp\left(-c^2\right),\\ &v_i=\xi_i-u_i,\quad c_i=\frac{v_i}{\sqrt{T}},\quad c^2=c_\alpha c_\alpha,\quad S_i=\frac{2q_i}{nT^{3/2}}. \end{aligned} \tag{2}$$

Here rarefaction parameter δ defines the degree of gas rarefaction and is inversely proportional to the Knudsen number Kn; summation of over the repeated Greek indexes is assumed. For a monatomic gas the Prandtl number $\mathrm{Pr}=2/3$. The hard-sphere intermolecular interaction $\mu=\sqrt{T}$ is used in all calculations.

The kinetic equation (2) has to be augmented with the boundary conditions. Let $\boldsymbol{n}=(n_1,n_2,n_3)$ be the unit normal vector to a boundary surface, pointing in the outward direction to the surface. Assuming the diffuse molecular scattering

boundary condition with complete thermal accommodation to the surface temperature T_w, the distribution function of reflected molecules is given by:

$$f(t,\boldsymbol{x},\boldsymbol{\xi}) = \frac{n_w}{(\pi T_w)^{3/2}} \exp\left(-\frac{\xi^2}{T_w}\right), \quad \xi_n = \xi_\alpha n_\alpha > 0. \tag{3}$$

The number density of reflected molecules n_w is found from impermeability condition stating that the mass flux through the walls is equal to zero:

$$n_w = N_i/N_r, \quad N_i = -\int\limits_{\xi_n<0} \xi_n f\, d\boldsymbol{\xi}, \; N_r = +\int\limits_{\xi_n>0} \xi_n \frac{1}{(\pi T_w)^{3/2}} \exp\left(-\frac{\xi^2}{T_w}\right) d\boldsymbol{\xi}. \tag{4}$$

On the inflow and outflow boundaries of the spatial domain the distribution function of the incoming molecules is prescribed as the locally-Maxwellian one with the corresponding values of the macroscopic variables.

3. Sequential single-block method of solution

The steady-state solution of the six-dimensional boundary-value problem (2)–(4) is found by means of the implicit time-marching algorithm, developed in [38,39]. The first step in the numerical solution procedure is to replace the infinite domain of integration in the molecular velocity space $\boldsymbol{\xi}$ by a finite computational domain, which is then discretized using the non-uniform mesh with $N_{\xi_1} \cdot N_{\xi_2} \cdot N_{\xi_3} \equiv N_\xi$ cells. Both Cartesian and cylindrical coordinate systems can be used in the velocity space, depending on application. The velocity distribution function is then defined in centres $\boldsymbol{\xi}_\alpha$ of the resulting velocity mesh. The kinetic equation (2) is replaced by a system of N_ξ time-dependent advection equations for each of f_α:

$$\frac{\partial}{\partial t} f_\alpha = -\boldsymbol{\xi}_\alpha \nabla f_\alpha + J_\alpha, \quad J_\alpha = \nu\left(f^{(S)} - f\right)_\alpha, \tag{5}$$

which are connected by the macroscopic parameters in the function $f^{(S)}$ from the model collision integral J. Here ∇ is the gradient operator in the physical space (x, y, z).

3.1. Calculation of macroscopic quantities

The direct approximation of expressions (1) for macroscopic quantities yields a non-conservative numerical method that violates the discrete mass, momentum, and energy conservation laws. A detailed analysis of this problem is provided in [37]. There are several approaches for construction of the conservative methods for model kinetic equations, see [39] for a short discussion on the subject. The present work uses the most recent method [36,37], which is based on the direct numerical approximation of the integral conditions used in the derivation of the S-model collision integral [29,30]. Let ω_α be the weights of the composite quadrature rule used for integration in $\boldsymbol{\xi}$ space. In each spatial cell the eight macroscopic quantities are found as a solution of the following system of equations (subscript i of the spatial mesh is omitted for simplicity):

$$\boldsymbol{H}(\boldsymbol{W}) = \sum_\alpha \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \xi^2 \\ \boldsymbol{v}v^2 \end{pmatrix}_\alpha \left(f_\alpha^{(S)} - f_\alpha\right)\omega_\alpha + \begin{pmatrix} 0 \\ \boldsymbol{0} \\ 0 \\ 2\,\mathrm{Pr}\,\boldsymbol{q} \end{pmatrix} = \boldsymbol{0}, \quad \boldsymbol{W} = \begin{pmatrix} n \\ \boldsymbol{u} \\ T \\ \boldsymbol{q} \end{pmatrix}. \tag{6}$$

The system is solved by means of the Newton iterations of the form:

$$M\left(\boldsymbol{W}^{s-1}\right)\left(\boldsymbol{W}^s - \boldsymbol{W}^{s-1}\right) = -\boldsymbol{H}\left(\boldsymbol{W}^{s-1}\right), \quad s = 1, 2, \ldots, \; M = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{W}}. \tag{7}$$

The initial guess $\boldsymbol{W}^0$ for the iterations is obtained from the direct quadrature discretisation of (1).

In the special case $\mathrm{Pr} = 1$ (BGK model [5]) the function $f^{(S)}$ no longer contains the heat flux vector making the last three equations in (6) unnecessary. If these three equations are omitted, the procedure (6) for macroscopic parameters coincides with the ones proposed in [23,24,11] from different considerations for the BGK model.

The difficult part of the Newton method (7) is the calculation of the Jacobian matrix M, which is given by the discrete integral sums of the derivatives of $f^{(S)}$ with the respect to the macroscopic variables. The same applies to methods proposed in [24,11]. The exact evaluation of M is both costly and cumbersome to code, more so for more complicated kinetic models, such as the R-model of the diatomic gas [28,19]. To make this more computationally efficient, the numerical integration in the expression for M is replaced by the exact (analytical) one so that the Jacobian matrix is approximately expressed as an explicit function of macroscopic variables:

$$M \approx \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_1 & n & 0 & 0 & 0 & 0 & 0 & 0 \\ u_2 & 0 & n & 0 & 0 & 0 & 0 & 0 \\ u_3 & 0 & 0 & n & 0 & 0 & 0 & 0 \\ \frac{3}{2}T + u_\alpha u_\alpha & 2nu_1 & 2nu_2 & 2nu_3 & \frac{3}{2}n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{7}{3} - \mathrm{Pr} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{3} - \mathrm{Pr} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{3} - \mathrm{Pr} \end{pmatrix}. \tag{8}$$

The use of the approximation (8) simplifies the numerical algorithm at the cost of some degradation in the convergence speed. Although the convergence to the solution is no longer quadratic, the resulting iterative procedure requires significantly less CPU time to converge up to $\delta = 1000$ than the method based on the exact evaluation of the Jacobian matrix.

3.2. Advection scheme

The next step in the description of the numerical method is to outline the procedure to solve each of the kinetic equations (5) assuming the model collision integral is known from Eq. (6). Introduce in the physical variables a computational mesh consisting of N_{space} elements (spatial cells) V_i. Let $|V_i|$ be the cell volume, $|A|_{il}$ area of face l, $\Delta t = t^{n+1} - t^n$ time step, $f^n_{\alpha i}$ the spatial average of the velocity distribution function in the cell V_i at time t^n for the molecular velocity $\boldsymbol{\xi}_\alpha$, $g^n_{\alpha i} = f^{n+1}_{\alpha i} - f^n_{\alpha i}$ time increment of the distribution function. Also denote by $\sigma_l(i)$ the cell index of the cell adjacent to the face l of cell V_i. For the given index α of the velocity mesh the implicit finite-volume method is written as

$$g^n_{\alpha i} = \Delta t L^{n+1}_{\alpha i}, \quad L^{n+1}_{\alpha i} = -\frac{1}{|V_i|}\sum_l \Phi^{n+1}_{\alpha il} + J^{n+1}_{\alpha i}, \ i = 1, 2, \ldots N_{\mathrm{space}}. \tag{9}$$

Here $\Phi^{n+1}_{\alpha il}$ is the high-order accurate approximation to the interface flux for the face l of the cell i. The right-hand side of the scheme (9) is linearized around the lower time level $t = t^n$. Two conventional simplifications are made in the calculation of the derivatives of L in the left side of the scheme: (i) the face fluxes are assumed to be first-order ones; (ii) the complicated dependence of the macroscopic variables on the discrete values of the distribution function via (6) is ignored. Then, the general implicit scheme (9) becomes a system of N_{space} linear equations for the temporal increment of the solution $g^n_{\alpha i}$:

$$\begin{aligned} &D_{\alpha i} g^n_{\alpha i} + \sum_l c_{\alpha i \sigma_l(i)} g^n_{\alpha \sigma_l(i)} = \Delta t L^n_{\alpha i}, \\ &D_{\alpha i} = 1 + \Delta t \nu^n_i + \frac{1}{2}\Delta t \sum_l \xi_{\alpha nl}(1 + \operatorname{sign}\xi_{\alpha nl})\frac{|A_{il}|}{|V_i|}, \\ &c_{\alpha i \sigma_l(i)} = \frac{1}{2}\Delta t \xi_{\alpha nl}(1 - \operatorname{sign}\xi_{\alpha nl})\frac{|A_{il}|}{|V_i|}, \end{aligned} \tag{10}$$

where $\xi_{\alpha nl}$ is the projection of the vector $\boldsymbol{\xi}_\alpha$ onto the outward unit normal to the face l of the cell V_i. If the face l of the cell belongs to a computational boundary, then the corresponding temporal increment of the distribution function is set to zero

$$g^n_{\alpha \sigma_l(i)} = 0. \tag{11}$$

In the right-hand side of the scheme the advection operator acting on the values of the velocity distribution function on the lower time level, is approximated as a sum of the intercell fluxes, which are calculated according to the conventional formula

$$\begin{aligned} &\Phi^n_{\alpha il} = \frac{\xi_{\alpha nl}|A_{il}|}{2}\left(f^- + f^+ - \operatorname{sign}(\xi_{\alpha nl})\left(f^+ - f^-\right)\right), \\ &f^- = f^n_{\alpha il}, \quad f^+ = f^n_{\alpha, \sigma_l(i), l_1}. \end{aligned} \tag{12}$$

Here l_1 is the number of the face of the cell $\sigma_l(i)$, adjacent to the face l of the cell i. The general formula (12) is modified if the face l is adjacent to a boundary surface and $\xi_{\alpha nl} < 0$ by applying the corresponding boundary condition.

