View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Cronfa at Swansea University

Cronfa - Swansea University Open Access Repository

This is an author produced version of a paper published in:
International Journal of Computational Fluid Dynamics

Cronfa URL for this paper:
http://cronfa.swan.ac.uk/Record/cronfa51144

Paper:
Evans, B., Hanna, M., Dawson, M. & Mesiti, M. (2019). High order parallelisation of an unstructured grid, discontinuous-Galerkin finite element solver for the Boltzmann–BGK equation. *International Journal of Computational Fluid Dynamics*, 1-9.
http://dx.doi.org/10.1080/10618562.2019.1651843

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder.

Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

http://www.swansea.ac.uk/library/researchsupport/ris-support/

High order parallelisation of an unstructured grid, discontinuous–Galerkin finite element solver for the Boltzmann-BGK equation

B. Evans[a,*], M. Hanna[a], M. Dawson[b], M. Mesiti[b]

[a] *College of Engineering, Swansea University, Swansea SA1 8EN, Wales, UK*
[b] *College of Science, Swansea University, Swansea SA1 8PP, Wales, UK*

Abstract

This paper outlines the implementation and performance of a high order parallelisation approach involving partitioning of both (unstructured grid) physical space and (discrete) velocity space domains for finite element solution of the Boltzmann-BGK equation. The numerical solver utilised is based on a discontinuous Taylor-Galerkin finite element approach. To the authors' knowledge this is the first time a 'high order' parallelisation, or 'phase space parallelisation', approach (i.e. parallelisation of both physical and velocity space domains) has been attempted in conjunction with a numerical solver of this type. In previous work domain decomposition was applied to the physical space alone leading to restrictions on parallelisation scalability. These restrictions have been overcome with the implementation detailed in this paper.

The developed algorithm has major advantages over continuum solvers in applications where strong discontinuities prevail (e.g. hypersonic flow) and/or in rarefied flow applications where the ratio of mean free path of molecules to the flowfield reference length scale (Knudsen number) is large. Previous work by the authors has outlined the range of applications that this solver is capable of tackling. This paper focuses on the parallelisation approach and performance.

The paper demonstrates that the high order, phase space parallelisation implemented is significantly more effective than previous implementations at exploiting High Performance Computing architectures. This opens up the range of potential application areas for this solution approach.

Keywords: Knudsen, Boltzmann-BGK, Computational Fluid Dynamics, Kinetic Theory, hypersonics, rarefied gas flow, discontinuous Galerkin

1. Introduction

The Boltzmann Equation provides a mathematical description of the statistical behaviour of non-equilibrium thermodynamic systems [1]. The Boltzmann Equation is an integro-differential equation that describes the transport of a scalar variable, the molecular velocity distribution function, f. This distribution function describes the probability of finding a molecule for a defined point in physical space for a given velocity state at a given time [2, 3]. The Boltzmann equation is based on a description of the fluid at the molecular level and, via integration of molecular properties across 'velocity space' (taking moments), the variation in macroscopic flow variables (pressure, temperature etc.) across physical space can be determined. If the the molecular velocity distribution

*Corresponding Author: Tel.: +441792602129
Email address: b.j.evans@swansea.ac.uk (B. Evans)

function, f, can be determined across phase space (physical and velocity space), all macroscopic state properties can be calculated without the requirement of making any of the classical continuum assumptions. Another key strength of a Boltzmann–based approach is that transport properties such as viscosity and thermal conductivity do not need to be defined via empirical relationships as is necessary in continuum approaches. Rather, these phenomena are naturally accounted for under the single governing equation.

The Boltzmann equation can take the following form:

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \frac{1}{Kn}Q(f, f^*), \tag{1}$$

where $f = f(\boldsymbol{r}, \boldsymbol{c}, t)$ is the molecular velocity distribution function across physical space, $\boldsymbol{r}$, velocity space, $\boldsymbol{c}$, and time, t. The molecular number density is represented by n, $\boldsymbol{F}$ describes any force fields that might be present (gravitational, electrostatic etc) and $Q(f, f^*)$ is the term accounting for molecular collisions. The term $Q(f, f^*)$ is a five-fold integral and the cause of major difficulties when attempting to attain a solution. The Boltzmann equation is a complex, non–linear, integro–differential equation and can be applied analytically only to highly simplified applications as shown by Kuznetsov [4]. For this work, the solver used is an extension to the finite element approach of Evans et al as exploited in previous works [2, 5, 6].

The dimensionless coefficient, Knudsen number, allows the classification of a fluid flow in terms of the degree of rarefaction [7]. The Knudsen number, Kn, is defined as, $Kn = \frac{\lambda}{L}$, where λ is the mean free path of molecules in the flow and L is a suitable reference length scale. From kinetic theory, the mean free path, λ, can be computed as $\lambda = \frac{k_B T}{\sqrt{2}\pi d^2 p}$ where k_B is the Boltzmann constant ($1.38 \times 10^{-23} J/K$), T is the gas temperature, p the gas pressure and d the molecular diameter.