3.3. Spatial reconstruction procedure

The calculation of the numerical fluxes $\Phi^n_{\alpha il}$ as defined by Eq. (12) requires the knowledge of the face averages of the distribution function $f^n_{\alpha il}$. Here the three different approaches are considered. For the first-order accurate method it is sufficient to set these values equal to the cell value

$$f^n_{\alpha il} = f^n_{\alpha i}, \quad l = 1, 2, \ldots. \tag{13}$$

Such procedure is still used in some methods [13]. It is well known, however, that the first-order method is quite inaccurate. There are two possibilities to obtain a higher-order accurate approximations to $f^n_{\alpha il}$. The original method [38,39] employs a high-order fully multidimensional spatial discretization based on the least-square reconstruction. In each spatial cell V_i the distribution function f_α is approximated locally by the piece-wise linear reconstruction polynomial

$$f \approx f^n_{i\alpha} + a_1 e_1(x) + a_2 e_2(x) + \cdots, \quad x \in V_i. \tag{14}$$

Here $e_k(x)$ are the polynomial spatial basis functions with zero mean, the coefficients a_k (degrees of freedom) are computed using the values of $f^n_{i_m\alpha}$ in the cells V_{i_m} from the so-called spatial reconstruction stencil. Here $m = 0, \ldots, M$ is the local spatial cell number in the stencil, with corresponding cells with spatial index i_m. In order to avoid non-physical (spurious) oscillations at discontinuities the reconstruction procedure is made non-linear (solution adaptive) by introducing the so-called slope limiter. The use of slope limiters was pioneered by Kolgan [14–16] and van Leer [45] for structured meshes (see also [46,27]) and subsequently extended to the unstructured meshes by other authors [35,4]. The final values $f^n_{i\alpha l}$, used in the calculations, are expressed via the unlimited face averages $p^n_{i\alpha l}$ of the reconstruction polynomial as

$$f^n_{i\alpha l} = f^n_{i\alpha} + \psi^n_{i\alpha} \cdot \left(p^n_{i\alpha l} - f^n_{i\alpha}\right), \quad p^n_{i\alpha l} = \sum_{m=0}^{M} \omega_{iml} f^n_{i_m\alpha}. \tag{15}$$

Coefficients ω_{iml} do not depend on the solution and are defined by the stencil only. The first-order scheme corresponds to $\psi^n_{i\alpha} \equiv 0$, whereas $\psi^n_i \equiv 1$ gives a linear (oscillatory) high-order method. Here, for steady-state calculation a smooth limiter from [49] is applied. The reconstruction procedure can be extended to polynomial reconstructions of any order of spatial accuracy. However, the test calculations have shown that the piece-wise linear (second-order) representation is the best compromise between the accuracy and computational cost.

The least-square reconstruction is robust, works well across all flow regimes and is significantly more accurate than the first-order method. However, it is known that the accuracy of the least-square approximation may degrade as the cell aspect ratio grows [26]. Additionally, for some flow problems, such as flows in long pipes, the three-dimensional reconstruction does not take into account the fact that the flow gradients are generally large in the direction normal to the surface and small along the pipe, with the exception of the entrance and exit regions. For the hexahedral meshes a locally structured reconstruction method can be used [42], in which the face average for face l is computed as

$$f^n_{\alpha il} = f^n_{\alpha i} + \phi(S_L, S_R)\Delta_{il}, \tag{16}$$

where Δ_{il} is the distance from cell centre to face centre, S_L, S_R are left and right estimates of solution slope, $\phi(x, y)$ is the slope limiter. The advantage of the form (16) is that in the one-dimensional case the reconstruction reproduces the linear function exactly even if the mesh spacing is not uniform. The disadvantage is that it is generally not strictly monotone. Experience suggests that in the extreme cases such as rapid gas flow into vacuum this reconstruction can be less robust than the more restrictive fully three-dimensional limiting process (15). For steady-state calculations it may be necessary to first obtain an initial approximation using either (13) or (15) and only then switch to (16).

In the present work the so-called monotonized central (MC) slope limiter is used [45]:

$$\phi(x, y) = \operatorname{minmod}\left(\frac{1}{2}(x + y), \beta \cdot \operatorname{minmod}(x, y)\right). \tag{17}$$

Here minmod is the modification of the original slope limiter of Kolgan

$$\operatorname{minmod}(x, y) = \frac{1}{2}\big(\operatorname{sign}(x) + \operatorname{sign}(y)\big)\min\big(|x|, |y|\big),$$

parameter $\beta = 1 \ldots 2$ controls the amount of steeping. The original paper suggests $\beta = 2$. However, experience shows that for steady-state computations the use of a smaller value gives superior convergence properties [50]. Following [50,6], the value $\beta = 1 \ldots 1.25$ is set in the present work.

The resulting second-order method (16) requires less computational memory and runs approximately 25% faster, than the fully three-dimensional one.

3.4. Solution of the linear system and convergence criteria

The direct numerical solution of the linear system (10) is a very slow operation with the computational cost proportional to N^3_{space}. Therefore, an approximate factorization of the system is carried out using the approach suggested in [21,22]. The system of equations (10) is solved approximately in two stages. Unlike the earlier implementations [39] the matrix-free approach is used. Firstly, the backward substitution is used to calculate intermediate values g^*:

$$D_{\alpha i} g^*_{\alpha i} = -\sum_{l:\ \sigma_l(i) < i} c_{\alpha i \sigma_l(i)} g^*_{\alpha \sigma_l(i)} + \Delta t L^n_{\alpha i}, \quad i = N_{\text{space}}, \ldots, 1. \tag{18}$$

Next, the direct substitution gives the final values:

$$D_{\alpha i} g^n_{\alpha i} = g^*_{\alpha i} - \sum_{l:\ \sigma_l(i) > i} c_{\alpha i \sigma_l(i)} g^n_{\alpha \sigma_l(i)}, \quad i = 1, \ldots, N_{\text{space}}. \tag{19}$$

The formulas (18), (19) do not require the storage of the matrix of the system (10). The computational cost of solving (18), (19) is linearly proportional to N_{space}. As a result, the cost of one time step of the implicit method is only 25% larger than the computational cost of an explicit method with the same spatial reconstruction procedure and the conservative calculation of macroscopic parameters.

In calculations, the value of the time step Δt is evaluated according to the expression $\Delta t = C \min_i d_i/\xi_0$, where C is the prescribed CFL number, d_i the characteristic linear size of the cell V_i. The numerical solution is deemed as converged to the steady state if the integral residual R_1 in the macroscopic conservation laws

$$R_1 = \sum_i \left| \boldsymbol{R}^n_i \right| \cdot |V_i|, \quad \boldsymbol{R}^n_i = \sum_\alpha \begin{pmatrix} 1 \\ \xi \\ \xi^2 \end{pmatrix}_\alpha L^n_{i\alpha} \omega_\alpha \tag{20}$$

drops below the specified tolerance ϵ_R.

4. Parallel solvers

For large-scale problems such as the ones reported here the calculations are carried out on modern high-performance clusters using Message Passing Interface (MPI). In the present work two approaches to extend the single-processor method to multiple processors via MPI are exploited.

4.1. Single-block parallelization

In the single-block strategy the spatial mesh is not partitioned, instead the velocity mesh is divided into blocks of equal size. It appears that this strategy was first used in [3] for explicit schemes. In the present method each processor is assigned a number of molecular velocities. The stages of the single-block parallel algorithm can be summarized as follows:

1. Each processor computes its part of the integral sums for the boundary condition (3).
2. Using MPI_AllReduce, the integral sums are summed up on the master processor, the densities of reflected molecules (4) are computed and sent to each of the processors for cell faces, adjacent to the solid boundaries.
3. For each velocity node, the kinetic equation is solved using the implicit method (10). The solution procedure includes spatial reconstruction, calculation of intercell fluxes and the model collision integral. Parts of the macroscopic integral sums (1) and residual vector (20) are calculated.
4. MPI_AllReduce command is used to sum the parts of the integral sums from each processors. The initial guess $\boldsymbol{W}^0$ for macroscopic quantities at the new time level is computed and sent to each processor.
5. The correction procedure is called (6), consisting of the following steps:
   (a) Calculation of parts of $\boldsymbol{H}^{s-1}$ in Eq. (7).
   (b) Summation of residual parts from all processors and broadcasting of the results using MPI_AllReduce.
   (c) Calculation of the next approximation $\boldsymbol{W}^s$.
   Iterations are repeated until convergence.
6. Verification of the global convergence criteria (20).

Experience shows that for transitional and nearly-continuum regime $\delta \geqslant 1$ the process converge rapidly within 2–3 iterations in the majority of cells. However, there are a small number of cells, for which up to 10 iterations may be required.

4.2. Discussion of the single-block approach

The advantages of the single-block approach to parallelization over existing methods [17] are its simplicity and fast convergence to steady state due to the use of the implicit time-marching method (10). For relatively coarse spatial meshes and single-precision data representation the parallel version of the algorithm was shown to scale well up to 512 cores [39]. However, on the most of the HPC systems parallelisation with respect to the velocity mesh may restrict the size of the spatial mesh. This is because all the unstructured mesh data such as connectivity, areas, volumes and reconstruction matrices must be kept by each processor. Experience shows that on a typical distributed-memory machine with 1 GB of RAM per core only the meshes with up to approximately $N_{\text{space}} = 2 \cdot 10^5$ cells can be used. Indeed, for double-precision calculations using N_{CPU} cores, the required amount of memory M in bytes for each core can be estimated as

$$M = 8 N_{\text{space}} \left(M_{\text{space}} + \frac{N_\xi}{N_{\text{CPU}}} \right). \tag{21}$$

Here $M_{\text{space}} \approx 10^3$ is the amount of memory, required by the second-order unstructured-mesh spatial solver (15) to store the mesh data as well as reconstruction data (list of cells in the stencil, reconstruction matrix, basis function values at faces, etc.), as well as ξ-independent variables, such as macroscopic data; N_ξ is the total number of cells in the velocity mesh. If the number of cores is sufficiently large $N_{\text{CPU}} \gg 1$, one can disregard the second term in the brackets in (21). Then, the minimum required memory per core can be estimated as $M \approx 8000 \cdot N_{\text{space}}$. Taking into account that the operating system also needs memory, the spatial mesh of at most $N_{\text{space}} = 10^5$ cells can be used if 1 GB of RAM per core is assumed.

The second disadvantage of the single-block approach is that its scalability stalls for sufficiently large spatial meshes due to the multiple use of MPI_AllReduce command during the calculation of macroscopic variables. The effect is stronger if the double-precision representation is used, which is essential for small Knudsen number calculations. This effect was studied in [40] for a similar numerical method, but in two spatial dimensions. As we intend to run the simulations on hundreds and possibly thousands of cores, a better parallel strategy may be required.

In the present work the computer implementation of the correction procedure (6) is modified in order to improve the overall efficiency. The convergence of the Newton-type iterations (7) is now controlled locally so that in the converged cells the residual vector is not recomputed. This saves computer time for recalculations of data as well as MPI_AllReduce calls.

4.3. Multi-block parallelization

The second approach to the construction of parallel solvers uses the partitioning of the spatial mesh into the blocks. For kinetic equations it has been used in a number of papers, e.g. [17], for the explicit advection schemes only. The implementation of high-order explicit advection schemes on multi-block unstructured meshes is well documented in the literature, see e.g. [8]. However, explicit schemes miss the key advantage of the implicit algorithm (10), namely fast convergence to steady state using large CFL numbers. The direct generalization of (10) to multi-block meshes is very difficult from the algorithmic point of view as well as for maintaining good MPI scalability of the code. For the two-dimensional structured meshes such implementation is reported in [25]. The authors used a special re-numeration of the spatial cells and asynchronous MPI communications to implement the parallel algorithm, which is identical to the sequential time marching. To the best of our knowledge, no extension of this algorithm to the three-dimensional unstructured meshes has been reported. Various simplified realizations of the parallel LU-SGS methods for multi-block unstructured meshes were studied in [34] as applied to the compressible Navier–Stokes equations. The authors noted a significant decrease in the convergence speed to the stationary solution as the number of spatial blocks grows. This sort of performance is not very suitable for the present method.

From the practical point of view, the preferred parallel version of the implicit method is such that does not use data exchange between blocks during the solution procedure (18), (19). Instead, the boundary condition (11) is used at block interfaces. The transient behaviour of the resulting parallel algorithm is not equivalent to the single-block method (either sequential or parallel). One of the goals of the present work is to access the impact of the mesh partitioning on the steady-state convergence of the implicit method.

The time advancing algorithm of the multi-block implicit method is now different from the single-block implicit one and consists of the following stages:

1. At each processor the integral sums for the boundary condition (3) are computed and the density of reflected molecules (4) are found. For the second-order method, the reconstruction procedure at this stages uses the cell values of the distribution function from the previous time level. Since we are not interested in the time-accurate transient, this is admissible.
2. For each velocity mesh node
   (a) For higher-order spatial schemes (15), (16) exchange of values of the distribution function in ghost cells is carried out.
   (b) The Reconstruction procedure is invoked to compute face values of the distribution function $f^n_{i\alpha l}$.
   (c) Exchange of face values at block boundaries is carried out.
   (d) Intercell fluxes $F^n_{i\alpha l}$ are computed according to (12).
   (e) The kinetic equation is solved using the implicit method (10) and the residual vector (20) is computed.
3. For each spatial cell the macroscopic parameters are computed according to (7) and steady-state convergence criteria (20) is checked.

To improve the parallel efficiency, all interprocessor exchanges are carried out using non-blocking send/receive MPI commands.

The estimate of the required amount of memory per core (21) is now rewritten as

$$M = 8\frac{N_{\text{space}}}{N_{\text{CPU}}}(M_{\text{space}} + N_\xi). \tag{22}$$

It is seen, that unlike (21), the required memory M decreases linearly with the number of cores N_{CPU}. Moreover, the parallel algorithm does not contain the calls to the slow command MPI_AllReduce. Instead, each processor (core) exchanges data with the direct neighbours only. Therefore, the parallel multi-block method is free of the two main bottlenecks of the single-block parallel algorithm.

5. Numerical examples

In this section numerical results are presented for the first-order method (13), three-dimensional TVD method (referred to as TVD3D) (15) and locally structured reconstruction method (16) (referred to as TVD1D). Both single-block and multi-block versions are used. The calculations have been carried out on the HPC "Lomonosov" of Lomonosov Moscow State University, Russia. "Lomonosov" supercomputer consists of both Intel Xeon X5570/X5670 CPUs and Nvidia X207 GPUs and has 1.7 PFlops of peak performance. The total number of Xeon cores, available for calculations, is 52 168. In the present work the runs were performed on 64 to 1024 cores of the machine (8 to 256 quad-core Xeon processors) with 3 GB of RAM per core.

5.1. Pipe flow

5.1.1. Formulation

Consider a rarefied gas flow through a circular pipe of length L and radius R, connecting two infinitely large reservoirs (volumes) filled with the same monatomic gas. Gas in reservoirs is kept under pressures $p_1 > p_2$, respectively, and at the same temperature $T_1 = T_2 = T_0$. The complete accommodation of momentum and energy of molecules occurs at the pipe surface, which is kept under the same constant temperature $T_w = T_0$. The values of pressure, temperature and viscosity in reservoir 1 are used as p_*, T_*, μ_*, whereas the pipe radius R is taken as the characteristic linear dimension l_*. The calculations presented here correspond to the short pipe $L/R = 1$ and two pressure ratios: $p_1/p_2 = 2$ (moderate drop) and $p_1/p_2 = \infty$ (flow into vacuum). The problem in this geometrical formulation with both reservoirs and the pipe is now included in the proposed database of benchmark problems in rarefied gas dynamics [33].

The main computed quantity is the mass flow rate $\dot{M}$ through the pipe

$$M = \int_{A(z)} \rho(x, y, z) w(x, y, z)\, dx dy.$$

Here the quantity $mn_1\beta_1 R^2$ is used as the scale to pass to non-dimensional value of $\dot{M}$. In the presentation of the results, it is more convenient to use the so-called reduced mass flow rate Q instead. The value of Q is defined as a ratio of the mass flow rate $\dot{M}$ at given value of the rarefaction parameter δ_1 and L/R_1 to its values M_0 in the free-molecular orifice flow [32,48]. In the non-dimensional variables Q is calculated as

$$Q = \frac{\dot{M}}{\dot{M}_0}, \quad \dot{M}_0 = \frac{\sqrt{\pi}}{2}. \tag{23}$$

For the free-molecular case $\delta_1 = 0$ the solution can be obtained using the integral equation of Clausing [7]. There were a number of solutions of this equation obtained by other authors afterwards as well as statistical solutions; for discussions see e.g. [18]. The dimensional mass flow rate is given by

$$\dot{M}_\infty = \pi R^2 \frac{p_1 - p_2}{\sqrt{2\pi (k/m) T_0}} W \tag{24}$$

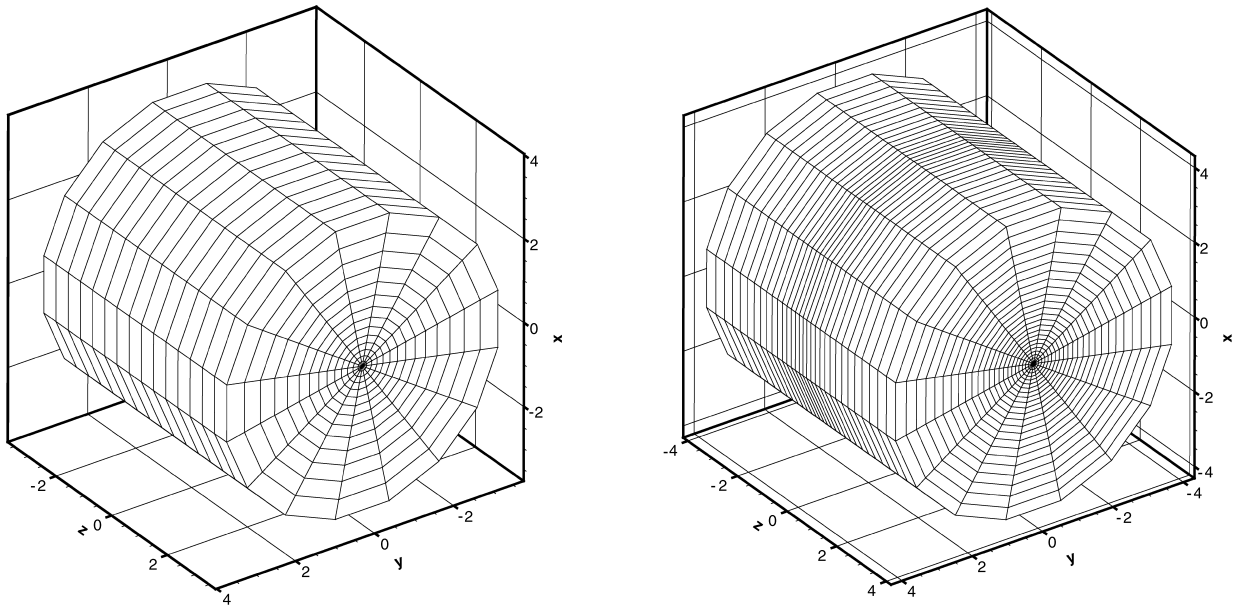
where W is the Clausing coefficient. In the non-dimensional variables one obtains

$$\dot{M}_\infty = \frac{1}{2}\sqrt{\pi}(p_1 - p_2) W, \qquad Q_\infty = (p_1 - p_2) W. \tag{25}$$