In this paper a simplified version of the full Boltzmann equation, known as the Boltzmann-BGK equation [8], is utilised. Rather than attempting to fully describe the molecular collisions, the effect of these collisions on the form of the distribution function is considered as a mathematical model. This is possible by exploiting the Maxwellian equilibrium distribution function [9] and the assumption that the effect of molecular collisions is to return a non-thermodynamic equilibrium system back to the Maxwellian state in a time-frame directly proportional to the molecular collision frequency. In this work it is also assumed that the effects of external force fields are negligible such that the governing equation takes the following form

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c} \cdot \frac{\partial(nf)}{\partial \boldsymbol{r}} = \nu(\boldsymbol{r}, t)((nf_0) - (nf)), \tag{2}$$

where $\nu(\boldsymbol{r}, t)$ is a term proportional to the molecular collision frequency and f_0 is the local Maxwellian equilibrium distribution function in its 2D form, $f_0(\boldsymbol{c}) = (\frac{\beta^2}{\pi})\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_0})^2)$, where $\boldsymbol{c_0}$ is the bulk velocity of the flow and $\beta = (2RT)^{-1/2} = \sqrt{m/(2k_B T)}$, R is the gas constant, T is the gas temperature, m is the molecular mass and k_B is the Boltzmann constant. The inclusion of the equilibrium distribution function in the BGK term means that the Boltzmann-BGK equation is still a non–linear, integro–differential equation because f_0 is a function of the fluid bulk velocity, $\boldsymbol{c_0}$ and the temperature, T, which are obtained by taking moments of f across velocity space. However, computationally, the BGK term is significantly less demanding than the full Boltzmann equation right-hand side term.

Taking moments of molecular property, ϕ, to achieve the mean, bulk value, $\bar{\phi}$, at a point in physical space is achieved by integrating the product of ϕ and the distribution function, f, across velocity space:

$$\bar{\phi} = \int_{-\infty}^{+\infty} \phi f(\boldsymbol{c}) d\boldsymbol{c}. \tag{3}$$

By setting ϕ to the appropriate molecular parameter, the bulk flow properties can be derived as follows:

- Density, ρ: $\phi = mn$ where m is the molecular mass and n is the number density
- Bulk velocity, $\boldsymbol{c_0}$: $\phi = \boldsymbol{c_i}$ where $\boldsymbol{c_i}$ is the molecular velocity ($i = x, y$ in 2D Cartesian coordinates)
- Static pressure, $\boldsymbol{p_i}$: $\phi = mn\boldsymbol{c_i c_j}$ (note that static pressure is a vector quantity for a non-equilibrium gas).
- 'Kinetic' temperature, T_k, is defined as $T_k = \frac{p}{R\rho}$.

2. Solution Approach and Algorithm Parallelisation Strategy

2.1. Discretisation of Phase Space

Since the molecular velocity distribution, f, is dependent on both the physical space (p-space) and velocity space (v-space), discretisation must be performed on both the physical and velocity space domains. In this paper, only two dimensional problems (in p-space) are considered. The p-space domain, denoted as Ω_r in Figure 1 (a), is discretised into an unstructured assembly of discontinuous, linear, triangular elements with nodes at the vertices. This choice of discretisation was made to exploit the natural capture of complex geometries using unstructured grids, to enable the natural capture of discontinuities in the solution field and to facilitate straight-forward parallelisation of the physical space using METIS-based domain decomposition [10].

The v-space is, in principle, infinite in extent. However, in practice, it can be limited to circular with radius, r_v, representing a limit on the maximum speed a molecule is capable of travelling within the model. This introduces an assumption that any effects caused by molecules travelling at speeds greater than this user-imposed maximum limit are negligible. In accordance with previous work using the same solver, a rule of thumb adhered to is that the maximum velocity in the v-space is 'at least several times the mean thermal molecular velocity' [2, 5, 6]. Discretisation of the v-space domain, using a Discrete Velocity Model (DVM) approach [11, 12], is achieved using a Lobatto quadrature approach as described in [2] resulting in a nodal distribution and associated Lobatto weighting distribution as shown in Figure 1 (b). It must be noted that this is not necessarily the most efficient quadrature approach for this domain but investigating the performance of alternative quadrature schemes in this context, as has been done by Li et al [11], was deemed beyond the scope of this work. The principles of the parallelisation approach outlined here will naturally extend to any chosen quadrature scheme.