For $L/R = 1$ the Clausing coefficient is equal to $W = 0.672$ so that $Q_\infty = 0.336$ for $p_1/p_2 = 2$ and $Q_\infty = 0.672$ for $p_2 = 0$. These values can be used to assess the accuracy of the methods in the free-molecular regime. It should be noted that such a regime is quite difficult for shock-capturing schemes as discontinuities of the distribution function are not tracked explicitly like in shock-fitting schemes, often used for two-dimensional calculations [31,1].

5.1.2. Mesh convergence study

It is very difficult to perform proper six-dimensional mesh convergence studies due to the very high computational cost of the simulations. Experience shows that the solution is most sensitive to the spatial accuracy of calculations, whereas the velocity space discretization has a smaller influence; fine velocity mesh is important for the rarefied flow regimes only. Therefore, in the present work first the required velocity mesh resolution is determined using the cylindrical arrangement of velocity nodes. For the case $p_1/p_2 = 2$ the coarse velocity mesh consists of 15 points in the radial direction $\zeta = \sqrt{\xi_1^2 + \xi_2^2}$ and 16 cells in the z and angular direction. The radial and z component of the velocities vary in the limits $0 \leqslant \zeta \leqslant 4$, $-3.5 \leqslant \xi_3 \leqslant 3.5$ with the ζ mesh clustered towards the origin. The second mesh consists of $25 \times 32 \times 32$ nodes with the same domain size. The calculations showed that the coarse mesh is sufficient for $\delta_1 > 1$, whereas for $\delta_1 \leqslant 1$ the finer mesh provides higher accuracy; the difference for the most difficult free-molecular flow regime is around 1%. Therefore, for the rest of the pipe flow studies the coarse mesh is used for $\delta_1 > 1$ and the fine mesh is for $\delta_1 \leqslant 1$.



(a) $15 \times 16 \times 16$ nodes (b) $25 \times 16 \times 32$ nodes

Fig. 1. Velocity meshes for the pipe flow.

For the flow into vacuum $p_1/p_2 = \infty$ a finer velocity mesh is required in all cases, consisting of $25 \times 16 \times 32$ nodes with the extended domain size $-4 \leqslant \xi_3 \leqslant 4$. Moreover, the ξ_3 mesh is clustered towards the origin. Such differences in velocity mesh construction are necessary in order to take into account a large temperature decrease in the vacuum reservoir and high gas velocity. The Simpson rule is used for odd number of cells in ζ direction whereas the trapezoidal rule is used for the even number of cells in ζ and for two other components of velocity mesh. Fig. 1 shows the $15 \times 16 \times 16$ nodes mesh for the moderate pressure drop and $25 \times 16 \times 32$ node mesh for the vacuum case.

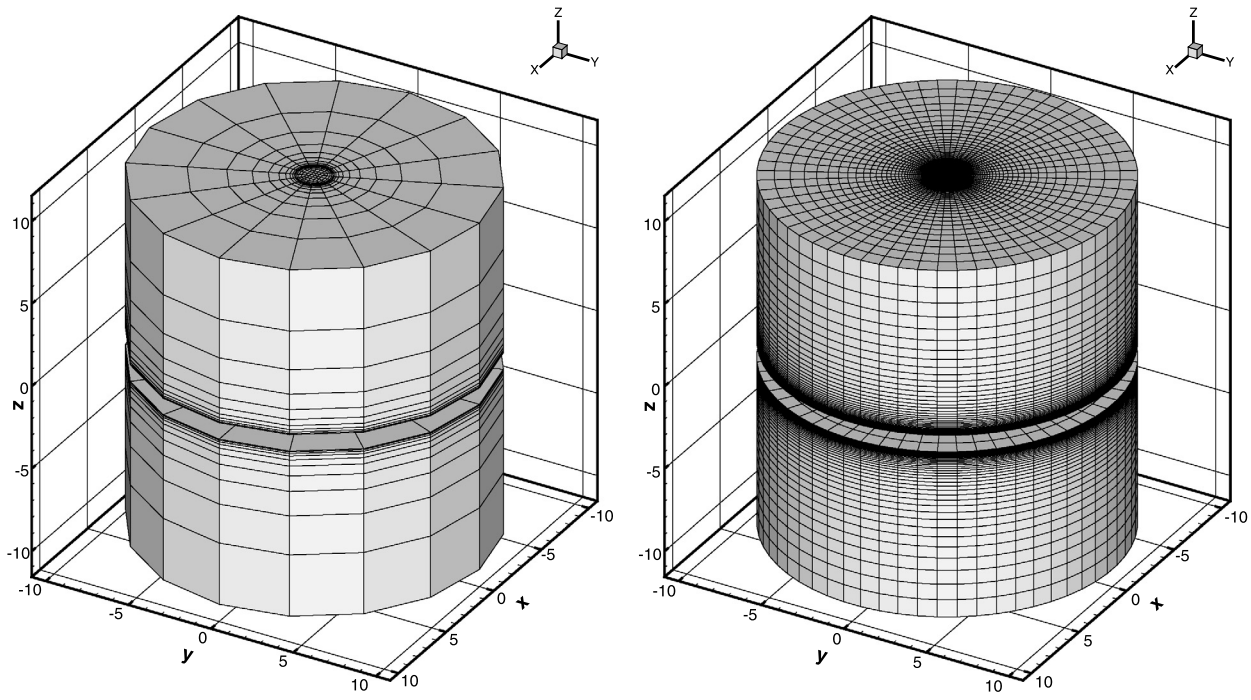
The definition of the parameters of the spatial mesh begins with the estimation of the required size of the reservoirs. The use of small domains for reservoirs decreases the computational effort required to obtain the solution, but may lead to the underestimation of the mass flow rates. This is especially so for relatively short pipes and/or $\delta_1 \gg 1$. Our test calculations (not shown here) indicate that the size parameters recommended in the DSMC studies [48] are quite adequate. In the present work, larger reservoirs of length and radius 10 are used to allow for more reliable studies for $\delta_1 \gg 1$.

Figs. 2, 3 illustrate the first (coarsest) and third (finest) spatial meshes. The first mesh consists of 5600 cells and is exceedingly coarse with only 10 cell intervals along the pipe, 116 cells in the cross-sectional plane of the pipe (240 cells for the reservoir cross section with the pipe). This should be compared with much finer cross-sectional meshes used for the infinite pipe calculations. The cell size in the direction normal to the pipe surface and exits is 0.06. The second and third meshes are constructed by approximately doubling the number of cells in each direction so that the total number of cells is 40 761 and 349 905, respectively. The spatial mesh resolution is selected to be deliberately coarse in order to better highlight the difference between spatial schemes (13), (15), (16).

To summarize the mesh description, it is noted that the finest six-dimensional computational mesh $(x, y, z, \xi_1, \xi_2, \xi_3)$ contains 1.3×10^9 cells for $p_1/p_2 = 2$ case and 4.5×10^9 cells for the flow into vacuum.

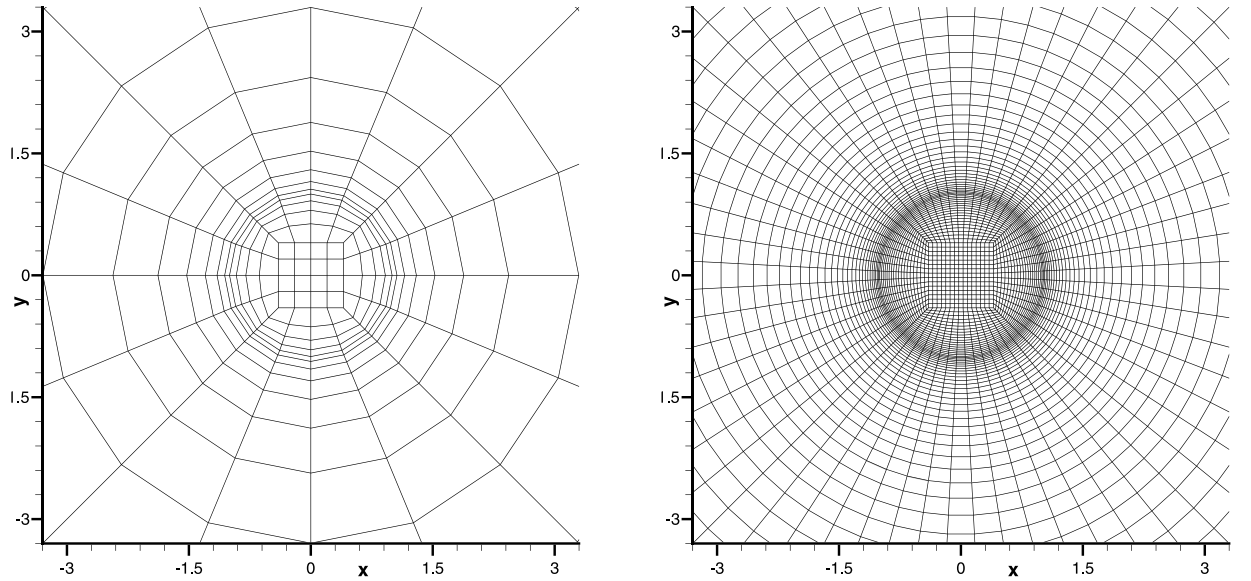
The convergence study for $p_2/p_1 = 0.5$ is reported in Table 1 for selected values of the rarefaction parameter $\delta_1 = 0$, 0.1, 1, 10 and 100. As a reference solution also provided the results of the DSMC studies from [48]. For the monatomic gas the well-resolved DSMC calculations can be considered as equivalent to the direct numerical solution of the Boltzmann equation with the exact collision integral. The S-model equation is exact for the free-molecular regime and provides the correct limit for $\delta_1 \gg 1$ (nearly continuum and continuum flows). For the rarefied and transitional flows the use of the approximate collision integral can lead to the deviation from the solution of the Boltzmann kinetic equation with the exact collision integral by a few percent. Therefore, one does not expect the numerical results from the S-model equation to be equal to the DSMC results [48] in the complete range of the rarefaction parameter values.

It is seen from Table 1 that the second-order accurate locally one-dimensional method (16) is the most accurate scheme for the present problem, including in the free-molecular regime. The difference between its results on two finest meshes is within 1%. For $\delta_1 = 100$ the present calculations give a somewhat higher flow rate than the DSMC data, which is however still within the stated computational error of [48]. For $\delta_1 = 1$ there is a discrepancy at the level of 3%, which is due to the use of the approximate collision integral. Overall, the middle spatial mesh of 40 761 is sufficient to obtain the accurate results in the complete range of the considered values of the rarefaction parameter. The use of the coarsest mesh gives



(a) 5600 cells (b) 350000 cells

Fig. 2. Spatial meshes for the pipe flow.



(a) 5600 cells (b) 350000 cells

Fig. 3. Cross-sectional cuts of the spatial meshes in the x–y plane.

acceptable values for $\delta_1 \leqslant 1$, but results in a significant error of up to 7% for $\delta_1 \geqslant 10$. However, on 16 cores it takes only 3 hours to obtain the converged solution for $\delta_1 = 100$ and hence this coarse mesh can be used for preliminary studies.

The accuracy of the three-dimensional spatial approximation (15) is comparable to that of the scheme (16) for the rarefied flow regime $\delta_1 \leqslant 1$. However, as the flow approaches the continuum, the three-dimensional least-square reconstruction becomes less accurate. Both second-order schemes are, however, superior to the simple first-order accurate method (13).

Table 1
Mesh convergence studies for the reduced flow rate Q defined in Eq. (23), for finite pressure ratio $p_1/p_2 = 2$. Results (i), (ii) and (iii) correspond to the spatial meshes with 5600, 40 761 and 349 905 hexahedra.

δ_1	First-order scheme (13)			TVD3D scheme (15)			TVD1D scheme (16)			DSMC, Ref. [48]
	(i)	(ii)	(iii)	(i)	(ii)	(iii)	(i)	(ii)	(iii)	
0.	0.327	0.333	0.335	0.333	0.335	0.336	0.335	0.336	0.336	0.336
0.1	0.334	0.340	0.343	0.342	0.344	0.344	0.344	0.345	0.345	0.343
1.	0.383	0.401	0.410	0.407	0.417	0.420	0.412	0.419	0.421	0.405
10.	0.626	0.734	0.802	0.800	0.855	0.873	0.832	0.871	0.876	0.866
100.	0.912	1.093	1.187	1.134	1.233	1.285	1.220	1.292	1.303	1.29

Table 2
Mesh convergence studies for the reduced flow rate Q defined in Eq. (23), for the flow into vacuum $p_1/p_2 = \infty$. Results (i), (ii) and (iii) correspond to the spatial meshes with 5600, 40 761 and 349 905 hexahedra; Ref. [42] uses 270 000 hexahedra.

δ_1	TVD1D scheme (16)			Ref. [42]	DSMC, Ref. [47]	Exp. data, Ref. [10]
	(i)	(ii)	(iii)			
0.	0.666	0.670	0.672	0.670	0.672	
0.1	0.678	0.683	0.684	0.681	0.680	0.675
1.	0.758	0.766	0.768	0.758	0.754	0.743
10.	1.035	1.061	1.066	1.058	1.062	1.06
100.	1.290	1.351	1.367	1.355	1.358	1.33
200.	1.331	1.406	1.425		1.412	
500.	1.331	1.454	1.474		1.449	

The difference is most striking for $\delta_1 \gg 1$, when the first-order method (13) has a 12% error even on the finest mesh. Since the computational cost of all three methods is comparable, the use of the first-order approximation is not recommended.

For the second case of flow into vacuum $p_1/p_2 = \infty$ only the most accurate locally one-dimensional scheme (16) is used. Table 2 contains the present results for all three spatial meshes, DSMC data from [47] and experimental data from [10]. The numerical results cover the range $0 \leqslant \delta_1 \leqslant 500$, corresponding to flow regimes from the free-molecular one to the nearly continuum. It is seen that the locally one-dimensional scheme clearly converges as the spatial mesh is refined and yields sufficient accuracy on the second mesh even for $\delta_1 = 500$. The free-molecular solution is computed with 0.3% error. The deviation from the DSMC data is now reduced to 2% for $\delta_1 = 1$ and well below 1% for other values of the rarefaction parameter. The comparison with the previous results based on an original version of the method [42] demonstrate a discrepancy at the level of 1%. However, it should mentioned that the calculations in [42] were run on a spatial mesh of 270×10^3 cells with a much stronger clustering towards the surface. Therefore, the present modification of the method maintains the same accuracy using the spatial mesh with 6 times fewer cells.

Comparison with the experimental data shows that the kinetic solution on the finest spatial mesh differs from the experimental data by 2–3% depending on the value of the rarefaction parameter δ_1. The DSMC solution agrees slightly better, mostly in the rarefied flow regime. Overall, the difference of the numerical and experimental results is comparable with the experimental measurements error of 2%. It is expected, that if the correct accommodation coefficient is used in the boundary condition, the discrepancy between experimental and computational results can be further decreased.

5.1.3. Comparison of single- and multi-block parallel solvers

For the multi-block solver the finer of the three spatial meshes with 349 905 cells is split into the required number of blocks using version 4.0 of METIS software package [12]. For 64 cores each spatial blocks contains 5467–5468 cells with between 1300 and 2000 ghost cells per block for the high-order method. For 1024 cores the number of cells in the block varies between 340 and 344 (load balance of approximately 1.01) and 200 to 300 ghost cells.

The steady-state convergence of both parallel versions of the method is assessed by computing the solution to the problem for $p_1/p_2 = 2$ at $\delta_1 = 1$, using the locally one-dimensional spatial scheme (16) and the velocity mesh with $15 \times 16 \times 16$ cells. The initial condition for all runs is provided by the locally-Maxwellian function, corresponding to the gas at rest with the linear density distribution along the pipe. Fig. 4 shows the behaviour of the maximum norm residual R_∞ and normalized flow rate Q for the single-block method and multi-block method run on 128 and 512 blocks. It is seen that mesh partitioning into spatial blocks affects the convergence behaviour of the method insignificantly, at least for the considered value of the rarefaction parameter.

The scalability of variants of the parallel method is assessed using 64, 128, 256, 512 and 1024 cores on the third (finest) spatial mesh. For this set of runs it is more convenient to use the velocity mesh with 16^3. For the single-block solver the velocity mesh is naturally split into the required number of blocks, equal to the number of CPU cores.

The wall clock times of each calculation are given in Table 3. It is seen that overall, the single-block method does not scale above 512 cores and the time required for one time step, grows sharply. This behaviour is due to the high computational cost of calling MPI_AllReduce command on the given spatial mesh. The analysis, carried out in [40] for the two-dimensional version of the single-block parallel method, showed that the condition for good scalability is the

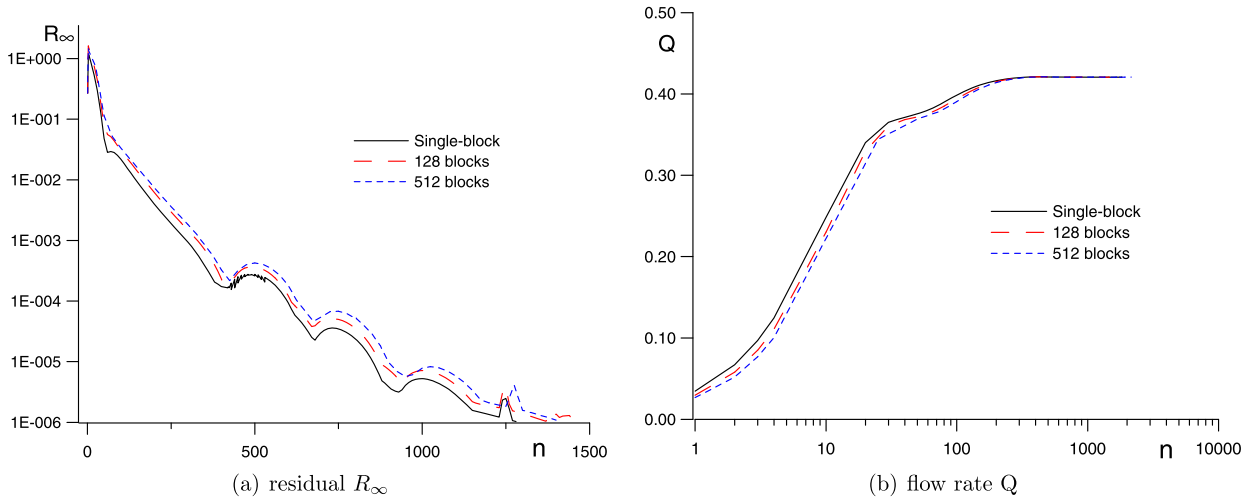


Fig. 4. The maximum norm residual R_∞ and normalized flow rate Q for $\delta_1 = 1$ as a function of number of spatial blocks.