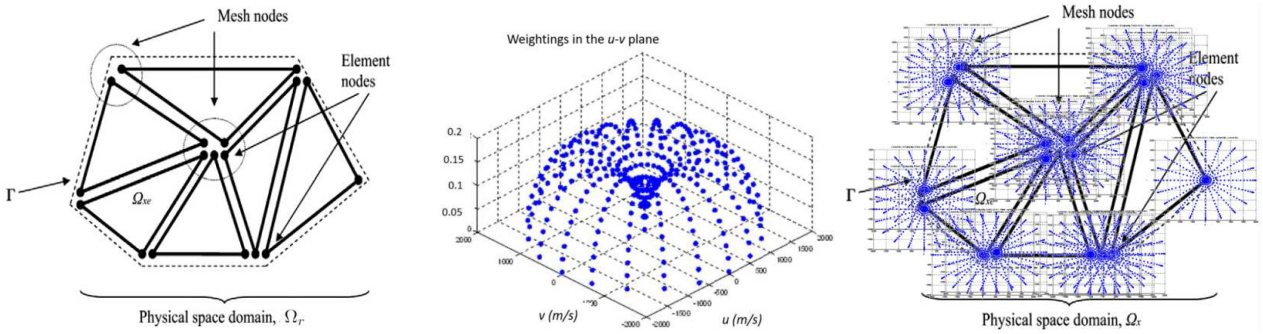
Figure 1 (c) shows the superposition of the v-space discretisation, in Figure 1 (b), onto each discontinuous element node in a (simplified) p-space domain. The total number of degrees of freedom in the problem, given this discretisation approach, is therefore $n_{DoF} = n_p \ . \ n_v$ where n_p is the number of p-space nodes ($n_p = 3.n_e$ where n_e is the number of p-space elements) and n_v is the number of v-space nodes/sample points.

2.2. Equation Discretisation

A Taylor-Galerkin approach is implemented for equation discretisation which leads to a two-step procedure for incrementing the initial solution to steady state as follows:

Step 1

A standard discontinuous finite element spatial discretisation, as discussed in section 2.1, is used and the discretised form of equation 2 is rearranged in order to define a 'half–timestep increment', $\Delta(nf)_{re,\boldsymbol{c}}$, on physical space element, re, as



(a) Discretisation of the physical space into an assembly of unstructured, discontinuous, triangular elements

(b) Lobatto coordinates and weights for a velocity space discretisation using 400 nodes

(c) Superposition of v-space domain onto each p-space discontinuous element node

Figure 1: Summary of the phase space discretisation space used in the Boltzmann-BGK solver

$$\Delta(nf)_{re,\boldsymbol{c}} = \frac{\Delta t}{2}\Sigma\Phi^m_{k,\boldsymbol{c}}N_k - \frac{\Delta t}{2}F^m_{ik,\boldsymbol{c}}\left.\frac{\partial N_k}{\partial r_i}\right]_{re,\boldsymbol{c}}, \tag{4}$$

where the Einstein summation convention is employed and $k(=1,2,3)$ extends over the three nodes of triangular element re, m is the timestep number, Δt is the local timestep governed by the Courant stability condition and $\Phi = \nu(\boldsymbol{r},t)((nf_0)-(nf))$ is the BGK collision term (RHS of equation 2). N_k is the standard, piecewise–linear finite element shape function associated with node k in physical space, subscript $\boldsymbol{c}$ represents the molecular velocity, subscript $i(=1,2)$ represents Cartesian coordinates (x,y) in 2D physical space and $F^m_{ik,\boldsymbol{c}} = F\left((nf)^m_{k,\boldsymbol{c}}\right)$ is the flux of $(nf)^m_{k,\boldsymbol{c}}$ which is simply the product of $(nf)^m_{k,\boldsymbol{c}}$ and the molecular velocity, $\boldsymbol{c}$, i.e.

$$F^m_{ik,\boldsymbol{c}} = \boldsymbol{c}(nf)^m_{k,\boldsymbol{c}}. \tag{5}$$

This allows a 'half timestep' physical space element flux to be computed by the piecewise–linear discontinuous representation

$$F_i^{m+\frac{1}{2}}]_{re,\boldsymbol{c}} = \boldsymbol{c}\left((nf)^m_{k,\boldsymbol{c}} + \Delta(nf)_{re,\boldsymbol{c}}\right)N_k, \tag{6}$$

where the definition of the flux, F, is again the product of nf (incremented by $\Delta(nf)_{re,\boldsymbol{c}}$) and the molecular velocity, $\boldsymbol{c}$. For a more detailed derivation of equations 4, 5 and 6 the reader is referred to previous work by the authors in [2].

Step 2

Full timestep nodal increments are computed according to

$$M_L]_{re}\Delta(nf)_{k,\boldsymbol{c}} = \Delta t M_L]_{re}\Phi^{m+\frac{1}{2}} + \Delta t\int_{\Gamma_{re}} F^{m+\frac{1}{2}}_{n,\boldsymbol{c}}N_k \mathrm{d}\Gamma_{re} - \Delta t\int_{\Omega_{re}} F^{m+\frac{1}{2}}_{ik,\boldsymbol{c}}\frac{\partial N_k}{\partial r_i}\mathrm{d}\Omega_{re,\boldsymbol{c}}, \tag{7}$$

where $M_L]_{re}$ is the standard, lumped, 3x3 physical space element mass matrix, $F^{m+\frac{1}{2}}_{n,\boldsymbol{c}}$ denotes the normal component of the upstream flux at the physical space element edges for a velocity of $\boldsymbol{c}$, Γ_{re} is the physical space

element boundary and Ω_{re} is the physical space element. Appropriate choices must be made for inter-element fluxes at domain boundaries depending on the appropriate boundary conditions being applied (inflow, outflow or wall). In order to appreciate the communication required across velocity space during the application of wall boundary conditions, the corresponding algorithm is outlined in the following section.

2.3. Wall Boundary Condition

The condition that must be enforced at a solid wall is zero mass flux across the boundary. In a kinetic theory description, this is expressed as

$$\int_{\Gamma_r} \int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}} \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}\Gamma_r = 0, \tag{8}$$

where $F_{n,\boldsymbol{c}} = (\boldsymbol{c}.\boldsymbol{n}) f(\boldsymbol{c}, \boldsymbol{r}, t)$ and Γ_r is the p–space domain boundary. This condition is ensured by an appropriate modelling of molecular collisions with the wall. We make the assumption that a certain fraction, α, of molecules are adsorbed by the wall and remitted in thermodynamic equilibrium with the wall (diffuse reflection). The remaining fraction, $(1-\alpha)$, are not adsorbed by the wall and simply reflect directly back into the domain (specular reflection). The term α is known as the 'adsorption coefficient'. The distribution function of the net reflected flux of molecules is, therefore, constructed as

$$f(\boldsymbol{c}, \boldsymbol{r}, t) = (1-\alpha) R f(\boldsymbol{c}, \boldsymbol{r}, t) + \alpha M f(\boldsymbol{c}, \boldsymbol{r}, t), \qquad \text{for} \quad \boldsymbol{c}.\boldsymbol{n} \leq 0, \tag{9}$$

where $Rf(\boldsymbol{c}, \boldsymbol{r}, t) = f(\boldsymbol{c} - 2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}), \boldsymbol{r}, t)$, $Mf(\boldsymbol{c}, \boldsymbol{r}, t) = \eta(\boldsymbol{r}, t) M_w(\boldsymbol{c})$ and $\boldsymbol{n}$ is the outward facing unit normal at the wall. If T_w is the wall temperature, then M_w is determined as

$$M_w = \exp\left(-\frac{c^2}{2\, R\, T_w} \right). \tag{10}$$

The parameter η is used to enforce the condition in equation (8), i.e. it is used to ensure conservation of mass at the the wall, which implies that

$$\eta(\boldsymbol{r}, t) = \frac{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})>0} f(\boldsymbol{c}, \boldsymbol{r}, t) |\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{c}}{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})\leq 0} M_w(\boldsymbol{c}) |\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})| \, \mathbf{d}\boldsymbol{c}}. \tag{11}$$

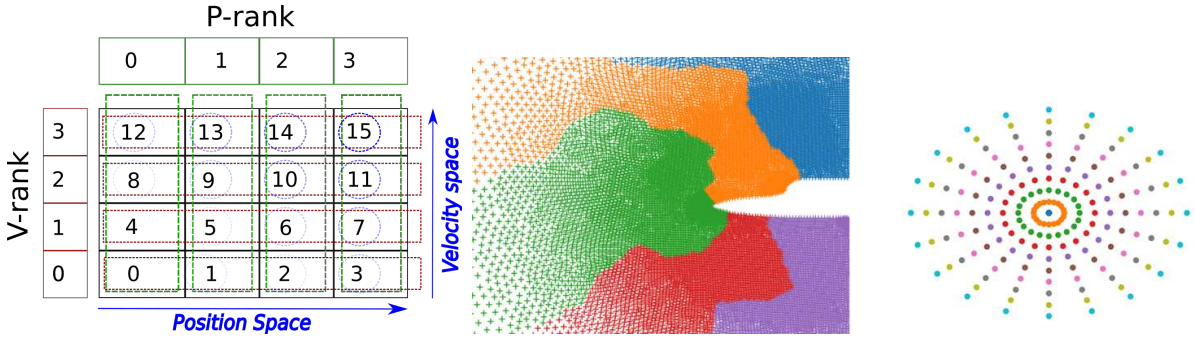
For further details on the application of this boundary condition and for a proof of its mass conserving properties, the reader is referred to [2].

2.4. Algorithm Structure

Algorithm 1 shows a pseudo-code outlining the algorithm structure emphasising the points within the algorithm where communication is required across the p-space domain and across the v-space domain.