Table 3
Wall clock time in seconds of single- and multi-block solvers with the spatial scheme (16). Spatial mesh with 349 905 cells and 16^3 velocity mesh are used.

Cores	64	128	256	512	1024
Single-block	48	27	15	8.6	20
Multi-block	54	23	12	7	5.2

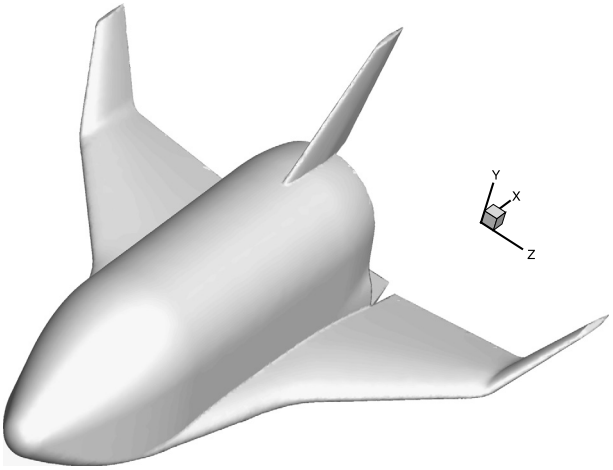


Fig. 5. Re-entry space vehicle geometry.

use of relatively coarse spatial mesh and finer velocity mesh. This was confirmed in [39] where the single-block method was shown to scale well up to 512 cores, running on the spatial mesh of $\approx 5.5 \times 10^4$ cells. However, for many flow problems this requirement on the number of cells in spatial and velocity meshes is not feasible. However, the new adaptive implementation of the macroscopic correction procedure allows the single-block method to scale well up to 512 cores on a significantly finer spatial mesh as compared to the earlier work [39]. The multi-block parallel version scales up to 1024 cores, although the gains are relatively small going from 512 to 1024 cores. Moreover, it is faster for the most cases except for 64 cores.

5.2. Supersonic flow over a model re-entry space vehicle

The geometry of the re-entry space vehicle (RVS) is provided in initial graphics exchange specification (IGES) format and is shown in Fig. 5. The aerodynamic shape of the RSV consists of a fuselage with spherical nose bluntness, two wings with dogtooth extension and end edges, and a vertical keel and fuselage flap. The length of the fuselage is 9000 mm, the radius of the nose is 450 mm, the upper surface diameter is 2600 mm. The total length of the RVS with the flap is 10 000 mm.

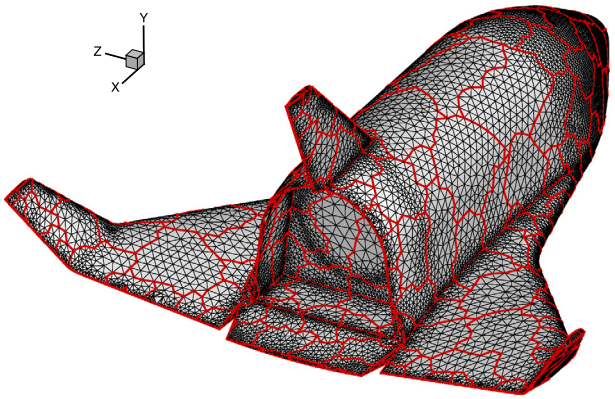


Fig. 6. Surface mesh for the RVS calculations.

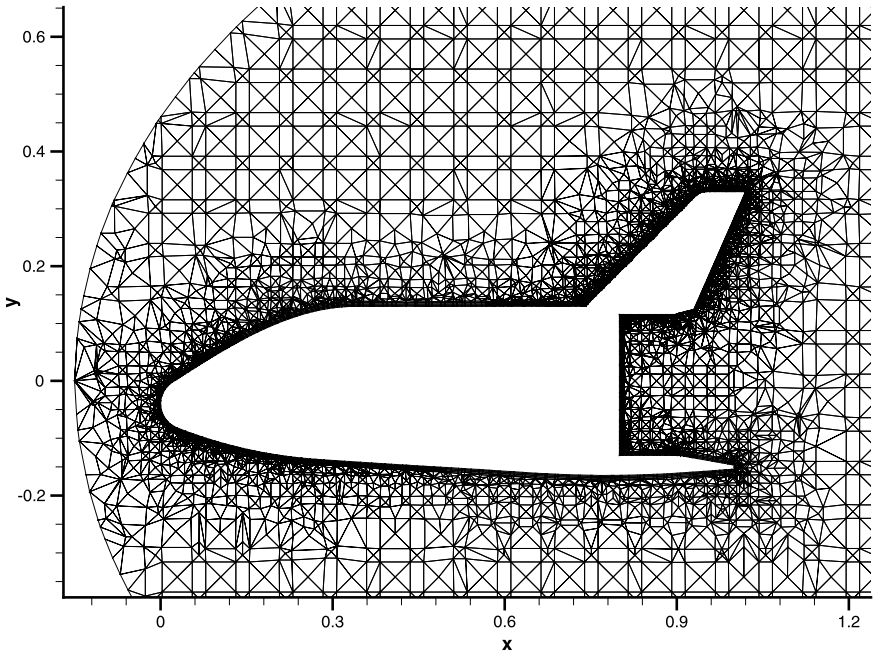


Fig. 7. Volume mesh cut around the vehicle for the RVS calculations.

The hybrid computational mesh in the physical space x, y, z is initially constructed in the dimensional variables. Its use allows to capture all essential flow features without spending considerable effort on generation of a multi-block structured mesh. The surface of the RVS is covered by the triangles with the appropriate refinements to the sharp features of the geometry as well as to the blunted nose. Fig. 6 illustrates the surface mesh and its decomposition into 256 blocks, used in calculations. The four layers of prismatic cells of 20 mm heights are used near the surface of the RVS, whereas the rest of the computational domain is filled with tetrahedrons. Fig. 7 shows the cut through the volume mesh at the symmetry plane. The total number of spatial cells is $N_{\text{space}} \approx 533 \times 10^3$, including approximately 386×10^3 tetrahedrons, 147×10^3 prisms and 551 pyramids. Each block contained approximately 2100 internal cells and between 900 and 2800 so-called ghost cells, required by the second-order TVD3D method (15). Such a large variation in the number of ghost cells seems to be due to the strong clustering of cells towards surface combined with the relatively large number of blocks for the considered number of spatial cells.

The total length of the vehicle is chosen as the spatial scale l_*, whereas the free-stream values of pressure and temperature are set as p_*, T_*. The surface temperature was fixed and equal to the free-stream temperature. The non-dimensional spatial mesh is obtained from the initial dimensional mesh by dividing over l_*. Calculations were carried out for the non-dimensional free-stream velocity number $u_\infty = 2$. The rarefaction parameter is set to $\delta_\infty = 1000$, which approximately corresponds to the altitude of 100 km (1 cm mean free path). The velocity mesh consisted of 24^3 nodes. The total number of cells in the 6-dimensional mesh is thus approximately 6.9×10^9.

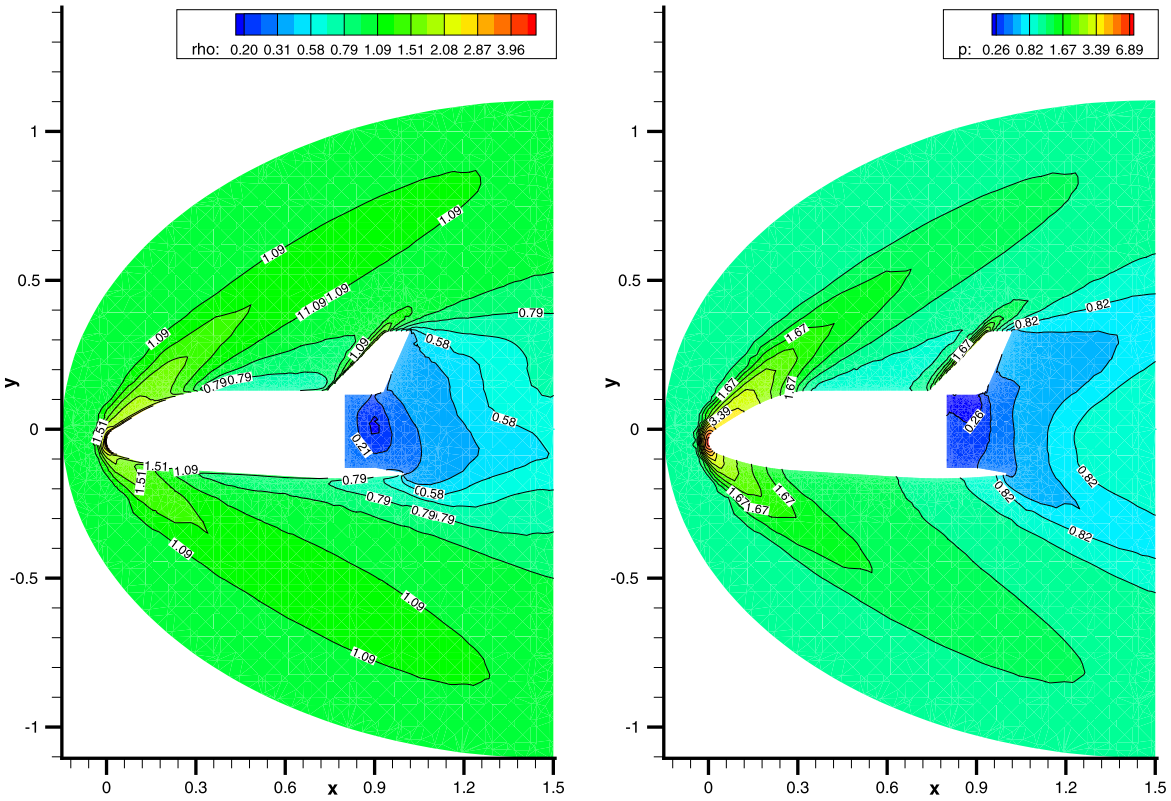


Fig. 8. Flow visualisation in the x–y plane: density (left) and pressure (right).

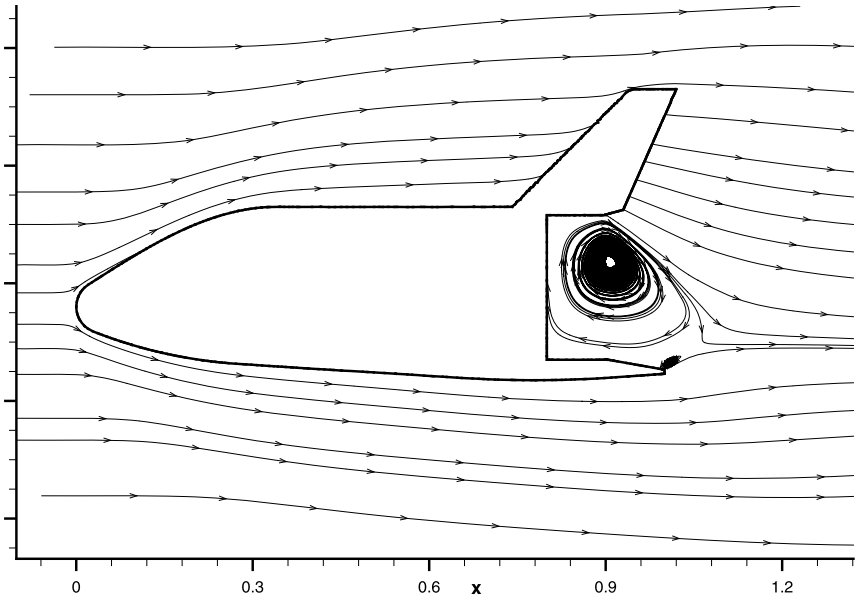
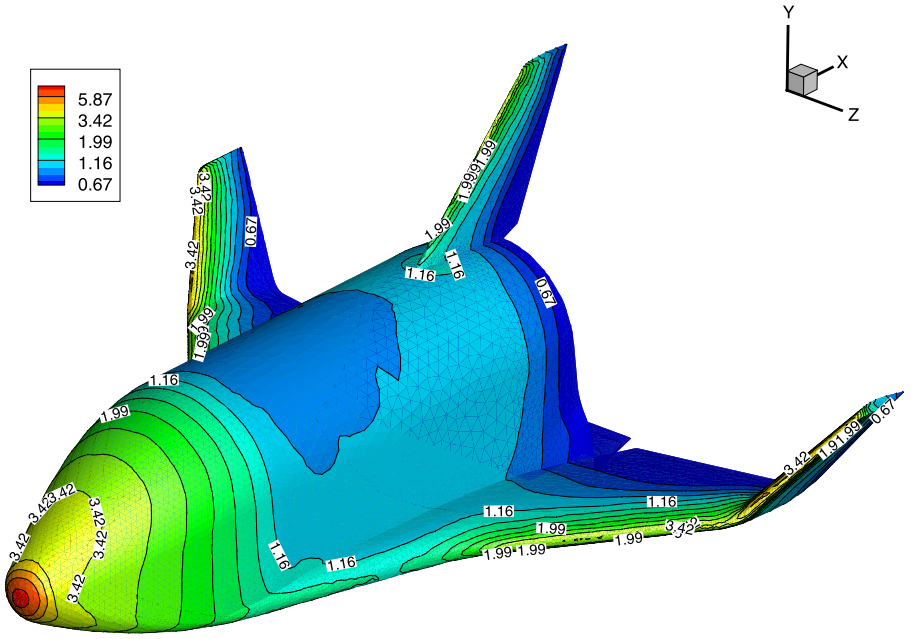
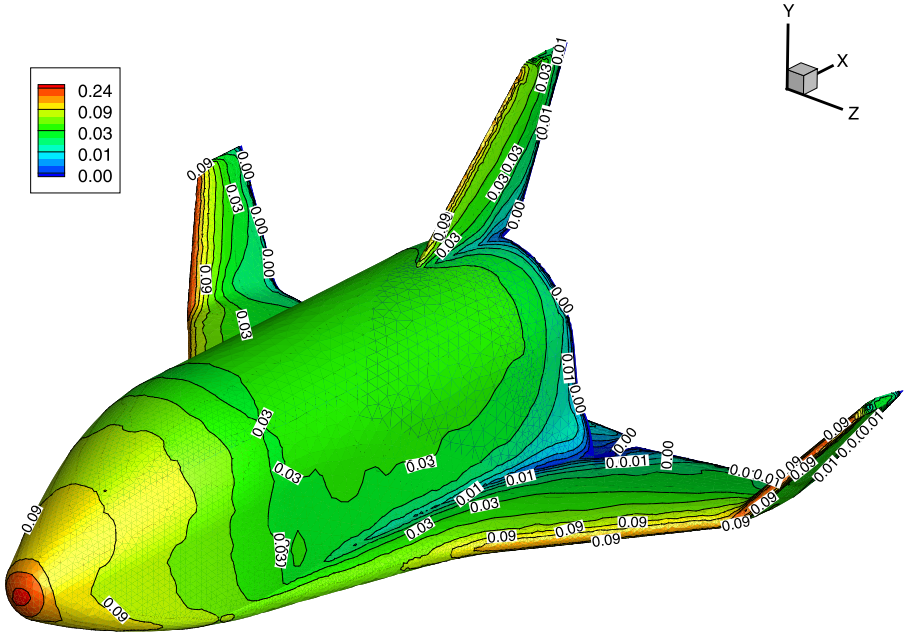


Fig. 9. Flow visualisation in the x–y plane: streamlines.

The calculation process is organized as follows. The free-stream values of all macroscopic variables are used as the initial guess. The first-order solution is constructed, using the first-order scheme. The time step is chosen according to the CFL number of 25. Then, the second-order solution is computed using the TVD3D scheme and first-order solution as the initial guess. One time step takes approximately 120 seconds. The complete steady-state convergence requires several thousand iterations.



(a) pressure



(b) normal heat flux

Fig. 10. Surface distribution of pressure and normal component of heat flux.

Figs. 8, 9 provide a general representation of the flow field. Shown are the isolines of density and pressure on the symmetry plane as well as streamlines. Fig. 10 depicts surface distribution of pressure as well as normal heat flux to the surface. Overall, a typical flow pattern of a supersonic rarefied gas flow over a cold body is observed. The flow is characterized by the formation of the bow shock wave in front the vehicle as well as sharp drop in pressure and density in the wake close to the tail. The rarefaction parameter related to the blunted nose is $\delta_{\text{nose}} \approx 45$, hence the bow shock wave is quite diffused. Pressure and heating at the largest at the stagnation point on the nose. Along the cylindrical surface of the fuselage pressure and density change quite slowly. However, there is significant heating of the front edges of the wings and

the keel, which is expected. At this value of the rarefaction parameter a recirculation zone is formed after the tail above the flap. Finally, there is no visible loss of symmetry in the surface plots even though the unstructured mesh is not symmetric.

6. Conclusions

A numerical framework for modelling the three-dimensional steady rarefied gas flows on the basis of the Boltzmann kinetic equation with the model collision integrals has been reviewed and extended to multi-block unstructured meshes. It is shown that steady-state convergence properties of the proposed method do not degrade with mesh partitioning into increasing number of blocks. Moreover, the multi-block implicit algorithm scales better, than the corresponding single-block one. The newly proposed method is shown to be applicable to external flows over re-entry space vehicles of complex aerodynamic shape, for which its ability to use hybrid unstructured meshes, implicit time marching and good scalability combine into a versatile computational fluid dynamics tool. Future work include the extension of the method to diatomic gases on the basis of the R-model kinetic equation [28,19], hypersonic flows and even higher orders of spatial accuracy using a non-intrusive defect-correction strategy [9], where the higher-order corrections are included as a source term into the low-order scheme.