Algorithm 1 Pseudo code describing a single timestep iteration of the Boltzmann-BGK solver.

for all edges on the boundary **do**
  Compute contributions to $\eta(\boldsymbol{r}, t)$ (Eq.11, numerator)
end for
Communicate contributions to flux conservation parameter **(reduction across v-space)**
for all edges on the boundary **do**
  Compute $\eta(\boldsymbol{r}, t)$, (Eq. 11)
end for
for all elements in the current p-space partition **do**
  for all nodes in element **do**
    Compute macroscopic flow variables at each element node **(reduction across v-space)**
    Apply BGK collision term
  end for
end for
for all points in current v-space partition **do**
  for all elements in current p-space partition **do**
    Compute the half-timestep p-space element flux based on current point in v-space
  end for
  for all elements in the current p-space partition **do**
    for all nodes in element **do**
      Update Nodal distribution functions (Eq. 7) - local contribution
      Compute remote contribution to nodal distribution functions (Eq. 7).
      Apply boundary contitions **(if at wall boundary communication across v-space)**
    end for
  end for
  Communicate data for nodal distribution function update **(all-to-all communication across p-space)**
  for all elements in the current p-space partition **do**
    for all nodes in element **do**
      Update Nodal distribution functions - remote contribution (Eq. 7)
    end for
  end for
end for



(a) MPI ranking system used in the parallelisation procedure. Rows correspond to partitions in v-space, columns to partitions in p-space.

(b) P-space partitioning where the centres of various elements with the same colour correspond to being processed via the same processor

(c) V-space partitioning where velocity sampling points with the same color correspond to being processed with the same processor

Figure 2: Overview of the phase space parallelisation approach used for algorithm acceleration on parallel HPC architectures

2.5. Phase Space Parallelisation

The important contribution to the methodology introduced in this work is the parallelisation of the algorithm across both physical and velocity space. In previous works in which a finite element scheme has been used for solution of the Boltzmann-BGK equation parallelisation has relied on physical space domain decomposition alone [2, 13]. This has led to restrictions on scalability. Recent work by Ho et al [12] in which a Boltzmann-BGK solver has been shown to scale to multiple thousands of cores is restricted to structured physical space grids. The use of unstructured grids to discretise the physical space allows a natural application to more complex geometries but does require a more sophisticated approach to mesh partitioning (described below) and to the derivation of the inter-element fluxes (as described in equations 4 through 6).

For this parallelisation scheme, each MPI rank is identified by two integer numbers: the 'p-rank' and the 'v-rank', i.e. the index in the physical space domain decomposition and the index in the velocity space domain decomposition. This is shown in Figure 2 (a). MPI ranks with the same p-rank index contain identical physical space partitions, but contain solution information for different points in v-space and the converse is true for MPI ranks with the same v-rank index.

Processes in the same row exchange data during p-space element edge (inter-element) flux communications (Algorithm 1, line 25), while processes in the same column exchange data during the computation of integrals across v-space (Eq.3, Algorithm 1, line 10) and computing the flux conservation parameter η (Eq.11, Algorithm 1, line 4).

For the p-space, the number of partitions is selected and METIS is used to partition the domain such as to minimise the communication requirement between partition edges [10]. Figure 2 (b) shows how the mesh used in the case study detailed in section 3.1 is partitioned for 5 p-ranks. Computing updates to the distribution function at a node in the p-space mesh requires evaluation of a flux between elements. Since the Boltzmann–BGK equation numerical solution algorithm is 'local' in p-space, this flux depends only on solution values in neighbouring elements, and thus requires only 'nearest neighbour' communication. The flux term of the adopted discretised form of the Boltzmann–BGK equation (Eqs.4,7) is also *mostly* 'local' in v-space, as all the v-space points can be treated independently (with the notable exception of the collision term, Φ). This suggests that, in

principle, a v-space domain decomposition scheme can be adopted. This would enable each node/sampling point in v-space to be assigned to a different process. In addition to the BGK collision term and the computation of η (Eq.11), another exception to v-space locality on the object boundary is due to molecular specular reflection, which changes the direction of the movement of the molecule (but not the speed in the wall reference frame), causing an intra v-space flux to appear (see Eq.9). It is, in principle, possible to take this into account, but it is considerably simpler to exclude partitioning of v-space in the angular direction. This ensures that processes are not required to exchange any data due to particle reflections on the object boundary (Algorithm 1, line 22), keeping the v-space parallelisation straightforward. The operations that needed to be modified to allow the parallelisation in v-space are the computation of the moments (equation 3, discretised form in [2]), where each process must compute the contribution to the moment relative to its own v-space partition, and the sum of the partial results for each v-space partition must be communicated to all processes involved. This computation can be performed independently for each node in p-space. Figure 2 (c) shows how the v-space is partitioned for a typical 20x10 Lobatto quadrature. There are 10 different processes handling the v-space. It can be seen that every process handles equally 20 points in the v-space hence equal distribution of processing requirements. In this partitioning scheme, the total number of processes (ranks) is equal to the product of the number of p-space partitions (np_p) and the number of v-space partitions (np_v).