Acknowledgements

The work of V.T. and S.U. was financially supported by the Russian government under grant “On Measures to Attract Leading Scientists to Russian Educational Institutions” (contract No. 11.G34.31.0072). V.T. and M.D. acknowledge the support of the Italian Ministry of Research (MIUR) and the British Council under the project British–Italian Partnership Programme for Young Researchers 2008–2009. M.D. was financed by the European Research Council (ERC) under the European Union Seventh Framework Programme (FP7/2007–2013) with the research project STiMulUs, ERC Grant agreement No. 278267. The use of high-performance computers of Lomonosov Moscow State University and Moscow Institute of Physics and Technology, Russia, is gratefully acknowledged.

References

[1] K. Aoki, K. Kanba, S. Takata, Numerical analysis of a supersonic rarefied gas flow past a flat plate, Phys. Fluids 9 (4) (1997) 1144–1161.
[2] V. Aristov, A. Frolova, S. Zabelok, R. Arslanbekov, V. Kolobov, Simulations of pressure-driven flows through channels and pipes with unified flow solver, Vacuum 86 (11) (2012) 1717–1724 (special issue “Vacuum Gas Dynamics: Theory, Experiments and Practical Applications”).
[3] V. Aristov, S. Zabelok, A deterministic method for solving the Boltzmann equation with parallel computations, Comput. Math. Math. Phys. 42 (3) (2002) 406–418.
[4] T. Barth, P. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, in: 28th Aerospace Sciences Meeting, 1990, AIAA paper No. 90-0013.
[5] P. Bhatnagar, E. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (511) (1954) 1144–1161.
[6] S. Bosnyakov, I. Kursakov, A. Lysenkov, S. Matyash, S. Mikhailov, V. Vlasenko, J. Quest, Computational tools for supporting the testing of civil aircraft configurations in wind tunnels, J. Prog. Aerosp. Sci. 44 (2008) 67–120.
[7] P. Clausing, Über die stromung sehr verdunnter gase durch röhren von beliebiger länge, Ann. Phys. 32 (1932) 961–989.
[8] M. Dumbser, M. Käser, V. Titarev, E. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, J. Comput. Phys. 226 (2007) 204–243.
[9] A. Filimon, M. Dumbser, C. Munz, High-order finite volume schemes based on defect corrections, Z. Angew. Math. Mech. 93 (6–7) (2013) 423–436.
[10] T. Fujimoto, M. Usami, Rarefied gas flow through a circular orifice and short tubes, J. Fluids Eng. 106 (4) (1984) 367–373.
[11] A. Gusarov, I. Smurov, Gas-dynamic boundary conditions of evaporation and condensation: numerical analysis of the Knudsen layer, Phys. Fluids 14 (12) (2002) 4242–4255.
[12] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1998) 96–129.
[13] Y. Kloss, D.V. Martynov, F. Cheremisin, Computer simulation and analysis of the Holweck pump in the transient regime, Tech. Phys. 82 (4) (2012) 451–456.
[14] V. Kolgan, Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics, Trans. Central Aerohydrodyn. Inst. 3 (6) (1972) 68–77 (in Russian).
[15] V. Kolgan, Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack, Trans. Central Aerohydrodyn. Inst. 6 (2) (1975) 1–6 (in Russian).
[16] V. Kolgan, Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous solutions of gas dynamics, J. Comput. Phys. 230 (7) (2011) 2384–2390.
[17] V. Kolobov, R. Arslanbekov, V. Aristov, A. Frolova, S. Zabelok, Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement, J. Comput. Phys. 223 (2007) 589–608.
[18] Y. Koshamarov, Y. Ryzhov, Applied Rarefied Gas Dynamics, Mashinostroenie, Moscow, 1977 (in Russian).
[19] I. Larina, V. Rykov, Kinetic model of the Boltzmann equation for a diatomic gas with rotational degrees of freedom, Comput. Math. Math. Phys. 50 (12) (2010) 2118–2130.
[20] Z.-H. Li, H.-X. Zhang, Numerical investigation from rarefied flow to continuum by solving the Boltzmann model equation, Int. J. Numer. Methods Fluids 42 (4) (2003) 361–382.
[21] I. Men’shov, Y. Nakamura, An implicit advection upwind splitting scheme for hypersonic air flows in thermochemical nonequilibrium, in: A Collection of Technical Papers of 6th Int. Symp. on CFD, vol. 2, Lake Tahoe, Nevada, 1995, p. 815.
[22] I. Men’shov, Y. Nakamura, On implicit Godunov’s method with exactly linearized numerical flux, Comput. Fluids 29 (6) (2000) 595–616.
[23] L. Mieussens, Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics, Math. Models Methods Appl. Sci. 8 (10) (2000) 1121–1149.
[24] L. Mieussens, Discrete-velocity models and numerical schemes for the Boltzmann–BGK equation in plane and axisymmetric geometries, J. Comput. Phys. 162 (2) (2000) 429–466.

[25] P. Pavlukhin, I. Menshov, Efficient parallel implementation of the LU-SGS method for gas dynamics problems, Sci. News MGTU GA 165 (2011) 46–54 (in Russian).
[26] N. Petrovskaya, A. Wolkov, The impact of grid geometry on the accuracy of higher order finite-volume and finite element schemes, Math. Model. 22 (3) (2010) 145–160 (in Russian).
[27] A. Rodionov, Complement to the Kolgan project, J. Comput. Phys. 231 (2012) 4465–4468.
[28] V. Rykov, A model kinetic equation for a gas with rotational degrees of freedom, Fluid Dyn. 10 (6) (1975) 959–966.
[29] E. Shakhov, Approximate kinetic equations in rarefied gas theory, Fluid Dyn. 3 (1) (1968) 112–115.
[30] E. Shakhov, Generalization of the Krook kinetic relaxation equation, Fluid Dyn. 3 (5) (1968) 95–96.
[31] E. Shakhov, Transverse flow of a rarefied gas around a plate, Fluid Dyn. 7 (6) (1972) 961–966.
[32] E. Shakhov, The axisymmetric non-linear steady flow of a rarefied gas in a pipe of circular cross section, Comput. Math. Math. Phys. 36 (8) (1996) 1123–1131.
[33] F. Sharipov, Benchmark problems in rarefied gas dynamics, Vacuum 86 (11) (2012) 1697–1700 (special issue “Vacuum Gas Dynamics: Theory, Experiments and Practical Applications”).
[34] D. Sharov, H. Luo, J.D. Baum, R. Löhner, Implementation of unstructured grid GMRES+LU-SGS method on shared-memory, cache-based parallel computers, in: 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2000, pp. 10–13 (AIAA-2000-927).
[35] N. Tillaeva, A generalization of the modified Godunov scheme to arbitrary unstructured meshes, Trans. Central Aerohydrodyn. Inst. 17 (2) (1986) 18–26 (in Russian).
[36] V. Titarev, Towards fully conservative numerical methods for the nonlinear model Boltzmann equation, Preprint NI03031-NPA, Isaac Newton Institute for Mathematical Sciences, University of Cambridge, Cambridge, UK, 2003, p. 13.
[37] V. Titarev, Conservative numerical methods for model kinetic equations, Comput. Fluids 36 (9) (2007) 1446–1459.
[38] V. Titarev, Implicit numerical method for computing three-dimensional rarefied gas flows using unstructured meshes, Comput. Math. Math. Phys. 50 (10) (2010) 1719–1733.
[39] V. Titarev, Efficient deterministic modelling of three-dimensional rarefied gas flows, Commun. Comput. Phys. 12 (1) (2012) 161–192.
[40] V. Titarev, Implicit high-order method for calculating rarefied gas flow in a planar microchannel, J. Comput. Phys. 231 (1) (2012) 109–134.
[41] V. Titarev, Rarefied gas flow in a circular pipe of finite length, Vacuum 94 (2013) 92–103.
[42] V. Titarev, E. Shakhov, Computational study of a rarefied gas flow through a long circular pipe into vacuum, Vacuum 86 (11) (2012) 1709–1716 (special issue “Vacuum Gas Dynamics: Theory, Experiments and Practical Applications”).
[43] A. Vaganov, S. Drozdov, A. Kosykh, G.G. Nersesov, I. Chelysheva, V.L. Yumashev, Numerical simulation of aerodynamics of winged re-entry space vehicle, TsAGI Sci. J. 40 (2) (2009) 131–149.
[44] A. Vaganov, S. Drozdov, S. Zadonsky, I. Chelysheva, A. Kosykh, G. Nersesov, V. Yumashev, The study of aerodynamics of the aerospace vehicle with a deflected balancing flap, TsAGI Sci. J. 40 (5) (2009) 517–534.
[45] B. van Leer, Towards the ultimate conservative difference scheme V: a second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[46] B. van Leer, A historical oversight: Vladimir P. Kolgan and his high-resolution scheme, J. Comput. Phys. 230 (7) (2011) 2378–2383.
[47] S. Varoutis, D. Valougeorgis, O. Sazhin, F. Sharipov, Rarefied gas flow through short tubes into vacuum, J. Vac. Sci. Technol. 26 (1) (2008) 228–238.
[48] S. Varoutis, D. Valougeorgis, F. Sharipov, Simulation of gas flow through tubes of finite length over the whole range of rarefaction for various pressure drop ratios, J. Vac. Sci. Technol. A 27 (6) (2009) 1377–1391.
[49] V. Venkatakrishnan, On the accuracy of limiters and convergence to steady-state solutions, in: 31st Aerospace Science Meeting & Exhibit, Reno, NV, January 11–14, 1993, AIAA paper 93-0880.
[50] V. Vlasenko, On mathematical basis and principles of construction of numerical methodologies in the applied package EWT-TsAGI, Tr. TsAGI 2671 (2007) 20–85.