3. Application - Results and Discussion

The focus of this paper is the performance of the novel parallelisation implementation. For details on validation of the underlying solver algorithm, the reader is referred to previous work by the lead author in which validation across a range of test cases against continuum CFD, Direct Simulation Monte Carlo (DSMC) and analytical solutions are undertaken [6]. In the section below we outline the results from the baseline case used to test the performance of the parallelisation approach i.e. rarefied, hypersonic flow over a double ellipse.

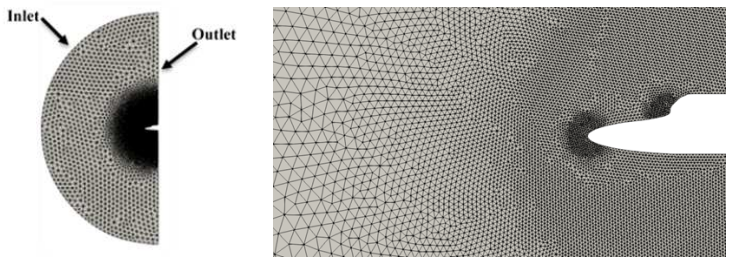
3.1. Rarefied, hypersonic flow over a double-ellipse

In this test case a generic hypersonic re-entry spacecraft leading body was considered under high Mach and low density conditions similar to that encountered during the reentry phase of a space flight in the upper atmosphere. The low pressure / low density atmospheric conditions of the upper atmosphere leads to a large molecular mean free path, λ, resulting in high Knudsen number. The geometry of the generic space vehicle leading body takes the form a double ellipse as can be seen in Figure 3. Details of the discretisation used and computational requirements for this baseline case are shown in Table 1.

Note that the 'spikes' appearing in the solutions in Figure 4 are numerical anomalies resulting from the particular quadrature scheme chosen which introduces 'preferred' radial directions for molecules. It might be possible to reduce the influence of these with alternative quadrature schemes which will be investigated in future work.

3.2. Parallelisation Performance

The exercise of describing and understanding the scaling properties of this parallelisation scheme is more complex than for the usual finite element parallelisation strategies, because of the dual nature of our approach, where we have two parameters we can vary, namely the number of p-ranks and the number of v-ranks. The



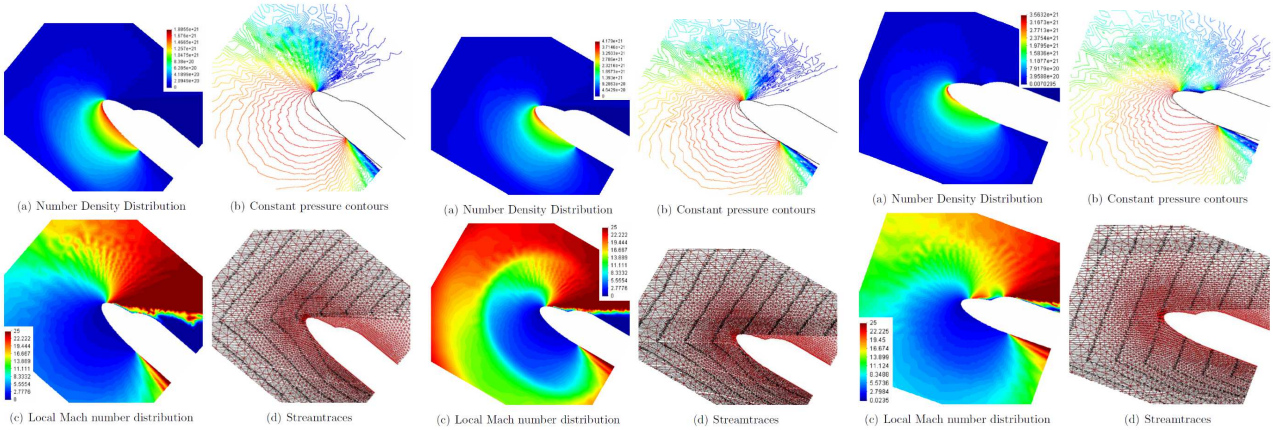
(a) Full p-space domain showing the inlet and outlet

(b) Zoom in showing the p-space mesh in the vicinity of the vehicle geometry

Figure 3: P-space mesh used for the hypersonic reentry case

P-space elements	28,304
P-space nodes	84,912
Lobatto order	80
V-space nodes	6,400
Degrees of Freedom (DoFs)	543,436,800
M_∞	25
AoA	$0° - 40°$
Kn	10
no. v-space partitions np_v	80
no. v-space partitions np_p	8
Number of cores	640
Runtime to steady state convergence	1.5 hrs

Table 1: Computational and flowfield parameters for the baseline hypersonic double ellipse case



(a) AoA = 40°

(b) AoA = 30°

(c) AoA = 20°

Figure 4: Molecular number density, pressure, Mach distributions and streamtraces for hypersonic reentry cases

scaling behaviour of the system is not only dependent on the total number of processes used, but also to how the total number of processes is partitioned between p-space and v-space. To appreciate the performance of the parallelisation scaling, the following observations must be made:

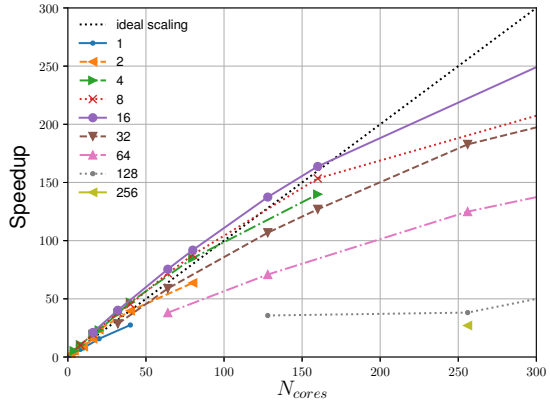
1. communications between v-space partitions occurs $\mathcal{O}(1)$ times per timestep;
2. communications between p-space partitions occur $\mathcal{O}(10)$-$\mathcal{O}(100)$ times per timestep (once per each point in the current velocity space partition);
3. the number of v-space partitions is limited to the order of the Lobatto quadrature;
4. the size of the messages to be exchanged during the p-space communication is proportional to the number of edges at the boundary of the p-space partition;
5. the size of the messages exchanged during v-space communications is proportional to the number of elements in the p-space partition, but the cumulative size of the messages is proportional to the number of v-space partitions;
6. The architecture of the HPC system used for this study has 16 cores per compute node;
7. Inter-node communication is an order of magnitude slower, in terms of sustained bandwidth and latency, than intra-node communication;

Taking into account all these effects is very complicated and well beyond the scope of this paper but we can, nonetheless, try to point out what are the limiting factors for the case studied.

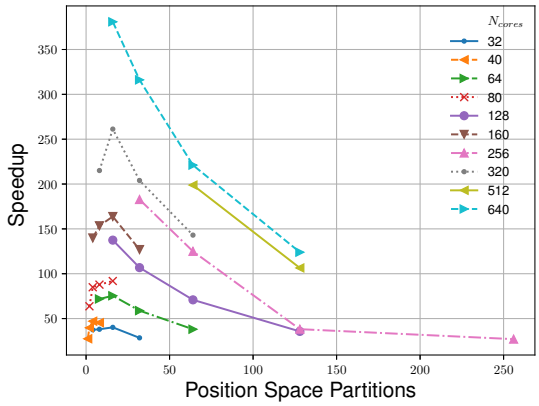
In Figure 5 the speedup is shown for the discretisation outlined in Section 3.1. The scaling behaviour for fixed number of p-space partitions np_p is shown in Figure 5 (a). For each value of np_p separately we see a standard scaling behaviour, with an efficiency (slope) dependent on the number np_p, and being maximal for $np_p = 16$. The same data is plotted also in Figure5 (b), where the lines connect data points relative to the same total number of processes. If the scaling were ideal, we would have perfectly horizontal lines here, but this is not the case. A possible explanation of this data is that we are close to the scalability limit in p-space, with p-space communications being the main bottleneck, which is alleviated when a set of processes in the same v-space partition sits on the same computing node, and all processes are able to communicate with minimal cost, i.e. with low latency and high bandwidth. This is indeed the case only when np_p is 16, which explains the maxima we see for all the lines in the plot.

In Figure 6 the speedup is shown for the same problem but run on a much finer p-space mesh. In Figure 6 (b) we notice that performance in this case seems much less affected by the balance of np_p and np_v than in the 'smaller' case. It is also worth noting that the speedup with 640 processes is generally lower than the one with 512 processes, and that the speedup with 320 processes is lower than the one with 256 processes. In order to fully explore this behaviour further analysis (beyond the scope of this paper) will be required.

We can conclude that, given a total number of cores available, is is a safe choice to set np_p to the number of cores in a node, in order to minimise the effect of the limitation in bandwidth and the network latency on p-space communication. In cases where the p-space partitions are still large, it may be advantageous to increase the number of p-space partitions at the expense of the number of v-space partitions in order to reduce the total quantity of data to be exchanged. A more precise analysis of the speedup expected for a given domain decomposition requires careful consideration of both the types of partition used, the amount of communication in a given problem, the mechanisms used by the MPI communication and the topology of the network.

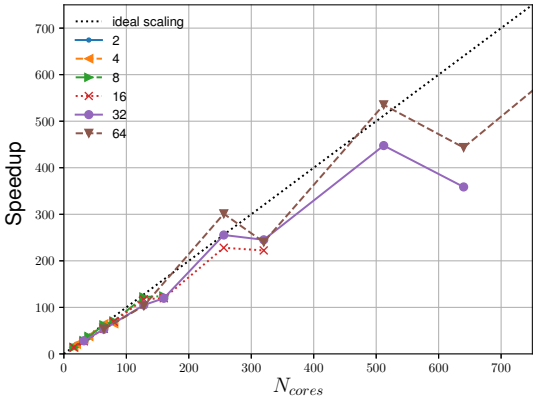


(a) Speedup as a function of total number of ranks, grouped by number of p-space partitions (np_p).

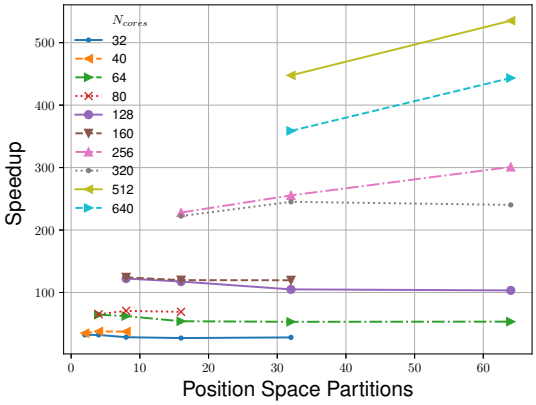


(b) Speedup as a function of number of p-space partitions (np_p), grouped by total number of ranks.

Figure 5: Scaling perfomance for the hypersonic reentry case outlined in section 3.1 (with $5.4 \cdot 10^8$ DoF,$8.5 \cdot 10^4$ p-space nodes). Speedup is computed w.r.t. one process.



(a) Speedup as a function of total number of ranks, grouped by number of p-space partitions (np_p).



(b) Speedup as a function of number of p-space partitions (np_p), grouped by total number of ranks.

Figure 6: Scaling perfomance for a larger problem with $7.7 \cdot 10^9$ DoF,$1.2 \cdot 10^6$ p-space nodes. Speedup is computed w.r.t. one process.

4. Conclusions and final remarks

One of the biggest disadvantages of a molecular gas dynamics based approach to solving fluid dynamics problems using a direct solution of the Boltzmann equation in previous work has been the very high computational requirements. The work outlined in this paper, to a large extent, overcomes that problem and significantly reduces the runtime requirement to simulate complex, rarefied gas flowfields. Phase space parallelisation was achieved in this work where both physical space and the velocity space are partitioned across the available cores of the compute system. This is an important step that now allows the possibility of extending this solver into 3D in future work. Other algorithm improvements that will be considered in future work include exploring alternative, high order quadrature schemes for velocity space integration.

Acknowledgements

The authors would like to thank the Zienkiewicz Centre for Computational Engineering for the generous Zienkiewicz Scholarship awarded to Mark Hanna to allow him to undertake this body of work. The authors acknowledge the support of the Supercomputing Wales project and the Swansea Academy of Advanced Computing, which are part-funded by the European Regional Development Fund (ERDF) via Welsh Government.

5. References

[1] Uffink, J., Boltzmann's Work in Statistical Physics, Stanford Encyclopedia of Philosophy, Stanford University, 2014

[2] Evans, B., Morgan, K., Hassan, O., A discontinuous finite element solution of the Boltzmann kinetic equation in collisionliess and BGK forms for macroscopic gas flows, App Math Mod 35, pp 996-1015, 2011

[3] Pieraccini, S., Puppo.,G, Implicit - Explicit Schemes for BGK Kinetic Equations, Journal of Scientific Computing, vol. 32 (1), pp 128, 2007

[4] Kuznetsov,M.M., Analytical solution of the Boltzmann equation in a Knudsen layer, J. App. Mech. Tech. Phys. 12(4), PP 604-607, 1971

[5] Evans, B., Walton, S.P., Aerodynamic optimisation of a hypersonic reentry vehicle based on solution of the Boltzmann-BGK equation and Evolutionary Optimisation, App Math Mod, 52, PP 215-240, 2017

[6] Evans, B., Nano-particle drag prediction at low Reynolds number using a direct Boltzmann-BGK solution approach, Journal of Computational Physics, 352, PP 123–141, 2017

[7] Niimi, t., High Knudsen Number Flows, Dept. of Micro/Nano Systems Engineering, Nagoya University

[8] Bhatnagar,p.l., Gross, E.P., Krook,M., Model for collision processes in gases, I Small amplitude processes in charged and neutral one-component systems, Phys. Rev., 94, pp 511-524, 1954

[9] Bird, G.A., Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford, 1994

[10] Karypis, G., Kumar, V., METIS - Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0 (Technical report), 1995

[11] Zhi-Hui Li, Han-Xin Zhang, Study on gas kinetic unified algorithm for flows from rarefied transition to continuum, Journal of Computational Physics, 193, pp 708-738, 2004

[12] Ho et al, A multi-level parallel solver for rarefied gas flows in porous media, Computer Physics Communications, 234, pp 14-25, 2019

[13] Wei Su, Alina Alexeenko, Guobiao Cai, A parallel runge-kutta discontinuous galerkin solver for rarefied gas flows based on 2D Boltzmann kinetic equations, Computers and fluids, 109, pp 123-136, 2015